Doctoral Dissertation
Doctoral Program in Physics (35.th cycle)

Unsupervised inference methods for protein sequence data

Luca Sesta

* * * * *

Supervisor
Prof. Andrea. Pagnani

Doctoral Examination Committee:
Prof. Martin Weigt, *Referee*, Ecole Normale Superieur, Paris
Prof. Francesco Zamponi, *Referee*, Ecole Normale Superieur, Paris
Prof. Matteo Osella, *External member*, Università degli Studi di Torino, Torino
Prof. Andrea Gamba, *Internal member*, Politecnico di Torino, Torino
Prof. Alfredo Braunstein, *Internal member*, Politecnico di Torino, Torino

Politecnico di Torino
May 12, 2023

I hereby declare that, the contents and organisation of this dissertation constitute my own original work and does not compromise in any way the rights of third parties, including those relating to the security of personal data.

. . . . . .

Luca Sesta
Turin, May 12, 2023

Summary

Recent years have seen an explosion of availability of protein sequence data. However, the vast majority of these data are unlabeled, that is, the sequences are not accompanied by supplementary information about their functional or structural properties. In this perspective, the development of statistical methods which are able to leverage this huge availability of sequence data to try to unveil the sequence function/structure relation represents an interesting chance for scientists, and especially for biophysicists and computational biologists.

Among the statistical methods developed to tackle sequence data, a relevant role has been played by statistical physics inspired strategies, such as the generalized Potts model, where protein sequences are interpreted as vectors of q-states *spin* variables, to which a scalar *energy* function is associated. In this framework, the general idea is to use the sequence data to determine the model constituent parameters, as in an inverse-problem of statistical physics. Such techniques have proven to be particularly effective in the context of *multiple sequence alignments* (MSA), for the determination of structural properties and in predicting mutational effects. A fundamental requirement for these models to be highly predictive is that they have to be global, or alternatively stated, epistatic. The minimal choice to achieve such feature is considering pairwise interactions between the protein residues, as it happens in the case of the *Direct Coupling Analysis* approach.

In this thesis, we present some novel unsupervised inference methods which are inspired by the DCA approach, but with the aim to extend them to protein sequence data which are produced by laboratory experiments. Considering the short time scales that characterize these experiments, especially when compared to the natural evolution process, such data turn out to be inherently out of equilibrium. We believe that incorporating (at the least effectively) this dynamical information into the statistical model might be beneficial to infer more efficiently and accurately the fine-grain structure of the fitness landscape, i.e. the functional (and structural) properties of the protein sequences in the vicinity of the ones tested experimentally.

The thesis outline goes as follows. In Ch. 1 we give a general biological introduction, describing what proteins are and why they are so important for living organisms. Then, we will introduce what an MSA is, and what information we can extract from this data structure. Finally, we will review the experimental techniques on which we will apply

the proposed inference methods.

These are treated in Ch. 3 and 4, and go under the names of *Annealed Mutational approximated Landscape* (AMaLa) and betaDCA respectively. The former was specifically conceived to be applied to sequence data generated from Directed Evolution experiments, whereas the second was meant as a more general model that could be applied to a wide variety of experimental settings. The distinctive feature of both methods is that they do not require accurate population information to infer meaningful models.

Another statistical physics inspired model which has recently sparked attention in the context of protein sequence data is represented by *Restricted Boltzmann machines* (RBM). In Ch. 5 of this thesis, we investigate the chance to employ *Expectation Propagation*, an iterative algorithm for approximating intractable probability distributions, to infer the constituent parameters of an RBM. The work related to this problem is still ongoing, and we present here the results obtained so far, postponing further analysis to future manuscripts.

Contents

Chapter 1

Biological Introduction

In this chapter we give a general biological introduction, focusing in particular on proteins. In Sec. 1.1 we present what proteins are and why they are considered a fundamental building block for the biological processes of living organisms. In Sec. 1.2, we introduce *multiple sequence alignments* (MSA), that are the typical data structure used to perform statistical inference based on protein sequences. Finally, in Sec. 1.3, we describe two kinds of protein evolution experiments, and we briefly present the adaptive immune system.

1.1 Proteins

In this section we introduce *proteins*, which are one of the most fundamental elements for the functioning of living beings. From a chemical perspective, a protein is a molecule made of specific fundamental building blocks, the amino acids (or peptides). Amino acids are organic compounds containing two chemical groups, the amino group ($-NH_2$) and the acidic carboxyl group (-COOH). On top of these two, a side chain group R differentiate twenty different amino acids, each with its peculiar chemico-physical properties, e.g.: polarity, charge, hydrophilicity or hydrophobicity. In Fig. 1.1, a list of the different amino acids divided according to their properties is shown.

Amino acids can bind together by means of the so called *peptide bond*, a covalent bond taking place between the nitrogen of the amino and the carbon of the carboxyl group, yielding a molecule of water as a byproduct. When multiple peptides bind together they produce a linear chain of peptide bonds, also called polypeptide backbone. The specific R groups are displayed on the sides of this chain-like structure, defining the protein sequence. Indeed, the very sequence of amino acids along the backbone is the simplest representation of a protein. Typical lengths in terms of peptide units span from 10 up to 1000.

Usually, proteins in living organisms are not found in a simple linear conformation, but are rather organized in a complex three-dimensional folded structure that minimizes

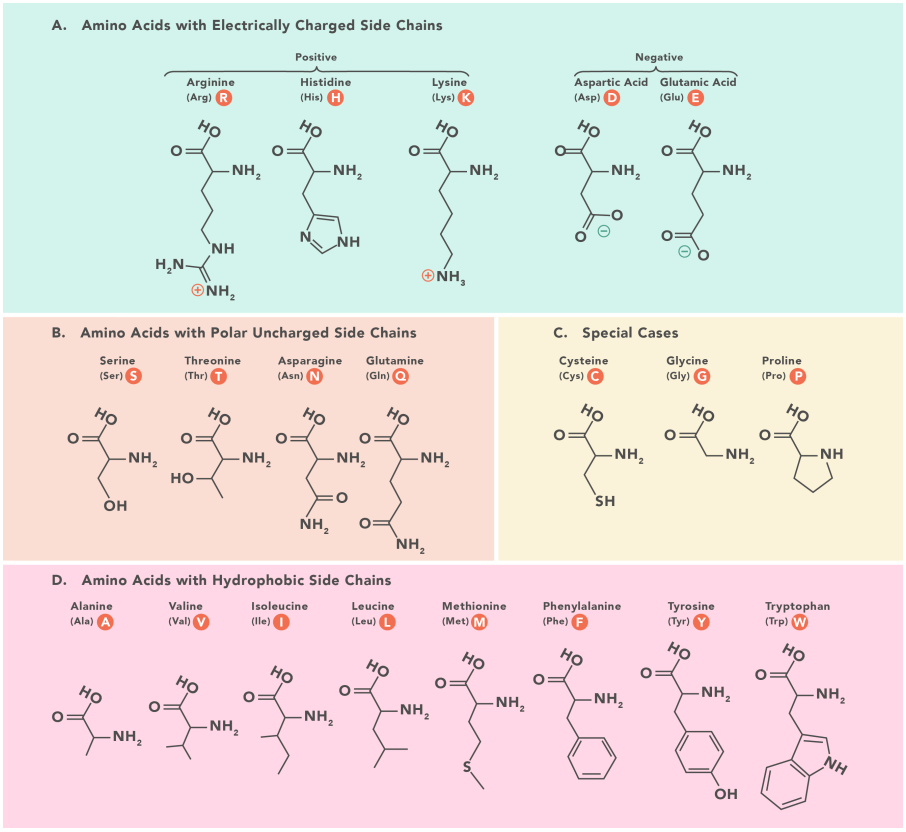


Figure 1.1: Schematic table classifying the twenty natural occurring amino acids according to their chemico-physical properties. Image taken from [158].

the Gibbs free energy. This originates from the chemical interactions among amino acids, e.g. ionic or hydrogen bonds and disulfide bridges. In particular, since proteins are usually found in aqueous environments, they tend to bury the hydrophobic amino acids into a stable core, and to expose the hydrophilic ones to the surface.

The three dimensional configuration that emerges as a consequence of this process is tightly related to the functional properties of the protein. More specifically, we can distinguish among several levels of structural organization.

- **Primary**: the very sequence of amino acids along the chain.
- **Secondary**: short range interactions between amino acids can create local structures that assume two possible shapes: α-helix and β-sheets.
- **Tertiary**: the actual three dimensional folded conformation of a protein. Long range interactions between amino acids play a crucial role in determining such structure.

- **Quaternary**: structure arising in inter-protein interactions defining protein complexes.

In Fig. 1.2 examples of all the organizational levels are reported.

Proteins perform a variety of different tasks in living beings. As enzymes, they have catalytic effects accelerating reactions. They also perform (or are involved into) signaling process, or are suited to bind to specific target (e.g. antibodies). Furthermore, proteins can be structural elements in tissues and cells, therein being static components or performing dynamical functions as for molecular motors (e.g. kinesin, myosin).

Due to the close relationship between structure and function, begin able to determine the protein folded configuration is a fundamental biological problem. However, determining experimentally the structure is a highly non-trivial and expensive task. Standard techniques rely on crystallized proteins, and employ X-ray diffraction in order to determine a representation of the structure in Fourier space. Consequently, being able to predict protein structure from the mere sequence of amino acids assumes a crucial role. Yet, directly simulating the folding process in a molecular dynamics fashion is a hard computational task. In this perspective, statistical-based methods represent an alternative approach to either impose some constraint to the folding simulation, or to predict the three dimensional structure altogether. In the recent years, astonishing progresses have been achieved thanks to machine learning approaches. In particular, AlphaFold2 [85], developed by Google during the 2020 **Critical Assessment of Protein Structure Prediction** (CASP), represents a powerful computational tool which is able to predict the protein structure from sequence information only.

1.1.1 Protein domains

Before going on, it is worth mentioning the concept of **protein domains**. Earlier, we discussed the different hierarchical organizations of a protein structure, and pointed out that the tertiary structure is the one playing a key role in determining the functional properties of a protein. However, it is not always necessary to focus on how the whole protein chain folds. Indeed, it is often possible to leverage the modular organization of proteins, which is based on protein domains, which are functional subunits that are found almost unchanged in a large variety of different proteins. Since these domains are tailored to perform specific tasks, they tend to maintain the same structure across different species. Then, one can think of a protein as a mixture of organized and structurally ordered regions (the protein domains), interspersed by relatively disordered ones.

In Fig. 1.3, we reported an example of a multi-domain protein.

In the following Sec. 1.2 we will focus on how to build useful data structures for the protein domains, pointing out the biological mechanisms that make this data so suitable for the application of statistical analysis.

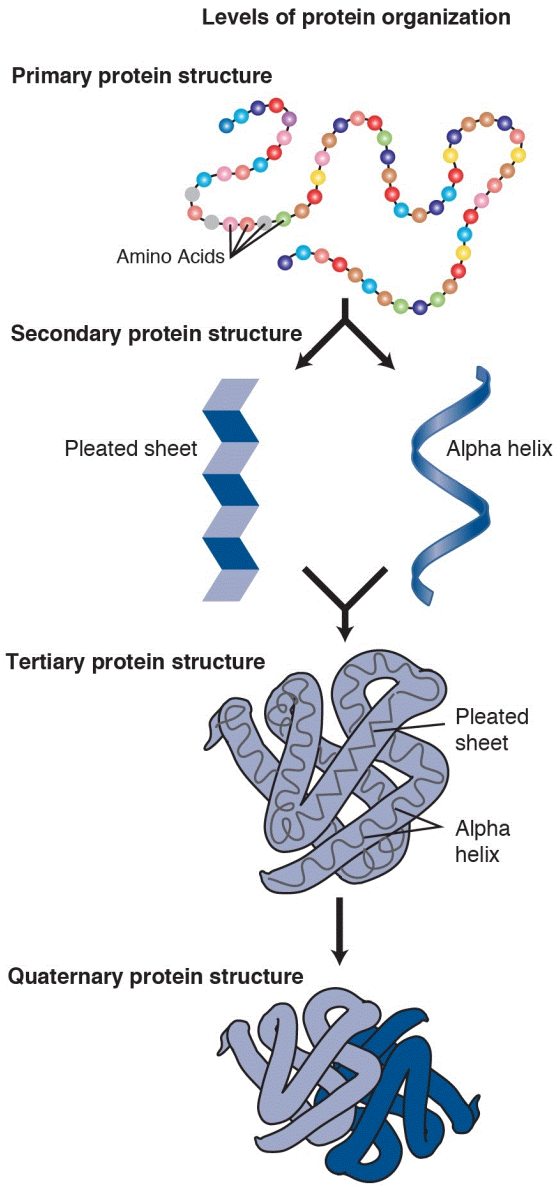


Figure 1.2: Schematic representation of the different level of organization of protein structure. From top to bottom: *primary* structure, the very sequence of amino acids along the chain; *secondary* structure, local β-sheet and α-helix conformations; *tertiary* structure, actual spatial organization in the three dimensional space; *quaternary* structure, inter-protein organization. Figure taken from National Human Genome Research Institute: https://www.genome.gov/genetics-glossary/Protein.

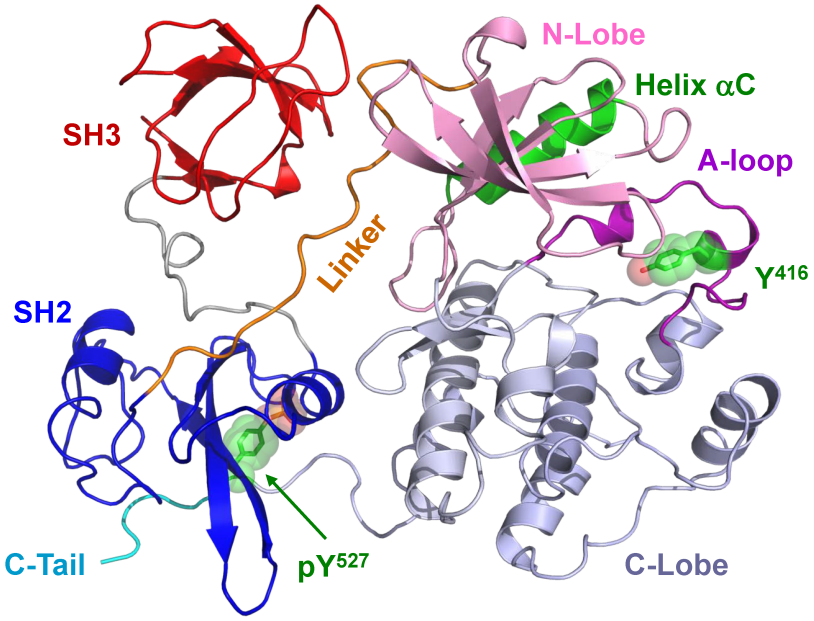


Figure 1.3: The figure displays the protein domain architecture of the protein c-Src Kinase, i.e. how the different domains within the protein are organized. In this specific case, it is possible to recognize the domain SH2 and SH3, and the peculiar position of the latter determines a down-regulation of kinase activity. Picture taken from [109].

1.2 Multiple Sequence Alignments

At the end of the Sec. 1.1 we mentioned how statistical-based methods might represent an appealing way to analyze protein data. Here, we discuss the specific data structure that allowed for an outbreak of applications of statistical approaches, namely, the so called **multiple sequence alignment** (MSA). In order to do so, we first need to introduce the concept of protein family.

1.2.1 Protein families

Proteins and protein domains evolve over time due to random mutations and selective pressure. In this perspective, a protein family is a collection of proteins (or domains) that share a common evolutionary history, starting from a common ancestral sequence. Such a set of sequences is said to be homologous, and what it is observed nowadays as the result of the evolution process coincides with the leaves of a phylogenetic tree.

As already mentioned in Sec. 1.1.1, a very interesting feature of homologous proteins is that, even if they became different from one another due to mutations, they still share the same function, and consequently tend to maintain the same structure. Thus, homologous sequences can be thought as different realizations of a same protein, from

which statistical information can be extracted.

Diversity among homologous sequences is introduced by means of several mechanisms: substitution, insertion and deletion. The accumulation of such mutations over very long time scales eventually produces sequences that, even if still maintain some similarities, possess an average sequence identity as low as 30%. Moreover, since both deletions and insertions alter the length of the sequences, an aligning process is required. This might introduce an additional symbol on top of the 20 naturally occurring amino acids, which is usually referred to as the *gap*, being indicated as $'-'$. The details of how such alignment is obtained are described in subsection 1.2.3, and the result of the process is the aforementioned MSA.

An MSA can be modeled as a matrix whose rows are the homologous sequences. Then, each row of the matrix is a vector of L components $\mathbf{S} = (\sigma_1, \sigma_2, \dots, \sigma_L)$, with L the common length of the aligned sequences. The variables σ's, live over a discrete alphabet of symbols $\{\sigma\} = \{A, C, G, \dots, Y\}$, with each letter uniquely identifying one amino acid, with the addition of the gap symbol. Eventually, the set of symbols can be mapped over the integer numbers from 1 to q, with q the size of the set, that in the case of MSA's usually coincide with $q = 21$. The number of unique sequences in the alignment is labeled with M.

MSA of protein families are collected into the PFAM database, where approximately 16000 alignments are available. These alignments contain a number of (non-unique) sequences that span between 10^2 and 10^5, with aligned sequences length L that goes from order 10 up to 500 residues.

1.2.2 Statistical features and biological signals

The relatively large number of sequences appearing in an MSA of a protein family allows to perform meaningful statistical analysis. In particular, we might ask ourselves which are the relevant biological information that is possible to extract from the statistical features of an MSA. We will specifically focus on two quantities: the single and two-site *frequencies* $f_i(a)$ and $f_{ij}(a, b)$:

$$\begin{cases} f_i(a) = \frac{1}{M} \sum_{m=1}^{M} \delta(\sigma_i, a)\,, \\ f_{ij}(a, b) = \frac{1}{M} \sum_{m=1}^{M} \delta(\sigma_i, a)\, \delta(\sigma_j, b). \end{cases} \tag{1.1}$$

In Eq. (1.1) we introduced the Kronecker delta function:

$$\delta(a, b) = \begin{cases} 1 & a = b, \\ 0 & a \neq b. \end{cases} \tag{1.2}$$

The single site frequency $f_i(a)$ is the normalized count of how many times amino acid a appears in column i, whereas the two-site frequency $f_{ij}(a, b)$ is the normalized occurrence of both amino acids a and b at column i and j. They are related to two important biological concepts: **conservation** and **coevolution**. If a residue has a key

role for the function the protein has to carry out, e.g. if it is an active site, one expects the $f_i(a)$ to be polarized only on the amino acids suitable for the associated task. Consequently, in that position one expects one or few amino acids to be highly conserved across sequences. The two-site frequencies, are instead able to unveil correlation between pairs of residues. If two sites are independent, one expects the joint frequency to factorize $f_{ij}(a,b) = f_i(a)f_j(b)$. If it is not the case, this might be an indicator of interaction between the two residues. Indeed, it is known that *causation* generates correlation. The converse however is not true, because correlations can be spurious, that is, they might arise from mediated indirect interactions. If two residues are directly interacting, they are said to be *co-evolving*. This may happen either if the two sites are collectively crucial to the protein functionality, or if they are in spatial proximity in the three dimensional structure. In this scenario, if a mutation alters one of the two interacting amino acids, the function or even the folding capability of the protein can be damaged. Then, the contact (or the correct functionality) can be restored in two ways. Either the mutated site gets back to the original amino acid, or a *compensatory* mutation happens for the other one. This process leaves a footprint as correlations in the MSA. However, as previously mentioned, correlations might not be a good proxy for interaction, and so statistical methods that are able to disentangle direct and indirect ones are necessary. Another common expression to refer to interacting effects among residues is *epistasis*. An epistatic model should at least include pairwise interaction, but higher order contributions are also possible. However, the peculiar feature of epistatic models is that they are global, or alternatively, context dependent. This ultimately indicates that the various protein sites are not independent.

1.2.3 Hidden Markov Models

MSA's are the fundamental data structure on which statistical methods for analyzing protein sequences are based. As previously mentioned, framing homologous sequences in a common alignment is not straightforward, for the evolution process might alter the sequence length with respect to the ancestor. Profile Hidden Markov Models (HMM) [41, 40] are a common statistical tool employed to align protein sequences belonging to the same family. Furthermore, they can be used to determine wether a specific sequence belongs or not to a family, or to find a matching family for a sequence whose membership is not known.

The word *profile* preceding HMM, indicates the fact that only single site frequencies are used to build the statistical model, which is consequently site-independent, i.e. neglecting correlations between different residues. HMM are based on a generalization of Markov chains, in which there is a distinction between emitted symbols and the state of the chain, namely, it is not possible to reconstruct the state that has emitted a symbol by mere observation of the latter. Consequently, a HMM is characterized by two sets of probabilities: transition and emission ones. If we label with k a possible state of the chain, $a_k(l)$ would be the transition probability to state l. On the other hand, $e_k(a)$ would

be the emission probability of the symbol a from the state k.

In the case of MSA, a representation of the employed HMM as a directed graph is reported in Fig. 1.4. We can identify three type of states: match M_j, insertion I_j and deletion D_j. The index j is associated to residues in the reference MSA. This means that when scanning an unaligned sequence, different sites can be assigned to the same *consensus* residue as either insertion or deletion following a matched state. Moreover, match and insert states are the only one to which emission probabilities are assigned, deletion one being silent, in the sense that they can just produce the gap symbol. It is worth mentioning that insert states can go into themselves, allowing for multiple insertions. Moreover, whereas these self-transitions are position independent, the same is not true for transition between deletion states. Two special match states exist, the *begin* and *end* one, that are used to indicate the starting and ending point of the sequence.

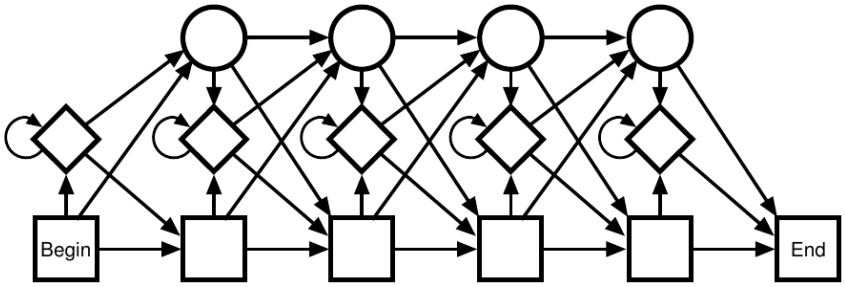


Figure 1.4: Representation of a HMM as a directed graph made of three kind of states: match (squares), insert (diamonds), deletion (circles). Figure taken from [40].

In order to learn the statistical features of a protein family, HMM's parameters are inferred over a seed of manually curated alignments. Once the parameters are fixed, one can look for homologous sequences so to build up a larger alignment. However, when novel unaligned sequences are taken into account, the state series is not known, as only the emitted symbols are available. To reconstruct the states sequence, it is possible to use the *Viterbi* algorithm, providing the most probable path in the state space. This might be useful also the compute the probability of a sequence given the model, defined as:

$$P(\mathbf{S}|\boldsymbol{\theta}) = \sum_{\pi} P(\mathbf{S}|\boldsymbol{\pi}, \boldsymbol{\theta}) P(\pi|\boldsymbol{\theta}). \tag{1.3}$$

Eq. (1.3) can indeed be estimated by retaining only the most probable path $\boldsymbol{\pi}^* = \text{argmax}_{\boldsymbol{\pi}} P(\mathbf{S}, \boldsymbol{\pi})$ in the summation. Alternatively, the so called forward algorithm can be used to directly estimate the marginal $P(\mathbf{S}|\boldsymbol{\theta})$. Once this probability is known, it can be used to compute log-odds ratios with respect to background distribution, in order to estimate how likely is for the sequence to belong to the family.

The HMMer suite [50] is the most common tool used to both infer HMM models from seed alignments and to create MSA from unaligned sequences once the model is inferred. MSA present on the PFAM database are in fact constructed via this tool. Although profile models are typically employed to construct protein family alignments, a recent work [111] built up a statistical modeling that was able to include also epistatic interactions among residues.

1.3 Experimental evolution

This thesis is mainly focused on the development of unsupervised inference methods on protein sequence data. Specifically we are interested in the case in which such data are produced by laboratory experiments of protein evolution. The natural *Darwinian* evolution process [31] takes place on incredibly long time scales, and it is driven by two fundamental mechanisms:

- **Mutagenesis**: it is the process that introduces diversity by modifying the genome of an organism. Point mutations alter just a single base of the genome, whereas multiple mutations modify multiple bases at one time. Since the genetic code is degenerate, mutations can be synonymous, i.e. the coded protein remains unaltered, or non-synonymous when the protein sequence is actually changed.
- **Selection**: it acts on the phenotype, typically at the amino acid level, selecting only the organisms possessing mutations that provide adequate traits for the environment they live in. In this perspective, only mutations that are neutral or beneficial with respect to the specific selective pressure are retained during the course of evolution.

For multicellular organisms, only mutations taking place in the so called *germline* can be transmitted to the offspring. They coincide with sperm and eggs cells, and are differentiated from somatic cells [114]. This distinction is now considered to be less sharp, and a germline can be thought of as the lineage of cells that has been transmitted among individuals along the course of evolution from the *last universal common ancestor* (LUCA) [173], which is the last progenitor of all organisms present nowadays on earth.

Simpler unicellular organisms such as bacteria do not need to transmit genetic information through sexual reproduction, for they reproduce via binary fission, a process in which the cell divides in two identical copies. Moreover, bacteria can also possess additional genetic material called plasmid. This extrachromosomal DNA molecule can be directly transmitted between individuals via a process called *conjugation*. This is of outmost importance, because it allows bacteria to rapidly exchange genetic information, a feature that is particularly relevant in the context of antibiotic resistance development. Furthermore, plasmids can be used as cloning-vectors, a property that will come out again when describing experimental evolution.

Even if they are not properly considered as living organisms, viruses provide another example in which the genetic information can vary rapidly over time. A striking example is provided by Sars-Cov-2, that in the last years showed the emergence of multiple strains with increasing transmissivity [178, 32], binding to human ACE-2 [3, 95] and antibody escape [74, 61, 123], as a consequence of mutations and selective pressure generated by interaction with humans.

In this work, we will be particularly interested to the study of evolution at the protein level. Our fundamental purpose is the determination of the *fitness landscape*, which is a map between the sequence space and the fitness score, or in other words, the functionality of the sequence with respect to a specific selection mechanism. For instance, in a binding experiment, fitness can be identified with the capability of the sequence to bind to the target. On the other hand, in an experiment probing bacterial antibiotic resistance, fitness is defined as the capability of the bacterium equipped with a specific protein sequence to generate copies of itself in an antibiotic enriched environment.

In the following, we will describe two specific kind of experiments for protein evolution: Deep Mutational Scanning (DMS) and Directed Evolution (DE). The last kind of experiments is also known as Genetic Drift (GD). Both types of experiments have the purpose to explore the sequence space whilst probing for a specific selection mechanism, and the main difference between the two lies in how diversity among sequences is introduced.

1.3.1 Deep Mutational Scanning

Deep mutational scanning (DMS) represents a prominent example of high-throughput screening experiments [54, 55]. The main purpose of these experiments is to assess the effect of mutations on an original protein named **wild-type**, from which an initial combinatorial library is generated. Such library systematically contains all the single and possibly part of the multiple mutations, and in some cases it is even possible to probe the whole mutational space. The presence of multiple mutations is particularly relevant for pointing-out epistatic effects.

Mutational effects are quantified by means of a genotype-to-phenotype platform, probing a specific functional feature. Examples of such platforms are cell-based assays [71, 174, 170] and phage display experiments [5, 6, 18].

In cell-based experiments the genetic information is often encoded in a plasmid, i.e. a ring of extrachromosomal genetic material, which is plug into a living organism such as bacterial or yeast cells. An example of a cell-based DMS experiment is the study of the fitness landscape of TEM-1 β-lactamase [51, 81]. In these experiments, the genetic information is plug into bacterial cells, which represent the natural host environment, since β-lactamases provide them with antibiotic resistance. Then bacteria are put into an antibiotic enriched environment, so to test the functionality of the different variants. In this context, fitness can be estimated a *minimum inhibitory concentration* (MIC) or more involved functions of the antibiotic concentration.

If one aims at studying viral proteins, these can be directly displayed on the surface of yeast cells, where they can be subsequently probed for a phenotypic character, e.g. binding affinity onto a target [157]. A possible strategy to extract quantitative phenotypic information from this yeast-display platform, is to rely on flow-cytometry techniques such as *fluorescence-activated cell sorting* (FACS). The different variants are indeed fluorescently labeled, and FACS proceeds by sorting each yeast cell into a bin according to its fluorescence. Then, the variants in each bin are sequenced, and the phenotypic measurement is computed as an average over the bins of the fluorescence distribution.

A drawback of cell-based scanning approaches, is that they often require to couple the quantitative trait that is aimed to be selected with cell growth [136], limiting the total number of phenotypes that can be actually tested.

In the case of phage display experiments, the genetic information is plug into a vector phage virus, which is able to display the protein of interest on its surface as a result of fusion with its coat protein. A prominent example of phage display is phage binding experiment, in which the displayed protein is probed for binding onto an immobilized target. In Fig 1.5 an example of a phage binding pipeline is reported.

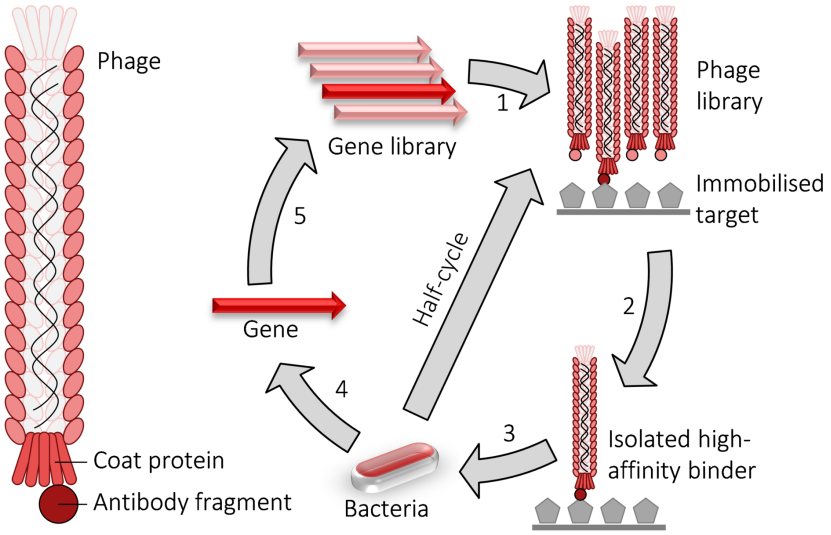


Figure 1.5: A sketch representation of a DMS experiment. Genes coding for the proteins of interest are inserted into the phages in order to be displayed and screened against binding on immobilized targets. Then, non-binders are eliminated from the experiment through a washing procedure, isolating the high-affinity ones. Afterwards, the surviving phages are inserted into bacteria to be multiplied thanks to bacterial growth. Finally, the amplified genes are extracted and the cycle can be repeated again. Image taken from https://en.wikipedia.org/wiki/Phage_display.

For all the kinds of platforms, a DMS experiment is characterized by two fundamental mechanisms: selection and amplification. The first allows to isolate the most

fitting variants, whereas the second generate libraries with a very large number of mutant sequences (up to order 1 million). Indeed, modern technologies allow DMS to be high-throughput experiments, so that a very large number of non-unique clones can be screened at the same time. These two steps can be either performed once or can be repeated over multiple rounds.

It is worth underlying that the variability in a DMS experiment is entirely introduced at the beginning, with the generation of the combinatorial library. If the experiment is high-throughput and the sequencing is realized in a not severe under-sampling regime, it is possible to estimate the fitness of a variant as the ratio between the associated abundances at neighboring rounds. These quantities are usually referred to as enrichment or depletion ratios. More precisely, it is common choice to rely on log-selectivities to estimate a sequence fitness:

$$\Theta_m = \frac{1}{t_f - 1} \sum_{t=t_1}^{t_f - 1} \log \frac{N_m^{t+1}}{N_m^t}, \tag{1.4}$$

where m identifies a specific sequence, N_m^t is the number of copies of that sequence at round $t = \{t_1, \dots, t_f\}$. A more refined definition of log-selectivity that include the effect of amplification and noise fluctuation will be given in Sec. 4.4. Therein, we will also describe a suitable statistical modeling for this kind of experiments.

1.3.2 Directed Evolution

Directed Evolution (DE) [1] experiments share several features with DMS. They also start from a unique original wild-type sequence, from which diversity is generated with the introduction of random mutations. Moreover, the generated sequences are subjected to a selective pressure, so to retain only the functional ones. The main difference lies in how and when mutations are introduced. If typically in DMS a combinatorial library containing all the single and eventually higher order mutations is used, in a DE experiment variants are introduced randomly, by means of *error-prone polymerase chain reaction* (epPCR) [28]. The mutation rate can be tuned by choosing the number of cycles of epPCR to be performed. Furthermore, mutagenesis is not solely carried out at the beginning of the experiment, but it is repeated before every round of selection. Consequently, it is reasonable to consider the mutation and the selection step (eventually together with amplification) as the fundamental building block of the experiment, and the wild-type sequence as the initial library. It is then evident how diversity is introduced in the entire course of the experiment, as opposed to what happens in DMS. This

[1] Depending on the specific intensity of the selective pressure exerted on protein variants, experiments with multiple introduction of mutations can also be referred to as Genetic Drift (GD). In particular, a GD experiment is meant to be realized in a relatively weak selective pressure regime, so that the constraint on sequence functionality is still imposed, but allowing for a broader exploration of the sequence space.

allows for a broader exploration of the sequence space around the wild-type, but on the other hand makes attempting to develop a dynamical modeling more difficult.

As for DMS, DE experiments require a genotype-to-phenotype platform, such as cell-based or phage display. In the recent years, cell-based approach for DE has proved to be particularly effective to study experimental evolution of β-lactamase protein in E. Coli [44, 160]. These experiments can be considered as being partially *in-vivo*, because the proteins are put in its natural living organism, though bacteria are subsequently grown in a culture medium. Such a strategy turned out to reproduce data sharing some features with correspondent homologous family, also allowing to partially assess contact prediction.

However, since DE provides only a local exploration of the sequence space around the wild-type, it would be in principle more suited to infer local fitness landscapes rather than global properties such as structural constraints. We will come back to this topic by giving a thorough discussion in Sec. 3.4. Moreover, DE experiments provide an interesting benchmark to test the relationship between fitness-landscape and molecular evolution [17, 130, 179] and for designing novel optimally functioning proteins [135, 177].

In Fig. 1.6 a pictorial representation of a DE experiment realized on a cell-based platform is reported.

1.4 Immune system: antibody evolution

Since in Ch. 4 we will deal with experimental data related to antibody sequences, we give here a brief and general description of the fundamental elements and concepts regarding the immune system.

The immune system is an ensemble of different molecules and sophisticated mechanisms allowing vertebrate organisms to protect themselves from external pathogens. One of its fundamental building blocks is represented by antibodies, on which we will focus in the present discussion. Antibodies are biomolecules made by two pairs of two kind of proteins, referred to as heavy and light chain. In Fig. 1.7, we show an example of an IgG antibody. The antibody molecule can be divided in two subparts: a variable region called Fab, which is responsible for binding onto antigens, and a constant region referred to as Fc, allowing the antibody to bind to B-cells or other kind of molecules such as macrophages. Indeed, antibodies are actually produced by B-cells, which are in turn generated in the marrow. Once B-cells have matured, they expose B-cell receptors (BCR), i.e. the antibodies, on their surface.

The main task of antibodies is to recognize antigens, which can be seen as signalers of the pathogen presence. An antigen can coincide for instance with protein fragments of the pathogen, as it is the case for surface proteins of viruses, e.g. the spike protein of Sars-Cov-2. Once the antibody binds to the antigen, it acts as a tag for the organism, signaling that the molecule it has bound needs to be destroyed. Alternatively, it might

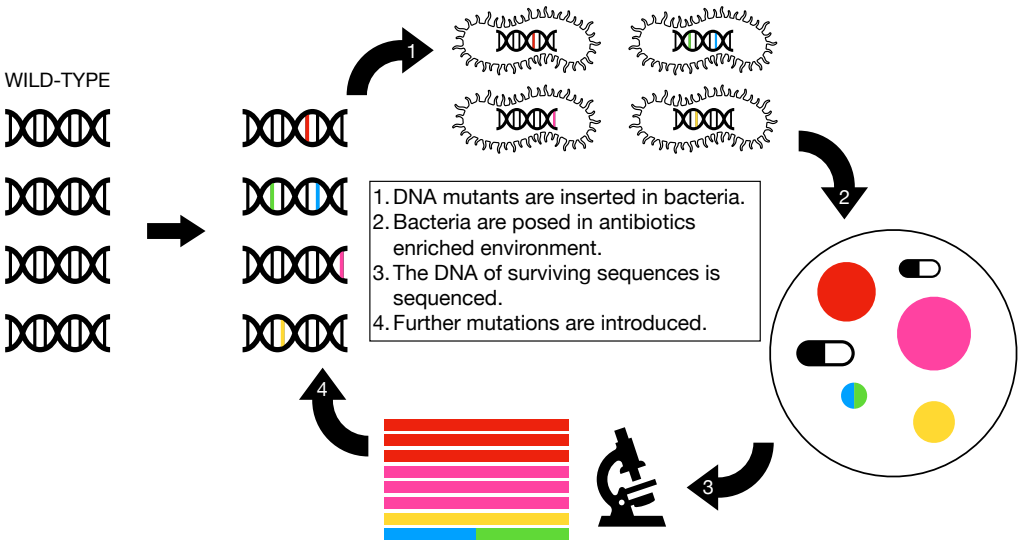


Figure 1.6: Pictorial representation of a DE experiment based on a cell platform, specifically bacterial cells. The initial configuration is a library made of identical wild type DNA sequences. Then, random mutations are introduced, so to generate brand-new variants. Such mutant genomes are subsequently inserted into the bacterial cells, where they will encode for the corresponding protein sequences. The bacteria are then posed in an antibiotic enriched environment, where their reproductive capacity depends on the functionality of the encoded amino acidic sequence. The surviving bacteria are then extracted, and their DNA is sequenced. Finally the process starts again introducing further mutations among the extracted DNA sequences.

avoid the pathogen to penetrate into the host cells, by binding onto its proteins which are responsible for entering the cells.

It is possible to distinguish between two different kinds of response of the immune system, an *innate* and an *adaptive* one. The first is a consequence of the so called naïve immune repertoire, i.e. the collection of antibody which is produced from the genetically inherited materials. The possible ensemble of antibodies of such repertoire is already huge, thanks to a phenomenon known as genetic recombination [112, 79], concerning how antibody chains are generated. In particular, the genetic material encoding for the heavy chains possesses a modular organization defined by four units: V, D, J and C. Each of these units can be found in a certain number of different copies, so that several modules can be combined together in many different ways. On top of this, further diversity is introduced by adding or deleting bases at the junction between the units. The light chains are generated in a similar modular way, even if the D segment is lacking.

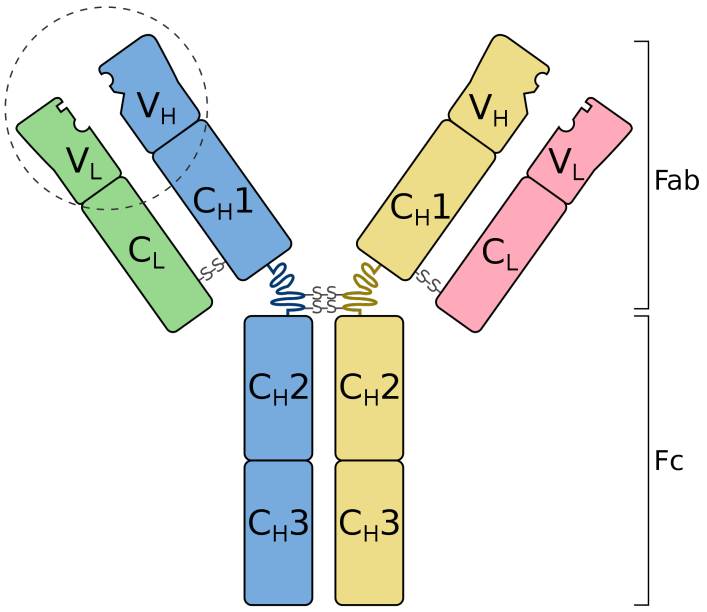


Figure 1.7: Schematic representation of a monomeric antibody. The Fab region coincides with the upper part, defined by the union between light and heavy chains. Therein, the binding active site is highlighted with a circle. The variable regions $V_{H,L}$ are found in the Fab part of the antibody. On the other hand, the Fc region is defined by constant parts of the heavy chain. Image taken from https://en.wikipedia.org/wiki/Antibody.

Even if the potential diversity of the naïve repertoire is already very large, it is still not able to protect the organism against never encountered pathogens, which are dealt with by the adaptive immune system. The general idea is that, when a novel pathogen enters the organism, new antibodies which are tailored for the specific antigen need to be produced. This process is achieved by *affinity maturation*, which is composed by two mechanisms: somatic hypermutation (SHM) [37] and clonal selection. Affinity maturation takes place in the germinal centers [169], which are found in lymph nodes. Germinal centers can be divided in two regions, namely a light and a dark zone. In the latter, B-cell that migrated into the germinal centers undergo SHM, increasing their genetic diversity in their variable regions known as *complementarity determining region* (CDR). Then, they move to the light zone, where the different antibodies compete with respect to antigen binding, so that only high-affinity binders are isolated. SHM and clonal selection can be repeated over multiple rounds, eventually leading to the production of *plasma* B-cells and *memory* B-cells. The former cannot switch their immunoglobulin type, and are highly specific for a single antigen. Thus, they serve as an immediate protection again re-exposition to the antigen. On the other hand, memory B-cells can undergo again SHM, so that they can also protect against mutants of the

original antigen [126].

In light of this discussion, we can interpret organism exposition to antigens as an in-vivo experiment, in which the initial library is given by the naïve repertoire and the antibody repertoire after the exposition coincides with the sequence distribution after the action of the selection process. In Ch. 4 we will use a novel inference method to characterize the statistical difference between the unimmunized/immunized antibody repertoires of mice.

Chapter 2

Statistical analysis of protein sequence

In this chapter we will give an overview of the statistical methods that have been developed to study protein sequence data, focusing in particular on *Direct Coupling Analysis* (DCA) methods. DCA methods have proved themselves to be able to tackle a variety of different tasks, among which we can mention contact prediction, sampling in sequence space, generation of new sequences, mutational effects prediction, family assignment and fitness landscape reconstruction. In Sec. 2.1 we described the *maximum-entropy* principle as a valuable tool to define statistical models. In Sec. 2.2 we introduce the *generalized Potts model* (GPM), which is the basis of the DCA approach. Afterwards, Sec.2.3 gives an overview of possible strategies to infer the GPM. Finally, in Sec. 2.4 we give an overview of the possible application of the GPM.

2.1 Maximum-entropy principle

In this section we discuss the *maximum-entropy* principle [82], which provide us a solid theoretical background for choosing an appropriate statistical model. This approach proved to be successful for a wide range of biological applications [143, 107, 14]. We first need to introduce the Shannon entropy of a probability (density):

$$\begin{cases} S(P) = -\sum_{y \in Y} P(y) \log P(y), \\ S[p] = -\int_{-\infty}^{+\infty} \mathrm{d}x\, p(x) \log p(x). \end{cases} \tag{2.1}$$

The two different definitions correspond to whether the random variable is discrete or continuous. In the first case, the possible values y are defined over a discrete alphabet, that can be mapped over the integers $Y = \{1, \dots, N\}$, N being the total number of symbols. If the variables are discrete, it is possible to associate a probability to each

outcome $P(y)$. On the other hand, if the variable is defined over a continuous domain, the Shannon entropy is defined as a functional of the probability density function $p(x)$.

In this thesis, we will mainly deal with discrete random variables, for which the Shannon entropy fulfills the inequality $0 \leq S(P) \leq \log N$, that is, it is positive definite and bounded above by the logarithm of the number of symbols. These two limiting values coincide with two specific choices of the probability, i.e. the deterministic and the uniform one. Only one possible outcome is allowed for the former $P(y) = \delta(y, n)$, whereas all the outcomes have the same probability $P(y) = 1/N$ for the latter. Indeed, the entropy can be thought of as missing information or amount of surprise. If the process is deterministic, you already possess all the information and you are never surprised by the outcome of the drawing process. On the other hand, the uniform distribution is the one characterized by the maximum unpredictability, and consequently surprise.

The maximum-entropy principle provides a recipe to define a probability function $P(y)$ given a set of constraints, which are observables over the data. In doing so, maximizing the entropy entails having the least possible information and assumptions made about how the data should be distributed. Analogously, this coincides with taking the least constrained distribution, and reflects our ignorance about the underlying mechanism that generated the data. To formalize this idea, imagine we have a dataset of M discrete outcomes $\hat{Y} = \{\hat{y}_1, \dots, \hat{y}_M\}$. Associated to it, we have a set of $K + 1$ constraints $\mathscr{C} = \{c_0, \dots, c_K\}$, defined for a set of functions $\{f_0(y), \dots, f_K(y)\}$. The constraints read:

$$\langle f_k \rangle = \sum_{y \in Y} f_k(y) P(y) = \bar{f}_k = \frac{1}{M} \sum_{m=1}^{M} f_k(\hat{y}_m), \tag{2.2}$$

for each $k = 1, \dots, K$. In practice, we impose that each average value over the dataset of the function f's must be equal to the corresponding ensemble average over P. In order to find the probability function P, we need to maximize the corresponding Shannon entropy given the set of constraints, and to do so we rely on the Lagrange multipliers formalism, defining the objective:

$$\mathscr{F}(P, \boldsymbol{\mu}) = S(P) - \sum_{k=0}^{K} \mu_k \left(\sum_{y \in Y} f_k(y) P(y) - \bar{f}_k \right). \tag{2.3}$$

In order to maximize Eq. (2.3) we need to compute the derivative with respect to $P(x)$:

$$\begin{aligned} \frac{\partial \mathscr{F}(P, \boldsymbol{\mu})}{\partial P(x)} &= -\sum_{y \in Y} \delta(x, y) \log P(y) - \sum_{y \in Y} \delta(x, y) - \sum_{k=0}^{K} \mu_k \sum_{y \in Y} f_k(y) \delta(x, y) \\ &= -\log P(x) - 1 - \sum_{k=0}^{K} \mu_k f_k(x) = 0 \end{aligned} \tag{2.4}$$

where we used the identity $\partial P(y)/\partial P(x) = \delta(x, y)$. The normalization condition $\sum_{y\in Y} P(y) = 1$ is a constraint that must be always imposed to probability functions. We can choose the zeroth constraint c_0 to carry it out, calling the corresponding Lagrangian multiplier $\alpha = \mu_0$. We can then rewrite Eq. (2.4) as:

$$\begin{aligned} -\log P(x) - 1 + \alpha - \sum_{k=1}^{K} \mu_k f_k(x) = 0 \\ P(x) = \mathrm{e}^{\alpha-1}\mathrm{e}^{-\sum_{k=1}^{K}\mu_k f_k(x)}, \end{aligned} \tag{2.5}$$

and determine α by enforcing normalization $\sum_{y\in Y} P(y) = 1 = \mathrm{e}^{\alpha-1}\sum_{y\in Y}\mathrm{e}^{-\sum_{k=1}^{K}\mu_k f_k(y)}$ $\Rightarrow \mathrm{e}^{1-\alpha} = \sum_{y\in Y}\mathrm{e}^{-\sum_{k=1}^{K}\mu_k f_k(y)}$, so that the maximum-entropy functional form for the probability reads:

$$P(y) = \frac{\mathrm{e}^{-\sum_{k=1}^{K}\mu_k f_k(y)}}{\sum_{y'\in Y}\mathrm{e}^{-\sum_{k=1}^{K}\mu_k f_k(y')}}. \tag{2.6}$$

Each μ_k is meant to impose the constraint associated to the function $f_k(y)$. If the variables y are numerical, a common choice for f is the identity function, i.e. $f(y) = y$, so that the constraint becomes the mean of the variable. If the y's are instead categorical variables, which is the case for protein sequences, it is usual to impose the single and two-sites frequencies as constraints, as they were defined in Eq. (1.1).

The probability function provided by the maximum-entropy principle (Eq. (2.6)) can be thought as a Boltzmann distribution, which describes the statistical weight of configurations in a canonical ensemble. This is especially evident if we take the average value of the energy $E(y)$ as our sole constraint, indeed $P(y) \propto \exp[-\mu E(y)]$, which is exactly in the canonical form if we interpret the Lagrangian multiplier μ as the inverse temperature of the system $\mu = \beta = 1/k_B T$.

2.2 Generalized Potts Model

In the previous section we introduced the maximum-entropy principle as a tool to define reasonable statistical models. Here we derive a global epistatic model for protein sequences, following the maximum-entropy recipe, which is usually referred to as the *Generalized Potts Model* (GPM). We mentioned that the constraints that we need to impose are the one and two-site frequencies. In principle, one could impose statistical constraints of increasing order, as for instance the three-sites statistics. However, the higher is the order, the more data are needed in order to fix the model parameters with sufficient precision.

For clarity, let's write explicitly the expression of the constraints for the MSA:

$$\begin{cases} f_i(a) = P_i(a) = \sum_{\{\mathbf{S}\}} P(\mathbf{S})\delta(\sigma_i, a) = \langle \delta(\sigma_i, a) \rangle, \\ f_{ij}(a,b) = P_{ij}(a,b) = \sum_{\{\mathbf{S}\}} P(\mathbf{S})\delta(\sigma_i, a)\delta(\sigma_j, b) = \langle \delta(\sigma_i, a)\delta(\sigma_j, b) \rangle, \end{cases} \tag{2.7}$$

where $P(\mathbf{S})$ is the probability of observing a sequence $\mathbf{S}$, and represents the function we want to determine. The functional to be optimized given the constraints in Eq. (2.7) reads:

$$\begin{aligned} \mathcal{F}[P, \mathbf{h}, \mathbf{J}] = -\sum_{\{\mathbf{S}\}} \Bigg\{ P(\mathbf{S}) \log P(\mathbf{S}) + \sum_{i=1}^{L} h_i(\sigma_i) \left[P_i(\sigma_i) - f_i(\sigma_i) \right] \\ + \sum_{i=1}^{L-1} \sum_{j=i+1}^{L} J_{ij}(\sigma_i, \sigma_j) \left[P_{ij}(\sigma_i, \sigma_j) - f_{ij}(\sigma_i, \sigma_j) \right] + \left[P(\mathbf{S}) - 1 \right] \Bigg\}. \end{aligned} \tag{2.8}$$

We introduced a novel notation for the Lagrangian multipliers, which is more suitable for the protein sequence modeling. In order to enforce the single site frequencies we introduced a set of *fields* $\mathbf{h} = \{\mathbf{h}_1, \dots, \mathbf{h}_L\}$. Each $\mathbf{h}_i$ is a q components vector, resulting in a number of $L \times q$ fields parameters, with L the length of the protein and q the number of amino acids (eventually including the gap symbol). To impose the two-site frequencies instead, we introduce the set of *couplings* $\mathbf{J}$, yielding a number of parameters $\binom{L}{2} \times q \times q$. Actually, the overall number of parameters is not $\binom{L}{2}q^2 + Lq$ but it is smaller, since the normalization and two site frequencies constraints automatically enforce some of the the single site frequencies, so that the actual number of parameters is $\binom{L}{2}(q-1)^2 + L(q-1)$.

The sought probability function P is determined by the stationary point of Eq. (2.8):

$$\frac{\partial \mathcal{F}}{\partial P(\mathbf{S}')} = -\log P(\mathbf{S}') - 1 - \sum_{i=1}^{L} h_i(\sigma_i') - \sum_{i=1}^{L-1} \sum_{j=i+1}^{L} J_{ij}(\sigma_i', \sigma_j') = 0, \tag{2.9}$$

from which we derive:

$$P(\mathbf{S}) = \frac{1}{Z} \exp\left\{ \sum_{i=1}^{L} h_i(\sigma_i) + \sum_{i=1}^{L-1} \sum_{j=i+1}^{L} J_{ij}(\sigma_i, \sigma_j) \right\}. \tag{2.10}$$

The normalization Z is also referred to as the partition function of the system, and it contains all the relevant statistical properties of the system. It is defined as:

$$Z = \sum_{\{\mathbf{S}\}} \exp\left\{ \sum_{i=1}^{L} h_i(\sigma_i) + \sum_{i=1}^{L-1} \sum_{j=i+1}^{L} J_{ij}(\sigma_i, \sigma_j) \right\}, \tag{2.11}$$

and it is a summation over q^L possible configuration, a computation which is already unfeasible for relatively short protein (L $\sim$ 50). Interestingly, the statistical model defined in Eq. (2.10) can be interpreted as an equilibrium Boltzmann probability function, that is, $P(\mathbf{S}) = \exp\left(-H(\mathbf{S})\right)/Z$ in which we define an Hamiltonian energy function:

$$H(\mathbf{S}) = -\sum_{i=1}^{L} h_i(\sigma_i) - \sum_{i=1}^{L-1} \sum_{j=i+1}^{L} J_{ij}(\sigma_i, \sigma_j). \tag{2.12}$$

Such an energy function is the aforementioned GPM, that can be interpreted as a generalization of the standard Ising model, which corresponds to the special case $q = 2$. Let's comment the meaning of the parameters appearing in Eq. (2.12). From the perspective of the maximum principle, they are thought only as the free parameters that allow the probability function in Eq. (2.10) to reproduce the one and two sites frequencies. However, one could also write down Eq. (2.12) from scratch, relying on the *physical* interpretation of the parameters appearing therein. In this perspective, the maximum-entropy principle provide a principled way to justify the choice of such a model.

Furthermore, Eq. (2.12) can be thought of as a minimal model allowing to take into account global epistatic effects, thanks to the couplings $\mathbf{J}$. Indeed, a model in which these parameters are absent is also called an *independent site* or profile model, for the probability distribution factorizes over the different sites. This is the case of the HMM's traditionally used to align protein sequences. From a *microscopic* point of view, the fields are local functions acting on the single residues: the higher is the value of $h_i(a)$, the more conserved amino acid a would be in position i. On the other hand, the coupling parameters are meant to model interactions between pairs of residues, and in particular for each pair of sites $i-j$, it is possible to define a $q \times q$ matrix J_{ij}, having an entry for each possible combination of two amino acids. The "larger" these entries are, the stronger is the interaction between the residues, as it might result from the fact that the two sites are coevolving. The interesting feature of the coupling is that they quantify a direct interaction between residues, from which the name *Direct Coupling Analysis* (DCA), to indicate statistical models as in Eq. (2.10). In this way, we are able to take into account correlations between different positions of an MSA, at the same disentangling the spurious ones. In Sec. 2.4 we will see how to precisely employ the couplings in order to quantify the interaction between two residues, an information that is fundamental for the contact prediction problem.

2.2.1 Gauge invariance

To conclude this section about the GPM, we discuss an interesting property known as gauge invariance, for which the GPM is invariant under a class of gauge transformations. These transformations are characterized by the following property. If we identify with $\boldsymbol{\theta}$ the set of parameters defining the Potts Hamiltonian, i.e. the fields and couplings, a gauge transformation $f(\boldsymbol{\theta})$ turns the parameters into $\boldsymbol{\theta}' = f(\boldsymbol{\theta})$ so that $H_{\boldsymbol{\theta}'}(\mathbf{S}) - H_{\boldsymbol{\theta}}(\mathbf{S}) = c$, with c constant. Since adding a constant to Eq. (2.12) leaves the probability in Eq. (2.10) unchanged, all the models parametrized by $\boldsymbol{\theta}$ and $\boldsymbol{\theta}'$ are equivalent, because they produce the same single and two sites marginals. The explicit form of a gauge transformation is:

$$J'_{ij}(a,b) = J_{ij}(a,b) + V_{ij}(a) + U_{ij}(b)$$
$$h'_i(a) = h_i(a) - \sum_{j<i} U_{ji}(a) - \sum_{j>i} V_{ij}(a) + C_i \tag{2.13}$$

and it can be explicitly checked that it fulfills the required properties by direct substitution in Eq. (2.12). The redundancy of possible parameters defining the same probability function can be ascribed to the fact that conditions in Eq. (2.7), used to impose the constraints and defining the model structure, are in fact not independent, as mentioned in Sec. 2.1.

There are several possible ways to fix the gauge of the model. Here we present two possibilities: the *lattice gauge* [108] and the *zero-sum-gauge* [43]. The first is the one that is employed in the derivation of the mean field (MF) approximation of the GPM (see Sec. 2.3.1), and it entails:

$$\forall a, b \in \mathcal{Q}, \quad J_{ij}(a,q) = J_{ij}(q,b) = h_i(q) = 0. \tag{2.14}$$

The zero-sum-gauge is another common choice in the inference of the GPM. In order to achieve it, the gauging functions U, V and C are to be chosen as:

$$V_{ij}(a) = -\frac{1}{q}\sum_{b=1}^{q} J_{ij}(a,b) + \frac{1}{2q^2}\sum_{a,b=1}^{q} J_{ij}(a,b),$$
$$U_{ij}(b) = -\frac{1}{q}\sum_{a=1}^{q} J_{ij}(a,b) + \frac{1}{2q^2}\sum_{a,b=1}^{q} J_{ij}(a,b),$$
$$C_i = -\frac{1}{q}\sum_{a=1}^{q} h_i(a), \tag{2.15}$$

which leads to the following parameters transformation:

$$J_{ij}(a,b) \to J_{ij}(a,b) - \frac{1}{q}\sum_{b=1}^{q} J_{ij}(a,b) - \frac{1}{q}\sum_{a=1}^{q} J_{ij}(a,b) + \frac{1}{q^2}\sum_{a,b=1}^{q} J_{ij}(a,b),$$
$$h_i(a) \to h_i(a) - \frac{1}{q}\sum_{b=1}^{q} h_i(b) + \sum_{j\neq i}\left[\frac{1}{q}\sum_{b=1}^{q} J_{ij}(a,b) - \frac{1}{q^2}\sum_{a,b=1}^{q} J_{ij}(a,b)\right]. \tag{2.16}$$

With this choice the parameters also fulfill the condition:

$$\sum_{a=1}^{q} J_{ij}(a,b) = \sum_{b=1}^{q} J_{ij}(a,b) = \sum_{a=1}^{q} h_i(a) = 0. \tag{2.17}$$

This implies that the zero-sum-gauge is the one that minimizes the Frobenius norm of the coupling matrix, a property which turns out to be useful in the context of contact prediction.

2.3 Inference of the Generalized Potts Model

We have derived a functional form for the probability of observing a sequence in an MSA, theoretically justified by the maximum-entropy principle. However, we have not yet stated how the inference should be performed, that is, how to fix the value of the parameters $\mathbf{J}$ and $\mathbf{h}$. In principle, they should be chosen in such a way that the constraints in Eq. (2.7) are fulfilled. In a Bayesian framework, we can write down the log-likelihood associated to an MSA $\{\mathbf{S}^{(m)}\}_{m=1}^{M}$:

$$\mathscr{L}(\mathbf{h},\mathbf{J}) = \frac{1}{M}\sum_{m=1}^{M}\log P\left(\mathbf{S}^{(m)}\right) = \frac{1}{M}\sum_{m=1}^{M}\left[\sum_{i=1}^{L} h_i(\sigma_i^{(m)}) + \sum_{i=1}^{L-1}\sum_{j=i+1}^{L} J_{ij}(\sigma_i^{(m)},\sigma_j^{(m)})\right] - \log Z(\mathbf{h},\mathbf{J})\,. \tag{2.18}$$

Maximization of Eq. (2.18) provides the maximum-likelihood estimate of the GPM parameters. However, the computation of the partition function Z is exponential in the length of the protein sequence, since it scales as q^L. Such number becomes enormous even for relatively short protein sequences, e.g. for $L = 20$ we have $q^L = 10^{L\log_{10} q} \simeq 10^{21}$. In the following sections we will present several possible approximation schemes.

If we were able to compute the partition function, we could alternatively solve the inference problem by direct computation of the single and two sites marginals. Indeed:

$$\frac{\partial \log Z}{\partial h_j(a)} = \frac{1}{Z}\sum_{\mathbf{S}}\delta(\sigma_j,a)\exp\left\{\sum_{i=1}^{L} h_i(\sigma_i) + \sum_{i=1}^{L-1}\sum_{j=i+1}^{L} J_{ij}(\sigma_i,\sigma_j)\right\} = P_j(a), \tag{2.19}$$

$$\begin{aligned}\frac{\partial^2 \log Z}{\partial h_j(a)h_k(b)} &= -\frac{1}{Z^2}\left(\sum_{\mathbf{S}}\delta(\sigma_j,a)\mathrm{e}^{-H(\mathbf{S})}\right)\left(\sum_{\mathbf{S}}\delta(\sigma_k,b)\mathrm{e}^{-H(\mathbf{S})}\right) + \frac{1}{Z}\sum_{\mathbf{S}}\delta(\sigma_j,a)\delta(\sigma_k,b)\mathrm{e}^{-H(\mathbf{S})} \\ &= P_{jk}(a,b) - P_j(a)P_k(b) \equiv C_{jk}(a,b),\end{aligned} \tag{2.20}$$

where we introduced the connected correlation function $C_{ij}(a,b)$. Eqs. (2.19) and (2.20), together with the empirical frequencies, contain all the necessary information to solve the problem. Since it will be soon necessary, we point out that we could rewrite Eqs. (2.19) and (2.20) in terms of the *Helmholtz* free energy $F = -\log Z$, i.e. $-\partial F/\partial h_j(a) = P_j(a)$ and $-\partial^2 F/(\partial h_j(a)\partial h_k(b)) = C_{jk}(a,b)$.

2.3.1 Mean Field approximation

The expression *Mean Field* (MF), identifies a broad class of techniques that are used to approximate intractable probability functions [116]. Generally speaking, if the system is made of units labeled by an index i, the MF approximation considers the probability function of the system to be factorized over these units, i.e. $P = \prod_i P_i$. Here, we rather present a generalization of the standard MF approach, which is known in the

literature as the Plefka [124, 65] or *small coupling* expansion. The method is based on the introduction of a perturbative parameter α, so that Eq. (2.12) is modified to:

$$H(\mathbf{S};\alpha) = -\sum_{i=1}^{L} h_i(\sigma_i) - \alpha \sum_{i=1}^{L-1} \sum_{j=i+1}^{L} J_{ij}(\sigma_i, \sigma_j). \tag{2.21}$$

Such parameter is meant to be *small*, that is $\alpha \ll 1$. Moreover, we notice that the original model is recovered for $\alpha = 1$, whereas if $\alpha = 0$ it becomes a site-independent (profile) model. We subsequently need to introduce the *Gibbs potential*, which is defined as the Legendre transform of the *Helmholtz* free energy [162] $F = -\log Z$. Different conventions for such transformation can be chosen, but we will stick to the one which is more common in statistical physics. According to this definition, the Legendre transform turns convex functions into concave ones (rather than preserving convexity as the standard Legendre transform does). The relations connecting the two thermodynamical potentials are:

$$F(\alpha, \mathbf{h}) = \min_{\mathbf{Q}} \left[G(\alpha, \mathbf{Q}) - \sum_{i=1}^{L} \sum_{a=1}^{q-1} h_i(a) Q_i(a) \right], \tag{2.22}$$

$$G(\alpha, \mathbf{Q}) = \sup_{\mathbf{h}} \left[F(\alpha, \mathbf{h}) + \sum_{i=1}^{L} \sum_{a=1}^{q-1} h_i(a) Q_i(a) \right]. \tag{2.23}$$

With the vector $\mathbf{Q}$, we are labeling the single site marginal probabilities, which are the conjugate variables of the fields. If F is a differentiable function, the extremum condition can be rewritten as $\partial_{h_j(a)} [F(\alpha, \mathbf{h}) + \mathbf{h} \cdot \mathbf{Q}] = 0$, from which we recover Eq. (2.19). This condition should be used to express $\mathbf{h}$ as a function of $\mathbf{Q}$, via the inverse function of $\partial F / \partial h_j(a)$. In other words, given a specific value of the single site probability $Q_i(a)$, the compatible value of the field $h_i(a)$ is fixed via the derivative of the free energy F. From this, we can derive the following useful relation:

$$\begin{aligned} \frac{\partial G(\alpha)}{\partial Q_j(b)} &= \frac{\partial}{\partial Q_j(b)} \left[F(\alpha, \mathbf{h}(\mathbf{Q})) + \sum_{i=1}^{L} \sum_{a=1}^{q-1} h_i(a)\,(\mathbf{Q})\, Q_i(a) \right] \\ &= \underbrace{\frac{\partial F(\alpha, \mathbf{h}(\mathbf{Q}))}{\partial h_j(b)}}_{-Q_j(b)} \frac{\partial h_j(b)}{\partial Q_j(b)} + h_j(b) + Q_j(b) \frac{\partial h_j(b)}{\partial Q_j(b)} = h_j(b), \end{aligned} \tag{2.24}$$

and since we have shown earlier that $\partial P_i(a) / \partial h_j(b) = C_{ij}(a, b)$, we also obtain:

$$\frac{\partial h_j(b)}{\partial Q_i(a)} = (C)^{-1}_{ij}(a, b) = \frac{\partial^2 G(\alpha, \mathbf{Q})}{\partial Q_i(a) Q_j(b)}, \tag{2.25}$$

which can be understood as a fluctuation-dissipation relation.

We now want to perform the expansion of the Gibbs potential with respect to the parameter α:

$$G(\alpha) = G(0) + \alpha \frac{\mathrm{d}G(\alpha)}{\mathrm{d}\alpha}\bigg|_{\alpha=0} + \mathcal{O}(\alpha^2). \tag{2.26}$$

In order to perform the expansion, it is handier not to consider directly Eq. (2.23), but rather the not extremized potential $\mathcal{G}(\alpha) = -\log Z(\alpha) + \mathbf{h}\cdot\mathbf{Q}$. At the zeroth order in α, we have $\mathcal{G}(0) = -\log \sum_{\mathbf{S}} \exp^{\sum_{i=1}^{L} h_i(\sigma_i)} + \sum_{i=1}^{L}\sum_{a=1}^{q} h_i(a)Q_i(a)$. Extremization coincides with expressing the fields as a function of the marginals, by relying on $-\partial F/\partial h_j(b) = Q_j(b)$. If we do so we obtain:

$$Q_j(b) = \frac{\mathrm{e}^{h_j(b)}}{\sum_{a=1}^{q}\mathrm{e}^{h_j(a)}}, \tag{2.27}$$

which finally yields:

$$G(0) = \sum_{i=1}^{L}\sum_{a=1}^{q} Q_i(a)\log Q_i(a). \tag{2.28}$$

When computing the derivative with respect to α, it must be recalled that the complete equation for determining the fields is:

$$\begin{aligned} -\frac{\partial F(\alpha,\mathbf{h})}{\partial h_j(b)} &= \frac{\partial}{\partial h_j(b)}\log\left\{\sum_{\mathbf{S}}\exp\left[\alpha\sum_{i=1}^{L-1}\sum_{j=i+1}^{L}J_{ij}(\sigma_i,\sigma_j)+\sum_{i=1}^{L}h_i(\sigma_i)\right]\right\} \\ &= \frac{1}{Z}\sum_{\mathbf{S}}\delta(\sigma_j,b)\exp\left[\alpha\sum_{i=1}^{L-1}\sum_{j=i+1}^{L}J_{ij}(\sigma_i,\sigma_j)+\sum_{i=1}^{L}h_i(\sigma_i)\right] = Q_j(b). \end{aligned} \tag{2.29}$$

Since the field is determined by the inversion of Eq. (2.29), we see that it can be generically expressed as $h_j(b) \equiv h_j(b)\big(\alpha, Q_j(b)\big)$, which explicitly depends on α. The derivative of the pseudo Gibbs potential reads:

$$\begin{aligned} \frac{\partial\mathcal{G}(\alpha)}{\partial\alpha} &= -\frac{\mathrm{d}\log Z(\alpha)}{\mathrm{d}\alpha} + \sum_{i=1}^{L}\sum_{a=1}^{q}\frac{\mathrm{d}h_i(a)}{\mathrm{d}\alpha}Q_i(a) \\ &= \frac{1}{Z(\alpha)}\sum_{\mathbf{S}}\left[\sum_{i=1}^{L-1}\sum_{j=i+1}^{L}J_{ij}(\sigma_i,\sigma_j)+\sum_{i=1}^{L}\frac{\mathrm{d}h_i(\sigma_i)}{\mathrm{d}\alpha}\right]\mathrm{e}^{-H(\alpha)} + \\ &\qquad + \sum_{i=1}^{L}\sum_{a=1}^{q}\frac{\mathrm{d}h_i(a)}{\mathrm{d}\alpha}Q_i(a). \end{aligned} \tag{2.30}$$

When computing the derivative at $\alpha = 0$, extremization is still obtained by means of Eq. (2.27), and the ensemble average appearing in equation Eq. (2.30) are meant to be computed with respect to the profile model. We consequently obtain for the actual Gibbs potential:

$$\left.\frac{\mathrm{d}G(\alpha)}{\mathrm{d}\alpha}\right|_{\alpha=0} = -\sum_{i=1}^{L-1}\sum_{j=i+1}^{L}\left\langle J_{ij}\right\rangle_0 = -\sum_{i=1}^{L-1}\sum_{j=i+1}^{L}\sum_{a,b=1}^{q} J_{ij}(a,b)Q_i(a)Q_j(b). \tag{2.31}$$

Finally, we can put together Eqs. (2.28) (2.31) to express the approximation of the Gibbs potential at the first order:

$$G(\alpha,\mathbf{Q}) = \sum_{i=1}^{L}\sum_{a=1}^{q} Q_i(a)\log Q_i(a) - \alpha\sum_{i=1}^{L-1}\sum_{j=i+1}^{L}\sum_{a,b=1}^{q} J_{ij}(a,b)Q_i(a)Q_j(b) + \mathcal{O}(\alpha^2). \tag{2.32}$$

The original model is recovered by setting $\alpha = 1$. As previously mentioned, the number of parameters in the model is redundant, as it is evident from the degeneracy of the connected correlation function:

$$\sum_{b=1}^{q} C_{ij}(a,b) = \sum_{b=1}^{q} P_{ij}(a,b) - P_i(a)\sum_{b=1}^{q} P_j(b) = P_i(a) - P_i(a) = 0, \tag{2.33}$$

making the matrix not invertible. Such feature can be related to the gauge invariance of the GPM. A possible strategy to avoid this inconvenient is then to fix a gauge, and we choose here the so called lattice-gas gauge 2.2.1. The lattice-gas gauge amounts to express the single site marginals for the q-th amino acid as: $Q_i(q) = 1 - \sum_{a=1}^{q-1} Q_i(a)$. Then, we need to compute the derivative of the Gibbs potential so as to obtain a set of equations for the fields and couplings. By doing so, we get a set of auto-consistent equations:

$$\frac{\partial G(\mathbf{Q})}{\partial Q_k(c)} = h_k(c) = \log\left[\frac{Q_k(c)}{Q_k(q)}\right] - \sum_{j\neq k}\sum_{a=1}^{q-1} J_{kj}(c,a)Q_j(a). \tag{2.34}$$

$$\frac{\partial^2 G(\alpha,\mathbf{Q})}{\partial Q_k(c)\partial Q_l(d)} = (C)^{-1}_{kl}(c,d) = \begin{cases} -J_{kl}(c,d) & k \neq l \\ \frac{\delta(c,d)}{Q_k(c)} + \frac{1}{Q_k(q)} & k = l \end{cases} \tag{2.35}$$

If we plug the empirical values of the frequencies and connected correlations $f_i(a)$, $C^{\mathrm{EMP}}_{ij}(a,b) = f_{ij}(a,b) - f_i(a)f_j(b)$, into Eqs. (2.34) and (2.35), we are eventually able to determine the fields and couplings compatible with the observed dataset. The most expensive operation in this process is the inversion of the $L(q-1)\times L(q-1)$ correlation matrix, though the operation has to be performed only once. The MF approach proved

to be successful for the contact prediction problem [108] and it was one of the first methods to be applied to the inference of the GPM. However, it suffers some limitations. It is for instance not able to satisfy Eq. (2.7), i.e. to accurately reproduce the empirical statistics. Moreover, since the inferred couplings are usually very large, it is difficult to sample in sequence space via Markov chain Monte Carlo (MCMC) because this leads to a glassy-like energy landscape.

2.3.2 Gaussian DCA

An alternative approach for inferring the GPM parameters is the Gaussian DCA method [9]. Within its framework, sequences in an MSA are probabilistically modeled as independently drawn Gaussian vectors. The representation used to express protein sequences as vectors of real components is known as one-hot encoding. This is a standard technique used in machine learning to transform categorical data into numerical ones. For the case of protein sequences, we have usually $q = 21$ symbols (including the gap). Each symbol is mapped onto a $q-1$ components vector, which has all entries equal to 0 but one that is equal to 1, in such a way that every amino acid has a different representation. The q-th amino acid is instead associated to the null vector. A protein sequence can be identified with a $N = L(q-1)$ vector:

$$\mathbf{S} = (2, q-3, \dots, q) \rightarrow \mathbf{x} = \left(\underbrace{0,1,0,\dots,0}_{q-1}, \underbrace{0,\dots,1,0,0}_{q-1}, \dots \underbrace{0,0,\dots,0}_{q-1} \right). \tag{2.36}$$

The MSA is made of M sequences, so that the dataset can be globally modeled as a $M \times N$ matrix X, whose rows are the protein sequences $\mathbf{x}^{(m)}$, for $m = 1, \dots, M$. A multivariate Gaussian distribution is defined by the mean vector $\boldsymbol{\mu}$ and the covariance matrix Σ, which are the parameters we want to infer in a Bayesian framework. Given such parameters, the likelihood function associated to the data sample X is given by:

$$P(X|\mu, \Sigma) = \frac{1}{(2\pi)^{\frac{MN}{2}} \det(\Sigma)^{\frac{M}{2}}} \exp\left\{ -\frac{1}{2} \sum_{m=1}^{M} \left(\mathbf{x}^{(m)} - \boldsymbol{\mu}\right)^T (\Sigma)^{-1} \left(\mathbf{x}^{(m)} - \boldsymbol{\mu}\right) \right\} \tag{2.37}$$

It is also possible to define the empirical observables corresponding to the distribution parameters:

$$\overline{\mathbf{x}} = \frac{1}{M} \sum_{m=1}^{M} \mathbf{x}^{(m)}, \tag{2.38}$$

$$\overline{C}(X) = \frac{1}{M} \sum_{m=1}^{M} \mathbf{x}^{(m)^T} \mathbf{x}^{(m)} - \overline{\mathbf{x}}^T \overline{\mathbf{x}}. \tag{2.39}$$

If we were to infer the distribution parameters in a maximum-likelihood fashion, their best estimate would coincide with Eq. (2.38) and (2.39) for $\boldsymbol{\mu}$ and Σ respectively. However, the empirical correlation matrix is usually rank deficient and consequently, not invertible. This would lead to an ill-defined Gaussian distribution, and an alternative approach is demanded. The Bayesian framework allows to include information about the prior distribution of parameters (see section 2.3.6), in such a way that the posterior distribution does not coincide with the likelihood. Then, the parameter estimates can be computed as an average over such posterior distribution. A suitable choice for the prior is the Normal inverse Wishart (NIW) distribution, which is the conjugate prior of the multivariate Gaussian. Such prior can be written as $p(\boldsymbol{\mu}, \Sigma) = p(\boldsymbol{\mu}|\Sigma)p(\Sigma)$, where:

$$p(\boldsymbol{\mu}|\Sigma) = (2\pi)^{-\frac{N}{2}} \kappa^{\frac{N}{2}} \det(\Sigma)^{-\frac{1}{2}} - \frac{1}{2} \exp\left\{-\frac{\kappa}{2}(\boldsymbol{\mu} - \boldsymbol{\eta})^T(\Sigma)^{-1}(\boldsymbol{\mu} - \boldsymbol{\eta})\right\}, \tag{2.40}$$

which is a multivariate Gaussian over the mean vector $\boldsymbol{\mu}$, characterized by a prior mean $\boldsymbol{\eta}$ and covariance Σ/κ. The prior over Σ instead reads:

$$p(\Sigma) = \mathcal{N}\det(\Sigma)^{-\frac{-\nu+N+1}{2}} \exp\left\{-\frac{1}{2}\mathrm{Tr}\left(\Lambda\Sigma^{-1}\right)\right\}, \tag{2.41}$$

with $\mathcal{N}$ a normalization factor. The average values of $\boldsymbol{\mu}$ and Σ over the prior are $\langle\boldsymbol{\mu}\rangle_{p(\boldsymbol{\mu},\Sigma)} = \boldsymbol{\eta}$, $\langle\Sigma\rangle_{p(\boldsymbol{\mu},\Sigma)} = \Lambda/(\nu - N - 1)$. Since the NIW is the conjugate prior of the multivariate Gaussian, also the posterior $p(\boldsymbol{\mu}, \Sigma|X) \propto p(X|\boldsymbol{\mu}, \Sigma)p(\boldsymbol{\mu}, \Sigma)$ is a NIW, with parameters *updated* according to the observed data:

$$\begin{cases} \kappa' = \kappa + M \\ \boldsymbol{\eta}' = \frac{\kappa}{\kappa+M}\boldsymbol{\eta} + \frac{M}{\kappa+M}\overline{\mathbf{x}} \\ \nu' = \nu + M \\ \Lambda' = \Lambda + M\overline{C} + \frac{\kappa M}{\kappa+M}(\overline{\mathbf{x}} - \boldsymbol{\eta})^T(\overline{\mathbf{x}} - \boldsymbol{\eta}) \end{cases} \tag{2.42}$$

We notice how the number of prior observations κ is augmented by the number of observed data points M, and the same happens for the parameter ν. Finally, the maximum a posteriori value of the parameters $\boldsymbol{\mu}$ and Σ becomes:

$$\langle\boldsymbol{\mu}\rangle_{p(\boldsymbol{\mu},\Sigma|X)} = \boldsymbol{\eta}' = \frac{\kappa}{\kappa + M}\boldsymbol{\eta} + \frac{M}{\kappa + M}\overline{\mathbf{x}}, \tag{2.43}$$

$$\langle\Sigma\rangle_{p(\mu,\Sigma|X)} = \frac{\Lambda'}{\nu' - N - 1} = \frac{\Lambda + M\overline{C} + \frac{\kappa M}{\kappa+M}(\overline{\mathbf{x}} - \boldsymbol{\eta})^T(\overline{\mathbf{x}} - \boldsymbol{\eta})}{\nu + M - N - 1}. \tag{2.44}$$

We notice that the posterior mean is made of two contributions, the prior mean $\boldsymbol{\eta}$ and the empirical one $\overline{\mathbf{x}}$, which are weighted according to the correspondent number of observations κ and M. We now need to make some assumption about the prior parameters. If we consider them to be as uninformative as possible, it is conceivable to choose the parameters so that the prior describes a uniformly distributed sample. This

entails $\langle \mu_i \rangle_{p(\mu,\Sigma)} = \eta_i = 1/q$ for $i = 1, \dots, N$ for the mean vector. The expected value of the covariance matrix $U = \Lambda/(\nu - N - 1)$ will have all entries equal to 0 apart for $(q-1) \times (q-1)$ blocks having expression $\frac{1}{q}\left[\delta(a,b) - \frac{1}{q}\right]$, for $a, b = \{1, \dots, q-1\}$. Then, if we set $\nu = \kappa + N + 1$, we are able to interpret the ratio $\kappa/(\kappa + M)$ as the normalized *pseudocount* α (see section 2.3.6), that is, the statistical weight given to the prior uniform sample. We can then rewrite the posterior parameters estimate as:

$$\begin{cases} \langle \boldsymbol{\mu} \rangle_{p(\boldsymbol{\mu},\Sigma|X)} = \alpha\boldsymbol{\eta} + (1-\alpha)\overline{\mathbf{x}}, \\ \langle \Sigma \rangle_{p(\boldsymbol{\mu},\Sigma|X)} = \alpha U + (1-\alpha)\overline{C} + \alpha(1-\alpha)\,(\overline{\mathbf{x}} - \boldsymbol{\eta})^T\,(\overline{\mathbf{x}} - \boldsymbol{\eta})\,. \end{cases} \tag{2.45}$$

In the multivariate Gaussian framework, the couplings matrix is defined as the inverse of the covariance Σ, as it is expressed in Eq. (2.45). This is actually very similar to what happens in the MF scheme.

An objection that can be moved against this modeling is that considering protein sequences to be real valued vector is a very crude approximation. However, even if the $\mathbf{x}$ components can assume any value on the real axis, the data themselves are highly structured, as it is evident from the empirical correlation function between amino acids at a same site l:

$$\overline{C}_{(l-1)(q-1)+a,(l-1)(q-1)+b} = -\overline{x}_{(l-1)(q-1)+a}\overline{x}_{(l-1)(q-1)+b} < 0, \tag{2.46}$$

as it should be, because the occurrence of different amino acids at a certain site should be anti-correlated. If the pseudocount is small, the main contribution in Eq. (2.45) is indeed given by $\overline{C}$, so that the peculiar data structure is included in Σ.

2.3.3 Pseudo-likelihood

The pseudo-likelihood inference strategy has been first introduced in [8] with the acronym GREMLIN, and then revised in [43]. It represents an efficient method to approximate the global likelihood function by means of single site conditional probabilities:

$$P_r\left(\sigma_r|\sigma_{\setminus r}\right) = \frac{e^{h_r(\sigma_r) + \sum_{j \neq r} J_{rj}(\sigma_r,\sigma_j)}}{\sum_{a=1}^{q} e^{h_r(a) + \sum_{j \neq r} J_{rj}(a,\sigma_j)}}, \tag{2.47}$$

where the notation $\sigma_{\setminus r} = (\sigma_1, \dots, \sigma_{r-1}, \sigma_{r+1}, \dots, \sigma_L)$ stands for the amino acids at all residues but r. The great advantage of Eq. (2.48) is that it is possible to compute the normalization factor Z_r, since the summation is performed over q configurations only. The pseudo-likelihood is defined as the product over all the sites of the single conditional probabilities $\prod_{r=1}^{L} P_r\left(\sigma_r|\sigma_{\setminus r}\right)$. Then, the log-pseudo-likelihood associated to an MSA $\left\{\mathbf{S}^{(m)}\right\}_{m=1}^{M}$ will become:

$$\mathscr{L}_{\text{pseudo}} = \frac{1}{M}\sum_{m=1}^{M}\sum_{r=1}^{L}\log P(\sigma_r^{(m)}|\sigma_{\setminus r}^{(m)}). \tag{2.48}$$

An interesting feature of Eq. (2.48) is that it provides an asymptotic approximation of the global likelihood function, that is, it approaches the correct result for $M \to \infty$. Furthermore, rather than optimizing Eq. (2.48) altogether, it is possible to break the log-pseudo-likelihood in L independent contributions which can be optimized in parallel [42], because they depend on different sets of parameters $\mathbf{h}_r, \mathbf{J}_r = \{J_{ri}\}_{i\neq r}$:

$$g_r(\mathbf{h}_r, \mathbf{J}_r) = \frac{1}{M}\sum_{m=1}^{M}\log P(\sigma_r^{(m)}|\sigma_{\setminus r}^{(m)}), \tag{2.49}$$

providing a consistent speed up of the inference process. The drawback of this *asymmetric* approach is that it provides two different estimates for the couplings J_{ij}^i and J_{ji}^j, where the apices indicate from which specific g_r the parameter has been inferred. A possible solution is to define a symmetrized version of the coupling as $J_{ij} = \left(J_{ij}^i + J_{ji}^j\right)/2$.

The pseudo-likelihood has demonstrated to be particularly effective for the contact prediction problem [43] over a wide range of different of protein families. It must be noted that the inference procedure is conceptually different to what is prescribed by the maximum-entropy principle, even though the functional form of the distribution is the same. However, one can check a-posteriori that the inferred probability distribution is able to reproduce the empirical frequencies, even if these constraints have not been explicitly imposed during the learning.

In the following we will refer to pseudo-likelihood inference of the GPM as the PlmDCA approach.

In section 3.3 we will treat again the pseudo-likelihood approximation, since it is the inference method we will rely on for our novel statistical models of experimental sequence data.

2.3.4 Boltzmann Machine Learning

Boltzmann machine learning (BML) [2] is an inference scheme that allows to determine the model parameters by optimization of the global likelihood function, and it was first applied to infer the GPM in [92]. In the following, we will refer to BML in che context of the GPM as bmDCA. As we mentioned at the beginning of Sec. 2.3, the inherent issue of optimizing Eq. (2.18) is the necessity to compute the partition function Z. bmDCA circumvents this limitation by exploiting MCMC sampling. Indeed, the derivatives of the log-likelihood with respect to the GPM parameters read:

$$\begin{cases} \frac{\partial \mathscr{L}}{\partial h_k(a)} = \frac{1}{M}\sum_{m=1}^{M} \delta(\sigma_k^{(m)}, a) - \langle \delta(\sigma_k, a)\rangle = f_k(a) - \langle \delta(\sigma_k, a)\rangle \,, \\ \frac{\partial \mathscr{L}}{\partial J_{kl}(a,b)} = \frac{1}{M}\sum_{m=1}^{M} \delta(\sigma_k^{(m)}, a)\delta(\sigma_l^{(m)}, b) - \langle \delta(\sigma_k, a)\delta(\sigma_l, b)\rangle = f_{kl}(a,b) - \langle \delta(\sigma_k, a)\delta(\sigma_l, b)\rangle \,. \end{cases} \tag{2.50}$$

We recall that with the brackets $\langle\cdot\rangle$ we indicate ensemble averages with respect to the probability function in Eq. (2.10). MCMC sampling allows to determine these average values without computing the normalization factor Z. Then, a gradient ascent algorithm can be implemented for the determination of the model parameters, according to the update equations:

$$\begin{cases} h_k^{(t+1)}(a) = h_k^{(t)}(a) + \eta\left[f_k(a) - \langle \delta(\sigma_k, a)\rangle\right] , \\ J_{kl}^{(t+1)}(a,b) = J_{kl}^{(t)}(a,b) + \eta\left[f_{kl}(a,b) - \langle \delta(\sigma_k, a)\delta(\sigma_l, b)\rangle\right] , \end{cases} \tag{2.51}$$

which depend on the hyperparameter η, usually referred to as the *learning rate*. The choice of an adequate value of η is fundamental for a successful learning of the parameters. Too small a value leads to a slow exploration of the likelihood landscape, whereas a too large one generates back and forth oscillation preventing convergence. Unless one relies on Newton-like methods that automatically set the learning rate [69], a general strategy for determining the appropriate learning rate is lacking, and the specific optimal value changes from case to case.

Since the log-likelihood is a concave function of the model parameters, convergence to a global maximum is in principle guaranteed. Moreover, model parameters can be determined with arbitrary accuracy [161], although this requires a likewise elongation of the learning process. The major limitation is due to the number of samples that need to be collected in the MCMC simulation in order to obtain a precise estimate of the distribution averages. This makes bmDCA generally much slower than the previously presented methods, even though a recent implementation of bmDCA [11] employed a parameters reduction via sparsification. On the other hand, since the maximum-entropy conditions for the matching of empirical and ensemble frequencies are directly imposed into the update equation, these are quite precisely fulfilled by bmDCA, in contrast for instance to the MF method.

2.3.5 Autoregressive DCA

Recently, an inference method based on autoregressive networks has been applied to protein sequence data [164]. Differently from previously presented techniques, autoregressive DCA (arDCA) is a self-supervised inference method based on a shallow network, allowing for the exact computation of sequence probabilities, in contrast with bmDCA and PlmDCA that provide unnormalized probabilistic weights.

The main idea at the basis of arDCA is to use Bayes theorem to break down the joint probability of a sequence as:

$$P(\sigma_1, \sigma_2, \dots, \sigma_L) = P(\sigma_1)P(\sigma_2|\sigma_1)P(\sigma_3|\sigma_2, \sigma_1) \dots P(\sigma_L|\sigma_{L-1}, \dots, \sigma_1). \tag{2.52}$$

Differently from PlmDCA, where single site probabilities are conditioned over all the rest of the sequence, here the probability factor related to site i is conditioned only over $j < i$. Yet, inference can be performed similarly to PlmDCA, as the likelihood gradient can be computed exactly, allowing to avoid expensive computations as it happens in bmDCA, where it is necessary to estimate ensemble averages via MCMC. The single site conditional probabilities are parametrized according to:

$$P(\sigma_i|\sigma_{i-1}, \dots, \sigma_1) = \frac{\exp\left\{h_i(\sigma_i) + \sum_{j=1}^{i-1} J_{ij}(\sigma_i, \sigma_j)\right\}}{\sum_{a=1}^{q} \exp\left\{h_i(a) + \sum_{j=1}^{i-1} J_{ij}(a, \sigma_j)\right\}}, \tag{2.53}$$

which is referred to as soft-max regression. Even though the number of parameters in the model is the same of a generalized Potts Hamiltonian (and then usually much smaller than deep learning architecture), the two sets do not have the same interpretation, in particular for the coupling ones. Indeed, they cannot be directly interpreted as a *direct interaction* between i and j, but rather, the influence of j on i for $j < i$ is taken into account.

This leads to another subtle point of the method, that is, the lack of site permutation invariance. The specific order in Eq. (2.52) does matter, yielding different results from case to case. The strategy chosen in [164] is to start from the less entropic residues, since this possesses also a biologically motivated interpretation.

arDCA displays performances comparable to bmDCA for both the reproduction of the empirical frequencies observed in natural alignments and in the prediction of mutational effects (see Sec. 2.4.2).

2.3.6 Regularization

All the inference methods presented previously need some kind of *regularization*, either to avoid overfitting or to make the inference problem feasible. Both the MF and Gaussian method require the inversion of a covariance matrix for the determination of the couplings. However, the number of available independent data is usually not large enough to have an accurate estimate of the two sites frequencies, and consequently, correlations can be easily under or overestimated. In this perspective, the usage of pseudocounts is a possible solution to attenuate data scarcity.

The underlying idea is to include a Dirichlet prior for the empirical frequencies [40], augmenting the observed counts according to the concentration parameters of the prior. The most unbiased assumption is to consider that a-priori all the amino acids have the same probability to be observed, and consequently the prior probability is uniform, i.e. $1/q$. Then, the expected number of a-priori observed amino acids is given by the

product of the total number of pseudocounts λ and the prior probability, so that the empirical frequencies are corrected as:

$$f_i(a) = \frac{1}{M+\lambda}\left[\frac{\lambda}{q} + \sum_{m=1}^{M}\delta(\sigma_i, a)\right]$$
$$f_{ij}(a,b) = \frac{1}{M+\lambda}\left[\frac{\lambda}{q^2} + \sum_{m=1}^{M}\delta(\sigma_i, a)\delta(\sigma_j, b)\right]. \tag{2.54}$$

The previous equation can also be rewritten in terms of the normalized pseudocounts $\alpha = \frac{\lambda}{\lambda+M}$, as it was done in the Gaussian DCA approach in Sec. 2.3.2.

On the other hand, gradient based methods such as PlmDCA and bmDCA are often regularized by adding an l_2 penalty to the gradient. For the GPM, the l_2 regularization reads:

$$R(\mathbf{h},\mathbf{J}) = \lambda_J \|\mathbf{J}\|^2 + \lambda_h \|\mathbf{h}\|^2 = \lambda_J \sum_{i=1}^{L}\sum_{j>i}^{L}\sum_{a,b=1}^{q} J_{ij}(a,b)^2 + \lambda_h \sum_{i=1}^{L}\sum_{a=1}^{q} h_i(a)^2, \tag{2.55}$$

where we have introduced the hyperparameters λ_J and λ_h, that set the regularization strength. In a Bayesian framework, Eq. (2.55) can be interpreted as a Gaussian prior over the parameters, and its inclusion in the objective function also allows to make the inference process faster and easier. However, too strong a regularization might distort the inferred parameters. Moreover, even if in principle the regularization strength should go to zero for a data sample of diverging size, it is a common choice to set λ_J and λ_h to a value between 1.0e-3 and 0.01, independently to the sample size.

2.4 Generalized Potts Model applications

In this section, we review the most relevant results related to the application of the GPM to protein sequence data [30]. In the following, we will use equivalently the names GPM and DCA to refer to the maximum-entropy probabilistic model defined by the energy function Eq. (2.12).

2.4.1 Contact prediction

The problem of folding, i.e. predicting the tertiary structure from the very sequence of amino acids, has always been a milestone of computational biology. Recently, AlphaFold practically solved the problem by relying on a deep learning architecture [85]. However, one of the first significant advances in this direction was provided by the application of DCA methods [172, 108].

The experimental determination of a protein three dimensional structure can be realized by means of several techniques such as: X-ray crystallography, Nuclear Magnetic Resonance (NMR) and electron microscopy. All these methods are expensive and not trivial to be realized, and consequently the amount of accessible structures remained limited in time. On the other hand, the advances in sequencing technologies yielded an explosion of sequence data, allowing for the application of statistical-based methods.

Initially, the idea was to use statistical correlations detectable in MSA's of protein families [119, 53] as described in Sec. 1.2.2. Specifically, one of the used indicator was the *Mutual Information* [39] between residues i and j:

$$\mathrm{MI}_{ij} = \sum_{a,b=1}^{q} f_{ij}(a,b) \log \frac{f_{ij}(a,b)}{f_i(a) f_j(b)}. \tag{2.56}$$

If the two residues are uncorrelated then $f_{ij}(a,b) = f_i(a) f_j(b)$, yielding a null mutual information. A high MI_{ij}, indicates that the residues are highly correlated, as a possible consequence of structural or functional interactions. However, as we discussed in 1.2.2, such correlations can be spurious, and a tool to disentangle between direct and indirect interaction is required. Within the GPM approach, the direct interaction is quantified by the coupling parameters $\mathbf{J}$. Specifically, in order to quantify the amount of interaction between two residues, it is necessary to trace over all the possible amino acids. This is achieved by means of the Frobenius norm:

$$F_{ij} = \left\| J_{ij} \right\|^2 = \sum_{a,b=1}^{q} J_{ij}(a,b)^2. \tag{2.57}$$

Then, the possible residue pairs are sorted in a decreasing order of score F_{ij}, for the one with the higher score are intended to be more probably interacting. In doing so, only residue pairs such that $|i-j| > 4$ are retained, so to avoid to consider contacts deriving from the secondary structure. In order to assess if the pair of residues is actually in contact in the tertiary structure a reference coming from the *Protein Database Bank* is employed, where the information about the majority of experimentally resolved structures are collected, and identified by a four symbol alpha-numeric string. A contact is considered to be so if the corresponding residues are found at a distance smaller of a certain threshold, which is usually set at 5Å or 8Å.

Before moving on, it is necessary to treat some key technical points. When applying DCA to MSA's of homologous proteins, the underlying working hypothesis is that the sequences are independently sampled from an equilibrium distribution. However, phylogenetic relations among sequences introduce correlations which are not due to functional or structural constraints, hindering the determination of the coupling parameters. It is extremely hard to disentangle phylogenetic and coevolution correlations, and the research activity in this direction is still ongoing [132, 131, 78]

One of the most common strategies to attenuate the effect of phylogeny is the use of reweighting procedures, as they were introduced in [172]. The idea is to associate

to each sequence in the MSA a weight which is inversely proportional to the similarity with other sequences, according to:

$$w^{(m)} = \left[1 + \sum_{n \neq m} \Theta\left(\alpha - \frac{1}{L} \mathrm{h_D}\left(\mathbf{S}^{(m)}, \mathbf{S}^{(n)}\right)\right)\right]^{-1}, \tag{2.58}$$

where $\Theta(x)$ is the Heaviside function, $\mathrm{h_D}(\mathbf{S}, \mathbf{S}') = \sum_{i=1}^{L} \left[1 - \delta(\sigma_i, \sigma_i')\right]$ is the Hamming distance between sequences, and α is a threshold. For every sequence $\mathbf{S}^{(n)}$ having a fraction of mismatches smaller than α, the weight of the sequence $\mathbf{S}^{(m)}$ is downweighted by one. Typical values for the threshold parameter are around $\alpha = 0.2$. Then, the one and two point frequencies can be computed as weighted averages:

$$\begin{aligned} f_i(a) &= \frac{1}{M_{\mathrm{eff}}} \sum_{m=1}^{M} w^{(m)} \delta(\sigma_i^{(m)}, a), \\ f_{ij}(a, b) &= \frac{1}{M_{\mathrm{eff}}} \sum_{m=1}^{M} w^{(m)} \delta(\sigma_i^{(m)}, a) \delta(\sigma_j^{(m)}, b), \end{aligned} \tag{2.59}$$

with $M_{\mathrm{eff}} = \sum_{m=1}^{M} w^{(m)}$ the effective number of unique sequences. The reweighting procedure consistently improves the result of DCA for contact prediction.

Another common procedure to improve the contact prediction is the so called *average product correction* (APC). The idea is to correct the interaction score between residues i and j by averaging over the interaction they have with all other residues:

$$F_{ij}^{APC} = F_{ij} - \frac{F_{i\cdot} F_{\cdot j}}{F_{\cdot\cdot}}, \tag{2.60}$$

where the point stands for averaging over the other residues, e.g. $F_{i\cdot} = \frac{1}{L-1} \sum_{k \neq i} F_{ik}$ and $F_{\cdot\cdot} = \frac{2}{L(L-1)} \sum_{k>l} F_{kl}$. A clear interpretation of why the APC works well for the contact prediction problem is actually missing, even though it has been suggested that it might be related to entropic or phylogenetic corrections [22, 86].

The DCA method has been tested on a large variety of protein families, proving to be quite robust both with respect to the choice of the family and the specific inference method used. Quite notably, the inference method that provides the best result is PlmDCA 2.3.3, even compared to more accurate algorithms such as bmDCA 2.3.4. A possible explanation for this outcome, is that what matters the most for the contact prediction problem is not the specific value of the parameters, but rather, the magnitude hierarchy between them. In Fig. 2.1, an example of DCA performance in assessing contact prediction for the protein family PF00397 (WW-domain) is reported, in the form of a *sensitivity plot*. Such plot shows the *positively predicted value* (PPV) as a function of the possible residue pairs sorted for decreasing value of the Frobenius norm. The

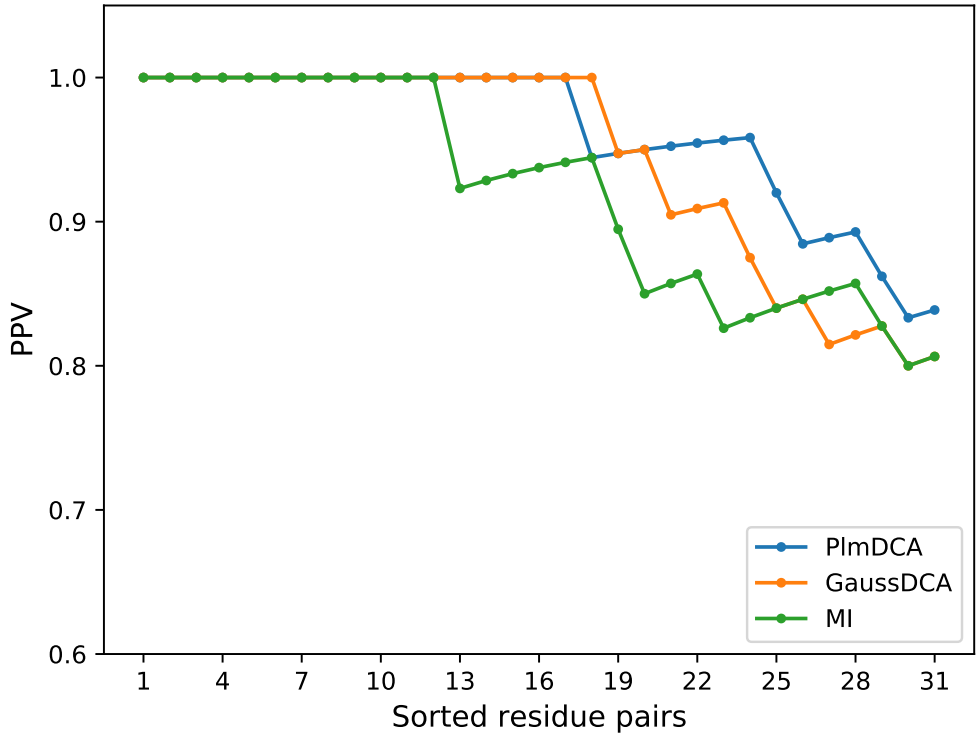


Figure 2.1: Example of DCA application to the contact prediction problem. The figure displays the sensitivity plots obtained for the WW domain using three different methods: PlmDCA 2.3.3, GaussDCA 2.3.2 and MI. Both direct-coupling methods exhibit superior performances compared to MI.

definition of the PPV is PPV=TP/(TP+FP), i.e. the fraction of true positive with respect to the total number of prediction, given by both true and false positive.

In the recent years, deep learning algorithms are progressively gaining more attention also for the analysis of protein sequence data, and specifically for the contact prediction and folding problem [171, 180]. In particular, the biggest leap has been provided by the team AlphaFold2 [85] during the CASP14 competition. Thanks to a neural network architecture, they have been able to resolve de-novo structures up to experimental precision, starting from the very sequence of amino acids. Notably, the neural network includes within its inputs also an alignment of homologous sequences, thus still leveraging coevolution information. This suggests that the integration of statistical-physics inspired models into deep architectures might be an interesting way to go for future applications to protein sequence data.

2.4.2 Mutational effects

Another interesting property of the DCA method, is that the energy function can be used as a proxy of log-probability, and consequently, as a score for sequences. In particular, this score can be used to quantify the effect of mutations on natural sequences, and also to determine how likely it is for that sequence to belong to a certain protein family.

The scenario we have in mind is that of a laboratory experiment of the kind introduced in Sec. 1.3, in which the sequence space in the vicinity of a wild-type is probed for a functional feature. Depending on the specific experimental technique employed, the exploration might be more or less broad, but still local when compared to an MSA of homologous sequences. Even though DCA is inferred on natural MSA's, the obtained landscape nonetheless provides an approximation of the local landscape in the neighborhood of the wild-type. In order to test this hypothesis, the statistical energy can be used as a proxy of the protein fitness. Specifically, the score of a variant $\mathbf{S}^{(mut)}$ can be computed as its energy difference with respect to the wild type:

$$\Delta E(\mathbf{S}^{(mut)}) = H(\mathbf{S}^{(mut)}) - H(\mathbf{S}^{(wt)}), \tag{2.61}$$

where $H(\mathbf{S})$ is the GPM of Eq. (2.12). Mutations (single or multiple) can be neutral, beneficial or deleterious. This reflects in the sign of Eq. (2.61), that becomes respectively null, negative or positive. Significantly, since the GPM is a global function of the sequence, even single mutations depend on the specific background, as it can be checked by inspection of Eq. (2.61) for the single site mutation $\sigma_i \to \sigma_i'$:

$$\Delta E(\mathbf{S}', \mathbf{S}) = H(\mathbf{S}') - H(\mathbf{S}) = h_i(\sigma_i) - h_i(\sigma_i') + \sum_{j \neq i} \left[J_{ij}(\sigma_i, \sigma_j) - J_{ij}(\sigma_i', \sigma_j) \right], \tag{2.62}$$

from which the fundamental role of the couplings $\mathbf{J}$ emerges.

The DCA approach has been applied to a variety of different experimental and biological contexts for the prediction of mutational effects. These include viral strains [23, 96, 133], bacterial cells [26, 49, 81], human proteins [45] and an ensemble of different datasets [75, 164]. One of the fundamental results outlined in these papers, is the necessity to use epistatic models in order to obtain meaningful predictions of mutational effects. As an example, in Fig. 2.2 Spearman correlations between mutational scores in Eq. (2.61) and experimental fitness measurements of the TEM-1 β-lactamase protein are reported, comparing a profile model with one including coupling parameters. From the correlation values, it is evident how the epistatic model is able to provide superior performances.

In Fig. 2.3, a summary of the performances of various DCA models for several datasets is presented and compared with a deep learning approach [129].

In this thesis, we aim to present some novel DCA-inspired approaches (chapters 3 and 4) for the unsupervised inference of accurate local fitness landscapes. Specifically,

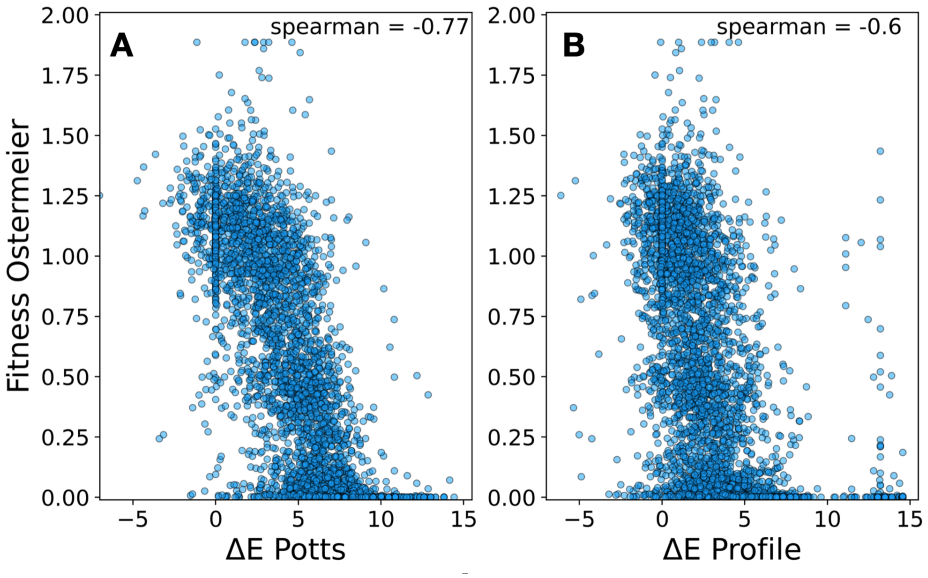


Figure 2.2: Comparison between profile and epistatic model in terms of mutational effects prediction. Both panels display the scatter plot between the statistical mutational score and the experimental fitness measurements of [51], and values of the correpsonding Spearman correlations are reported as inserts. On the left, scatter of the epistatic scores in Eq. (2.61), having a correlation $\rho_S = -0.77$. On the right, the same scatter obtained with a profile model, having a lower correlation $\rho_S = -0.6$. Figure taken from [15].

these methods try to develop an effective dynamical modeling of the underlying experimental process, thus leveraging the information contained in all sequenced rounds, as opposite to standard DCA approaches, that are usually inferred on homologous alignments, and when applied to local mutational datasets such as DMS and DE experiments, are not able to include the whole experimental information.

2.4.3 Protein-protein interactions

Another interesting open problem in the protein realm is the prediction of interaction partners, which can eventually fold together to make a protein complex. On top of predicting if two or more proteins interact with each other, it is interesting to determine how they will, i.e. the physical contacts among them. Notably, the DCA method can be employed for both this purposes.

Initially, DCA and other coevolutionary-based methods were used to predict contacts between proteins [172, 122, 76] that were known to be interacting. The problem of determining if paralogs within a species are interacting was subsequently tackled in

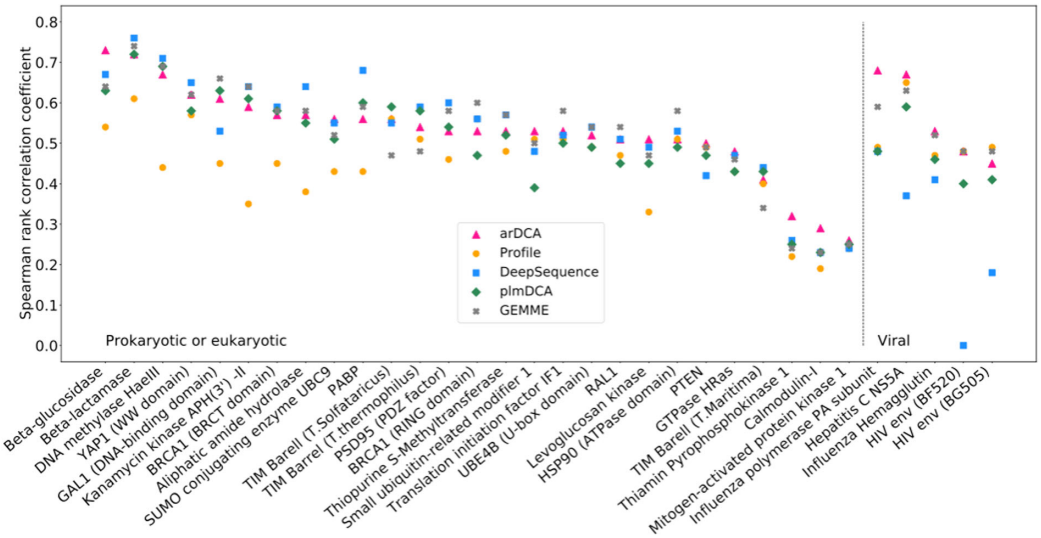


Figure 2.3: Performances comparison in terms of the prediction of mutational effects for various inference methods. Of particular interest are the results of PlmDCA [43, 42], arDCA [164] and DeepSequence [129], as a benchmark comparison between fully unsupervised against both shallow and deep architectures supervised learning. The performance is evaluated in terms of Spearman correlation between the mutational scores and the available experimental fitness measurements, for both eukaryotic and prokaryotic organisms, as well as viruses. Figure taken from [164].

[68, 16, 46].

The common idea used to face all these problems, is to define a probability for two concatenated sequences $(\mathbf{S}, \mathbf{S}')$ that may or not interact with each other:

$$P(\mathbf{S}, \mathbf{S}') = \frac{1}{Z} \mathrm{e}^{-H(\mathbf{S})-H'(\mathbf{S}')-H_{\mathrm{int}}(\mathbf{S},\mathbf{S}')}, \tag{2.63}$$

where $H_{\mathrm{int}}(\mathbf{S}, \mathbf{S}')$ is defined as:

$$H_{\mathrm{int}}(\mathbf{S}, \mathbf{S}') = -\sum_{i \in \mathbf{S}, j \in \mathbf{S}'} J_{ij}(\sigma_i, \sigma'_j). \tag{2.64}$$

Such energy contribution is supposed to model the coevolution between protein $\mathbf{S}$ and $\mathbf{S}'$. In principle, it would be possible to consider a higher number of interacting proteins. However, the number of interacting parameters in the model grows as $(L_1 + L_2 + L_N)^2$, with L_k the length of each protein, $k = 1, \dots, N$ and N the total number of possibly interacting partner. Consequently, one usually restricts to the case of pair-wise interactions.

As in standard DCA, the Frobenius norm of the inter-protein couplings can be used as a score for assessing how likely it is for the residues to be interacting. On the other

hand, a possible strategy to determine if two concatenated paralogs are actually interacting, is to use the mean value of the strongest couplings between the sequences as a score. Alternatively, this can achieved before-hand during the pairing process of MSA's, by an iterative reshuffling of the alignments determined according to coevolution signals [68, 16].

Interestingly, once a pair of proteins is guessed to be interacting (or already known to be so), the predicted contacts can be used to guide the process of predicting the actual spatial conformation of the interacting surface [144].

Recently, it has been shown that phylogenetic correlations, that usually hinder coevolution information when applying DCA methods, can actually be leveraged in the context of protein interaction prediction [97, 60, 67].

2.4.4 Sequence generation

Another appealing application of the GPM is the possibility to generate de-novo functional sequences. Indeed, once the probability function in Eq. (2.10) is learnt, it can be used to sample sequences. The sampling process is realized via a MCMC based on the energy function of Eq. (2.12), e.g. via the Metropolis-Hastings algorithm. The first indications that the pairwise maximum-entropy model could be a promising tool for functional sequence generation came from [140, 150]. In these works, a tool called *statistical coupling analysis* was employed to generate artificial sequences of the WW-domain ($L = 35$ residues). The method relies only on the statistical pattern related to the alignment of natural homologous sequences (PFAM family PF00397), and specifically, column-wise and column-paired frequencies. In order to generate the artificial sequences, a curated alignment of 120 natural sequences is taken as a starting point. Then, brand new alignments are generated in three different ways:

- **R** (random): sequences in the alignment are randomly reshuffled, consequently destroying all the statistical patterns, i.e. both conservation and correlations.
- **IC** (independent conservation): every column in the alignment is reshuffled independently, thus conserving the single residue statistics, but destroying the inter-column correlations.
- **CC** (coupled conservation): the IC dataset is used as a starting point for performing a Monte Carlo annealing process that leads the final alignment to possess the same pairwise statistics of the natural one.

For each set of sequences, a subset is selected for experimental testing for both thermodynamic stability [150] and binding affinity onto a cognate ligand [140]. Both the **R** and **IC** groups of sequences were not able to fold. On the other hand, 33% of the **CC** sequences folded into a stable structure. This outcome suggests that inter column correlation is a fundamental feature that has to be fulfilled in order to generate functional

sequences. Since the maximum-entropy modeling is based on the idea that the inferred model must be able to reproduce the pair-wise statistics, this makes it a good candidate for the generation of de-novo protein sequences.

Another interesting property of the Potts model, is that it allows to score single sequences with respect to folding, similarly to what was described in Sec. 2.4.2 for mutational effects. In this perspective, less accurate inference methods such as MF and PlmDCA can also be used, as it was shown in [8].

In Fig. 2.4 three different energy spectra are reported. The blue one corresponds to the energy distribution obtained via MCMC sampling based on an accurate Potts model inferred on the homologous alignment of the WW domain. The green one is obtained by sampling from the site-independent model and the red one by randomly extracting sequences. All the energies are computed with respect to the parameters inferred on the homology family. A sharp separation between random sequences and the other two can be observed. Moreover, even though the independent and full Potts energy spectra are partly overlapping, the folding sequences of datasets [140, 150] mostly fall in the second spectrum, or in the low energy region of the independent model, once more witnessing how energy can be used to discriminate between functional and non-functional sequences.

In order to generate de-novo functional sequences, the model parameters must be inferred very accurately, and consequently methods such as the *Adaptive Cluster Expansion* [29, 12], bmDCA or arDCA are required. Specifically, bmDCA was shown to be able to generate functional sequences in [139], whereas the arDCA approach was shown to be comparable to bmDCA in inference accuracy [164], at the same time allowing for a simpler sampling of protein sequences that does not require expensive MCMC simulations.

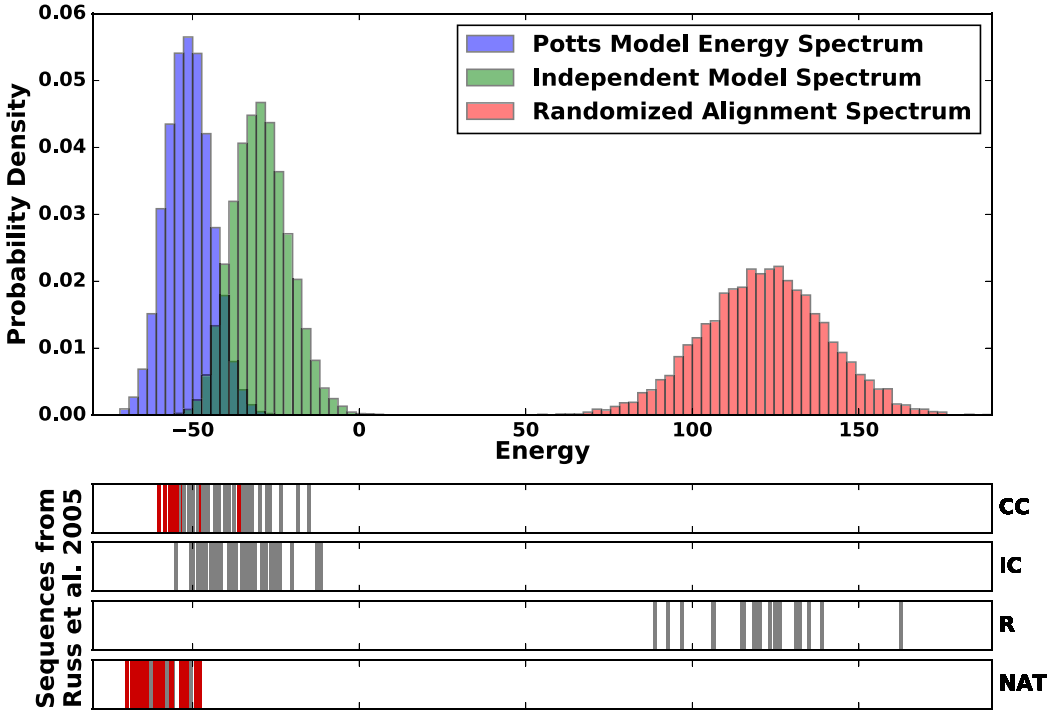


Figure 2.4: Top panel: WW domain Potts model energy spectra generated in three different ways. The blue histogram coincides with sequences generated by MCMC sampling using the aforementioned Potts model. The green histogram corresponds to sequences generated from a profile model, i.e. relying only on amino acid conservation. The red one is related to totally randomly generated sequences. Bottom panel: energy values of the sequences tested in [140, 150]. Red bars are for folding sequences, grey for non-folding ones.

Chapter 3

Annealed Mutational approximated Landscape (AMaLa)

In this chapter we present *Annealed Mutational approximated Landscape* (AMaLa), a novel inference method tailored for performing inference on DE experiments, as it was published on the International Journal of Molecular Science [145].

3.1 Motivations

In Sec. 1.3.2 we described the peculiar features of DE experiments, underlying how the main difference with other kinds of screening experiments lies in the extent at which mutagenesis is performed. This is particularly interesting, because it allows for a broader exploration of the sequence space in the neighborhood of the wild-type sequence with respect to DMS, yet still giving a local glimpse of such landscape when compared to MSA of homologous sequences.

The locality of the exploration makes these data optimal candidates for inferring accurate fitness landscapes in the vicinity of the wild-type sequence. Such fine scale information might be combined with landscapes inferred over homology data, as to obtain a more complete description of the fitness landscape [10], with potential applications to a variety of topics, such as protein engineering, study of evolutionary paths for functional shift, protein evolution and residue contact prediction.

Most of the computational strategies developed so far to analyze protein evolution data, mainly rely on two approaches: (i) DCA-inspired models of phylogenetically related sequences [48, 75, 105, 156, 87, 89, 15]; (ii) supervised machine learning approaches on sequenced samples of high-throughput functional assays or screening experiments. In this case, a statistical model of the mutants' fitness is inferred from a subset of the

sequencing data (training set) with machine learning techniques developed to solve a specific—generally non-linear—regression problem [24, 138, 121, 142, 176, 58].

Usually, standard DCA approaches are applied only to the last available round, for it is the one supposed to be closer to equilibrium. However, since this hypothesis is already not entirely verified for natural MSA data, which evolved over million years time scales, this is even more true for short-time laboratory data. Consequently, it seems reasonable to attempt to build a statistical model that is able, at least effectively, to embody the dynamics of the underlying experimental process.

In this perspective, alternative unsupervised strategies have been proposed [120, 47] to cope with all sequencing data coming from screening experiments. In [47], a probabilistic model is described, which takes into account three different steps always occurring in screening experiments: (i) selection, (ii) amplification, and (iii) sequencing (sampling). Although such models are very effective for describing DMS experiments, they rely on the variations of the variants' relative abundances across rounds, and hence on the sample of the same variants at different time-steps with sufficient statistics.

Several screening setups, and in particular the one of DE experiments, do not provide such data. In particular, the repeated introduction of new mutant sequences at each round of the experiment requires to disentangle such mutational effects from functional ones when accounting for the library variation. On top of this, the data provided by [160, 44] turn out to be in a strong undersampling regime, the majority of the counts being either zero or one, implying an inherent difficulty in applying population based statistical modelings. In this framework, AMaLa is a time dependent statistical model that allows to take into account all sequenced data, by effectively resolving the experimental evolutionary dynamics and without relying substantially on abundances information.

3.2 Modeling

AMaLa uses the sequencing samples of rounds of Directed Evolution experiments to learn a map between the protein amino acid sequence and the fitness associated with the selection process, generically indicated as the fitness landscape. Typically, fitness in these experiments is related to the binding affinity to a certain target or to more complex phenotypic traits, such as antibiotic resistance in bacterial strains.

We consider the probability of observing a generic sequence at a certain time (or round) t in the following form:

$$P^{(t)}(\mathbf{S}) = \frac{\mathrm{e}^{-\beta(t)E(\mathbf{S})-\nu(t)\mathrm{h}_{\mathrm{D}}(\mathbf{S},\mathbf{S}^{wt})}}{Z^{(t)}}, \tag{3.1}$$

where $\mathbf{S} = (\sigma_1, \dots, \sigma_L)$ is the protein sequence, L being its length, and the symbols σ's are defined over the amino acids alphabet $\sigma_i = \{A, C, \dots, Y\}$ with $i = 1, \dots, L$. Alternatively, the amino acids can be mapped onto the integers $\sigma_i = \{1, \dots, q\}$, with q the size of the amino acid alphabet. The normalization is defined as the sum over all possible

sequences $Z^{(t)} = \sum_{\mathbf{S}} \exp\left[-\beta(t)E(\mathbf{S}) - \nu(t)\mathrm{h_D}\left(\mathbf{S}, \mathbf{S}^{wt}\right)\right]$, requiring in principle to sum over q^L configurations.

Eq. (3.1) is a Boltzmann-like probability made of two contributions, respectively modeling selection and mutagenesis; both can be interpreted as made of an energetic time independent function of the sequence, and a *temperature* like time dependent pre-factor. Specifically, the function $E(\mathbf{S})$ is the one assumed to contain the functional properties associated to the sequence:

$$E(\mathbf{S}) = -\sum_{i=1}^{L} h_i(\sigma_i) - \sum_{i=1}^{L-1}\sum_{j=i+1}^{L} J_{ij}(\sigma_i, \sigma_j), \tag{3.2}$$

which is in the form of a generalized Potts model defined by a set of fields and couplings $\theta_E = \{\mathbf{h}, \mathbf{J}\}$. These are the parameters we ultimately aim to determine through the inference process. The pre-factor $\beta(t)$ plays the role of an annealing temperature, constraining the exploration of the fitness landscape to be gradually closer to the minima of $E(\mathbf{S})$ as the experiment proceeds.

On the other hand, the function $\mathrm{h_D}(\mathbf{S}, \mathbf{S}^{wt}) = \sum_{i=1}^{L}\left[1 - \delta(\sigma_i, \sigma_i^{wt})\right]$ coincides with the Hamming distance between sequence $\mathbf{S}$ and the wild-type, and it models the action of mutagenesis. In Sec. 3.2.2 we will describe in details the reasons behind the choice of this functional form and the behavior of the parameter $\nu(t)$.

In order to define a likelihood function associated to the whole dataset, we assume that the model probabilities (Eq. (3.1)) of observing a given sequence at different times are statistically independent. An alternative approach would consider the dynamics as a Markov process, describing the transition probabilities defining the whole trajectory in sequence space. However, such a strategy seems to be computationally intractable, as one should sum over all possible paths connecting two sequences at subsequent times. Here, we consider a factorized time-dependent log-likelihood:

$$\mathscr{L}[\boldsymbol{\theta}_E, \boldsymbol{\beta}, \boldsymbol{\nu}] = \sum_{t=\{t_1,\dots,t_f\}} \sum_{m=1}^{M^{(t)}} w^{(m,t)} \log P^{(t)}(\mathbf{S}^{(m,t)}), \tag{3.3}$$

which is a function of the energetic parameters $\boldsymbol{\theta}_E$ and of the time dependent parameters $\boldsymbol{\beta}$ and $\boldsymbol{\nu}$, since the actual experimental data correspondent to the sequenced rounds are plugged into Eq. (3.1). The weights $w^{(m,t)} = N^{(m,t)} / \sum_{m'=1}^{M^{(t)}} N^{(m',t)}$ are defined as the normalized abundances $N^{(m,t)}$ of every unique sequence $m = 1, \dots, M^{(t)}$ appearing at round time t. In order to determine the model parameters we follow a maximum-likelihood approach, seeking for the optimal set of $\{\overline{\boldsymbol{\theta}}_E, \overline{\boldsymbol{\beta}}, \overline{\boldsymbol{\nu}}\}$ that extremizes Eq. (3.3). In Sec. 3.3 we will discuss the details of the inference pipeline.

3.2.1 Modeling selectivity

As already previously stated, the function encoding the fitness landscape information is a GPM energy $E(\mathbf{S})$. We treated thoroughly this model in Sec. 2.2, but let us recall some fundamental features. Eq. (3.2) can be interpreted as a fully connected graphical model. A priori, it is indeed not possible to tell which are the most important couplings, even if, an interesting method that allows to sparsify the model recently came out, ruling out some connections in the graph [125].

Since a necessary condition for a protein functionality is that it must be able to fold into a three-dimensional stable structure, $E(\mathbf{S})$ should in principle also contain structural information. In this perspective, dealing with a fully connected model is advantageous, because it allows to consider all the possible contacts between protein residues. Moreover, the generalized Potts model is epistatic, as the effect of a mutation on fitness depends on the specific sequence context. Such epistatic effects might indeed be relevant not only for structural features, but also when dealing with protein functionality.

The details of the selection process modeling are reported in Sec. 4.2. Here, we underline how the fundamental hypothesis is the possibility to write down the selection probabilities related to the transition between rounds $t-1$ and t as $Q_{t,t-1}(\mathbf{S}) \propto \mathrm{e}^{-\alpha_{t,t-1}E(\mathbf{S})}$. The prefactors α's model the selective pressure between neighboring rounds. Consequently, the probability at t can be expressed as $P_t(\mathbf{S}) \propto \mathrm{e}^{-\sum_{t'=t_0+1}^{t}\alpha_{t',t'-1}E(\mathbf{S})}P_{t_0}(\mathbf{S}) = \mathrm{e}^{-\beta(t,t_0)E(\mathbf{S})}P_{t_0}(\mathbf{S})$, where we set $\beta(t,t_0) = \sum_{t'=t_0+1}^{t}\alpha_{t',t'-1}$. If there are no mutations, the fundamental Fisher's theorem states that the selective pressures are a decreasing function of time [57]. This is not necessarily the case for DE experiments, where the genetic diversity is augmented at each round.

In the ideal scenario of an infinite number of rounds, the series defining the fictitious inverse temperature β might converge or not. If it does not, then the probability concentrates on the ground state of E. Otherwise, the probability gets peaked around a set of particularly functional sequences, as in a low-temperature sampling process defined by $\mathrm{e}^{-\beta(\infty)E(\mathbf{S})}$.

3.2.2 Mutagenesis: the Jukes-Cantor model

In this section we discuss a generalization of the Jukes-Cantor (JC) model [110], that was originally introduced for the description of DNA neutral evolution, that is, in the absence of selective pressure. Although the model was thought for a four symbols alphabet, namely the basis A, C, G and T in the genome, it can be generalized to any arbitrary number of symbols q. For our case of interest, the model is formulated with $q = 20$, as if mutations were considered to occur directly at the amino acid level rather than on the genome.

Here, we rely on a continuous time formulation of the JC model, referring the reader to appendix C for a discussion of the discrete time version. In this scenario, the simplest possible assumption would be to consider all transitions between amino acids to be

equally probable, yielding a unique mutation rate μ.

The JC formalism models neutral evolution as a Markov process. Consequently, the probability of observing a sequence at time t, can be written in terms of a transition probability between t and t' (with $t' < t$) and a one time probability as:

$$P(\mathbf{S}, t) = \sum_{\mathbf{S}'} P(\mathbf{S}, t|\mathbf{S}', t')P(\mathbf{S}', t'). \tag{3.4}$$

Then, mutagenesis is taken as a site independent process, so that Eq. (3.4) factorizes over the protein sites $i = 1, \dots, L$. Moreover, evolution is assumed to be starting from a single original sequence at $t = 0$, namely, the wild-type sequence. Evolutionary trajectories can then be considered from $t = 0$ up to generic time t, given the probability of the initial state:

$$P(\mathbf{S}, 0) = \prod_{i=1}^{L} \delta(\sigma_i, \sigma_i^{wt}), \tag{3.5}$$

yielding $P(\mathbf{S}, t) = P(\mathbf{S}, t|\mathbf{S}^{wt}, 0)$. In order to determine transition probabilities, we can write down the single site dynamical equation:

$$P(\sigma, t + \Delta t) = P(\sigma, t) - P(\sigma, t)\mu\Delta t + \sum_{\sigma' \neq \sigma} P(\sigma', t)\mu\Delta t, \tag{3.6}$$

where we introduced the mutation rate μ. Eq. (3.6) can be rewritten in a vectorial form, gathering the P's for each possible value of σ. If we do so, we obtain $\mathbf{p}(t + \Delta t) = \mathbf{p}(t) + M\mathbf{p}(t)\Delta t$, introducing the $q \times q$ matrix M:

$$M = \begin{pmatrix} -\mu & \mu & \dots & \mu \\ \mu & -\mu & \dots & \mu \\ \vdots & & \ddots & \vdots \\ \mu & \mu & \dots & -\mu \end{pmatrix}. \tag{3.7}$$

Rearranging Eq. (3.6) and taking the limit $\Delta t \to 0$, we obtain a differential equation for the single time probability $\dot{\mathbf{P}}(t) = M\mathbf{P}(t)$, which can be readily solved as $\mathbf{P}(t) = \mathrm{e}^{tM}\mathbf{P}(0)$, where e^{tM} is the exponential matrix of tM. Given an invertible matrix S, the exponential matrix satisfies $\mathrm{e}^{SMS^{-1}} = S\mathrm{e}^{M}S^{-1}$. Since M is diagonalizable, we can directly solve the differential equation in the basis of M's eigenvectors, and then transforming back the solution to the original basis. By doing so, we obtain the following expression for the transition probability of amino acid σ to go from k to l:

$$P(\sigma = l, t) = P(\sigma = l, t|\sigma = k, 0) = W_{kl}^{(t)}(\mu) = \begin{cases} \frac{1-(q-1)\mathrm{e}^{-\mu t}}{q} & l = k, \\ \frac{1-\mathrm{e}^{-\mu t}}{q} & l \neq k, \end{cases} \tag{3.8}$$

where we introduced the shorthand notation $W_{kl}^{(t)}(\mu)$ for the probability of a transition $k \to l$ over a time t given a mutation rate μ. Finally the probability of observing a sequence at Hamming distance d from the wild-type will be equal to:

$$\begin{aligned} P^{(t)}\left(\mathbf{S}|\mathrm{h_D}(\mathbf{S},\mathbf{S}^{wt})=d\right) &= \left[\frac{1+(q-1)\mathrm{e}^{-\mu t}}{q}\right]^{L-d}\left[\frac{1-\mathrm{e}^{-\mu t}}{q}\right]^{d} \\ &= \left(\frac{1+(q-1)\mathrm{e}^{-\mu t}}{q}\right)^{L}\exp\left\{-d\ln\left[\frac{1+(q-1)\mathrm{e}^{-\mu t}}{1-\mathrm{e}^{-\mu t}}\right]\right\} \\ &= \frac{\mathrm{e}^{-\nu(t)d}}{Z^{(t)}}, \end{aligned} \tag{3.9}$$

which yields the functional form appearing in Eq. (3.1), and it also provides the analytical expression of the time dependent parameter $\nu(t)$:

$$\nu(t) = \ln\left[\frac{1+(q-1)\mathrm{e}^{-\mu t}}{1-\mathrm{e}^{-\mu t}}\right], \tag{3.10}$$

depending on the number of symbols q and the mutation rate μ only. It is interesting to compute the two asymptotic behaviors for $t \to 0, +\infty$:

$$\begin{aligned} \nu(t) &\overset{t\to 0}{\longrightarrow} +\infty, \\ \nu(t) &\overset{t\to +\infty}{\longrightarrow} 0. \end{aligned} \tag{3.11}$$

As $t \to 0$, no sequence but the wild-type is present in the population. In order to impose the constraint $\mathrm{h_D}(\mathbf{S},\mathbf{S}^{wt}) = 0$ in Eq. (3.9), $\nu(t)$ needs indeed to diverge. On the other hand, the purely mutational process can be thought as a free diffusion from the wild-type sequence. The rate at which the distribution broadens in sequence space is given by ν, and asymptotically, every possible sequence becomes accessible, yielding a uniform probability $P^{(\infty)}(\mathbf{S}) = (1/q)^L$.

3.3 AMaLa inference

The goal of the inference procedure is to determine the selective energy parameters $\boldsymbol{\theta}_E$. To do so, it is also necessary to fix the values of the time dependent parameters $\boldsymbol{\beta}$ and $\boldsymbol{\nu}$. As previously mentioned, we rely on a maximum likelihood approach in order to achieve this task. More precisely, we employ the pseudo-likelihood approximation introduced in Sec. 2.3.3, that allows to compute the partition function in order $O(qL)$.

If the experimentalist has sequenced T rounds of the experiment, the time dependent parameters will be defined as a T component vectors: $\boldsymbol{\beta} = \left(\beta(t_1), \beta(t_2), \dots, \beta(t_f)\right)$ and

$\boldsymbol{\nu} = \big(\nu(\mu, t_1), \nu(\mu, t_2), \dots, \nu(\mu, t_f)\big)$. For now, let's take $\boldsymbol{\beta}$ and $\boldsymbol{\nu}$ as fixed. In order to find the optimal set of parameter $\bar{\boldsymbol{\theta}}_E$ we need to extremize the objective function $g(\boldsymbol{\theta}_E)$, which is defined as the sum of the pseudo-likelihood and the regularization contribution. Specifically, since we use the asymmetric version of the pseudo-likelihood approximation, we are left with L independent optimizations of the objective functions $g_r(\mathbf{h}_r, \mathbf{J}_r)$:

$$\begin{aligned} g_r(\mathbf{h}_r, \mathbf{J}_r, \boldsymbol{\beta}, \boldsymbol{\nu}) &= -\sum_{t=\{t_1,\dots,t_f\}} \sum_{m=1}^{M^{(t)}} w^{(m,t)} \ln P(\sigma_r = \sigma_r^{(m,t)} | \sigma_{\setminus r} = \sigma_{\setminus r}^{(m,t)}) + R_r(\mathbf{h}_r, \mathbf{J}_r) \\ &= -\sum_{t=\{t_1,\dots,t_f\}} \sum_{m=1}^{M^{(t)}} w^{(m,t)} \Bigg\{ \beta(t) \Bigg[h_r(\sigma_r^{(m,t)}) + \sum_{i \neq r} J_{ri}(\sigma_r^{(m,t)}, \sigma_i^{(m,t)}) \Bigg] + \nu(t) \delta(\sigma_r^{(m,t)}, \sigma_r^{wt}) \\ &\quad - \ln \Bigg[\sum_{a=1}^{q} \exp \Bigg\{ \beta(t) \Bigg[h_r(a) + \sum_{i \neq r} J_{ri}(a, \sigma_i^{(m,t)})) \Bigg] + \nu(t) \delta(a, \sigma_r^{wt}) \Bigg\} \Bigg] \Bigg\} \\ &\quad + \lambda_h \sum_{a=1}^{q} h_r(a)^2 + \lambda_J \sum_{i \neq r} \sum_{a,b=1}^{q} J_{ri}(a, b)^2. \end{aligned} \tag{3.12}$$

In order to find the minimum of Eq. (3.12) it is necessary to set to zero its derivatives:

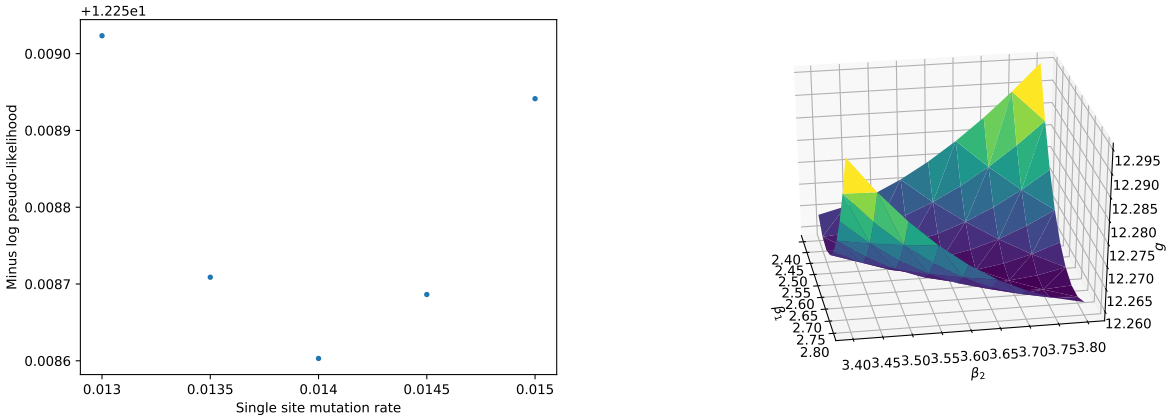
$$\begin{aligned} \frac{\partial g_r(\mathbf{h}_r, \mathbf{J}_r, \boldsymbol{\beta}, \boldsymbol{\nu})}{\partial h_r(c)} &= -\sum_{t=\{t_1,\dots,t_f\}} \sum_{m=1}^{M^{(t)}} w^{(m,t)} \beta(t) \Bigg\{ \delta(c, \sigma_r^{(m,t)}) - \frac{e^{\beta(t)\left[h_r(c) + \sum_{i \neq r} J_{ri}(c, \sigma_i^{(m,t)}))\right] + \nu(t)\delta(c, \sigma_r^{wt})}}{Z_r} \Bigg\} \\ &\quad + 2\lambda_h h_r(c) = 0, \end{aligned} \tag{3.13}$$

$$\begin{aligned} \frac{\partial g_r(\mathbf{h}_r, \mathbf{J}_r, \boldsymbol{\beta}, \boldsymbol{\nu})}{\partial J_{rj}(c, d)} &= -\sum_{t=\{t_1,\dots,t_f\}} \sum_{m=1}^{M^{(t)}} w^{(m,t)} \beta(t) \Bigg\{ \delta(c, \sigma_r^{(m,t)}) \delta(d, \sigma_j^{(m,t)}) \\ &\quad - \frac{\delta(d, \sigma_j^{(m,t)}) e^{\beta\left[h_r(c) + \sum_{i \neq r} J_{ri}(c, \sigma_i^{(m,t)}))\right] + \nu(t)\delta(c, \sigma_r^{wt})}}{Z_r} \Bigg\} + 2\lambda_J J_{rj}(c, d) = 0. \end{aligned} \tag{3.14}$$

Optimization of the objective functions is performed in parallel via the Julia implementation of the package NLopt [83].

At this point, we still need to understand how to fix the time dependent parameters. When mutagenesis and selection act simultaneously, nor the modeling presented in 3.2.1 or 3.2.2 are exacts, because the two processes do not commute with each other. Thus, to make the model more flexible, we decide to infer both $\boldsymbol{\beta}$ and $\boldsymbol{\nu}$, rather than plugging in theoretically expected values. To do so, we still rely on a maximum likelihood approach.

- $\boldsymbol{\nu} = \big(\nu(\mu, t_1), \dots, \nu(\mu, t_f)\big)$. In order to infer the components of the JC parameter ν, we employ the functional form in Eq. (3.10), performing a scan over a set of possible values of the mutation rate μ, at fixed $\boldsymbol{\beta}$. For each value of μ, we optimize the objective in Eq. (3.12) with respect to $\boldsymbol{\theta}_E$, computing the resulting total pseudo-likelihood. The optimal $\overline{\mu}$ is the one yielding the extremal objective. In Fig. 3.1 an example of the (minus) log-pseudo-likelihood as function of the mutation rate is reported.
- $\boldsymbol{\beta} = \big(\beta(t_1), \dots, \beta(t_f)\big)$. The inference of the $\boldsymbol{\beta}$ components is performed similarly to $\boldsymbol{\nu}$, which is supposed to be fixed. Eq. (3.12) is not contemporarily a convex function with respect to $\boldsymbol{\beta}$ and $\boldsymbol{\theta}_E$, and consequently it is not possible to optimize jointly with respect to both. In this perspective, the possible alternatives are: performing a scan over a set of values for each different $\boldsymbol{\beta}$ component, or seeking the pseudo-likelihood maximum by an alternate gradient ascent approach, in which the $\boldsymbol{\beta}$ and $\boldsymbol{\theta}_E$ parameters are updated alternatively. In both cases, it is possible to set the first $\boldsymbol{\beta}$ component to $\beta(t_1) = 1$, which amounts to a rescaling of the subsequent components. In appendix B we compute the derivative of the objective with respect to β and, in Fig. 3.1, an example of the scanning approach optimization is shown.



(a) (b)

Figure 3.1: Scan of the minus log-pseudo-likelihood as a function of: single site mutation proability p (closely tied to the mutation rate) (a), two free components of the $\boldsymbol{\beta}$ vector (b). In both cases the optimal value coincides with the minimum of the minus log-pseudo-likelihood.

The procedure just explained carries some subtleties. First of all, we infer one set of time dependent parameters at the time while keeping the values of other fixed. This is in principle not consistent, because for instance, any change of $\boldsymbol{\beta}$ affects $\boldsymbol{\nu}$ and vice versa.

Thus, the formally correct strategy would be to repeatedly infer $\boldsymbol{\nu}$ and $\boldsymbol{\beta}$ alternatively, until they both converge to a stable value. In order to avoid this quite costly inference pipeline, we adopt the following procedure. We first fix a set of values for $\boldsymbol{\beta}$ according to the linear scaling, i.e. $\boldsymbol{\beta}_0 = (1, t_2/t_1, \dots, t_f/t_1)$, and we infer the mutation rate given such values. Once μ is fixed, we infer the optimal set of $\boldsymbol{\beta}$ components, checking that they do not differ too much from the initial choice. This guarantees that a subsequent inference of the mutation rate would not yield an outcome which is significantly different from the previous one, as we explicitly checked on *in-silico* experiments.

Another possible scenario that is worth treating, is when $\beta(t_f) \to +\infty$ during optimization. In this case, the strategy is to set $\beta(t_f) = 1$ and $\beta(t_1) = 0$, inferring the remaining components with these two fixed, and constrained in the interval $[0,1]$.

3.4 Results on DE experiments

We tested AMaLa on three recently published DE experiments: two are described in [160] and one in [44]. The proteins mutated and selected in these experiments belong to the β-lactamase family (PSE-1 and TEM-1) and acetyltransferase family (AAC6). The β-lactamase is responsible for the hydrolysis of antibiotics such as penicillin, ampicillin and carbenicillin while the acetyltransferase is responsible for the catalysis of kanamycin via acetylation. The experiments alternate rounds of variants selection and mutagenesis steps where part of the population is randomly mutated through error-prone PCR. The fitness selection is obtained by exposing bacterial cultures containing the plasmids library to a certain concentration of ampicillin in the case of PSE-1 and TEM-1 (fixed for the former and variable for the latter), and kanamycin for AAC6. In all three experiments, only a subset of the rounds is sequenced.

We used two strategies to test the inferred fitness landscape: (i) by direct comparison of the predicted fitness with experimental measures of the same phenotype probed in the DE experiment; (ii) through indirect assessment of the predicted 3D structure of the protein, using the inferred epistatic interactions of the learned model. The first strategy can be applied only to TEM-1, since, to the best of our knowledge, there are no published high-throughput measures of kanamycin and ampicillin resistance for the other two proteins (AAC6, PSE-1). Moreover, being able to use DE experiments to predict a protein structure is an interesting research perspective in itself, and the main goal of both [160] and [44].

In Tab. 3.1, we report the values of the inferred $\boldsymbol{\beta}$ and single site mutation probability p for the various experiments of [160, 44], together with the regularization multiplier employed $\boldsymbol{\lambda} = (\lambda_h, \lambda_J) = (2\lambda, \lambda)$, as they were introduced in Sec. 2.3.6.

3.4.1 Prediction of mutational effects

High-throughput measurements of ampicillin resistance (viz. the same phenotypic trait under selective pressure in [44]) of single site mutants of TEM-1 are presented in [51],

Table 3.1: Values of AMaLa's time dependent parameters $\boldsymbol{\beta}$ and single site mutation rate p, as inferred on real DE data, together with the regularization multiplier λ (see Eqs. (2.55) (3.12)).

Protein	λ	$\boldsymbol{\beta}_{\text{opt}}$	p_{opt}
PSE-1	0.01	(1.0,1.7)	0.05
AAC6	0.005	(1.0,1.44,1.89)	0.05
TEM-1	0.01	(0.0,1.0,1.0)	0.017

whereas resistance measures to amoxicillin are presented in [81] (it has to be noted that the wild-type sequence in the experiment of [44] (PDB entry 1ZG4) and the one in [51] (Uniprot-P62593) have two mismatches). For the sake of clarity, we will refer to the various considered experiments as: DE-FAN for the DE data of [44], DMS-FIR and DMS-JAC for the DMS data of [51] and [81] respectively. In DMS-JAC, the fitness of the different variants is estimated as the minimum inhibitory concentration of the antibiotic necessary to neutralize the mutants, compared to the wild-type. In DMS-FIR, variants fitness is estimated as a weighted average of different antibiotic concentrations:

$$f_m = \frac{\sum_{p=1}^{13} c_p^{(m)} \log_2(a_p)}{\sum_{p=1}^{13} c_p^{(m)}}, \tag{3.15}$$

where m is the sequence index, p runs over the different antibiotic concentration, and $c_p^{(m)}$ is the number of counts of sequence m at the p-th antibiotic concentration. Eq. (3.15) actually coincides with the unnormalized fitness, the normalized version with respect to the wild-type sequence being $\phi_m = 2^{f_m}/2^{f_{wt}}$, where f_{wt} is the fitness of the wild-type sequence. In order to assess if the method was able to infer a meaningful fitness landscape, we compute the correlation of the minus selective energies E and the available fitness measurements. To be more precise, we preliminarily perform a mapping procedure of the energies onto the fitness measurements, in such a way that it was possible to employ a linear statistical estimator of the correlations such as the Pearson coefficient. Such procedure was introduced in [49], and goes as follows: the $-\Delta E_k = E(\mathbf{S}^{wt}) - E(\mathbf{S}^{(k)})$ related to K sequences for which we have associated fitness measurements are sorted in a decreasing order, and the same is done for the related fitness measurements. Consequently, we now have a rank index n, going from 1 for the highest to K for the lowest value. Finally, mutational scores are mapped over fitness measurements with the same ranking, i.e. $-\Delta E_k^{(n)} \to \phi^{(n)}$, regardless of the sequence the n-th fitness measurement $\phi^{(n)}$ is related to. The same procedure is also used to map DMS-FIR onto DMS-JAC.

The statistical energy score inferred by AMaLa on the dataset of DE-FAN highly

correlates with the DMS-FIR fitness measurements, with a Pearson correlation coefficient larger than $\rho = 0.8$, suggesting that the method is able to learn a reliable fitness landscape. This results is compared with the correlation obtained by applying PlmDCA to the MSA correspondent to the last round of the experiment, as it is reported in Fig. 3.2. It is also interesting to compare these results with the approach outlined in [49], where a Boltzmann learning DCA-based approach is applied to the PFAM β-lactamase family (PF13354). In this case, the correlation of the experimental minimum inhibitory concentration with the statistical energy score shows Pearson correlation coefficient $\rho \sim 0.7$, as shown in Fig. 3.3 panel (a). Therein, we also report in panel (b) a scatter plot of $-\Delta E$ of the inferred model against the fitness measurements of DMS-FIR.

DE-FAN data and the MSA of homologous sequences contained in PF13354 provide us with two very different datasets: the first one is a *local* exploration around the wild-type, with sequences selected to medium-low level of ampicillin selective pressure (average sequence identity of 85%), whereas the second, not surprisingly considering the extremely long time-scale involved in the evolutionary process, shows a remarkably high degree of variability (average sequence identity of 19%). Both can be used to learn a statistical model (AMaLa for DE-FAN, PlmDCA for PF13354) providing two distinct sets of model parameters that, remarkably, correlate with each other in terms of statistical energy score (see panel (a) of Fig. 3.4), and to the fitness measurements. Interestingly, the parameters of the two models do not correlate with each other (see panels (b)) in Fig. 3.4) and consequently, they provide very different contact predictions when used to infer structure information as outlined in the next section. We do not have a clear interpretation of this intriguing result.

3.4.2 Contact prediction

DCA is a powerful tool to extract structural properties from MSA's of evolutionary-related protein sequences. However, to show its full potential, MSA's of at least 10^3 sequences must be used. For many protein families, the number of homologous sequences available from public databases (e.g. PFAM or UNIPROT) is not sufficient to obtain a reliable folding structure using DCA predictions. Thus, the question of whether one can use artificially created sequences from DE experiments to extract structural information, has a very interesting practical purpose as discussed in [160] and [44]. In both papers, the authors apply two similar pseudo-likelihood based inference strategies (the PlmDCA algorithm in [44] and EV-coupling algorithms in [160]) to learn a GPM from the last sequenced round of the experiment. Only one of the two experimental work [160] reaches a precision sufficient to correctly fold the protein.

Here, we propose a different approach that leverages the sequencing information from all rounds of the DE experiment. AMaLa, instead of focusing only on the final round of the in-vitro Darwinian dynamics, indeed utilizes the whole time series. We hypothesize that being able to analyze all available data (as opposite to the use of just the last sequenced round) through a model that explicitly, albeit effectively, takes into

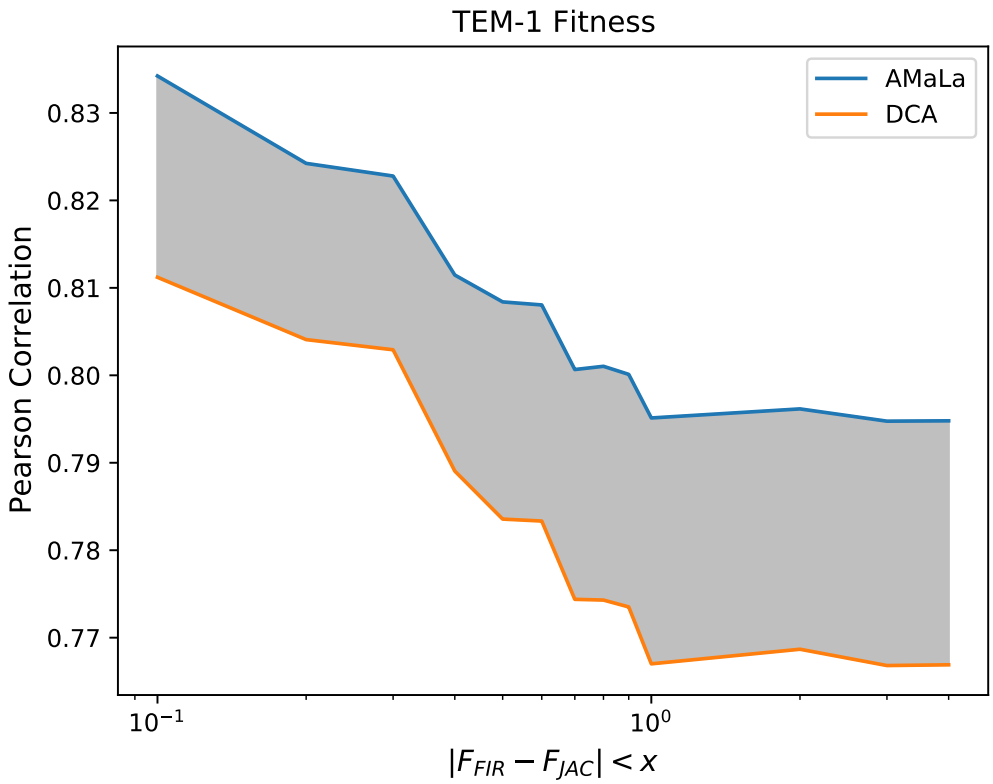
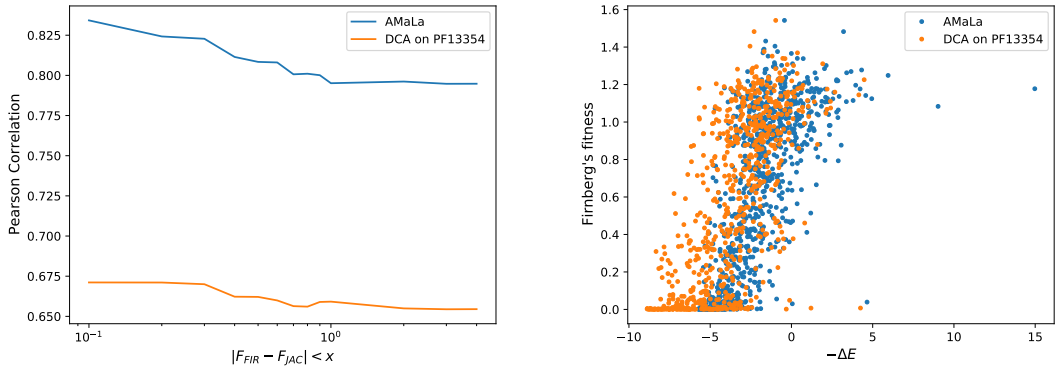


Figure 3.2: Comparison between AMaLa and PlmDCA for the reconstruction of TEM-1 β-lactamase DMS, from the experimental data of DE-FAN. AMaLa is inferred using all sequenced rounds, whereas PlmDCA is inferred over the last round only. The plot displays the Pearson correlation between energies and the measurements of DMS-FIR. Such measurements are preliminarily mapped over DMS-JAC data, following the same procedure proposed in [49]. Moreover, also the inferred energies are mapped over the fitness measurements. This allows to rely on the Pearson metrics to assess correlations between the two quantities. The discrepancy threshold between the two datasets is reported on the horizontal axis. Only data points that differ less than x are used to compute correlations for a given vertical slice. From the plot it emerges how reducing the discrepancy provides a performance improvement in terms of correlation.

account both mutation and selection steps, could in principle generate a more accurate model of the selection process, providing at the same time better structural information.

We assessed the quality of the DCA scores derived from AMaLa and PlmDCA by comparing the predicted contact map with the true one obtained by the PDB structure of the protein (see Sec. 2.4.1). The results are shown in Fig. 3.5. From the sensitivity plots, we see that, independently of the inference strategy, the predictions for PSE-1 are more accurate than the ones for AAC6. However, if we concentrate on AAC6, AMaLa



(a) (b)

Figure 3.3: Correlation between inferred energies and the DMS-FIR dataset. The same mapping procedure explained in Fig. 3.2 is used. Panel (a) shows the trend of the Pearson correlation obtained as a function of this discrepancy threshold. Namely, correlations are referred to energies inferred over DE-FAN dataset via AMaLa (blue line), and over PFAM PF13354 protein family via PlmDCA (orange). The plot displays a significant discrepancy between the two curves. On panel (b) the scatter between minus the energies (not mapped), and the fitness measurements of DMS-FIR is reported, for a discrepancy threshold between minimum inhibitory concentrations equal to $x = 0.8$.

predictions turn out to be more accurate. As the study of controlled artificial datasets presented in Sec. 3.5 seems to indicate, we expect AMaLa to provide better results with respect to PlmDCA when two conditions occur: (i) selection has a relatively weak effect compared to mutation, (ii) not too many rounds of the experiment are performed, so that the Jukes-Cantor modeling of the mutation process remains a good approximation. Indeed, this scenario entails that the selective factor $e^{-\beta(t)E(\mathbf{S})}$ can be interpreted as a perturbative contribution on the almost exact JC modeling. The first of the two conditions certainly holds for both proteins, since antibiotic concentration is slightly above the minimum inhibitory one (6μg/ml for PSE-1 and 10μg/ml for AAC6), while mutation rates are approximately the same for the two. Consequently, since the PSE-1 experiment takes place over 20 rounds, while AAC6 just over 8, we expect to obtain better results in comparison with PlmDCA for the latter rather than the first.

Nonetheless, the results obtained on PSE-1 show comparable precision of the two approaches, with AMaLa correctly predicting a higher number of contacts before the first error. Besides, looking at the predicted contact map, the contacts predicted by PlmDCA are mainly close to the polypeptide backbone, while the AMaLa ones are spread over all the contact map, providing long-range predictions that are more important for constrained molecular-dynamics simulations.

In complete analogy with what already observed in [44], when the same approach is used for TEM-1 dataset of Fantini et al., neither model is able to provide statistically

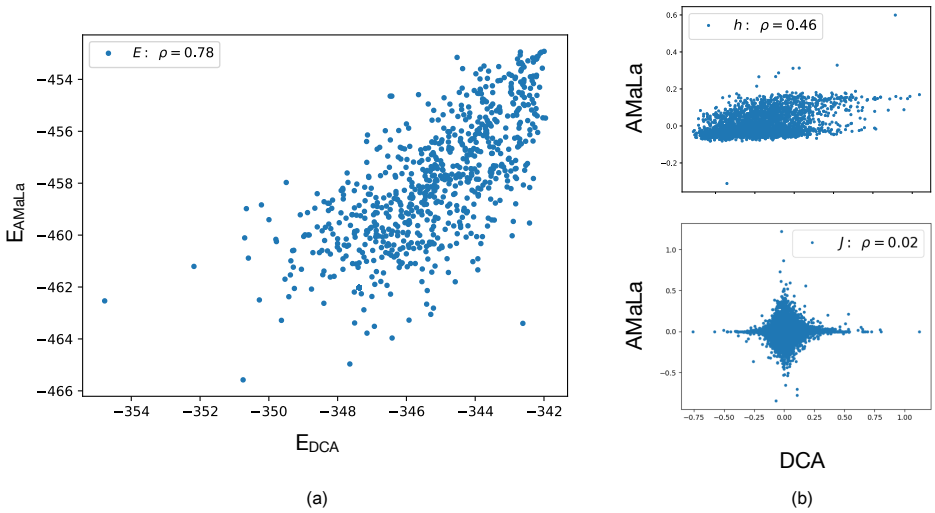


Figure 3.4: Model comparison between standard PlmDCA performed over the homology family (PF13354) and AMaLa, inferred over DE-FAN data restricted to residues corresponding to the homologs alignment. Panel (a) shows the scatter between the resulting total energies. Panel (b) displays the scatter plots of individual parameters. In the upper plot is reported the scatter among single site fields **h**, and in the lower the one among pair interaction couplings **J**. Even if energetic parameters displays separately either low ($\rho_h = 0.46$), or no correlation at all ($\rho_J = 0.01$), the resulting energies are nonetheless significantly correlated, the Pearson coefficient being $\rho_E = 0.77$. This underlines how the quantity encoding the relevant phenotypic information is indeed the total energy.

relevant contact predictions. The reason can be related to the different choice of the trade-off between selection strength and mutation rate compared to Stiffler et al., as pointed out in [15]. It is remarkable that, while the model predicts correctly the fitness direct measurements as shown in Fig. 3.2 and 3.3, it fails at providing structural information.

Interestingly, in [160] the authors report that the ep-PCR introduces approximately 3-4% amino acid substitutions per round from which we can estimate a mutation rate of $p_{\text{true}} \simeq 0.035$. We can compare it with the maximum-likelihood values inferred by AMaLa, that are $p_{\text{infer}} = 0.05$ for PSE-1, $p_{\text{infer}} = 0.055$ for AAC6, both comparable with the experimentally estimated one.

3.5 *In-silico* DE experiments

The results on AAC6, PSE-1, and TEM-1 clearly indicate how different experimental conditions (in particular the choice of the mutation rate and the selective pressure)

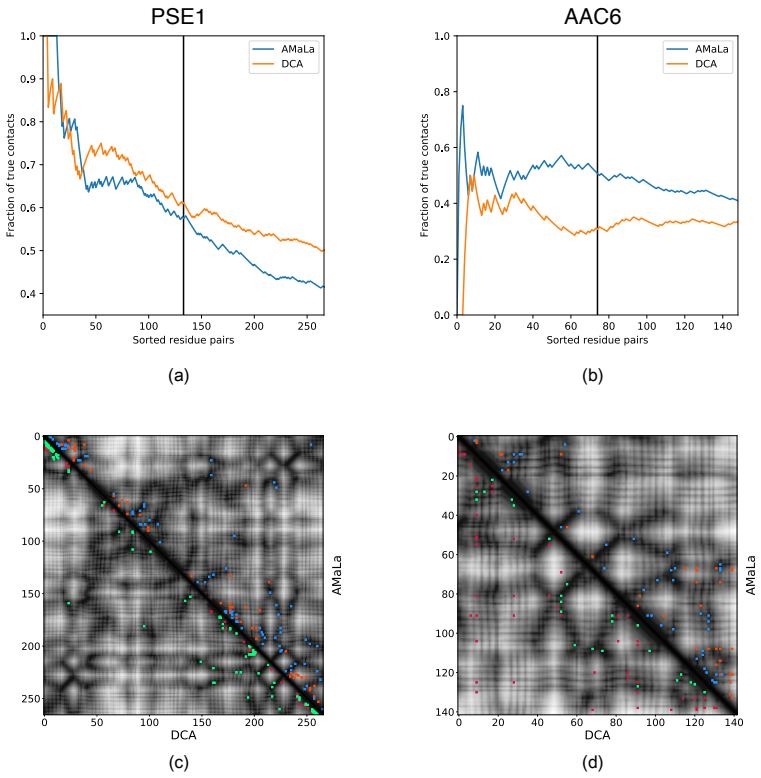


Figure 3.5: Top: sensitivity plot for contact prediction, via parameters inferred on PSE-1 and AAC6 [160] datasets. Blue curve: the score is computed as the Frobenius norm of the couplings inferred with AMaLa method. Orange curve: the score is computed as the Frobenius norm of the couplings inferred with standard pseudo-likelihood maximization approach. On panel (a) we have the result for PSE-1. At the $L/2$-th ranked residue pair AMaLa provides $\mathrm{AUC}(L/2) = 0.71$, $\mathrm{PPV}(L/2) = 0.58$, whereas PlmDCA yields $\mathrm{AUC}(L/2) = 0.72$, $\mathrm{PPV}(L/2) = 0.61$. Panel (b) shows the sensitivity plot for AAC6. In this case AMaLa yields at half of the length $\mathrm{AUC}(L/2) = 0.51$, $\mathrm{PPV}(L/2) = 0.51$, whereas for PlmDCA we have $\mathrm{AUC}(L/2) = 0.34$, $\mathrm{PPV}(L/2) = 0.31$. Bottom: contact maps up to $L/2$ predictions. In the upper-right half is reported the results related to AMaLa, whereas in the lower-left is the prediction provided by PlmDCA. Correctly predicted contacts are colored in green/blue, while wrong prediction are reported in red/orange for PlmDCA/AMaLa respectively. Panel (c) reports the result for PSE-1. Even if DCA provides both higher AUC and PPV, AMaLa seems to predict more long range contacts. A similar outcome, although less pronounced, can be appreciated in panel (d), which shows the contact map related to AAC6.

impact on the ability of the inference algorithm to predict either functional or structural properties. In particular, the interplay between the mutation rate and the selective

pressure determines the different dynamical regimes where the assumptions at the basis of the modeling could be more or less verified. To understand the limits of AMaLa, we simulated in-silico DE experiments at different regimes of selective pressure and mutagenesis, as thoroughly discussed in the following subsection 3.5.1.

3.5.1 Experiment simulation

To simulate DE experiments we define a dynamical process that mimics the mutation and selection steps occurring in a real experiment. The two fundamental parameters in the generation of synthetic data are: (i) the mutation probability p, (ii) the strength of the selective pressure $\tilde{\beta}$. Increasing it the selective pressure increases. The observable quantities are akin to the actual laboratory experiment: $N^{(m,t)}$ is the number of clones of variant m present at round t for $t \in \{t_1, \dots, t_f\}$. The total number of clones is kept fixed along the simulation and equal to $\sum_{m=1}^{M^{(t)}} N^{(m,t)} = N_{\text{tot}} = 2 \times 10^7$.

Mutations are drawn with the following strategy. Firstly, the number of sites to be mutated is extracted according to a binomial process defined by the mutational probability p:

$$P(\#\text{mut} = k) = \binom{L}{k} p^k (1-p)^{L-k}. \tag{3.16}$$

Then, for each selected site the new mutations are extracted uniformly over the possible $1/(q-1)$ different amino acids. This process either generates new variants or increases the abundances of already present ones.

Afterwards, we simulate the selection step by associating a *survival* probability $P_{\mathcal{S}}(\mathbf{S}^{(m)})$ to each variant m via a Fermi-Dirac probability in the rare binding regime [47]:

$$P_{\mathcal{S}}(\mathbf{S}) = \frac{1}{1 + e^{\tilde{\beta}[E_{\text{teacher}} - \tilde{\mu}]}}, \tag{3.17}$$

which is defined by the teacher energy E_{teacher}, which possesses the same functional form of Eq. (3.2) and defines the ground-truth fitness landscape. The auxiliary parameters $\tilde{\beta}$ and $\tilde{\mu}$, respectively play the role of an inverse temperature and a chemical potential. In particular, the choice of the latter is fundamental to guarantee that the simulation is performed in the rare binding regimes, i.e. when Eq. (3.17) can be approximated by a Boltzmann weight proportional to $\exp\left\{-\tilde{\beta}\left[E_{\text{teacher}}(\mathbf{S}^{(m)}) - \tilde{\mu}\right]\right\}$. Typical numerical values employed in the simulations are around $\tilde{\mu} \sim -18.6$, for which the chemical potential is lower than the wild-type sequence energy.

From the set of variants produced by the mutation process, a subset $n^{(m,t)}$ of surviving clones is selected according to a binomial process defined by:

$$P_B(n^{(m,t)} | N^{(m,t)}) = \binom{N^{(m,t)}}{n^{(m,t)}} P_{\mathcal{S}}(\mathbf{S}^{(m)})^{n^{(m,t)}} (1 - P_{\mathcal{S}}(\mathbf{S}^{(m)}))^{N^{(m,t)} - n^{(m,t)}}. \tag{3.18}$$

Finally, the population of clones that survived the selection step is amplified up to a fixed number N_{tot} according to the following multinomial distribution:

$$P_A(\mathbf{N}^{(t)}|\mathbf{n}^{(t)}) = \frac{N_{\text{tot}}!}{\prod_{m'=1}^{M^{(t)}} N^{(m',t)}!} \prod_{m=1}^{M^{(t)}} \left(\frac{n^{(m,t)}}{n_{\text{tot}}}\right)^{N^{(a,t)}} . \tag{3.19}$$

In addition, we randomly sample $R_{\text{tot}} = 10^6$ sequences out of the $\mathbf{N}^{(t)}$ present variants to introduce the sampling noise and simulate the effect of the sequencing.

In Fig. 3.6, a pictorial representation of the whole pipeline for the generation of simulated data is reported.

The setup of the simulation parameters was chosen with the aim to be as close as possible to a real experiment and not to introduce unnecessary artificial features. The teacher model for the ground truth fitness landscape is obtained by the inference of a Potts model on a Deep Mutational Scan (DMS) experiment [56]. The inference method used to obtain the teacher model is the one presented in [47], which has been shown to be able to infer accurate fitness landscapes. In the considered DMS experiment, the WW domain of the hYAP65 protein has been mutated and selected to bind to its cognate peptide ligand. The mutated part of the protein has a length of $L = 25$ amino acids.

In all experiments we kept the teacher energy parameters and the initial wild-type sequence fixed. We used subsets of variable size among the total of simulated rounds (typically including between 2 and 5 rounds).

3.5.2 Results on synthetic data

The performances of the inference method are assessed in terms of the correlation between teacher and student energies, computed over a test set of sequences not used to train the model. In Fig. 3.7 we display the retrieval of the true fitness as a function of the mutation rate (panel (a)) and selective pressure (panel (b)). In both cases we observe the existence of an optimal value for both tuned parameters. Interestingly, above the optimal mutation rate, the correlation tends to flatten at a value which is not far from the optimal one, ensuring that AMaLa's sweet spot for inference (at fixed selective pressure) is in general towards a high mutation rate regime. Just as a reference to real DE experiments, the mutation rate reported in [160] is $p_{\text{true}} \simeq 0.035$.

Unfortunately, we do not have access experimentally to a quantitative assessment of the strength of the selective pressure, making a direct comparison with experiments difficult. The method works at intermediate selective pressure as the selection tend to undermine its assumptions (see section 3.2). Indeed, when the selection strength is too low (depending on the time scale of the experiment), the sequence dynamics is dominated by genetic drift and, not surprisingly, the correlation between teacher and student degrades. The degradation of the performance observed for higher selective pressure is due to a combination of effects: on the one hand, we expect that in the limit of $\beta \to \infty$ only the lowest energy sequence generated in the mutation step would

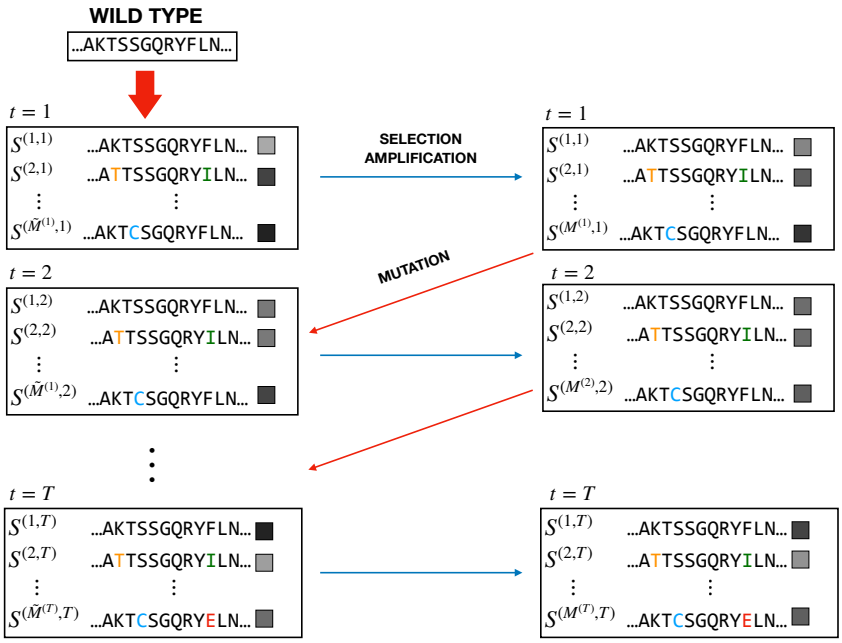


Figure 3.6: Schematic representation of the pipeline for the generation of in-silico data. The starting point is a host of copies of the wild-type sequence. The total abundance N_{tot} remains constant through the entire simulation. Then, mutagenesis is performed over this collection of wild-types, obtaining a library of $\tilde{M}^{(1)}$ unique sequences. Mutated residues are colored, and the gray-scale boxes indicate the abundances related to each unique sequence (increasing population going from black to white). This library is then subjected to both a selection and an amplification step. As a consequence, the abundances vary according to sequences fitness. Moreover, since the number of unique sequences may change, we have a new library size labeled by $M^{(1)}$. The described steps represent the fundamental unit of the simulation, which is then realized by cycling multiple times this block. Consequently, we obtain two temporal series of alignments: one which stems from mutagenesis $\{\tilde{\mathbf{N}}^{(1)}, \tilde{M}^{(1)}; ...\tilde{\mathbf{N}}^{(T)}, \tilde{M}^{(T)}\}$, and the other from selection and amplification $\{\mathbf{N}^{(1)}, M^{(1)}; ...\mathbf{N}^{(T)}, M^{(T)}\}$. Since experimental libraries are typically sequenced after selection only, we retain just the second series of alignments to construct our data sample. To be more precise, a further subsampling process is performed yielding the trajectory of *reads* $\{\mathbf{R}^{(1)}, M^{(1)}; ...\mathbf{R}^{(T)}, M^{(T)}\}$.

survive, making any inference unfeasible. On the other hand, at intermediate but high selective pressure, we expect that the consensus sequence starts drifting significantly from the initial wild-type sequence, making the mutational contribution in Eq. (3.1) an inaccurate description of the purely mutational step.

In DE experiments, one of the limiting factor is the number of selection rounds that can be sequenced (and that therefore can be used for the inference). In the following,

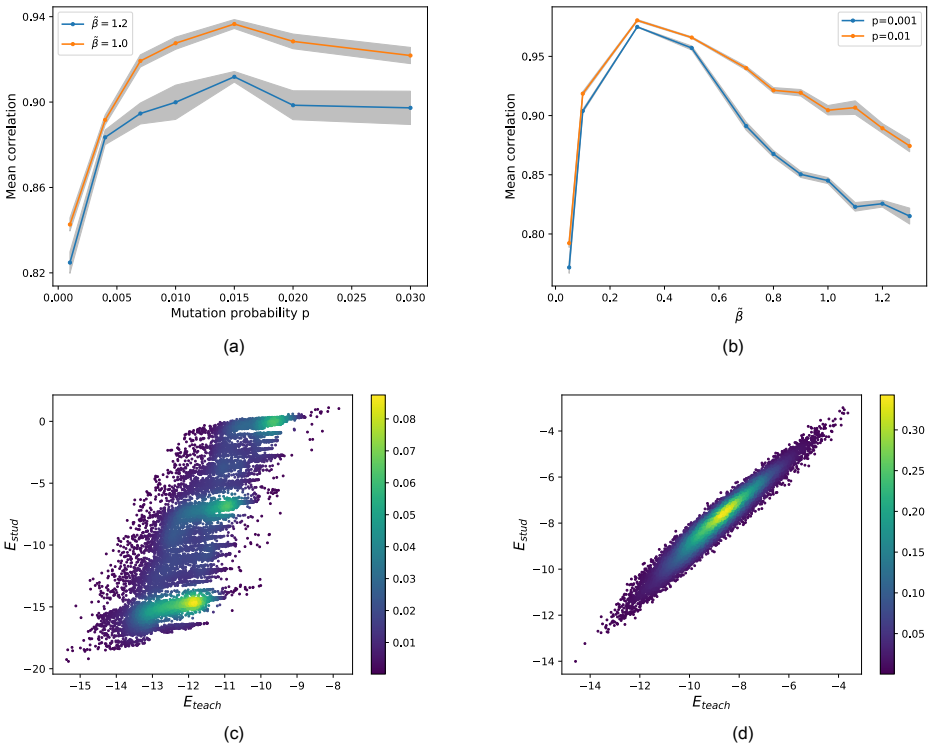


Figure 3.7: Simulated experiments varying the mutation rate and the selective pressure. On the top panels the Pearson correlation between true and predicted fitness (teacher and student model energy) are shown. In order to estimate statistical fluctuations, several replicas of the experiment have been realized for each point (N_{sim} = 20-40), reporting mean and standard deviation. On panel (a) the mutation rate is varied for two choices of the selective pressure: $\tilde{\beta}_{\text{high}} = 1.2$ (blue), $\tilde{\beta}_{\text{low}} = 1.0$ (orange). Conversely, on panel (b) the selective pressure is varied at two fixed mutation rates: $p_{\text{low}} = 0.001$(blue), $p_{\text{high}} = 0.01$(orange). An optimal mutation rate seems to emerge, with the mean Pearson coefficient which flattens for higher mutations rate. Performances appear to decrease with increasing selective pressure. Moreover, the curve coinciding with p_{high} displays significantly higher correlations. The bottom panels show two examples of density scatter plot between true (x-axis) and inferred (y-axis) energies over the test set. Two limiting cases are shown: high selective pressure and low mutation rate in panel (c) ($p = 0.001$ and $\tilde{\beta} = 1.2$), and low selective pressure and high mutation rate in panel (d) ($p = 0.05$ and $\tilde{\beta} = 0.5$) where AMaLa recovers the right fitness landscape. In the former case the Pearson's correlation is $\rho = 0.81$, while in the latter is $\rho = 0.97$.

we will assume we can afford only between two and five rounds of sequencing, and we ask which rounds bring the larger information content.

As shown in Fig. 3.8 panel (a), the correlation between the teacher and student energies of the test set increases as a function of the last round time for PlmDCA, whereas AMaLa performance behaves just in the opposite way: early rounds give better results. This finding is particularly interesting as it suggests that by using AMaLa one could achieve better inference results by performing just a limited number of rounds, i.e. with a lower experimental effort. However, AMaLa overall performances are always better than PlmDCA for any choice of the sequencing round.

Furthermore, the in-silico experiments can be used to investigate the generalization power of the learned fitness landscape beyond the local region of sequence space probed by the experiment. More specifically, how far from the wild-type an inference strategy is still able to predict the fitness? To answer to this question we trained both AMaLa and PlmDCA on rounds (2,4,8). Then, we tested the teacher-student energy correlation over randomly extracted sequences at Hamming distance up to the whole sequence length (here $L = 25$). As shown in Fig. 3.8 panel (b), we can see that for both low and high mutation rate regimes: (i) over the whole range of Hamming distance from the wild-type sequence, AMaLa always shows higher correlation with the teacher energies, (ii) PlmDCA performances seem to degrade more slowly as a function of the distance from the wild-type sequence.

We further investigated the features of the simulated data, specifically focusing on the Hamming distance from the wild-type sequence. Namely, in Fig. 3.9, we show the trend of the average Hamming distance from the wild-type in panel (a), and the Hamming distance of the consensus sequence from the wild-type in panel (b), both as a function of round time. In panel (a), we compare the outcome of the simulation with the values of the average Hamming distance expected for a purely mutational process. For a small mutation probability, as it is for this specific simulation in which $p = 0.01$, we notice how this trend is basically linear in time. On the other hand, we point out how, for this specific simulation, selection act as a mild modification of the purely mutational process in terms of Hamming distance. The trend of the Hamming distance between the consensus sequence and the wild-type in panel (b) displays a peculiar step behavior. Notably, selection generates a drift effect that pushes the consensus progressively further from the wild-type, although this is not necessarily a general feature, but depends on the specific choice of the teacher parameters and on the interplay between selection and mutagenesis.

While finalizing this work, we became aware of a similar approach described in [15]. Their strategy relies on a simultaneous treatment of selection and mutagenesis. The fitness approximated landscape is inferred over the homologous alignment, specifically via bmDCA of a GPM. Such energy provides a proxy for fitness, and a tool to probe context dependent mutations, for the energy function includes couplings between different residues. Indeed, a MCMC is implemented to generate a library which mimics the one that would have been obtained in a real DE experiment. The elementary step of this MCMC includes both mutation and selection. The energy variation of single site mutations with respect to the wild-type defines the acceptance probability (which depends

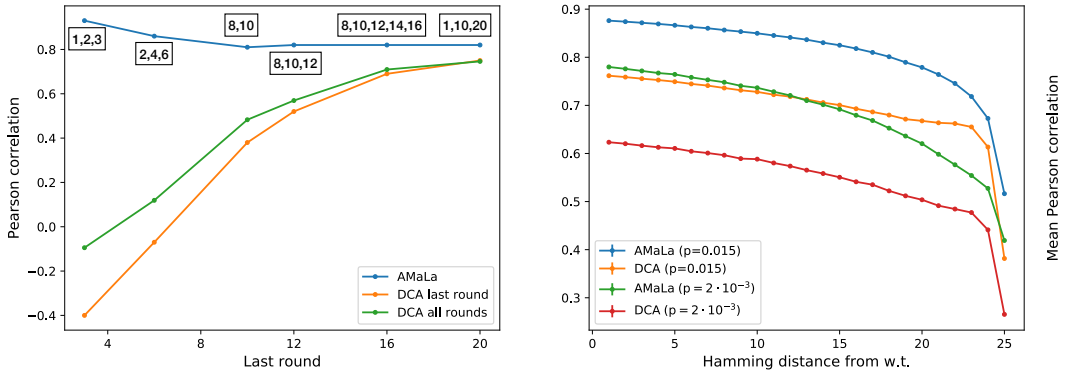
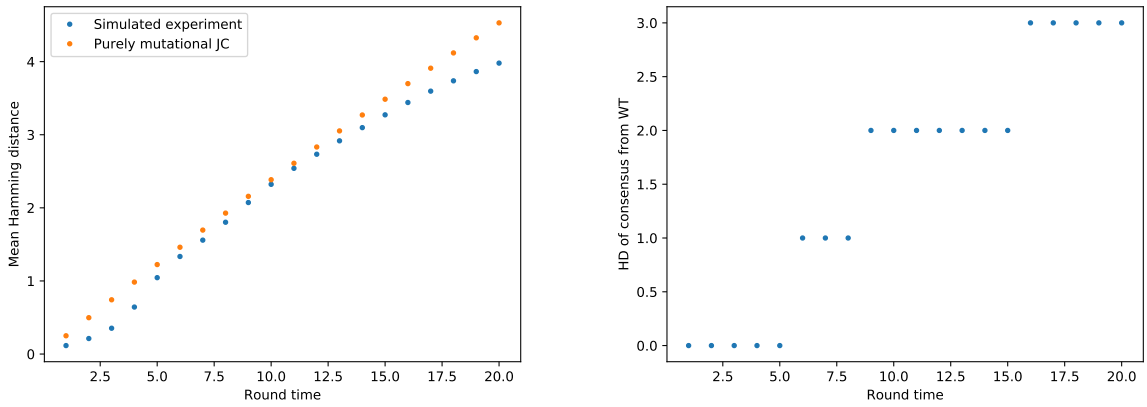


Figure 3.8: Dependence of the inferred signal on the number of rounds and Hamming distance from the wild-type. Panel (a): Pearson correlations for different rounds choices. Comparison between PlmDCA on the last round (or all rounds) and AMaLa. The sequenced rounds are: (1, 2, 3); (2, 4, 6); (8, 10); (8, 10, 12); (8, 10, 12, 14, 16); (1, 10, 20). PlmDCA significantly depends on the number of performed rounds, not significantly inferring the fitness landscape up to round $\sim$ 16. On the contrary, AMaLa provides predicted energy functions highly correlated with the fitness even for low number of performed selection rounds. Panel (b): degradation of the mean Pearson correlation between inferred and true energies, as a function of the Hamming distance from the wild-type sequence. Two different simulations are considered: high ($p = 0.015$) and low ($p = 0.002$) mutation probability. AMaLa predictions are systematically better than PlmDCA, while the latter display a slower decrease in correlation augmenting the distance from wild-type.

only on the a.a. sequence). On the other hand, the proposed mutations are restricted to the allowed single mismatch transitions among codons $\mathbf{c}^i = (c_1^i, c_2^i, c_3^i)$, thus involving the genomic sequence. This may suggest a possibility to improve AMaLa itself: the Hamming distance in the Jukes-Cantor contribution in Eq. (3.1) may be computed over the genome alignment. In this away forbidden transitions among a.a.'s are automatically excluded, but at the same time also multiple transitions are allowed, even if exponentially suppressed. Remarkably, the findings of [15] with respect to the optimal regime for a DE experiment agrees with the results we derived from the application of AMaLa to both in-silico and experimental data.

3.6 Conclusion and perspectives

In this chapter we presented AMaLa, an unsupervised inference method tailored for DE, a specific kind of laboratory evolution experiments. The most relevant features of the



(a) (b)

Figure 3.9: Panel (a): evolution of the average Hamming distance for in-silico generated data (blue points), compared to the analytical prediction of the purely mutational process (orange points) with the same single site mutation probability $p = 0.01$. On the x-axis is the round time, whereas on the y-axis we have the empirical average $\overline{\mathrm{h_D}(\mathbf{S}, \mathbf{S}^{wt})}$ (blue) and the purely mutational theoretical average $\left\langle \mathrm{h_D}(\mathbf{S}, \mathbf{S}^{wt}) \right\rangle$ (orange), according to Eq. (C.7). Panel (b): evolution of the Hamming distance of the consensus sequence from the wild-type. The trend displays a step behavior, remaining constant for some rounds and then increasing of a unit. Along the constant distance interval, the consensus sequence may nonetheless vary.

presented methods are: (i) the possibility to include all the experimentally sequenced rounds in the inference process via an effective modeling of the underlying dynamical process (ii) the fact that the statistical model is practically independent of enrichment information, allowing it to be applied to experiments characterized by severe undersampling regimes.

In our work, we showed how AMaLa can be successfully applied to DE data both for fitness landscape reconstruction and for assessing contact prediction. In particular, we pointed out how local datasets as the one provided by DE experiments might provide a better inference platform when compared to MSA of homologous sequences for fine scale fitness landscape reconstruction. The converse can instead be stated for contact prediction. In this perspective, the AMaLa method represents a promising tool for inferring accurate local fitness landscapes, or to provide residue contact predictions for those proteins missing an homology alignment.

Furthermore, our analysis suggests what might be the optimal experimental conditions for extracting relevant information from the data. These coincide with a low selective pressure and high mutation probability regime, allowing for a broad exploration of the sequence space, at the same time assuring sequence functionality.

Concerning the future perspectives, several improvements and extensions of the model are possible. Firstly, it would be interesting to reframe the mutational model at the genome level, in order to correctly take into account transitions between amino acids. Similarly, the modeling of the selection part could be ameliorated so to describe more faithfully the selection mechanism of the specific considered experiment. For instance, in the case of cell-based bacterial platforms probing antibiotic resistance, the response to antibiotics is typically described by a sigmoidal function. Such knowledge could be incorporated into the modeling to improve its predictive power.

The presented statistical model is specifically suited for laboratory evolution experiments that start from a unique wild-type sequence. It would be interesting to attempt to generalize the formalism to the case in which the initial configuration is a general library made of a set of sequences. A possibility in this perspective is provided by the inference method presented in the next chapter 4.

The natural next-step would be to develop a more faithful description of the experiment dynamics, that might allow to go beyond the approximate selection-mutation independent scheme which is at the basis of AMaLa. In this regard, population genetic like models represent an interesting approach, which was successfully applied to viral strain evolution [151, 152, 93]. The most significant factor undermining the straightforward application of such methods is the fact that they significantly rely on the availability of accurate population measurements, which are usually missing in the context of DE experiments.

Chapter 4

Inference on screening experiments: betaDCA

In this chapter we will describe betaDCA, a novel unsupervised inference method for protein sequence data produced by screening experiments. Such inference method is closely related to AMaLa Ch. 3, although possessing some crucial differences.

4.1 Motivations

Over the last few years, the development of increasingly accurate high-throughput biochemical assays with massive parallel sequencing techniques has established large-scale genetic screening as a fundamental tool for the investigation of the relationship between evolution and fitness [34, 6, 99, 81, 153, 101, 54, 102, 115, 137, 154, 1, 90, 136, 159, 175, 84, 94, 98, 38, 62, 141, 155, 134].

As more high-throughput sequencing data of screened libraries are available, new computational methods for accurate statistical modeling of the genotype–phenotype association are actively developed. The majority of these methods rely on *enrichment ratios* in order to extract relevant phenotypic information from the data. However, as already mentioned in Sec. 3.1, there are experimental conditions for which such ratios are either very noisy or simply not available. In particular, the AMaLa method has been developed for overcoming this issue in the case of Directed Evolution experiments, in which mutagenesis does not allow for a simple enrichment ratio inference scheme.

For DMS experiments, the main obstacle to develop a population based statistical modeling is represented by subsampling of the variants' pool, which can be caused by a multitude of different experimental setups. For instance, an initial library size that is too large compared to the sequencing coverage, as it is often the case when dealing with the sequencing antibodies repertoire [13]. Moreover, too strong a selective pressure can isolate only very fit sequences, not allowing for a broad survey of the sequence space. Analogously, a very high number of selection rounds can generate similar effects in

the last steps of the experiment. Finally, it could be that the very shape of the fitness landscape does not allow for a proper exploration of the sequence space, as it might happen if the dynamics get trapped in a very sharp local fitness maximum.

In this perspective, betaDCA represents a simple alternative to populations based methods for fitness landscape estimation in all the cases in which accurate abundance information are not accessible, or when the data are characterized by an elevated noise component. To do so, we model the experiment as an annealing process (see Sec. 3.2) from an initial sequences distribution. The annealing process progressively isolates the most fit sequences for the probed selective trait by lowering a statistical temperature.

In Sec. 4.2, we will give a detailed explanation of the method, discussing its advantages and limitations. Afterwards, we assess the capability of betaDCA to reconstruct meaningful fitness landscapes for various kinds of experimental setups in Sec. 4.4, making comparisons with alternative available methods such as: PlmDCA, *deterministic rare binding* (DRB) [47] and AMaLa.

4.2 Modeling

In this section, we describe a general mathematical framework and a statistical inference approach that applies to several experimental setups and biological systems. Datasets that can be used include, among others, protein screening experiments with one or multiple panning rounds, whether they include or not mutagenesis, repertoire sequencing samples at different times and infection stages and others. In all these cases, we observe a set of samples of protein variants (or several proteins) under selective pressure, which shapes the sequences distribution over time.

To describe the method we consider, for simplicity, a protein screening experiment without mutations that takes place over several panning rounds $t \in \{t_0, ..., t_f\}$, where $t_0 = 0$ refers to the initial unscreened library. The model is defined by the probability of observing a sequence $\mathbf{S}$ at time t, $P_t(\mathbf{S})$, and the survival probability $Q_{t,t-1}(\mathbf{S})$ between round $t-1$ and t. The fundamental hypothesis of the model concerns the functional dependence of the survival probability:

$$Q_{t,t-1}(\mathbf{S}) \propto \exp\left[-\alpha_{t,t-1} E(\mathbf{S})\right], \tag{4.1}$$

which is defined by two quantities, a time dependent factor $\alpha_{t,t-1}$ modeling the selective pressure between the two rounds, and a time independent function $E(\mathbf{S})$, associating an *energy* to the protein sequence $\mathbf{S}$. The functional form of Eq. (4.1) is inspired by a population genetics formalism [113], in which the argument of the exponential directly coincides with the sequence fitness (apart from a temporal dependence). Consequently, we assume that the energy E contains the information about the functional properties of the protein variants, and we parametrize it as a GPM:

$$E(\mathbf{S}) = \sum_{i=1}^{L} h_i^E(\sigma_i) + \sum_{i=1}^{L-1} \sum_{j=i+1}^{L} J_{ij}^E(\sigma_i, \sigma_j). \tag{4.2}$$

Whether or not the energy contains epistatic contributions depends on the specific considered dataset. We indicate the set of parameters defining Eq. (4.2) with $\boldsymbol{\theta}^E$. Employing Eq. (4.1), we can express $P_t(\mathbf{S})$ as:

$$\begin{aligned} P_t(\mathbf{S}) &= Q_{t,t-1}(\mathbf{S}) P_{t-1}(\mathbf{S}) \\ &= P_{t_0}(\mathbf{S}) \prod_{t'=t_0+1}^{t} Q_{t',t'-1}(\mathbf{S}) \\ &\propto P_{t_0}(\mathbf{S}) \left[\mathrm{e}^{-E(\mathbf{S})}\right]^{\sum_{t'=t_0+1}^{t} \alpha_{t',t'-1}}. \end{aligned} \tag{4.3}$$

Note that the product runs over all the rounds of the experiment, and not only the sequenced ones. Eventually, we were able to express $P_t(\mathbf{S})$ as a product of the initial configuration probability $P_{t_0}(\mathbf{S})$ and the factor $\mathrm{e}^{-E(\mathbf{S})}$, the latter raised to the sum of the selective pressures at each transition. We redefine such sum as:

$$\beta(t, t_0) = \sum_{t'=t_0+1}^{t} \alpha_{t',t'-1}. \tag{4.4}$$

Eq. (4.4) can be interpreted as a fictitious inverse temperature, accounting for the overall selective pressure between t_0 and t. In the absence of mutations, Fisher's fundamental theorem of evolution states that the α's are a decreasing function of time [36]. In the following we will indicate the fictitious inverse temperature as $\beta(t, t_0) \equiv \beta(t)$. Employing Eq. (4.4), we can rewrite Eq. (4.3) as:

$$P_t(\mathbf{S}) \propto \mathrm{e}^{-\beta(t)E(\mathbf{S})} P_{t_0}(\mathbf{S}), \tag{4.5}$$

from which it emerges how selection acts as a simulated annealing stochastic process defined by the temperature $\beta(t)$, pushing the distribution towards the minima of E. Notably, we do not need any explicit assumption on the specific temporal dependence of the inverse temperature, as the β factors are directly inferred from the data. Fig. 4.1 shows a pictorial representation of the overall modeling of the experimental screening process.

At $t_0 = 0$, $P_0(\mathbf{S})$ is the distribution of the variants in the initial library, which is unrelated to the selection process, for no round of screening has been performed at that stage. We describe the distribution of the initial variants by means of a further energy function $G(\mathbf{S})$, so that the time dependent probability finally becomes:

$$P_t(\mathbf{S}) = \frac{\mathrm{e}^{-\beta(t)E(\mathbf{S}) - G(\mathbf{S})}}{Z_t}, \tag{4.6}$$

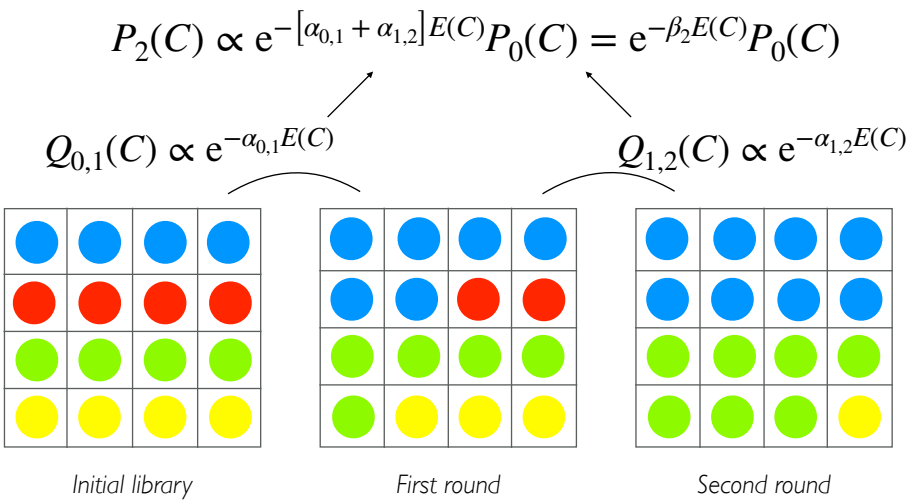


Figure 4.1: A simplified portrayal of the modeling of the selection process. Each color represent a different variant. Starting from the initial distribution of variants, the probability of observing a sequence in a subsequent round is shaped by the selection process, defined by the energy function $E(\mathbf{S})$. The selective pressure at each transition is encoded in $\alpha_{t-1,t}$, and the overall fictitious inverse temperature is given by the sum of all transitions $\beta(t) = \sum_{t'=1}^{t} \alpha_{t'-1,t'}$. As in a simulated annealing stochastic process where the temperature is progressively lowered, $\beta(t)$ increases with time, thus constraining the variant probability distribution to peak around the fittest sequences.

where Z_t is a time dependent normalization factor $Z_t = \sum_{\{\mathbf{S}\}} \exp\left[-\beta(t)E(\mathbf{S}) - G(\mathbf{S})\right]$, the sum running over all possible sequence configurations.

The statistical modeling described so far resembles the one outlined in Sec. 3.2. The crucial difference between the two lies in how the *round zero* library is modeled. Indeed, if AMaLa automatically imposes that at $t = 0$ the library collapses onto a wild-type sequence (Eq. (3.5)), subsequently taking into account the dynamics of the mutation process, betaDCA ascribes the whole dynamics to the selection process, considering the contribution related to $P_0(\mathbf{S})$ to be fairly time independent. In particular, we assume the initial library statistics to be described by another GPM:

$$G(\mathbf{S}) = -\sum_{i=1}^{L} h^G(\sigma_i) - \sum_{i=1}^{L-1}\sum_{j=i+1}^{L} J_{ij}^G(\sigma_i, \sigma_j). \tag{4.7}$$

In order to distinguish between the two GPM's, we add explicit apexes to the two sets of parameters $\boldsymbol{\theta}^E = \{\mathbf{h}^E, \mathbf{J}^E\}$, $\boldsymbol{\theta}^G = \{\mathbf{h}^G, \mathbf{J}^G\}$. Depending on the specific experimental setup, it might suffice to consider Eq. (4.7) as a profile model, for the parameterizations

of the E and G energies need not to be the same.

For DE experiments, the first available sequenced round is taken as if it were the initial library, so that for these data $t_0 \neq 0$. In particular, one pretends that the variants sequenced at t_0 did not undergo any round of selection, discarding the information that could be in principled extracted from selection up to that round.

To model the selection process, we introduced an energy function $E(\mathbf{S})$ that associates to each sequence $\mathbf{S}$ a fitness energy value that is time-independent as it has been done for the AMaLa modeling (Eq. (3.2)). As a consequence, the selection process favors variants based on the same fitness over the whole experiment. While this property is factual for most screening experiments, it might turn into a working approximation in other contexts, such as for instance viral strain evolution, in which the interaction between the virus and the host, as well as the action of vaccines makes the fitness time dependent [100].

4.3 betaDCA inference

In order to infer the model parameters $\boldsymbol{\theta}^E$, $\boldsymbol{\theta}^G$ and $\boldsymbol{\beta}$ we need to define the objective function to be optimized, which in our case coincides with the log-likelihood of the observed data. Such data are related to the subset of rounds of the experiment that have been sequenced, which we indicate as $\{\tau_0, \tau_1, \dots, \tau_f\} \subset \{0, t_1, \dots, t_f\}$.

$$\begin{aligned}\mathscr{L}\left[\boldsymbol{\theta}^E, \boldsymbol{\theta}^G, \boldsymbol{\beta}\right] &= \sum_{t\in\{\tau_0,\dots,\tau_f\}} \sum_{m=1}^{M^{(t)}} w^{(m,t)} \log P_t(\mathbf{S}^{(m)}) \\ &= -\sum_{t\in\{\tau_0,\dots,\tau_f\}} \sum_{m=1}^{M^{(t)}} w^{(m,t)} \left[\beta(t) E(\mathbf{S}^{(m)}) + G(\mathbf{S}^{(m)}) + \log Z_t\right], \end{aligned} \tag{4.8}$$

where as in Eq. (3.3), $w^{(m,t)} = N^{(m,t)} / \sum_{m'=1}^{M^{(t)}} N^{(m',t)}$ coincides with the normalized number of counts. Differently from the AMaLa modeling though, the annealing temperature has now a further component, as also the initial library (or the round taken as it) is part of the dataset. Namely, we have $\boldsymbol{\beta} = \left(\beta(\tau_0), \beta(\tau_1), \dots, \beta(\tau_f)\right) = (\beta_0, \beta_1, \dots, \beta_T)$, with T equal to the number of sequenced rounds. As in Sec. 3.3, Eq. (4.8) is approximated with a log-pseudo-likelihood objective function, so to avoid the necessity to compute the complete partition function Z_t.

At fixed $\boldsymbol{\beta}$ the single site objective function can be expressed as the sum of the minus-log-pseudo-likelihood and two regularization functions on the energetic parameters:

$$g_r\left(\boldsymbol{\theta}_r^E, \boldsymbol{\theta}_r^G; \boldsymbol{\beta}\right) = -\sum_{t=\{\tau_0,\dots,\tau_f\}} \sum_{m=1}^{M^{(t)}} w^{(m,t)} \log P_t\left(\sigma_r = \sigma_r^{(m,t)} | \boldsymbol{\sigma}_{\setminus r} = \boldsymbol{\sigma}_{\setminus r}^{(m,t)}\right) + R_r^E\left(\mathbf{h}_r^E, \mathbf{J}_r^E\right) + R_r^G\left(\mathbf{h}_r^G, \mathbf{J}_r^G\right), \tag{4.9}$$

where the regularization functions $R_r^E\left(\boldsymbol{\theta}_r^E\right)$ and $R_r^G\left(\boldsymbol{\theta}_r^G\right)$ are l_2 contributions:

$$\begin{aligned} R_r^E\left(\boldsymbol{\theta}_r^E\right) &= \lambda_h^E \sum_{a=1}^{q} h_r^E(a)^2 + \lambda_J^E \sum_{i\neq r} \sum_{a,b=1}^{q} J_{ri}^E(a,b)^2 \\ R_r^G\left(\boldsymbol{\theta}_r^G\right) &= \lambda_h^G \sum_{a=1}^{q} h_r^G(a)^2 + \lambda_J^G \sum_{i\neq r} \sum_{a,b=1}^{q} J_{ri}^G(a,b)^2. \end{aligned} \tag{4.10}$$

The two sets of regularization multipliers $\boldsymbol{\lambda}^E$ and $\boldsymbol{\lambda}^G$ do not generally need to be equal.

As for the AMaLa modeling, before inferring the time independent parameters we need to fix the inverse annealing temperature. For the present case, $\boldsymbol{\beta}$ has $T+1$ components, because the initial library is also included. However, the first two components, related respectively to the initial library and to the first screening round, are fixed to $\beta(\tau_0) = 0$ and $\beta(\tau_1) = 1$. Indeed, the initial library has not yet been subjected to any selection process, and at the same time, we have the freedom to rescale the selective pressure at any round according to the first one.

Then, it is necessary to infer the remaining $T-2$ components $\left\{\beta(\tau_2), \dots, \beta(\tau_f)\right\} = \{\beta_2, \dots, \beta_T\}$. To do so, we follow the same approach outlined in Sec. 3.3. We can perform repeated optimizations of the objective function over a grid of points. Each of these points is defined by a possible combinations of the chosen values for the inverse temperature components, and it amounts to find the optimal sets of $\boldsymbol{\theta}^E$ and $\boldsymbol{\theta}^G$ at fixed $\boldsymbol{\beta} = \left(0, 1, \beta_2^{(i)}, \beta_3^{(j)}, \beta_T^{(l)}\right)$, with the indeces i, j spanning over the possible values for each component. The optimal inverse annealing temperature is defined as: $\overline{\boldsymbol{\beta}} = \text{argmin}_{i,j,\dots,l}\left\{\min_{\boldsymbol{\theta}_E,\boldsymbol{\theta}_G}\left[\sum_{r=1}^{L} g_r\left(\boldsymbol{\theta}_E, \boldsymbol{\theta}_G; \left(0,1,\beta_2^{(i)}, \beta_3^{(j)}, \dots, \beta_T^{(l)}\right)\right)\right]\right\}$.

Alternatively, a gradient descent algorithm can be implemented, in such a way to update the $\boldsymbol{\beta}$ components and the energetic parameters $\boldsymbol{\theta}^E$ and $\boldsymbol{\theta}^G$ asynchronously. This alternate optimization is related to the fact that Eq. (4.9) is not contemporarily convex with respect to $\boldsymbol{\theta}^E$ and the annealing temperature 3.3. The two approaches are totally equivalent, and yield the same optimal $\overline{\boldsymbol{\beta}}$.

4.4 Results

In this section, we will present the results obtained from the application of betaDCA to different type of screening experiments, such as DMS, antibody repertoire sequencing

and DE experiments, as the versatility of the method represents its main strength. Since both DMS and DE experiments were described in Sec. 1.3, we redirect the reader there for a thorough description of these two experimental settings.

4.4.1 Deep Mutational Scanning

In the analysis of the inference of betaDCA on DMS experiments, we considered three different datasets [56, 18, 175].

In [56], the phenotypic probed trait is the binding affinity of the human WW-domain ($L = 25$) with its peptide ligand. More than 6×10^5 unique variants are generated in the initial library, which comprises almost all single point mutations, a fourth of the double and almost 2% of all three point mutations. Then, six rounds of phage display screening are performed, and rounds 3 and 6 are sequenced, together with the initial library.

In [18], a short segment of the CDR3 antibody heavy-chain (L=4) is probed for binding against two targets, polyvinylpyrrolidone (PVP) or a short DNA loop of 9 nucleotides. Around the CDR3 segment, 24 different libraries are constructed according to possible scaffolds determined by the V_H variable region, which are subsequently screened for three rounds of selection.

Also Wu et al. [175] focused on four protein residues, exhaustively generating all possible mutations. Such residues belong to the IgG-binding domain (GB1), that is screened for binding onto an immunoglobulin fragment target for a single round of selection.

In all these experiments, accurate measurements of abundances are available. Consequently, it is possible to estimate a sequence fitness in terms of enrichment ratios between neighboring rounds. In particular, we employ the empirical quantity *log-selectivity* $\boldsymbol{\Theta}$ as a reference for experimental fitness:

$$\log\left[\frac{N^{(m,t)}}{N^{(m,t-1)}}\right] = \Theta^{(m)} + \alpha^{(m,t)} + \epsilon^{(m,t)}. \tag{4.11}$$

Since in a DMS experiment the whole diversity is introduced in the initial library, an index m running over the $M^{(0)}$ unique sequences at the beginning of the experiment is sufficient to identify them univocally at any subsequent rounds. The log-ratio of the abundances is modeled by means of three different contribution: one is the log-selectivity $\Theta^{(m)}$, which is supposed to be a time independent feature. Then, $\alpha^{(m,t)}$ serves to quantify the amplification factor, which may vary among rounds. Finally, the term $\epsilon^{(m,t)}$ is needed in order to account for stochastic fluctuations. All three sets of parameters are inferred via linear regression on the empirical data.

In order to assess the performances of the method, we split the data into a *train* and *test* set, respectively of sizes 4/5 and 1/5 of the whole dataset. The model is learnt on the training set only, whereas the test one is used as a benchmark. In particular, for these three DMS experiments betaDCA's accuracy is estimated in terms of the Pearson correlation between the inferred selective energies E and the empirical log-selectivities

over the test set sequences. In Fig. 4.2, we show the comparison of betaDCA performances with DRB, a state of the art method to perform inference on DMS screening experiments.

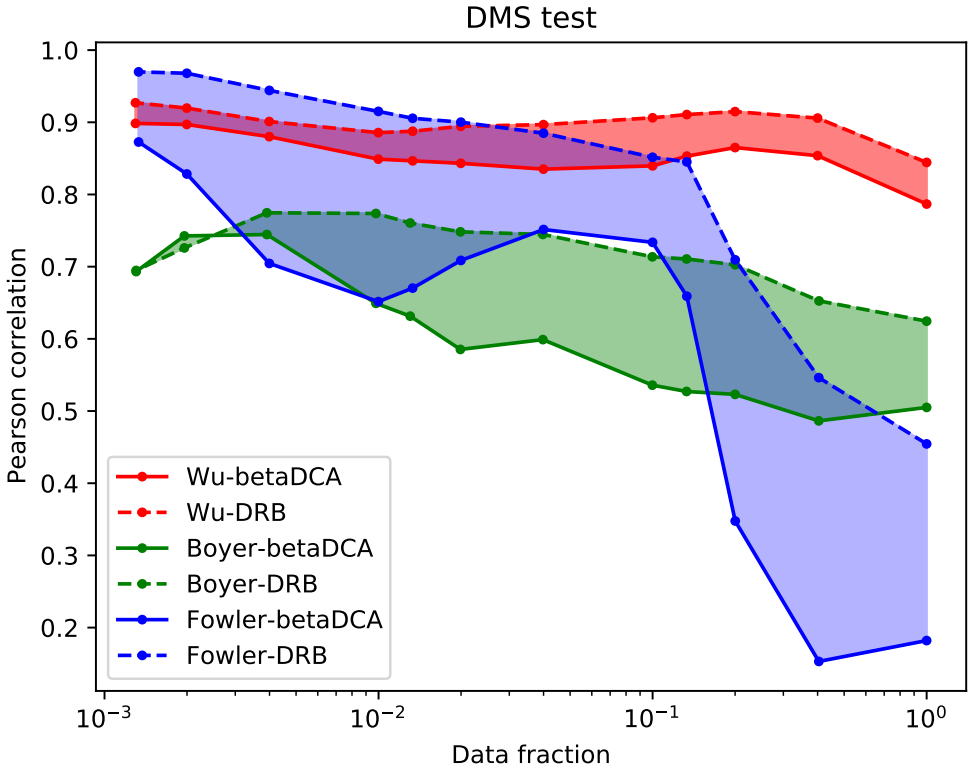


Figure 4.2: Pearson correlation between the inferred selective energies and empirical log-selectivities computed on the test set sequences for betaDCA and DRB. The correlation is plotted as a function of the retained data fraction. Sequences are pruned by progressively leaving out the one having a higher uncertainty of empirical log-selectivity, as it can be computed by fitting Eq. (4.11). betaDCA trend is plotted as points+solid lines, whereas DRB is identified by points+dashed lines. The three experiments are distinguished with different colors: red [175], green [18] and blue [56]. Shaded regions point out the discrepancy between the two methods for the same dataset.

For all three experiments, DRB provides the best correlations with empirical selectivities, since the method is able to leverage the abundances dynamics, that are accurately provided by all [175, 56, 18]. Interestingly, even if systematically worse, betaDCA provides Pearson correlations comparable with DRB, especially for the data of [175], in which the broad coverage of the sequence space compensates the lack of enrichment information in the betaDCA method. Our claim is that betaDCA might be able to eventually surpass the performances of DRB on less accurate datasets, as for instance in

severe undersampling or extremely noisy regimes. This is the case for antibody repertoire datasets, which will be treated in the next section 4.4.2.

We also tested the performances of PlmDCA inferred over the last round only. The method is not able to produce positive correlations for any of the considered experiments, and the trend as a function of the number of retained sequences is erratic and dominated by random fluctuations. Such an outcome is somewhat expected, since the DCA approach is thought to be applied to quasi-equilibrium MSA, which are quite far from the data produced by a DMS.

4.4.2 Antibody Repertoire Sequencing

In Sec. 1.4, we gave a brief and general introduction about the immune system, focusing on antibodies and how they evolve as a consequence of the exposition of an organism to a specific pathogen, so that this process could be interpreted as a complex in-vivo screening experiment. In this perspective, betaDCA represents an interesting tool to quantify the statistical difference in the repertoire before and after exposure to the pathogen. To be more precise, by applying the method to antibody repertoire sequencing data, we want to infer the probability of an antibody to be the outcome of an immune response. Once the model is trained, we obtain a parametrization of this probability function that can be used to design novel antibodies with high affinity to the target.

In order to test the betaDCA method we employ two kinds of datasets: a *negative* set related to the naïve repertoire, and a *positive* set, in which the repertoire is sampled after exposition to a specific antigen. Both kind of repertoires belong to mice of the BALB/c strain. Such animals are generated by an in-bred process that lead them to possess the same genetic material. Consequently, samples from separate mice can be considered as different realizations of the same repertoire, up to differences due to the phenomenon of genetic recombination.

The negative or background dataset comprises immunoglobulin G (IgG) secreted by plasmablast cells of three unimmunized mice [88]. The data are publicly available at the *Observed Antibody Space* [91], and includes around 20 thousands sequences of IgG heavy chains. On the other hand, the positive dataset refers to the repertoires of mice immunized with respect to two different antigens: Tetanus toxoid (TT) and Glucose-6-Phosphate Isomerase (GPI). These data were produced by Gerard et al. in [66], employing a microfluidic platform named CelliGO, which is used to isolate IgG's possessing a high binding affinity towards the antigen. Alternatively, one could directly sample the repertoire after immunization [7].

The basic mechanism of the platform is the following: single B-cells are encapsulated into oil droplets together with two different biomarkers and some paramagnetic colloidal nanoparticles. The biomarkers are necessary to verify if the B-cells are able to secrete IgG, and if they do, whether the antibody is able to bind to the antigen. The droplets are scanned one by one, passing firstly through a magnetic field which aligns

the beads, and then through a laser producing a fluorescent signal, allowing to distinguish among the phenotypic properties of the IgG's. In order to realize single-cell sequencing, the sorted cells are subsequently re-compartmentalized in another droplet, where the B-cell membrane undergoes a lysis process, and the available RNA is reverse transcribed to cDNA, ultimately providing the V_H antibody sequences.

After the sorting and the sequencing process, some IgG clones are selected to be tested against binding onto the respective antigens, i.e. TT and GPI. Specifically, 27 clones were selected from sorting against TT and 13 for GPI. The chosen phenotypic quantities are the EC50 for TT and the dissociation constant K_d for GPI. The former is defined as the antibody concentration for which the *response* is half of the possible maximum, where the response is usually defined by a change of color of the testing solution, which is more intense the more antibodies bind onto the antigens. On the other hand, dissociation constants are obtained from titering curves realized at different antigen concentrations, K_d being defined as the antigen concentration such that the concentration of bound antibody equals half the concentration of unbound ones.

Given these two unimmunized/immunized repertoire dataset, the general idea is to model the probability of observing an antibody in the positive set as the product of the following two probabilities: the background probability, i.e. the probability to observe an antibody in the negative set (the unimmunized repertoire), and the selection probability that describes the overall effective process of the immune response (together with the microfluidic platform in this case). In panel (a) of Fig. 4.3, we report a cartoon representation of the immunization process with respect to its effect on the mapping between fitness and sequence space, where fitness has to be intended as the antibody affinity for binding the antigen. The sequence space is fictitiously represented as a two-dimensional space, whereas fitness is visualized by means of contour lines. The selection process acts pushing antibodies to a region of sequence space characterized by a higher affinity with respect to the target.

The first task we tested was to assess the ability of the model to discriminate between binders and not binders. For this purpose, we split the positive and background datasets into a training and a test set to validate the classification predictions. This choice was due to the lack of a large list of IgG labeled as binders of the two antigens (TT and GPI). Thus, we use the positive and background sets as a proxy for binders/not binders labels. Fig. 4.3 panels (b) and (e) show the results of the classification problem. The sequences with low background energy are likely to be present in the unimmunized repertoire, while the selection energy accounts for the probability of an antibody being part of the immune response to the specific antigen. The model can discriminate remarkably well the binders (of the positive set) and non-binders (of the background set), as displayed by the ROC curves (panel (e)) in the test sets of both targets (AUC 0.98 for TT and 0.89 for GPI).

We subsequently considered the phenotype measurements realized in [66], namely the EC50 values for the TT immunization dataset and the dissociation constants K_d for the GPI exposed one. The antibodies employed to realize such measurements are

sampled fairly in the sequence space from the positive dataset, and lay on the high-selectivity model energy region. The results show that inferred selection energy correlates with K_d GPI measures, while there is no significant correlation between selection energy and EC50 in the TT case (see Fig. 4.3 panels (c) and (d)).

In summary, the tests we performed show that: (i) the models energies discriminate well between the positive and the negative sets: the method predicts with high fidelity whether a sequence comes from an unimmunized repertoire or it is the output of an immunized and binders-enriched one; (ii) The inferred selection energy contains information on the experimental binding energy in one case (GPI, Kd measures), while in the other (TT), the selection energy does not correlate with the EC50 values.

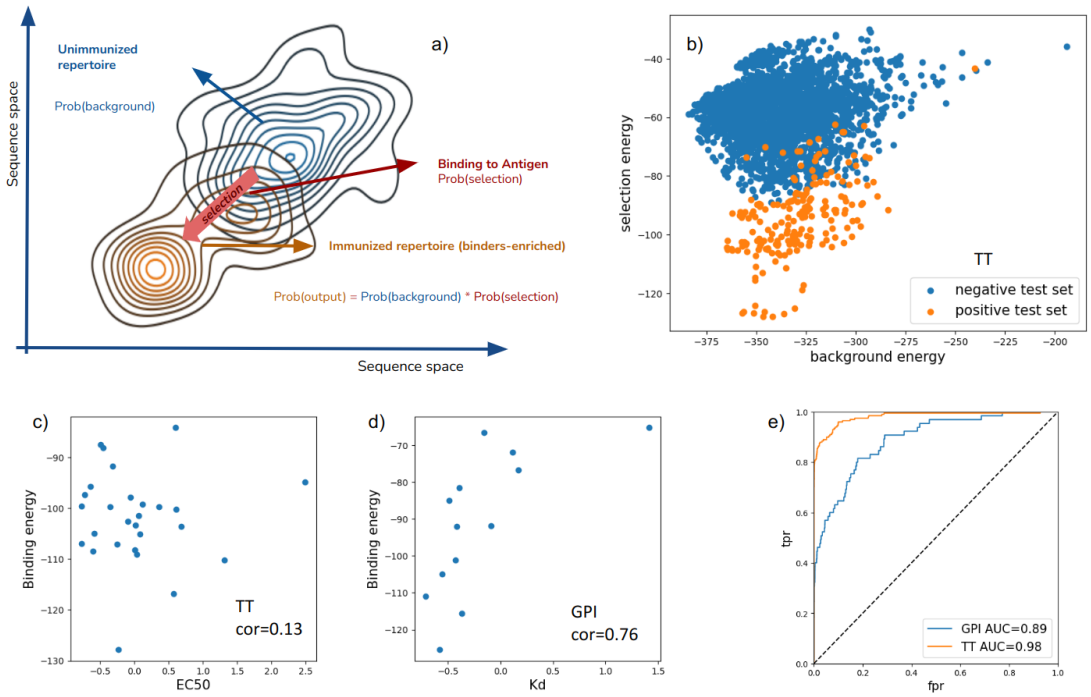


Figure 4.3: Main results obtained by the application of betaDCA to antibody repertoires. Panel (a): pictorial representation of the immunization process. The exposition to the antigen acts as a selective pressure that shifts the sequences towards a region of high affinity for antigen binding. Panel (b): discrimination between negative and positive test by employing the two model energies E and G. Panel (c) and (d): scatter plot between selection energy E and phenotypic traits such as EC50 and dissociation constant K_d. The latter is significantly correlated with the selective energies, whereas the former is not. Panel (e): ROC curves related to the discrimination between negative and positive sequences for the exposition to the two antigens GPI and TT. Values of the AUC are reported in the plot insert.

4.4.3 Directed Evolution

The flexibility of betaDCA allows to apply it also to experiments in which new protein variants are introduced at each round via mutagenesis. In this case, as discussed in [145], we cannot compute the enrichment ratios and selectivities, due to the introduction of mutations and to the severe undersampling regime in which such experiments are realized. Consequently, an approach which is not population-based is required. If AMaLa explicitly models the mutational process by means of a Jukes-Cantor like contribution (see Sec. 3.2.2), this is not the case for betaDCA, in which the G Hamiltonian explicitly models only the statistics of the initial library. Thus, the method can be applied heuristically, checking a-posteriori its capability to infer a meaningful landscape. Practically speaking, for a DE dataset, the first round of the experiment is considered as the initial library, as if that was the starting point of the experiment. Then, since G is learnt over all the sequenced rounds, sequences that were not present in the initial library can be interpreted as if they were actually there from the very beginning, but were not observed due for instance to undersampling effects. In Fig. 4.4, we report a scatter plot between G energies and the Hamming distance of a random pool of sequences from the AAC6 experiment from the corresponding wild-type. The two quantities are strongly correlated, with a Pearson correlation coefficient 0.94, suggesting that the G energy is actually able to effectively model the initial library of the experiment.

We basically repeat the same analysis carried out in 3.4, but for sake of clarity, we briefly recall the main features of the analyzed experiments and the testing strategies. The authors of [44, 160] screen proteins responsible for antibiotic resistance in bacteria: TEM-1 and PSE-1 variants of the β-lactamase family and AAC6 protein of the acetyltransferase family. Starting from a wild-type protein, error-prone PCR creates new mutants at each round. Subsequently, the library undergoes a selection step in which bacteria equipped with the mutants are exposed to an antibiotic-rich environment. This cycle of mutagenesis and screening is repeated multiple times, and for a subset of the panning rounds a sample of the library is sequenced.

We performed two different tests to assess the inferred model. In the case of TEM-1 β-lactamase, the model energy is directly compared with independent fitness measurements related to antibiotic resistance. Specifically, we considered two experimental papers [81, 51]. In [81] variants fitness is quantified in terms of *minimum inhibitory concentration* (MIC), that is, the minimum antibiotic concentration necessary to neutralize bacteria equipped with that variant. On the other hand, in [51], the authors directly measured the gene fitness as a weighted average of 13 different antibiotic concentrations, with the weights defined by the number of copies of each variant at the various concentrations. For our analysis, we mapped the measurements of [51] onto those of [81], following the procedure outlined in [49] (see Sec. 3.4.1 for a thorough discussion).

In Fig. 4.5 we reported the results obtained in terms of fitness landscape reconstruction, comparing betaDCA with both AMaLa and PlmDCA. Although the inferred

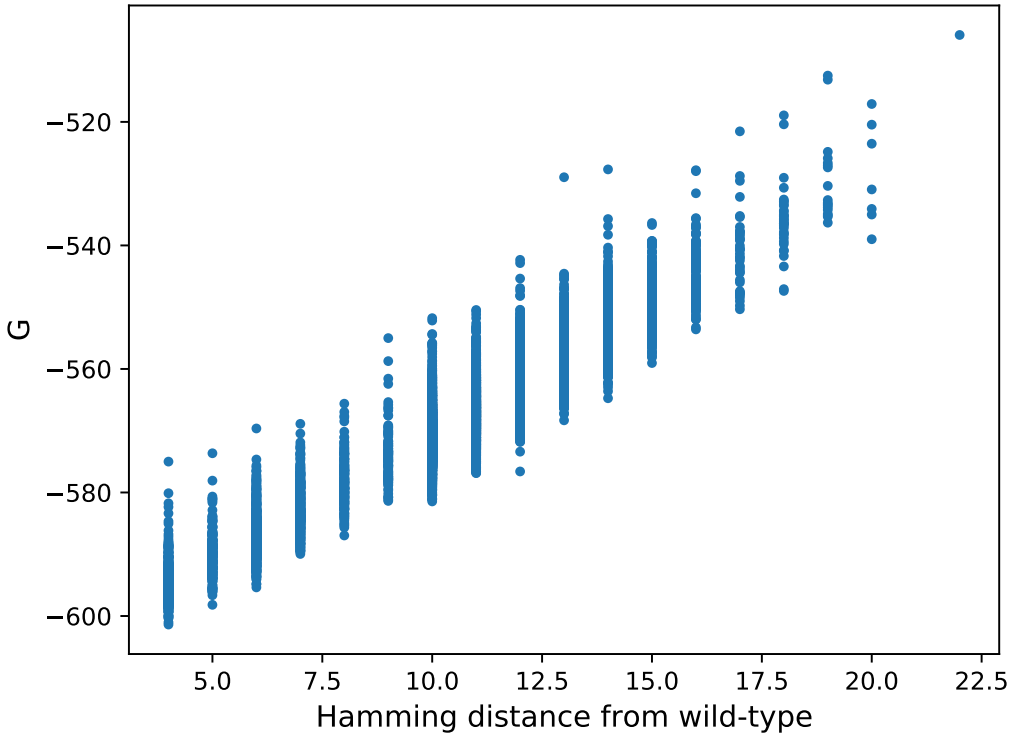


Figure 4.4: Scatter plot between G energies and Hamming distances from the wild-type, for a random pool of sequences generated in the AAC6 experiment of [160]. The two quantities are strongly correlated, as it should be if G is meant to model the experiment initial library.

energies correlate well with these independent experimental fitness measurements, the performances are worse than both AMaLa and PlmDCA.

When testing over PSE-1 and AAC6 data, for which fitness measurements are not available, the test involves the prediction of the protein structure contact map and the comparison with crystallographic studies of the protein, as described in Sec. 2.4.1. The obtained results in terms of sensitivity plots and contact maps are reported for both proteins in Fig. 4.6. Interestingly, and as opposed to the outcome for β-lactamase fitness, betaDCA displays performance comparable or slightly superior to those of AMaLa for contact prediction assessment. From panels (c) and (d) it emerges how the method is able to predict more long range contacts when compared to PlmDCA, as it was the case for AMaLa.

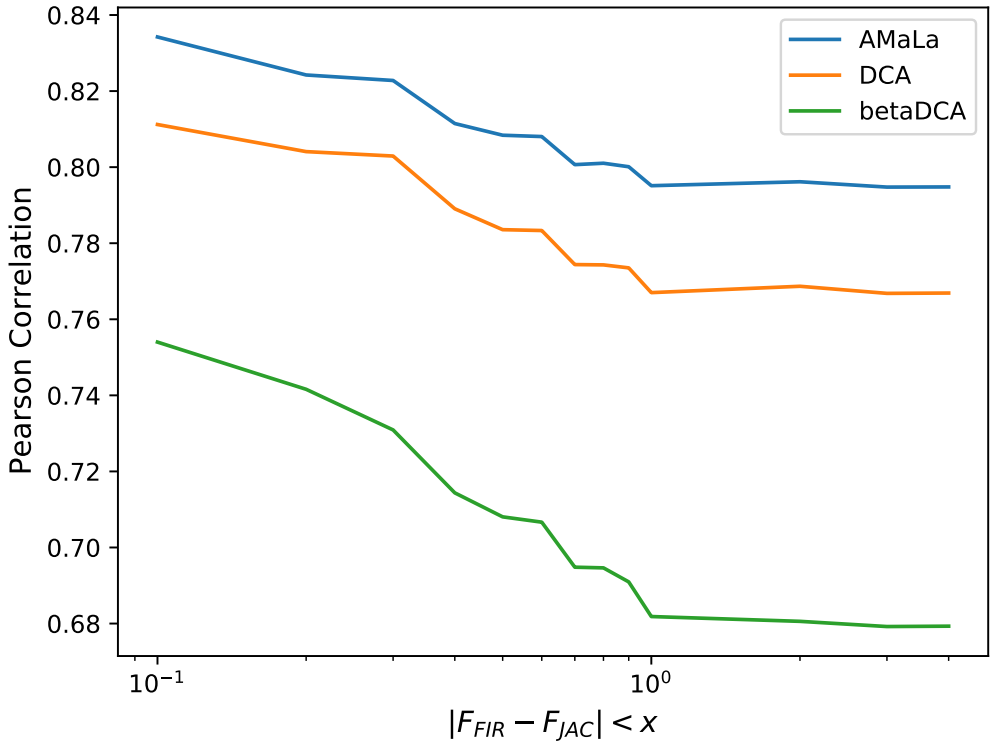


Figure 4.5: Analogous of Fig. 3.2, with the addition of the corresponding betaDCA curve. On the vertical axis we report the Pearson correlation between inferred selective energies E and the fitness measurements of [51], both mapped following the procedure outlined in [49]. On the horizontal axis, the discrepancy between the measurements of [81] and [51] is reported. Moving from right to left, disagreeing data points are progressively excluded.

4.5 Conclusions

In this chapter we presented betaDCA, an unsupervised inference methods for protein sequence data generated by screening experiments. The basic assumption of the method is that a time dependent selection process shapes the statistics in sequence space as an annealing process, in which a statistical temperature is progressively lowered. The statistics of the initial experiment configuration is also included in the modeling, which is consequently defined by two GPM energies: E accounting for the selection process, and G describing the unscreened library.

The main strength of the method is the possibility to apply it to scenarios in which accurate population measurements are missing, so that it is not possible to rely on enrichment ratios information. In this regard, betaDCA proved to be particularly effective

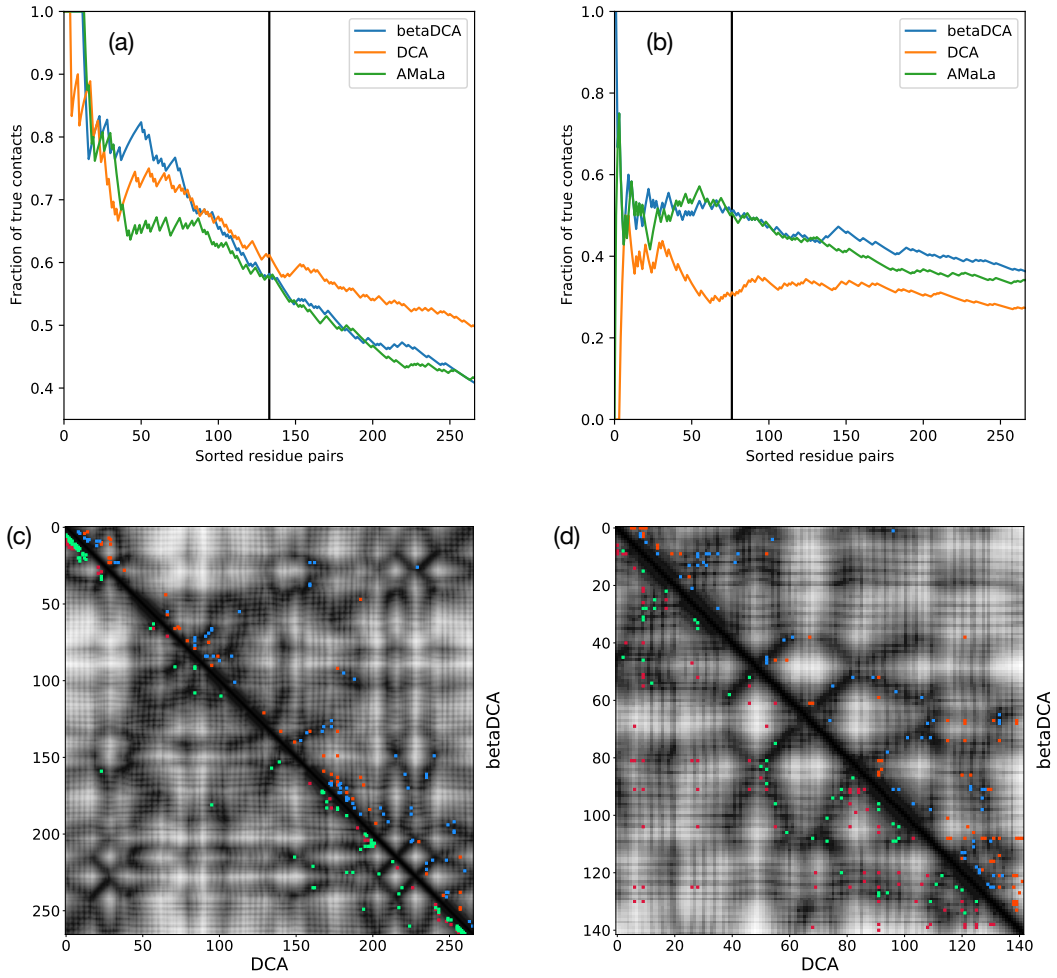


Figure 4.6: Analogous of Fig. 3.5, with the addition of the corresponding betaDCA curve. Top: sensitivity plot; the betaDCA approach provides AUC($L/2$) = 0.71, PPV($L/2$) = 0.58 for PSE-1, and AUC($L/2$) = 0.71, PPV($L/2$) = 0.58 for AAC6. Bottom: contact map up to $L/2$ predictions. In the upper-right half is reported the results related to betaDCA, whereas in the lower-left is the prediction provided by PlmDCA. Correctly predicted contacts are colored in green/blue, while wrong prediction are reported in red/orange for PlmDCA/betaDCA respectively. Panel (c) reports the result for PSE-1 and panel (d) for AAC6.

on antibody repertoire immunization data, which are usually characterized by significant noise level and for which it is not possible to access to accurate population trajectories. The method is able to successfully discriminate between sequences of the negative and positive set, and the selective energy E displays a good correlation with

the measurements of the dissociation constant K_d. When accurate abundance measurements are available, the method is still able to meaningful functional landscape, though with generally worse performances with respect to alternative methods that are able to leverage such population information. betaDCA proved as well to be a versatile method, for it can be successfully applied to DE experimental data, even though the mutational process is not explicitly considered in the modeling. In particular, the method provides strikingly good result for the contact prediction problem of the proteins studied in [160], whereas correlations with TEM-1 β-lactamase fitness turn out to be inferior with respect to both AMaLa and PlmDCA.

A possible interesting extension of the method would be the inclusion of mutational effects if present, introducing a suitable time dependence in the Hamiltonian G. This might be of particular interest, because it would represent a generalization of the AMaLa modeling to the case in which the initial library is not made of wild-type sequences only.

Chapter 5

Learning Restricted Boltzmann Machines via Expectation Propagation

In this chapter, we will present the ideas and some preliminary results related to the application of *Expectation Propagation* (EP), an iterative algorithm for approximating intractable probability distributions, to the inference problem of *Restricted Boltzmann Machines* (RBM). Thus, in Sec. 5.1 we describe RBM's architecture, pointing out their strengths and limitations. In Sec. 5.2, we analyze the EP algorithm, highlighting how it can be applied to infer RBM's. Finally, in Sec. 5.3 we present the results obtained by applying the EP based inference method to the MNIST dataset.

5.1 Restricted Boltzmann Machines

An RBM is an artificial neural network architecture that can be represented as an undirected bipartite graph made of two different type of nodes: the visible and the hidden units. The visible units usually coincide with the observed data, whereas the hidden units are necessary to identify and encode features of the data, and to model interaction among the visible units. We identify with N the number of visible units, and with M the number of hidden units, where typically $M < N$. The architecture is said to be restricted because the visible units do not interact directly with each other, but they are only coupled to hidden units. Similarly, also the hidden units do not interact among themselves. Consequently, an RBM can be identified by the following set of parameters: the weights $w_{i\mu}$, connecting visible and hidden units; the potentials or priors g_i acting on the visible units, and U_μ acting on the hidden ones. The indexes i and μ run respectively over visible and hidden units, $i = 1, \dots, N$ and $\mu = 1, \dots, M$. RBM were originally introduced in [149] with the name *Harmonium*, and can be interpreted as

a special case of a Boltzmann Machine, as they were introduced in [2]. RBM gained popularity after Hinton et al. [72] introduced an efficient learning algorithm known as *contrastive divergence* (CD), which we will describe in the following section 5.1.1.

In Fig. 5.1, we show an example of an RBM architecture.

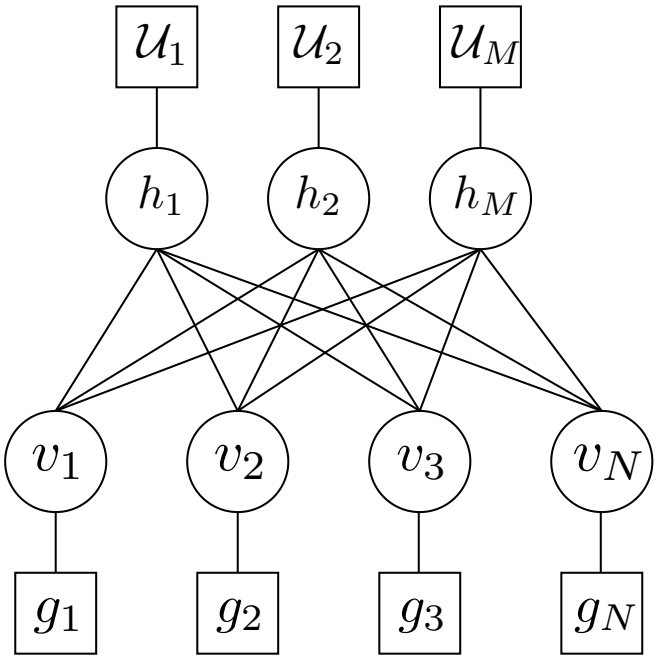


Figure 5.1: Graphical representation of an RBM. The visible nodes are linked to the potentials g_i whereas the hidden ones are linked to U_μ. Moreover, each visible unit is linked to all hidden nodes, but no internal connections are present among variables of the same type.

The graphical representation is associated to a joint probability function over visible and hidden nodes:

$$P(\mathbf{v}, \mathbf{h}) = \frac{1}{Z} \exp\left[-E(\mathbf{v}, \mathbf{h})\right], \tag{5.1}$$

where Z is the normalization factor, and $E(\mathbf{v}, \mathbf{h})$ is an energy function:

$$E(\mathbf{v}, \mathbf{h}) = -\sum_{i,\mu=1}^{N,M} v_i w_{i\mu} h_\mu + \sum_{i=1}^{N} g_i(v_i) + \sum_{\mu=1}^{M} U_\mu(h_\mu), \tag{5.2}$$

which is defined by the $N \times M$ matrix w, and the N and M components vectors of potentials $\mathbf{g}$ and $\mathbf{U}$. The type of the visible potential depends on the nature of the considered data, e.g. binary for images, Potts-field for categorical data. On the other

hand, the choice of the hidden potential is key in determining the properties of the RBM. In particular, if one wants to introduce collective interactions between the visible units, it is necessary to choose a non-quadratic potential U.

Indeed, the marginal distribution over the data is obtained by integration over the hidden units:

$$P(\mathbf{v}) = \int \mathrm{d}^M h P(\mathbf{v}, \mathbf{h}) = \frac{1}{Z} \exp\left[-\sum_{i=1}^{L} g_i(v_i) + \sum_{\mu=1}^{M} C_\mu\left(I_\mu(\mathbf{v})\right) \right], \tag{5.3}$$

where we introduced the cumulant generating function $C_\mu\left(I_\mu(\mathbf{v})\right)$:

$$C_\mu\left(I_\mu(\mathbf{v})\right) = \log \int \mathrm{d}h_\mu \mathrm{e}^{-U_\mu(h_\mu) + I_\mu(\mathbf{v}) h_\mu}, \tag{5.4}$$

and the input function for the hidden unit μ:

$$I_\mu(\mathbf{v}) = \sum_{i=1}^{N} v_i w_{i\mu}. \tag{5.5}$$

From these definitions, we see that if the hidden potentials are quadratic:

$$U_\mu(h_\mu) = \frac{\gamma_\mu}{2} h_\mu^2 - \theta_\mu h_\mu, \tag{5.6}$$

integration in Eq. (5.4) can be carried out analytically, yielding a marginal distribution:

$$P_{\text{Gauss}}(\mathbf{v}) \propto \exp\left[-\sum_{i=1}^{N} g_i(v_i) + \sum_{\mu=1}^{M} \frac{\left(I_\mu(\mathbf{v}) + \theta_\mu\right)^2}{2\gamma_\mu} \right]. \tag{5.7}$$

Eq. (5.7) contains couplings between the visible units defined as $J_{ij} = \sum_{\mu=1}^{M} \frac{1}{2\gamma_\mu} w_{i\mu} w_{j\mu}$, and represents in fact a fully connected model with pairwise interactions. However, as soon as the hidden potentials are non-quadratic, arbitrarily high order interactions among visible units are generated. Some common kind of hidden potentials in the context of RBM are:

- **Bernoulli**: $U_{\text{Ber}}(h) = -\theta h$, $h = \{0,1\}$.
- **Gaussian**: $U_{\text{Gauss}}(h) = \frac{\gamma}{2} h^2 - \theta h$.
- **ReLU**: $U_{\text{ReLU}}(h) = \begin{cases} \frac{\gamma}{2} h^2 - \theta h & h \geq 0, \\ +\infty & h < 0. \end{cases}$

Another interesting property of RBM is conditional independence, i.e. both $P(\mathbf{v}|\mathbf{h})$ and $P(\mathbf{h}|\mathbf{v})$ factorize over the respective nodes. This property derives from the peculiar RBM architecture, which possesses no connection among homologous nodes. It is particularly useful when performing sampling, because each type of units can be drawn independently. Specifically, alternate Gibbs-sampling [64] over visible and hidden units appears as the natural choice for generating equilibrium configuration from $P(\mathbf{v}, \mathbf{h})$. The sampling strategy goes as follows:

- Choose a random visible configuration $\mathbf{v}$, either extracting it from the data or generating it from scratch. From this, compute the hidden units associated inputs $\mathbf{I}^{\mathrm{h}} = w^T\mathbf{v}$.
- Sample independently the hidden units according to $P(h_\mu|\mathbf{v}) \propto \exp\left[-U_\mu(h_\mu) + I_\mu(\mathbf{v})h_\mu\right]$. This step amounts to *feature extraction*.
- Compute the visible inputs associated to the extracted hidden units $\mathbf{I}^{\mathrm{v}} = w\mathbf{h}$.
- Extract a new visible configuration according to $P(v_i|\mathbf{h}) \propto \exp\left[-g_i(v_i) + I_i(\mathbf{h})v_i\right]$.

The repetition of the previous steps eventually leads to an equilibrated sample of both $\mathbf{v}$ and $\mathbf{h}$. The presented steps also highlight the relation between visible and hidden units. Different data inputs activate different hidden units, providing a featural and lower-dimensional representation. Conversely, one can generate specific visible configurations by selecting the appropriate subset of hidden units to be activated. In this perspective, we can define two further quantities: the *transfer function* and mean hidden units activation. The former is defined as:

$$H_\mu(\mathbf{I}_\mu) = \mathrm{argmax}_{h_\mu} P(h_\mu|\mathbf{v}). \tag{5.8}$$

If the hidden unit potential derivative is an invertible function, the transfer function can also be express as $\left(U'_\mu\right)^{-1}(I_\mu)$. On the other hand, the average hidden activity is defined as:

$$\left\langle h_\mu \right\rangle_{P(h_\mu|\mathbf{v})} = \frac{\int \mathrm{d}h_\mu\, h_\mu \mathrm{e}^{-U_\mu(h_\mu)+I_\mu(\mathbf{v})h_\mu}}{\int \mathrm{d}h_\mu\, \mathrm{e}^{-U_\mu(h_\mu)+I_\mu(\mathbf{v})h_\mu}} = \frac{\partial C_\mu}{\partial I_\mu}\left(I_\mu(\mathbf{v})\right). \tag{5.9}$$

For a quadratic hidden potential, both the transfer function and the average activity are a linear function of the activation input $\mathbf{I}^{\mathrm{h}}$, whereas non-quadratic potentials generate non-linear responses.

In the following section we will describe how to train an RBM in an unsupervised fashion.

5.1.1 Learning RBM

Training or learning an RBM means inferring all its constituent parameters, that is, the set of weights w and the parameters defining the visible $\mathbf{g}$ and hidden $\mathbf{U}$ potentials. For the moment, we consider the potentials parameters as given, and we focus upon weights inference.

We follow a maximum likelihood approach, in which the objective function is defined as:

$$\mathscr{L}[w] = \frac{1}{D}\sum_{d=1}^{D} \log P(\mathbf{v}^{(d)}) = \frac{1}{D}\sum_{d=1}^{D} \log \int \mathrm{d}^M h\, P(\mathbf{v}^{(d)}, \mathbf{h}), \tag{5.10}$$

where the index $d = 1, \dots, D$ runs over the data points $\mathbf{v}^{(d)}$, which are plugged into the marginal visible probability. In order to infer the weight parameters, we have to find the set $\hat{w}$ which maximizes Eq. (5.10), and thus we need to compute the gradient:

$$\frac{\partial \mathscr{L}[w]}{\partial w_{jv}} = \frac{1}{D}\sum_{d=1}^{D} v_j^{(d)} \langle h_v \rangle_{P(h_v|\mathbf{v}^{(d)})} - \langle v_j h_v \rangle_{P(\mathbf{v},\mathbf{h})} = \overline{v_j^{(d)} \langle h_v \rangle_{P(h_v|\mathbf{v}^{(d)})}} - \langle v_j h_v \rangle_{P(\mathbf{v},\mathbf{h})}\,. \tag{5.11}$$

We notice that the gradient is the difference between two cross correlations: the ensemble average $\langle v_j h_v \rangle_{P(\mathbf{v},\mathbf{h})}$ and the average of the hidden unit over the conditional probability, whose product with the visible unit is itself averaged over the dataset $\overline{v_j^{(d)} \langle h_v \rangle_{P(h_v|\mathbf{v}^{(d)})}}$. We indicate averages computed with respect to probabilities as $\langle O \rangle$, and with respect to the data as $\overline{O}$.

The major difficulty lies in the computation of the ensemble cross correlation, which in principle requires to compute the normalization of the joint distribution. MCMC sampling methods can be used to overcome this issue and to estimate the sought moments. In this perspective, the contrastive divergence (CD) [72] and persistent contrastive divergence (PCD) [163] methods have been developed with the aim to approximate the likelihood gradient in Eq. (5.11).

Contrastive approaches

The CD algorithm was initially developed by Hinton et al. [72] and allowed for the first successful application of an RBM architecture to a dataset of consistent size, namely the MNIST one. The underlying idea is to modify the standard MCMC approach, in which the Monte Carlo chain is initialized from a random visible configuration. Instead, CD initializes the Monte Carlo chain from a datapoint $\mathbf{v}^{(d)}$ and performs only a fixed number of Gibbs sampling steps k, without necessarily reaching equilibrium. Typical values of the number of steps lie between 1 and 5, so that the obtained samples are typically strongly out of equilibrium. The name contrastive divergence is due to the fact that the algorithm quantifies the divergence between the data statistics and the

probability defined by the RBM. If $P(\mathbf{v})$ approximates well the empirical statistics, the divergence, i.e. the difference in the gradient, should go to zero.

Mathematically, if the maximum likelihood approach coincides with the minimization of the Kullback-Leibler (KL) divergence between the empirical distribution $P_{\mathscr{D}}(\mathbf{v}) = \sum_{d=1}^{D} \delta(\mathbf{v} - \mathbf{v}^{(d)})$ and the one defined by RBM:

$$D_{\mathrm{KL}}(P_{\mathscr{D}} \| P) = \sum_{\mathbf{v}} P_{\mathscr{D}}(\mathbf{v}) \log \frac{P_{\mathscr{D}}(\mathbf{v})}{P(\mathbf{v})}, \tag{5.12}$$

CD amounts to minimizing the KL difference $\mathrm{KL}(P_{\mathscr{D}} \| P)$ - $\mathrm{KL}(P_k \| P)$, in which $P_k(\mathbf{v})$ is the distribution obtained after k steps of Gibbs sampling. To summarize, the CD pipeline can be schematized ad follows:

- For each data point $\mathbf{v}^{(d)}$ $d = 1, \dots, D$, initialize a Markov chain with initial state $\mathbf{v}^0 = \mathbf{v}^{(d)}$.
- Perform k steps of alternate Gibbs sampling starting from $P(\mathbf{h}^0|\mathbf{v}^0)$ and proceeding sampling back and forth until drawing $P(\mathbf{h}^k|\mathbf{v}^k)$.
- Approximate the likelihood gradient as $\nabla_{w_{jv}} \mathscr{L} \simeq \left\langle v_j^0 \langle h_v \rangle_{P(h_v|\mathbf{v}^0)} \right\rangle_{P_0} - \left\langle v_j^k h_v^k \right\rangle_{P_k}$, where $P_0 = P_{\mathscr{D}}$ and the average over P_k is meant to be computed over the D sample obtained after k steps of Gibbs sampling.

The great advantage of CD is that it allows for a much faster computation of the gradient, although the parameter estimate is biased, i.e. the optimal set of weights $\hat{w}_{ML}$ and $\hat{w}_{CD}$ do not generally coincide [25].

An alternative version of the CD algorithm is its persistent version [163]. The difference between the two approaches is that at each parameters update, the Monte Carlo chains are not reinitialized from the data sample $\mathbf{v}^{(d)}$, but rather from the last sample obtained from the previous update. The intuition behind this procedure is that, if the weight parameters are slowly varying, then the sample $\mathbf{v}$ from a previous update is already likely to be drawn from the model distribution, and few Monte Carlo steps are necessary to reach equilibrium. While this is advantageous with respect to standard CD when the MCMC mixing rates are slow, it is also true that the approximation becomes exact only in the limit of vanishing learning rate (see Sec. 2.3.4), and possible divergence issues have been reported for finite learning rate values [52].

Moreover, the persistent MCMC might be problematic in situations in which the model distribution becomes multimodal. Indeed, since the exploration remains local, the algorithm can get stuck on a particular maximum not visiting the others, especially when the probability landscape becomes very sharp.

Tempering strategies for MCMC have been proposed to this aim [35, 27], and will be briefly discussed in the following section.

Parallel tempering MCMC

Parallel tempering is an algorithm for MCMC simulations tailored for sampling efficiently the probability distribution is defined by a rough energy landscape, and was introduced in [70] in the context of biomolecules.

The method simulates in parallel N replicas of the system at different temperatures, alternating standard local moves and global updates, in which two configurations at different temperature are exchanged. This global swap yields faster decorrelations within replicas configurations, i.e. a shorter mixing time, and eventually allows the system to escape energy barriers.

For the specific case of RBM, the inverse temperature goes from $\beta_1 = 0$ to $\beta_N = 1$, i.e. the N-th replica coincides with the original system, and it represents the target distribution. If the temperature parameter multiplies only the weights w, then the statistical weights associated to each replica are proportional to:

$$P_\beta(\mathbf{v}, \mathbf{h}) \propto \exp\left\{\beta \mathbf{v}^T w \mathbf{h} - \sum_{i=1}^{N} g_i(v_i) - \sum_{\mu=1}^{M} U_\mu(h_\mu)\right\}. \tag{5.13}$$

If instead, the inverse temperature multiplies the whole energy, the statistical weight of replica r needs to be modified to:

$$P_{\beta_r}(\mathbf{x}) \propto \exp\{-\beta_r E(\mathbf{x}) - (1 - \beta_r) E_0(\mathbf{x})\}, \tag{5.14}$$

where $\mathbf{x} = (\mathbf{v}, \mathbf{h})$, and E_0 is an energy function not including interacting terms. In both cases, the acceptance probability for swapping two configurations is given by the Metropolis-Hastings rule:

$$A_r = \min\left\{1, \frac{P_{\beta_r}(\mathbf{x}_{r+1}) P_{\beta_{r+1}}(\mathbf{x}_r)}{P_{\beta_r}(\mathbf{x}_r) P_{\beta_{r+1}}(\mathbf{x}_{r+1})}\right\}. \tag{5.15}$$

Global swaps are allowed only between neighboring replicas $r \leftrightarrow r + 1$, as the acceptance probability drops exponentially with the energy and temperature differences. After a full Monte Carlo sweep, i.e. after k-steps of Gibbs sampling and replicas swapping, the configuration at $\beta_N = 1$ is taken as a sample from the model probability. Alternatively, the swap probability can be restricted to visible marginals, yielding an augmented acceptance ratio.

The adoption of tempered techniques partially solves the problem of scarce configurational exploration related to both CD and PCD. However, if entropic barriers are present, even PT is not able to reach good sampling of the whole space. To overcome this further limitation, a new sampling algorithm named *augmented* PT was developed in [165].

Before moving to inference methods that are not based on sampling, it is worth mentioning a systematic study of MCMC based strategies, which has been carried out in [33]. The authors focus on the interplay between the number of steps k used to estimate

the gradient in Eq. (5.11) and the mixing time associated to the model distribution. In particular, when k is small ($k \sim 10$), the learning dynamics is strongly out of equilibrium, and this reflects in the properties of the learned model. Indeed, when sampling from the inferred distribution, it is possible to define an optimal sampling time t_G with respect to statistical estimators such as the log-likelihood, the error on the second moment and so on. For out of equilibrium sampling methods, one has $t_G \sim k$, as strong memory effects emerge. On the other hand, RBM learned with a large number of Monte Carlo steps, e.g. $k \sim 10^{4-5}$, are able to learn an equilibrium distribution that has a high chance to generate samples close to the dataset. To sum up, short k-steps MCMC schemes can be used if one is to merely generate good quality sample in a short time, whereas k should surpass the mixing time of the model probability at each learning epoch if the aim is to reproduce the equilibrium distribution of the dataset.

Non-sampling methods

It is also possible to train RBM via methods not based on sampling, as for instance the MF method, which we already treated when discussing the GPM in Sec. 2.3.1, or higher order terms of the Plefka's expansion as in the TAP (Thouless-Anderson-Palmer) approximation. In particular, a TAP approximation inference scheme has been proposed in [59] in the context of binary RBM.

The backbone of the method is the aforementioned Plefka's expansion of the Gibbs free energy, similarly to what has been described in Sec. 2.3.1. The starting point is to rewrite the log-likelihood as a sum of two free energy contributions:

$$\mathscr{L}[w] = \overline{\log\left(\sum_{\mathbf{h}} \mathrm{e}^{-E(\mathbf{v}^{(d)},\mathbf{h})}\right)} - \log Z = -\overline{F^c(\mathbf{v}^{(d)})} + F, \tag{5.16}$$

with $F^c(\mathbf{v}^{(d)})$ the *clamped* free energy over configuration d and $F = -\log Z = -\log \sum_{\mathbf{x}} \mathrm{e}^{-E(\mathbf{x})}$ the actual free energy of the model. The Plefka's expansion provides a systematic method to estimate the free energy in a weak coupling regime. Since the considered variables are binary (or more precisely boolean), the Legendre transform is performed with respect to *magnetizations* $\langle \mathbf{x} \rangle = \mathbf{m}$ and an auxiliary field $\mathbf{q}$:

$$G(\mathbf{m}) = \sup_{\mathbf{q}} \left[F(\mathbf{q}) + \sum_{i=1}^{L} q_i m_i \right] = \sup_{\mathbf{q}} \left[-\log\left(\sum_{\mathbf{x}} \mathrm{e}^{-E(\mathbf{x}) + \sum_{i=1}^{L} q_i x_i} \right) + \sum_{i=1}^{L} q_i m_i \right]. \tag{5.17}$$

Conversely, the Helmholtz free energy is obtained by anti-transforming $F(\mathbf{q}) = \inf_{\mathbf{m}} \left[G(\mathbf{m}) - \sum_{i=1}^{L} q_i m_i \right]$. The original free energy of the model is recovered for $\mathbf{q} = 0$, i.e. $F \equiv F(\mathbf{q} = 0) = \min_{\mathbf{m}} [G(\mathbf{m})]$. Consequently, the model free energy can be computed by expanding $G(\mathbf{m})$ in powers of a perturbative parameter (possibly the temperature), and then extremizing it with respect to magnetizations.

The stationarity condition for the Gibbs free energy $\nabla_{\mathbf{m}} G(\mathbf{m}) = 0$ provides a set of auto-consistent equations for the magnetizations. If the expansion is performed with respect to the coupling parameters, one obtains at second order:

$$m_i^{\mathrm{v}} = \sigma\left[a_i + \sum_{\mu=1}^{M} w_{i\mu} m_\mu^{\mathrm{h}} - w_{i\mu}^2 m_\mu^{\mathrm{h}} \left(m_i^{\mathrm{v}} - \frac{1}{2}\right)\left(1 - m_\mu^{\mathrm{h}}\right)\right], \tag{5.18}$$

where $\sigma(x) = (1 + \mathrm{e}^{-x})^{-1}$ is the sigmoid function, and $m_i^{\mathrm{v}} = \langle v_i \rangle$, $m_\mu^{\mathrm{h}} = \langle h_\mu \rangle$.

Eq. (5.18) belongs to a set of coupled non linear equations, which can be solved iteratively until convergence. When writing Eq. (5.18), we assumed that the model energy coincides with $E(\mathbf{x}) = -\sum_{i\mu} v_i w_{i\mu} h_\mu - \sum_{i=1}^{N} a_i v_i - \sum_{\mu=1}^{M} b_\mu h_\mu$, so that both visible and hidden units are boolean variables.

Once the auto-consistency equations converged, the fixed point magnetizations $\mathbf{m}^*$ can be plugged into the Gibbs free energy expansion, providing an approximation of $F = -\log Z$. Consequently, the approximate gradient can be computed as:

$$\frac{\partial \mathscr{L}[w]}{\partial w_{j\nu}} \simeq -\overline{\frac{F^c(\mathbf{v}^{(d)})}{\partial w_{j\nu}}} + \frac{\partial G(\mathbf{m}^*)}{\partial w_{j\nu}}, \tag{5.19}$$

in which $G(\mathbf{m})$ can be expanded up to arbitrarily high order in powers of w. In [59], it has been shown that the second order expansion was able to provide results of comparable quality with respect to both CD and PCD.

In the rest of the chapter, we aim to develop another inference method for RBM which is not based on sampling, but rather, on an iterative algorithm for approximating probability distributions known as *Expectation Propagation* (EP).

5.1.2 Applications of RBM

The introduction of the CD [72] algorithm sparked interest in the possible usage of RBM. Among the first applications, RBM were stacked one onto the other in order to learn deep-belief networks, which were subsequently fine-tuned by means of back-propagation [73]. Later on, the advances in supervised deep learning strategies made pre-training unnecessary, and RBM were replaced by other architectures as generative models.

Recently, a novel interest in RBM has aroused both from a theoretical perspective and for their successful application to biological sequence data. Indeed, RBM represent a minimal model of neural network architecture with the capability to introduce arbitrarily high-order interactions among data, and can be thought as a generalization of the Hopfield model [77] for pattern storage and recognition. In this perspective, Tubiana et al. [168] performed a replica-symmetric computation of a random ensemble of RBM's, in order to study the possible different operational regimes, correspondent to different phases.

The modeling considers the weights $w_{i\mu}$ defining the RBM's as quenched random variables representing the source of disorder. The visible units are considered as binary, i.e. $g_i(v_i) = -gv_i$, whereas hidden units are distributed according to a ReLU potential. Both priors are considered as uniform, that is, possessing the same parameters for each unit. The replica trick allows to compute an approximation of the free energy of the model, from which the phase diagram of the system can be studied.

In particular, the authors underline the importance of the so called *compositional* regime, in which each data input activates a significant, but not too large, number L of hidden units ($1 \ll L \ll M$), as opposed to the ferromagnetic and spin glass phases, characterized respectively by only one, and a large amount of incoherently activated hidden units. The fundamental features allowing to reach the compositional phase are: weights sparsity, quantified by the probability p that a weight is non-zero, and the specific value of the threshold parameter of the ReLU potential.

Stepping away from a purely theoretical viewpoint, in the recent years RBM have been successfully applied to a plethora of biological contexts, among which we can mention MSA of homologous protein sequences [167, 166, 147], immune system features such as TCR specificity [20, 19] and prediction of antigen presentation by the HLA-I MHC [21], screening experiments of aptamers binding capabilities [36].

In the context of protein families of homologous sequences, the appeal of RBM is represented by their capabilities to learn sequence motifs, leading to the possibility to cluster protein sequences according to different criteria, e.g. structural or functional properties and phylogenetic relations. Moreover, RBM are generative, so that they could in principle allow to generate sequences with specific properties, by selection of the proper hidden units.

5.2 Expectation Propagation

In this section we describe the iterative algorithm *Expectation Propagation* (EP), originally introduced by T.P. Minka in [104], even if it was already introduced in the specialized Gaussian case in the context of statistical physics of disordered systems under the name *adaTAP*, by M. Opper and O. Winther [118, 117]. Throughout this thesis we will mainly rely on the Gaussian version of EP, but a more general formulation based on exponential families is also possible.

5.2.1 Gaussian EP

Let's analyze the Gaussian formulation of the EP algorithm. Our goal is to approximate an intractable probability distribution of the form:

$$P(\mathbf{x}) = \frac{1}{Z} G(\mathbf{x}) \prod_{i=1}^{N} \psi_i(x_i), \tag{5.20}$$

where $G(\mathbf{x})$ is a multivariate Gaussian distribution:

$$G(\mathbf{x}) = \frac{\det(A)^{1/2}}{(2\pi)^{N/2}} \exp\left[-\frac{1}{2}\mathbf{x}^T A\mathbf{x} + \mathbf{x}^T\mathbf{m}\right], \tag{5.21}$$

defined by a precision matrix A and a mean vector $(A)^{-1}\mathbf{m}$, whereas the univariate factors $\psi_i(x_i)$ represent the set of intractable priors. The idea of the EP method is to approximate Eq. 5.20 with a full multivariate Gaussian distribution $Q(\mathbf{x})$:

$$Q(\mathbf{x}) = \frac{1}{Z_Q}G(\mathbf{x})\prod_{i=1}^{N}\phi_i(x_i) = \frac{1}{(2\pi)^{N/2}\det(\Sigma)^{1/2}}\exp\left[-\frac{1}{2}\left(\mathbf{x}-\langle\mathbf{x}\rangle\right)^T(\Sigma)^{-1}\left(\mathbf{x}-\langle\mathbf{x}\rangle\right)\right]. \tag{5.22}$$

In Eq. (5.22) the intractable priors have been substituted by a set of univariate Gaussian factors $\phi_i(x_i) = \mathcal{N}(x_i; a_i, d_i) = \frac{1}{\sqrt{2\pi d_i}}\exp\left[-\frac{(x_i-a_i)^2}{2d_i}\right]$, with a_i and d_i coinciding respectively with the mean and the variance of the Gaussian. $Q(\mathbf{x})$ can also be rewritten as a global multivariate Gaussian defined by a covariance matrix $\Sigma = \langle\mathbf{x}\mathbf{x}^T\rangle - \langle\mathbf{x}\rangle\langle\mathbf{x}^T\rangle = (A+D)^{-1}$ and a mean vector $\langle\mathbf{x}\rangle = \Sigma(\mathbf{m}+D\mathbf{a})$, where we introduced the matrix D, which is defined as:

$$\begin{pmatrix} \frac{1}{d_1} & 0 & \dots & 0 \\ 0 & \frac{1}{d_2} & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \frac{1}{d_N} \end{pmatrix}, \tag{5.23}$$

which is a $N \times N$ diagonal matrix having the inverse variances of the univariate factors as its elements. Within this formulation, it is straightforward to derive the normalization factor $Z_Q = (2\pi)^{N/2}\det(\Sigma)^{1/2}$.

The free parameters of the EP algorithm are the univariate Gaussians means and variances $\{\mathbf{a}, \mathbf{d}\}$, which must be optimally determined so that $Q(\mathbf{x})$ is an approximation of the original distribution $P(\mathbf{x})$. From a theoretical perspective, the parameters are fixed by minimizing the *local* inclusive Kullback-Leibler (KL) divergence between the two distributions. The global inclusive KL-divergence has expression $D_{\mathrm{KL}}(P\|Q) = \int \mathrm{d}^N x\, P(\mathbf{x})\log\frac{P(\mathbf{x})}{Q(\mathbf{x})}$ and it is generally difficult to directly minimize it. Minka [104] showed that minimizing a local version of the inclusive KL-divergence is equivalent to using message passing algorithms, such as *belief propagation* (BP) and EP. For our case of interest, the KL-divergence we aim at minimizing is the one between the full Gaussian approximation Q and the so called *tilted* distribution $Q^{(k)}$, which is defined as:

$$Q^{(k)}(\mathbf{x}) = \frac{1}{Z_{Q^{(k)}}}G(\mathbf{x})\left(\prod_{i\neq k}\phi_i(x_i)\right)\psi_k(x_k) = \frac{Z_Q}{Z_{Q^{(k)}}}Q(\mathbf{x})\frac{\psi_k(x_k)}{\phi_k(x_k)}. \tag{5.24}$$

Thus, the local KL-divergence becomes:

$$D_{\mathrm{KL}}(Q^{(k)}\|Q) = \int \mathrm{d}^N x\, Q^{(k)}(\mathbf{x}) \log \frac{Q^{(k)}(\mathbf{x})}{Q(\mathbf{x})}. \tag{5.25}$$

It is possible to show that minimizing Eq. (5.25) with respect to a_k and d_k amounts to impose the moment matching conditions:

$$\begin{cases} \langle x_k \rangle_{Q^{(k)}} = \langle x_k \rangle_Q \,, \\ \langle x_k^2 \rangle_{Q^{(k)}} = \langle x_k^2 \rangle_Q \,. \end{cases} \tag{5.26}$$

The strategy of the EP algorithm is to update iteratively the univariate Gaussian parameters $\{\mathbf{a}, \mathbf{d}\}$ according to Eq. (5.26) until they reach stable values. To better understand the EP update, we rewrite the tilted distribution as $Q^{(k)}(\mathbf{x}) \propto Q^{\setminus k}(\mathbf{x})\psi_k(x_k)$, where we introduced the cavity distribution:

$$Q^{\setminus k}(\mathbf{x}) \propto \exp\left[-\frac{1}{2}\left(\mathbf{x} - \langle \mathbf{x} \rangle^{(k)}\right)^T (\Sigma^{(k)})^{-1} \left(\mathbf{x} - \langle \mathbf{x} \rangle^{(k)}\right)\right], \tag{5.27}$$

where the cavity covariance matrix is defined as $\Sigma^{(k)} = \left(A + D^{(k)}\right)^{-1}$, with $D^{(k)}$ the same matrix as D but with the k-th diagonal entry set to zero. Correspondingly, the cavity average is given by $\langle \mathbf{x} \rangle^{(k)} = \Sigma^{(k)}\left(\mathbf{m} + D^{(k)}\mathbf{a}\right)$. Similarly, we can also express Q as $Q(\mathbf{x}) \propto Q^{\setminus k}(\mathbf{x})\phi_k(x_k)$, so that the average values become:

$$\begin{aligned} \langle x_k^\alpha \rangle_Q &\propto \int \mathrm{d}^N x\, x_k^\alpha Q^{\setminus k}(\mathbf{x})\phi_k(x_k) \\ &\propto \int \mathrm{d}x_k\, x_k^\alpha Q^{\setminus k}(x_k)\phi_k(x_k), \end{aligned} \tag{5.28}$$

where $Q^{\setminus k}(x_k)$ is the marginal of the cavity distribution and $\alpha = 1,2$. Thus, the average values are computed with respect to the product of two univariate Gaussians $Q^{\setminus k}(x_k)\phi_k(x_k)$, for which we can use the known formula:

$$\begin{cases} m = s\left[\frac{m_1}{s_1} + \frac{m_2}{s_2}\right], \\ s = \left[\frac{1}{s_1} + \frac{1}{s_2}\right]^{-1}, \end{cases} \tag{5.29}$$

where $m_{1,2}$ are the means and $s_{1,2}$ the variances of the two Gaussians, and m and s are those corresponding to the product between the two. Employing Eq. (5.29) to compute the integrals in Eq. (5.28) one gets:

$$\begin{cases} \langle x_k \rangle_Q = \left[\frac{1}{d_k} + \frac{1}{\Sigma_k} \right]^{-1} \left(\frac{a_k}{d_k} + \frac{\mu_k}{\Sigma_k} \right), \\ \langle x_k^2 \rangle_Q = \left[\frac{1}{d_k} + \frac{1}{\Sigma_k} \right]^{-1} + \langle x_k \rangle_Q^2, \end{cases} \tag{5.30}$$

where we labeled $\langle \mathbf{x} \rangle_k^{(k)} = \mu_k$ and $\Sigma_{kk}^{(k)} = \Sigma_k$. The actual EP update equations are obtained by expressing Eq. (5.26) as a function of a_k and d_k:

$$\begin{cases} a_k = \langle x_k \rangle_{Q^{(k)}} + \frac{b_k}{\Sigma_k} \left(\langle x_k \rangle_{Q^{(k)}} - \mu_k \right), \\ d_k = \left[\frac{1}{\langle x_k^2 \rangle_{Q^{(k)}} - \langle x_k \rangle_{Q^{(k)}}^2} - \frac{1}{\Sigma_k} \right]^{-1}. \end{cases} \tag{5.31}$$

Following Eq. (5.31), two possible update strategies can be pursued. Either EP parameters are updated sequentially, i.e. after each moments computation the cavity matrix is also updated before moving to the next, or the same $\Sigma^{(k)}$ and $\langle \mathbf{x} \rangle^{(k)}$ are used to perform every update. The second strategy is known as parallel update, since the single operations can be performed independently.

In order to simplify some computations, it is possible to rely on the so called Sherman-Morrison formula [146], which allows to relate two matrices differing for just a single entry, as it is the case for Σ and $\Sigma^{(k)}$:

$$\Sigma^{(k)} = \Sigma + \frac{\Sigma \mathbf{e}_k \mathbf{e}_k^T \Sigma}{d_k - \Sigma_{kk}}, \tag{5.32}$$

where $\mathbf{e}_k$ is the vector of the canonical basis with all components zero but the k-th. From Eq. (5.32) we obtain the marginal cavity mean and variance as:

$$\begin{cases} \Sigma_k = \frac{d_k \Sigma_{kk}}{d_k - \Sigma_{kk}} \\ \mu_k = \frac{d_k \langle x_k \rangle - a_k \Sigma_{kk}}{d_k - \Sigma_{kk}} \end{cases} \tag{5.33}$$

It is worth stressing that the tilted moments need to be computed explicitly for the given priors $\boldsymbol{\psi}$:

$$\langle x_k^\alpha \rangle_{Q^{(k)}} \propto \int \mathrm{d}^N x \, x_k^\alpha Q^{\setminus k}(\mathbf{x}) \psi_k(x_k) = \int \mathrm{d}x_k \, x_k^\alpha \frac{\mathrm{e}^{-\frac{(x_k - \mu_k)^2}{2\Sigma_k}}}{\sqrt{2\pi\Sigma_k}} \psi_k(x_k), \tag{5.34}$$

The EP parameters are updated until convergence is reached, according to the tolerance parameter ϵ:

$$\epsilon_t = \max_{k=1,\dots,N} \left\{ \left| \langle x_k \rangle_{Q_t^{(k)}} - \langle x_k \rangle_{Q_{t-1}^{(k)}} \right| + \left| \langle x_k^2 \rangle_{Q_t^{(k)}} - \langle x_k^2 \rangle_{Q_{t-1}^{(k)}} \right| \right\}. \tag{5.35}$$

In particular, iterations stop when $\epsilon_t < \epsilon_{\text{stop}}$.

In Alg. 1 we summarize the main points of the parallel implementation of the EP algorithm, whose key element is the **moment** function, corresponding to Eq. (5.34). Therein, we also introduced the damping parameter γ, which is necessary to stabilize EP dynamics.

Algorithm 1 Parallel EP update

procedure EP($A, \mathbf{m}, \{\psi_1, \dots, \psi_N\}$)
  Initialize $\mathbf{a}^{\text{old}}$, $\mathbf{d}^{\text{old}}$ and $\Delta av = 1$
  while iter < maxiter && $\Delta av > \epsilon$ **do**
    $\mathbf{av} = 0$
    $\mathbf{var} = 0$
    $\Sigma = (A + D)^{-1}$
    $\overline{\mathbf{x}} = \Sigma\,(\mathbf{m} + D\mathbf{a})$
    for $k = 1, \dots, N$ **do**
      $\mu_k = \frac{d_k \langle x_k \rangle - a_k \Sigma_{kk}}{d_k - \Sigma_{kk}}$
      $\Sigma_k = \frac{d_k \Sigma_{kk}}{d_k - \Sigma_{kk}}$
      $\langle x_k \rangle_{Q^{(k)}}, \langle x_k^2 \rangle_{Q^{(k)}} = \mathbf{moment}(\mu_k, \Sigma_k, \psi_k)$
      $\Delta av \leftarrow \max\left(\Delta av, \left|\langle x_k \rangle^{(k)} - av_k\right|\right)$
      $av_k \leftarrow \langle x_k \rangle_{Q^{(k)}}$
      $var_k \leftarrow \langle x_k^2 \rangle_{Q^{(k)}} - \langle x_k \rangle^2_{Q^{(k)}}$
      $d_k^{\text{new}} \leftarrow \left[\frac{1}{var_k} - \frac{1}{\Sigma_k}\right]^{-1}$
      $a_k^{\text{new}} \leftarrow av_k + \frac{d_k}{\Sigma_k}\left(av_k - \langle x_k \rangle^{(k)}\right)$
      $a_k^{\text{old}} \leftarrow \gamma a_k^{\text{old}} + (1-\gamma) a_k^{\text{new}}$
      $d_k^{\text{old}} \leftarrow \gamma d_k^{\text{old}} + (1-\gamma) d_k^{\text{new}}$
  return **av**, **var**

5.2.2 EP for RBM inference

In this section we specialize EP to the problem of inferring the constituent parameters of an RBM. In order to set a specific stage, we consider the case of a binary network, in which both the visible and the hidden units are boolean variables, i.e. $v_i, h_i \in \{0,1\}$. In this scenario, the energy function associated to the RBM can be written as:

$$E(\mathbf{v}, \mathbf{h}) = \mathbf{v}^T w \mathbf{h} + \mathbf{v}^T \theta^{\text{v}} + \mathbf{h}^T \theta^{\text{h}}, \tag{5.36}$$

in which we introduce the set of parameters defining the RBM: the $N \times M$ weight matrix w and the respectively N and M dimensional binary fields θ^{v} and θ^{h}. Within the

EP framework, we assume that the model probability $P(\mathbf{v}, \mathbf{h}) \propto \exp\left[-E(\mathbf{v}, \mathbf{h})\right]$ can be expressed as:

$$P(\mathbf{x}) = \frac{1}{Z} \mathrm{e}^{-\frac{1}{2}\mathbf{x}^T W \mathbf{x}} \prod_{p=1}^{N+M} \psi_p(x_p), \tag{5.37}$$

where $\mathbf{x} = (\mathbf{v}, \mathbf{h})$ is the concatenation of visible and hidden units vectors, and we introduced the $(N+M)\times(N+M)$ matrix W, which plays the role of the precision matrix A of the multivariate Gaussian term $G(\mathbf{x})$, and is defined as:

$$W = \begin{pmatrix} 0 & -w \\ -w & 0 \end{pmatrix}. \tag{5.38}$$

The functions ψ are the set of priors imposing the binary constraints. Indeed, according to the multivariate Gaussian part, $\mathbf{x}$ is in principle a real valued vector of zero mean $\mathbf{m} = 0$. EP aims at approximating Eq. (5.37) with the multivariate Gaussian ansatz $Q(\mathbf{x})$:

$$Q(\mathbf{x}) \propto \mathrm{e}^{-\frac{1}{2}\mathbf{x}^T W \mathbf{x}} \prod_{p=1}^{N+M} \phi_p(x_p; a_p, d_p) \tag{5.39}$$

$$= \mathrm{e}^{-\frac{1}{2}\left[\mathbf{x}^T W \mathbf{x} + (\mathbf{x}-\mathbf{a})^T D (\mathbf{x}-\mathbf{a})\right]} \tag{5.40}$$

so that $\Sigma = (W + D)^{-1}$ and $\langle \mathbf{x} \rangle_Q = \Sigma D \mathbf{a}$. The matrix D contains the inverse variances of the EP univariate factors:

$$D = \begin{pmatrix} D^{\mathrm{v}} & 0 \\ 0 & D^{\mathrm{h}} \end{pmatrix}, \tag{5.41}$$

where $D^{\mathrm{v}} = \mathrm{diag}_N\left(\frac{1}{d_1}, \dots, \frac{1}{d_N}\right)$ and $D^{\mathrm{v}} = \mathrm{diag}_M\left(\frac{1}{d_{N+1}}, \dots, \frac{1}{d_{N+M}}\right)$. Alternatively, the variances vectors can be seen as a concatenation of a visible and a hidden part $\mathbf{d} = (\mathbf{d}^{\mathrm{v}}, \mathbf{d}^{\mathrm{h}})$, akin to the mean vector $\mathbf{a} = (a_1, \dots, a_N, a_{N+1}, \dots, a_{N+M}) = (\mathbf{a}^{\mathrm{v}}, \mathbf{a}^{\mathrm{h}})$.

The EP algorithm requires to invert the precision matrix A to obtain the covariance matrix Σ, since the cavity, and consequently the tilted moments, are computed from it. For the specific case of RBM, the precision matrix has a peculiar block structure, that can leveraged so to optimize the inversion process:

$$
\begin{aligned}
A^{-1} = \Sigma &= \begin{pmatrix} D^{\mathrm{v}} & -w \\ -w^T & D^{\mathrm{h}} \end{pmatrix}^{-1} \\
&= \begin{pmatrix} (D^{\mathrm{v}})^{-1} + (D^{\mathrm{v}})^{-1} w \left(A^{\mathrm{h}}\right)^{-1} w^T (D^{\mathrm{v}})^{-1} & (D^{\mathrm{v}})^{-1} w \left(A^{\mathrm{h}}\right)^{-1} \\ \left(A^{\mathrm{h}}\right)^{-1} w^T (D^{\mathrm{v}})^{-1} & \left(A^{\mathrm{h}}\right)^{-1} \end{pmatrix} \\
&= \begin{pmatrix} (A^{\mathrm{v}})^{-1} & \left(D^{\mathrm{h}}\right)^{-1} w^T (A^{\mathrm{v}})^{-1} \\ (A^{\mathrm{v}})^{-1} w \left(D^{\mathrm{h}}\right)^{-1} & \left(D^{\mathrm{h}}\right)^{-1} + \left(D^{\mathrm{h}}\right)^{-1} w^T (A^{\mathrm{v}})^{-1} w \left(D^{\mathrm{h}}\right)^{-1} \end{pmatrix}
\end{aligned} \tag{5.42}
$$

where we introduced the precision matrices of the visible and hidden units $A^{\mathrm{v}} = D^{\mathrm{v}} - w\left(D^{\mathrm{h}}\right)^{-1} w^T$, $A^{\mathrm{h}} = D^{\mathrm{h}} - w^T (D^{\mathrm{v}})^{-1} w$. Each of the two expressions of the covariance matrix in Eq. (5.42) can be used. Specifically, since generally for the RBM $N > M$, it is convenient to employ the one that only requires to invert A^{h} (inversion of the D's matrices are trivial because they are diagonal). Following this strategy, we passed from a computational time of $O\left((N+M)^3\right)$, related to the inversion of the full precision matrix A, to a computational complexity of order $O(M^3)$ or $O(N^3)$, depending on the specific choice made.

Even if the previously presented trick allows for a speed up of the EP algorithm, a sequential update would still scale at most as $O(M^3(N+M))$. In the context of RBM, we devised an update strategy that we name block-EP, which is inspired from the alternate Gibbs sampling between visible and hidden units used in Monte Carlo learning strategies.

The idea of the block update scheme is to alternatively perform a parallel EP update (Alg. 1) within the visible and hidden units, and a sequential update between the two blocks. Thus, given the joint probability $Q(\mathbf{x})$, we first need to compute the marginals associated to the visible and hidden blocks. To do so, it is convenient to first rewrite the multivariate Gaussian in Eq. (5.40) as:

$$
Q(\mathbf{v}, \mathbf{h}) = \frac{1}{(2\pi)^{\frac{N+M}{2}} \det(\Sigma)^{\frac{1}{2}}} \mathrm{e}^{-\mathbf{v}^T w \mathbf{h} - \frac{1}{2}(\mathbf{v}-\mathbf{a}^{\mathrm{v}})^T D^{\mathrm{v}} (\mathbf{v}-\mathbf{a}^{\mathrm{v}}) - \frac{1}{2}(\mathbf{h}-\mathbf{a}^{\mathrm{h}})^T D^{\mathrm{h}} (\mathbf{h}-\mathbf{a}^{\mathrm{h}})}, \tag{5.43}
$$

where we highlighted the decomposition in visible and hidden block. Then, the visible units marginal is obtained by integration over the hidden variables:

$$
\begin{aligned}
Q(\mathbf{v}) &\propto \mathrm{e}^{-\frac{1}{2}(\mathbf{v}-\mathbf{a}^{\mathrm{v}})^T D^{\mathrm{v}}(\mathbf{v}-\mathbf{a}^{\mathrm{v}})} \int \mathrm{d}^M h \mathrm{e}^{-(\mathbf{v}^T w)\mathbf{h} - \frac{1}{2}(\mathbf{h}-\mathbf{a}^{\mathrm{h}})^T D^{\mathrm{h}}(\mathbf{h}-\mathbf{a}^{\mathrm{h}})} \\
&\propto \mathrm{e}^{-\frac{1}{2}(\mathbf{v}-\mathbf{a}^{\mathrm{v}})^T D^{\mathrm{v}}(\mathbf{v}-\mathbf{a}^{\mathrm{v}})} \int \mathrm{d}^M h \mathrm{e}^{-\frac{1}{2}\mathbf{h}^T D^{\mathrm{h}}\mathbf{h} + \left[D^{\mathrm{h}}(\mathbf{a}^{\mathrm{h}})^T - w^T\mathbf{v}\right]^T \mathbf{h}} \\
&\propto \mathrm{e}^{-\frac{1}{2}(\mathbf{v}-\mathbf{a}^{\mathrm{v}})^T D^{\mathrm{v}}(\mathbf{v}-\mathbf{a}^{\mathrm{v}})} \mathrm{e}^{\frac{1}{2}\overline{\mathbf{h}}^T (D^{\mathrm{h}})^{-1}\overline{\mathbf{h}}} \\
&\propto \mathrm{e}^{-\frac{1}{2}(\mathbf{v}-\langle\mathbf{v}\rangle)^T A^{\mathrm{v}}(\mathbf{v}-\langle\mathbf{v}\rangle)}
\end{aligned}
\tag{5.44}
$$

where we defined $\overline{\mathbf{h}} = D^{\mathrm{h}}\left(\mathbf{a}^{\mathrm{h}}\right)^T - w^T\mathbf{v}$. The marginal precision matrix and mean vector have explicit expression:

$$
\begin{cases}
A^{\mathrm{v}} = D^{\mathrm{v}} - w\left(D^{\mathrm{h}}\right)^{-1} w^T, \\
\langle\mathbf{v}\rangle = (A^{\mathrm{v}})^{-1}\left[D^{\mathrm{v}}\mathbf{a}^{\mathrm{v}} - w\mathbf{a}^{\mathrm{h}}\right].
\end{cases}
\tag{5.45}
$$

It is worth pointing out that the expression of the marginal precision matrix could also be derived by Eq. (5.42). Indeed, the marginal of a multivariate Gaussian is still a multivariate Gaussian, whose covariance matrix and mean vector are the subparts related to the considered block. If we repeat the same procedure for the hidden marginal we obtain:

$$
\begin{cases}
A^{\mathrm{h}} = D^{\mathrm{h}} - w^T (D^{\mathrm{v}})^{-1} w, \\
\langle\mathbf{h}\rangle = \left(A^{\mathrm{h}}\right)^{-1}\left[D^{\mathrm{h}}\mathbf{a}^{\mathrm{h}} - w^T\mathbf{a}^{\mathrm{v}}\right].
\end{cases}
\tag{5.46}
$$

We now have all the necessary ingredients to write down the block-EP algorithm, which is reported in Alg. 2. We specialize the pseudocode to the case in which the intractable probability to be approximated coincides with Eq. (5.37). We notice how the block algorithm has a computational complexity of order $O(N^3 + M^3)$, as it requires the inversion of the two precision matrices A^{v} and A^{h}. It is thus suboptimal with respect to the parallel update, even if it provides improved accuracy, for at the second inversion EP parameters are partially updated.

Gradient computation for binary RBM

Once the EP algorithm has converged to a fixed point, the multivariate probability function $Q(\mathbf{x})$ can be used to approximate the likelihood gradients. For the case of binary RBM, on top of Eq. (5.11), we also need to compute the derivatives of the likelihood with respect to the binary fields:

Algorithm 2 Block EP update for RBM

procedure BLOCKEP($w, \{\psi_1, \dots, \psi_N\}$)
  Initialize $\mathbf{a}^{\mathrm{v}}$, $\mathbf{a}^{\mathrm{h}}$, $\mathbf{d}^{\mathrm{v}}$, $\mathbf{d}^{\mathrm{h}}$ and $\Delta av = 1$
  while iter < maxiter && $\Delta av > \epsilon$ **do**
    $\mathbf{av}^{\mathrm{v}}, \mathbf{av}^{\mathrm{h}} = 0$
    $\mathbf{var}^{\mathrm{v}}, \mathbf{var}^{\mathrm{h}} = 0$
    $D^{\mathrm{v}} = \mathrm{diag}_N\left(\frac{1}{d_1^{\mathrm{v}}}, \dots, \frac{1}{d_N^{\mathrm{v}}}\right)$
    $D^{\mathrm{h}} = \mathrm{diag}_M\left(\frac{1}{d_1^{\mathrm{h}}}, \dots, \frac{1}{d_M^{\mathrm{h}}}\right)$
    $\Sigma^{\mathrm{v}} = \left(D^{\mathrm{v}} - w\left(D^{\mathrm{h}}\right)^{-1} w^T\right)^{-1}$
    $\langle \mathbf{v} \rangle_Q = \Sigma^{\mathrm{v}} \left(D^{\mathrm{v}} \mathbf{a}^{\mathrm{v}} - w \mathbf{a}^{\mathrm{h}}\right)$
    for k=1,...,N **do**
      $\mu_k = \frac{d_k^{\mathrm{v}} \langle v_k \rangle_Q - a_k^{\mathrm{v}} \Sigma_{kk}^{\mathrm{v}}}{d_k^{\mathrm{v}} - \Sigma_{kk}^{\mathrm{v}}}$
      $\Sigma_k = \frac{d_k^{\mathrm{v}} \Sigma_{kk}^{\mathrm{v}}}{d_k^{\mathrm{v}} - \Sigma_{kk}^{\mathrm{v}}}$
      $\langle v_k \rangle_{Q^{(k)}}, \left\langle v_k^2 \right\rangle_{Q^{(k)}} = \mathbf{moment}(\mu_k, \Sigma_k, \psi_k)$
      $\Delta av \leftarrow \max\left(\Delta av, \left|\langle v_k \rangle_{Q^{(k)}} - av_k^{\mathrm{v}}\right|\right)$
      $a^{\mathrm{v}}, d^{\mathrm{v}} = \mathbf{update}\left(\langle v_k \rangle_{Q^{(k)}}, \left\langle v_k^2 \right\rangle_{Q^{(k)}}, \mu_k, \Sigma_k\right)$
      $D_{kk}^{\mathrm{v}} = \frac{1}{d_k^{\mathrm{v}}}$
      $av_k^{\mathrm{v}}, var_k^{\mathrm{v}} \leftarrow \langle v_k \rangle_{Q^{(k)}}, \left\langle v_k^2 \right\rangle_{Q^{(k)}} - \langle v_k \rangle_{Q^{(k)}}^2$
    $\Sigma^{\mathrm{h}} = \left(D^{\mathrm{h}} - w^T \left(D^{\mathrm{v}}\right)^{-1} w\right)^{-1}$
    $\langle \mathbf{h} \rangle_Q = \Sigma^{\mathrm{h}} \left(D^{\mathrm{h}} \mathbf{a}^{\mathrm{h}} - w^T \mathbf{a}^{\mathrm{v}}\right)$
    for k=1,...,M **do**
      $\mu_{N+k} = \frac{d_k^{\mathrm{h}} \langle h_k \rangle_Q - a_k^{\mathrm{h}} \Sigma_{kk}^{\mathrm{h}}}{d_k^{\mathrm{h}} - \Sigma_{kk}^{\mathrm{h}}}$
      $\Sigma_{N+k} = \frac{d_k^{\mathrm{h}} \Sigma_{kk}^{\mathrm{h}}}{d_k^{\mathrm{h}} - \Sigma_{kk}^{\mathrm{h}}}$
      $\langle h_k \rangle_{Q^{(k)}}, \left\langle h_k^2 \right\rangle_{Q^{(k)}} = \mathbf{moment}(\mu_{k+N}, \Sigma_{N+k}, \psi_k)$
      $\Delta av \leftarrow \max\left(\Delta av, \left|\langle h_k \rangle_{Q^{(k)}} - av_k^{\mathrm{h}}\right|\right)$
      $a^{\mathrm{h}}, d^{\mathrm{h}} = \mathbf{update}\left(\langle h_k \rangle_{Q^{(k)}}, \left\langle h_k^2 \right\rangle_{Q^{(k)}}, \mu_{N+k}, \Sigma_{N+k}\right)$
      $D_{kk}^{\mathrm{h}} = \frac{1}{d_k^{\mathrm{h}}}$
      $av_k^{\mathrm{h}}, var_k^{\mathrm{h}} \leftarrow \langle h_k \rangle_{Q^{(k)}}, \left\langle h_k^2 \right\rangle_{Q^{(k)}} - \langle h_k \rangle_{Q^{(k)}}^2$
  return $\mathbf{av}^{\mathrm{v}}, \mathbf{av}^{\mathrm{h}}, \mathbf{var}^{\mathrm{v}}, \mathbf{var}^{\mathrm{h}}$

$$\begin{cases} \frac{\partial \mathscr{L}}{\partial \theta_j^{\mathrm{v}}} = \overline{v_j} - \langle v_j \rangle_{P(\mathbf{x})} \\ \frac{\partial \mathscr{L}}{\partial \theta_\nu^{\mathrm{h}}} = \overline{\langle h_\nu \rangle_{P(h_\nu|\mathbf{v})}} - \langle h_\nu \rangle_{P(\mathbf{x})} \,. \end{cases} \tag{5.47}$$

As it could be expected, the derivatives with respect to the binary fields $\boldsymbol{\theta}^{\mathrm{v}}$ and $\boldsymbol{\theta}^{\mathrm{h}}$ are given by the difference between empirical and ensemble averages of the visible and hidden units respectively. In the previous equation, both the visible and hidden averages are approximated as $\langle v_j \rangle_P \simeq \langle v_j \rangle_Q$, $\langle h_\nu \rangle_P \simeq \langle h_\nu \rangle_Q$, i.e. with the averages of the multivariate Gaussian. The conditional average $\langle h_\nu \rangle_{P(h_\nu|\mathbf{v})}$ appears both in Eqs. (5.11) and (5.47). Thanks to the peculiar RBM structure, such average can be computed analytically, and for the specific case of a binary prior one gets:

$$\langle h_\nu \rangle_{P(h_\nu|\mathbf{v})} = \frac{\sum_{h_\nu=\{0,1\}} h_\nu \mathrm{e}^{h_\nu \theta_\nu^{\mathrm{h}} + I_\nu(\mathbf{v})}}{\sum_{h_\nu=\{0,1\}} \mathrm{e}^{h_\nu \theta_\nu^{\mathrm{h}} + I_\nu(\mathbf{v})}} = \frac{1}{1 + \mathrm{e}^{-\left[\theta_\nu^{\mathrm{h}} + I_\nu(\mathbf{v})\right]}}. \tag{5.48}$$

Such conditional average is a function of the field $\boldsymbol{\theta}^{\mathrm{h}}$ and of the input $I_\nu(\mathbf{v}) = \sum_{i=1}^{N} w_{i\nu} v_i$. Alternatively, one can again rely on the approximation $Q(\mathbf{x})$, leveraging the known formula for conditioning Gaussian distributions:

$$\langle h_\nu \rangle_{Q(\mathbf{h}|\mathbf{v})} = \langle h_\nu \rangle_{Q(\mathbf{x})} + \left(\Sigma^{\mathrm{vh}}\right)^T \left(\Sigma^{\mathrm{v}}\right)^{-1} \left(\mathbf{v} - \langle \mathbf{v} \rangle_{Q(\mathbf{x})}\right), \tag{5.49}$$

where Σ^{vh} and Σ^{v} are respectively the $N \times M$ cross-covariance and the $N \times N$ visible covariance blocks. What is left to compute in Eq. (5.11) is the ensemble cross correlation $\langle v_i h_\mu \rangle_P$. This term can be approximated via the covariance matrix Σ of the probability Q according to:

$$\langle v_i h_\mu \rangle_{P(\mathbf{v},\mathbf{h})} \simeq \langle x_i x_{N+\mu} \rangle_{Q(\mathbf{x})} = \Sigma_{i(N+\mu)} + \langle x_i \rangle_{Q(\mathbf{x})} \langle x_{N+\mu} \rangle_{Q(\mathbf{x})} \,. \tag{5.50}$$

Let's now focus on the EP update equations for the specific case of a binary RBM, where we need to compute the average values with respect to the tilted distribution $Q^{(k)}(\mathbf{x}) = \frac{1}{Z_{Q^{(k)}}} Q^{\setminus k}(x_k) \psi_k(x_k)$. First of all, we need to compute the normalization factor:

$$
\begin{aligned}
Z_{Q^{(k)}} &= \int \mathrm{d}x_k \, Q^{\setminus k}(x_k)\psi_k(x_k) \\
&= \frac{1}{\sqrt{2\pi\Sigma_k}} \int \mathrm{d}x_k \, \mathrm{e}^{-\frac{(x_k-\mu_k)^2}{2\Sigma_k}} \left[\frac{1}{1+\mathrm{e}^{\theta_k}} \delta(x_k) + \frac{1}{1+\mathrm{e}^{-\theta_k}} \delta(1-x_k) \right] \\
&= \frac{1}{\sqrt{2\pi\Sigma_k}} \left[\frac{\mathrm{e}^{-\frac{\mu_k^2}{2\Sigma_k}}}{1+\mathrm{e}^{\theta_k}} + \frac{\mathrm{e}^{-\frac{(1-\mu_k)^2}{2\Sigma_k}}}{1+\mathrm{e}^{-\theta_k}} \right] \\
&= \frac{\mathrm{e}^{-\frac{(1-\mu_k)^2}{2\Sigma_k}}}{\left(1+\mathrm{e}^{-\theta_k}\right)\sqrt{2\pi\Sigma_k}} \left[\frac{1+\mathrm{e}^{-\theta_k}}{1+\mathrm{e}^{\theta_k}} \mathrm{e}^{\frac{1-2\mu_k}{2\Sigma_k}} + 1 \right],
\end{aligned} \tag{5.51}
$$

where we used the notation $\mu_k = \langle x_k \rangle^{(k)}$, $\Sigma_k = \Sigma_{kk}^{(k)}$ and $\theta_k = \theta_k^{\mathrm{v}}$ if $k < N$, whereas $\theta_k = \theta_{k-N}^{\mathrm{h}}$ if $k > N$. We can now compute the average values:

$$
\begin{aligned}
\langle x_k \rangle_{Q^{(k)}} &= \frac{1}{Z_{Q^{(k)}}} \int \mathrm{d}x_k \, x_k Q^{\setminus k}(x_k)\psi_k(x_k) \\
&= \frac{1}{Z_{Q^{(k)}}} \frac{1}{\sqrt{2\pi\Sigma_k}} \frac{\mathrm{e}^{-\frac{(1-\mu_k)^2}{2\Sigma_k}}}{1+\mathrm{e}^{-\theta_k}} \\
&= \left[\frac{1+\mathrm{e}^{-\theta_k}}{1+\mathrm{e}^{\theta_k}} \mathrm{e}^{\frac{1-2\mu_k}{2\Sigma_k}} + 1 \right]^{-1}.
\end{aligned} \tag{5.52}
$$

Since for Boolean variables $\langle x \rangle = \langle x^2 \rangle$, Eq. (5.52) provides all the necessary information to run the EP algorithm.

Positive definiteness

In this section, we presented how Gaussian EP might be used to infer the constituent parameters of an RBM, specifically focusing on the case in which the single unit potentials (i.e. the priors), correspond to Boolean variables. The core of the method is to approximate the model probability $P(\mathbf{x})$ in Eq. (5.37) with a multivariate Gaussian ansatz $Q(\mathbf{x})$. In order for this probability to be well defined, it is necessary for the covariance matrix Σ to be *positive definite*, i.e. its eigenvalues must all be larger than zero (positive variances in the diagonal basis).

In the context of RBM, the precision matrix A possesses a peculiar block structure:

$$
A = \begin{pmatrix} D^{\mathrm{v}} & -w \\ -w^T & D^{\mathrm{h}} \end{pmatrix}, \tag{5.53}
$$

where w is the $N \times M$ weight matrix defining the off-diagonal part, and D^{v} and D^{h} are the diagonal matrices having as elements the inverse EP variances. Since $\Sigma = A^{-1}$, if the covariance matrix is to be positive definite, also the precision matrix has to, which is particularly evident if we consider that the eigenvalues of Σ are the inverse of those of A.

The positive definiteness condition must hold through the whole iterative update of EP parameters. In particular, we must guarantee that the initialization of the univariate variances $\mathbf{d}$ is compatible with this condition. In this perspective, we study the simple case in which all the elements of the variance vector are equal $\mathbf{d} = d(1, \dots, 1)$, and ask ourselves what is the maximum value of d for which A is positive definite.

To answer the question we can employ the useful relation $\mathrm{eig}(A + c\mathbb{I}) = \mathrm{eig}(A) + c$, where c is a constant and $\mathbb{I}$ is the identity matrix. Thus, we need to compute the eigenvalues of the matrix W:

$$\det\left(W - \lambda \mathbb{I}_{N+M}\right) = \det\begin{pmatrix} -\lambda \mathbb{I}_N & -w \\ -w^T & -\lambda \mathbb{I}_M \end{pmatrix} = 0. \tag{5.54}$$

Since also Eq. (5.54) is referred to a block matrix, we can rely on the known relation:

$$\det\begin{pmatrix} A & B \\ C & D \end{pmatrix} = \det(D)\det(A - BD^{-1}C). \tag{5.55}$$

Applying Eq. (5.55) to Eq. (5.54) one gets:

$$(-\lambda)^M \det\left(\frac{ww^T}{\lambda} - \lambda \mathbb{I}_N\right) = 0. \tag{5.56}$$

W is an $(N + M) \times (N + M)$ matrix of rank $2M$. Consequently, the multiplicity of the null eigenvalue $\lambda = 0$ is $N - M$ (for $N - M + 2M = N + M$). The first factor of Eq. (5.56) provides $\lambda = 0$ as a solution of multiplicity M. Then, the second factor is to provide the $2M$ non zero eigenvalues and the null eigenvalue with multiplicity $N-2M$ as further solutions. However, for the positive definiteness problem, we are interested to determine the larger eigenvalue of W in absolute value, which is certainly different from zero. Thus, supposing $\lambda \neq 0$ in the second factor of Eq. (5.56), the eigenvalue equation becomes $\det(ww^T - \lambda^2 \mathbb{I}_N) = 0$, i.e. the non-zero eigenvalues of W are determined by the square roots of the eigenvalues of the matrix ww^T, which is a symmetric positive definite rank-M matrix. In other words, $\pm\lambda_k^{(W)} = \pm\sqrt{\lambda_k^{(ww^T)}}$ for any $\lambda_k^{(W)} \neq 0$. Finally, the sought constant c is determined as $c = \sqrt{\lambda_{\max}^{(ww^T)}} + \epsilon$, with ϵ positive and arbitrarily small.

The previously presented discussion can be extended to the case in which we have two different constants c^{v} and c^{h} in the visible and hidden blocks along the diagonal. In this scenario the eigenvalue equation reads:

$$\det\begin{pmatrix}(c^{\mathrm{v}}-\lambda)\mathbb{I}_N & -w\\ -w^T & (c^{\mathrm{h}}-\lambda)\mathbb{I}_M\end{pmatrix}=0, \tag{5.57}$$

and applying again Eq. (5.55), Eq. (5.57) becomes:

$$(c^{\mathrm{h}}-\lambda)^M\det\left((c^{\mathrm{v}}-\lambda)\mathbb{I}_N-\frac{ww^T}{c^{\mathrm{h}}-\lambda}\right)=0. \tag{5.58}$$

Similarly to the previous case, we need to suppose $\lambda \neq c^{\mathrm{h}}$, checking a posteriori the consistency of this hypothesis. Doing so, we are left with the eigenvalue problem $\det(ww^T-\tilde{\lambda})=0$, where we set $\tilde{\lambda}=(c^{\mathrm{v}}-\lambda)(c^{\mathrm{h}}-\lambda)$. Thus, $\tilde{\lambda}$ are again the non-zero eigenvalues of ww^T, whereas the solution to the original eigenvalue problem is provided by expressing λ as a function of $\tilde{\lambda}$ (solving the second order equation). Finally, the non-zero eigenvalues of W read:

$$\begin{cases}\lambda_{k_+}^{(W)} &= \frac{1}{2}\left[c^{\mathrm{v}}+c^{\mathrm{h}}-\sqrt{4\lambda_k^{(ww^T)}+(c^{\mathrm{v}}-c^{\mathrm{h}})^2}\right],\\ \lambda_{k_-}^{(W)} &= \frac{1}{2}\left[c^{\mathrm{v}}+c^{\mathrm{h}}+\sqrt{4\lambda_k^{(ww^T)}+(c^{\mathrm{v}}-c^{\mathrm{h}})^2}\right].\end{cases} \tag{5.59}$$

In Eq. (5.59), only the first set of eigenvalues can become negative. We consequently need to impose:

$$\begin{aligned}&c^{\mathrm{v}}+c^{\mathrm{h}}-\sqrt{4\lambda_k^{(ww^T)}+(c^{\mathrm{v}}-c^{\mathrm{h}})^2}>0,\\ &\left(c^{\mathrm{v}}+c^{\mathrm{h}}\right)^2>4\lambda_k^{(ww^T)}+(c^{\mathrm{v}}-c^{\mathrm{h}})^2,\\ &c^{\mathrm{v}}c^{\mathrm{h}}>\lambda_k^{(ww^T)},\end{aligned} \tag{5.60}$$

where at the second line of Eq. (5.60) we squared both sides of the inequality. In particular, the inequality holds for every k if $c^{\mathrm{v}}c^{\mathrm{h}}>\lambda_{\max}^{(ww^T)}$, which in turns guarantees that the precision matrix is positive definite. We notice that the previous condition over c is recovered by setting $c^{\mathrm{v}}=c^{\mathrm{h}}=c$.

A peculiar behavior that emerged when running the EP algorithm on simple toy RBM architectures is related to the cavity variances $\Sigma_{kk}^{(k)}=\Sigma_k$, which became negative along the iterative process. Such behavior might look as an inconsistency of the method, however, the $Q^{\setminus k}$ are not actual probability distributions, but are only needed to define the tilted $Q^{(k)}$ distributions, that conversely must be well defined.

5.3 Inference of RBM on MNIST

As a preliminary test, we applied the EP-based learning strategy to infer an RBM model probability for the MNIST dataset, which represents a common benchmark for testing

the capabilities of deep and shallow networks. Each datapoint is a handwritten digit, encoded in a 28×28 pixels image, which can be also represented as an $N = 784$ components vector $\mathbf{y}^d$. Each pixel y_i^d is defined over a gray-scale that goes from 0 to 255, but for our purposes, we proceed by binarizing them rescaling every y_i^d between $[0,1]$, and defining the binarized version v_i^d according to a threshold 0.5:

$$\begin{cases} v_i^d = 1 & y_i^d \geq 0.5, \\ v_i^d = 0 & y_i^d \leq 0.5. \end{cases} \tag{5.61}$$

Consequently, we are left with a dataset composed of $D = 60000$ binary vectors $\mathbf{v}^d$, constituting the input of the visible layer. We choose a number of hidden layer units of $M = 400$, since it appears to be a common shared choice in the literature. To define the training dataset, we do not use the whole set of points, but we rather choose a random subset of 10^4 digits.

To begin the learning procedure we need to initialize the model parameters, that in the case of a binary RBM coincide with the weights w and the visible and hidden fields $\boldsymbol{\theta}^{\mathrm{v}}$ and $\boldsymbol{\theta}^{\mathrm{h}}$. Each weight component is initialized as an independent standard normal, rescaled by the square root of the number of visible units $w_{i\mu} \sim \mathcal{N}(0,1)/\sqrt{N}$. The visible binary fields are initialized according to the average density of the pixels $\rho^{\mathrm{v}} \simeq 0.23$, i.e. $\theta^{\mathrm{v}} = \log \frac{\rho^{\mathrm{v}}}{1-\rho^{\mathrm{v}}}$. The hidden fields are initialized in a similar manner, but with the average densities ρ_μ^{h} which are random uniform variable between 0 and 1.

Then, the learning process is defined by minimization of the objective in Eq. (5.10) (actually minus the likelihood), performing a gradient descent algorithm based on the gradients in Eqs. (5.11) and (5.47). Similarly to what was presented for BML, the method relies on the hyperparameter *learning rate* η, defining the pace at which parameters are updated. Choosing the correct value of the learning rate is fundamental to obtain a good learning process, and a trial and error strategy must be pursued. Usually, a significantly large learning rate is set at the beginning of the process (e.g. $\eta \sim 0.1$), so to accelerate the learning in this phase in which the landscape is supposed to be smooth, and then it is shrinked in the final stages of the learning in order to help convergence.

We decided to implement the learning in a stochastic gradient descent (SGD) fashion, introducing stochasticity in the optimization process. The idea of SGD is to use random subsets of the dataset in order to compute an approximation of the gradient of the objective function. Practically speaking, the training dataset is randomly divided in a number of *batches* of a certain size, updating the model parameters for each batch. An iteration over all batches is also referred to as an *epoch* of the learning process. The main advantage of SGD is that it allows for a faster computation of the gradient, since the empirical averages are computed over smaller sets. On the other hand, convergence is usually slower because of random fluctuations in the process. However, the introduced stochasticity can help better navigating the landscape when this is very rough [103], which is usually the case when training deep neural network architectures.

For our case, we set the size of the batch to $b_s = 500$ datapoints, so that each epoch is composed by 20 parameters updates.

During the learning process we monitor the trend of some benchmark quantities such as the likelihood, the error between empirical and model connected correlations, the weight sparsity $\hat{p}$ and the model effective inverse temperature W_2. To track these quantities we save the state of the RBM at logarithmically spaced learning times, measured in units of the number of updates. In Fig. 5.2 we show the aforementioned trends. Panel (a) displays the log-likelihood evolution both for the train and the test set. The two trends are almost perfectly overlapping, and they both display an inflection point around 100 updates. Overall, the log-likelihood appears to be a monotonically increasing function of the number of updates. In panel (b) we show the trend of the error on the connected correlation function, which is defined as:

$$\epsilon^{(2)} = \frac{2}{N(N-1)} \sum_{i=1}^{N-1} \sum_{j=i+1}^{N} \left| \Sigma^{\mathrm{v}}_{ij} - \left(\Sigma^{\mathrm{v}}_{\mathrm{MC}}\right)_{ij} \right|^2 , \tag{5.62}$$

where $\Sigma^{\mathrm{v}} = \overline{\mathbf{v}\mathbf{v}^T} - \overline{\mathbf{v}}\,\overline{\mathbf{v}}^T$ is the empirical connected correlation function, whereas $\Sigma^{\mathrm{v}}_{\mathrm{MC}}$ is the visible connected correlation function computed from a thermalized Monte Carlo simulation of the model probability P. After an initial monotonic decrease, the trend becomes more erratic. Such a behavior can be related to what happens in the intermediate phase of the learning, when EP displays the emergence of multiple attractors (see Sec. 5.3.2).

The trend of the weight sparsity is reported in panel (c). Such feature is quantified by means of the parameter $\hat{p}$ defined as:

$$\hat{p} = \frac{1}{MN} \sum_{\mu=1}^{M} \frac{\left(\sum_{i=1}^{N} w_{i\mu}^2\right)^2}{\sum_{i=1}^{N} w_{i\mu}^4}, \tag{5.63}$$

as it was introduced in [168]. An interesting property of $\hat{p}$ is that it provides a scale invariant score, i.e. $\hat{p}(\lambda w) = \hat{p}$ for weights rescaling $w \to \lambda w$. The weights will be sparser the lower the estimator $\hat{p}$ is. After an initial phase in which $\hat{p}$ slightly increases, it begins to decrease steadily from around iteration 100, reaching a value as low as $\hat{p} \sim 0.2$.

Finally, the trend of the effective inverse temperature is showed in panel (d). The estimating parameter W_2 is defined according to:

$$W_2 = \frac{1}{M} \sum_{i,\mu=1}^{N,M} w_{i\mu}^2, \tag{5.64}$$

and it was as well introduced in [168]. Even if the probabilistic model provided by the RBM is always set at temperature $T = 1$, the magnitude of the weights can be interpreted as an effective temperature of the model. Specifically, the larger the

modulus of the weights, the lower is the effective temperature, coherently with the definition provided by Eq. (5.64). The estimator W_2 displays a monotonically increasing trend, that appears exponential in log-scale, so that the growth is actually linear with the number of updates.

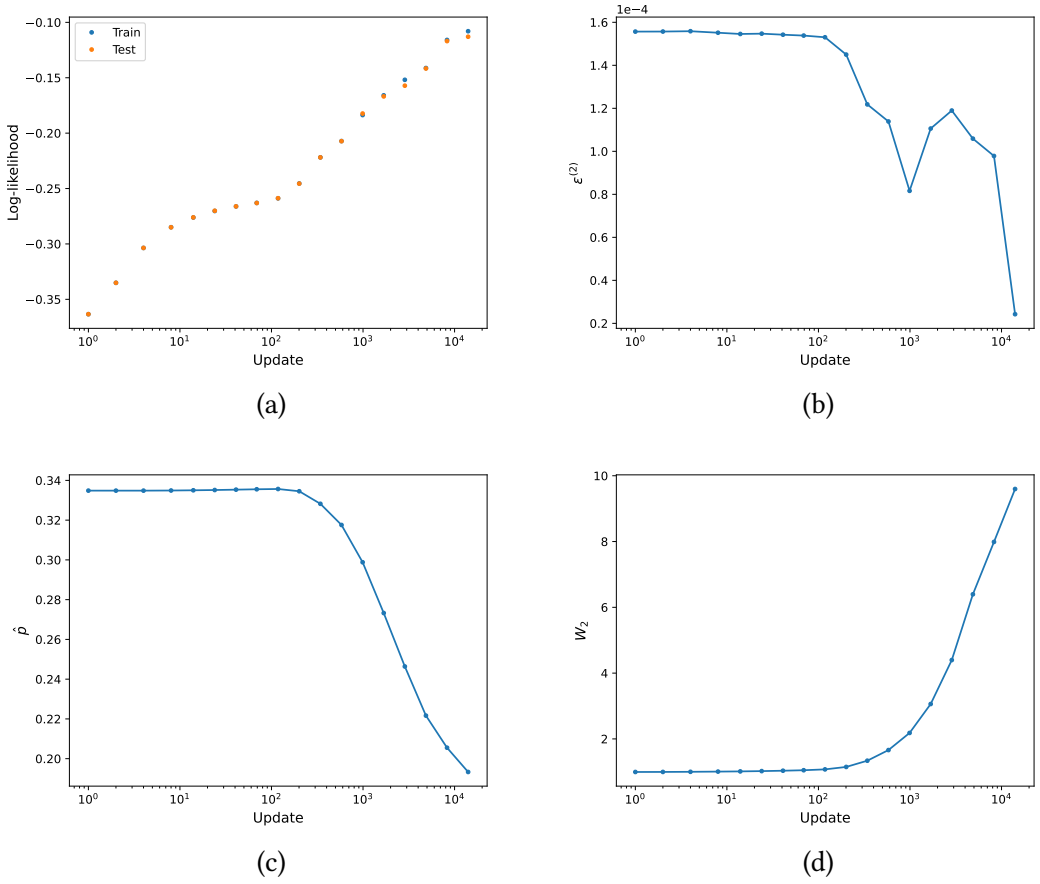


Figure 5.2: Global trends of some benchmarking quantities of the learning process, as a function of the number of the updates. All the plots are shown with the horizontal axis in log-scale. Panel (a): trend of the log-likelihood computed both over the train and the test sets. Panel (b): trend of the error on the second moment, i.e. discrepancy between the connected correlation function related to the model probability and the one of the empirical data. Panel (c): trend of weight sparsity estimator $\hat{p}$. Panel (d): trend of the effective inverse temperature estimator of the model W_2.

5.3.1 Early stage of the learning process

At the beginning of the learning the algorithm starts moving rapidly towards higher values of the likelihood, suggesting that this phase is characterized by a smooth landscape. The first statistical feature that is learnt by the RBM architecture is the average of the data, that after few iterations is pretty accurately reproduced by the model, as it can be noticed from Fig. 5.8a, where we show the scatter plots between the empirical statistics and the one provided by the model probability $P(\mathbf{v})$ after 24 updates of the weights. Since for Boolean variables the average allows to reconstruct the variance $\mathrm{Var}[v] = \langle v^2 \rangle - \langle v \rangle^2 = \langle v \rangle (1 - \langle v \rangle)$, also the diagonal of the covariance matrix is correctly reproduced by the model at this stage.

The weights remain Gaussian at this learning stage, though displaying an increasing variance. Moreover, if the visible components of the weights are visualized for a random subset of hidden units, no internal structure emerges. In panels (a) and (b) of Fig. 5.6, we show respectively a histogram of all the weights components in semi-log scale, and a graphical representation of a subset of 16 randomly extracted visible weights $\mathbf{w}_\mu$ after 14 SGD updates.

It is interesting to compare the capability of the method to approximate the likelihood gradient with alternatives based on Monte Carlo sampling. Specifically, we considered two methods: Random-k (Rdm-k), in which the Monte Carlo chain is initialized randomly and k Gibbs sampling steps are performed; contrastive divergence-k (CD-k), where the visible initial state is initialized from a datapoint configuration and again k sampling steps are realized. In Fig. 5.7, we show the comparison between the methods. A long thermalized Monte Carlo is used as a benchmark of correct estimate of the statistics necessary to compute the gradient, coinciding with the averages $\langle \mathbf{v} \rangle_P$ and $\langle \mathbf{h} \rangle_P$ and the cross correlations $\langle \mathbf{v}\mathbf{h}^T \rangle_P$. We stress that in making this comparison, Rdm-k and CD-k are not used as learning algorithms, but only to estimate the required statistics at a fixed valued of the RBM parameters. In Figs. 5.7a and 5.7b, we show the comparison between the different methods after 14 steps of weights update. We choose the number of Gibbs sampling steps to be $k = 500$, for both Rdm-k and CD-k. Remarkably, the estimates obtained with the EP approximation are in very good agreement with the outcome of the thermalized Monte Carlo, and better than those obtained with both alternative methods.

It is also interesting to study the transfer function defined in Eq. (5.8), a quantity that reflects if a hidden unit is activated or not. Specifically, we compute it for every hidden unit, averaging over the visible configurations in the training set. For Boolean variables, the transfer function has expression:

$$H_\mu \left(I_\mu(\mathbf{v}) \right) = \Theta \left(I_\mu(\mathbf{v}) + \theta^{\mathrm{h}}_\mu \right), \tag{5.65}$$

where $\Theta(x)$ is the Heaviside function. Thus, the hidden unit is activated if $I_\mu(\mathbf{v}) > -\theta^{\mathrm{h}}_\mu$ and not activated otherwise. In Fig. 5.9a we show the histogram of the average

transfer obtained after 14 updates of the weights, from which it emerges that the majority of the hidden units are either active or inactive for all the data points.

5.3.2 Emergence of multiple EP attractors

As the learning process goes on, an unexpected feature emerges: the convergence point of the EP parameters becomes not unique, and the specific reached attractor depends on the specific initialization of $\{\mathbf{a}, \mathbf{d}\}$. This feature is somewhat surprising, because EP is usually meant to provide a global approximation of the target probability P. Instead, when P becomes multimodal, EP begins to separately converge to the different modes, so that the actual approximation provided by the algorithm is to be intended as a *mixture* of multivariate Gaussians:

$$Q(\mathbf{x}) = \sum_{\alpha=1}^{K} \rho_\alpha Q_\alpha(\mathbf{x}), \tag{5.66}$$

where $\alpha = 1, \dots, K$ are the individual modes, ρ_α are the weights of the mixture, and each Q_α is a multivariate Gaussian having mean $\langle \mathbf{x} \rangle_{Q_\alpha}$ and covariance matrix Σ^α approximating a specific mode. The single mode statistics can be used to compute the first and second moment of the total mixture:

$$\langle \mathbf{x} \rangle_Q = \sum_{\alpha=1}^{K} \int \mathrm{d}^N x \, \rho_\alpha \mathbf{x} Q_\alpha(\mathbf{x}) = \sum_{\alpha=1}^{K} \rho_\alpha \langle \mathbf{x} \rangle_{Q_\alpha}, \tag{5.67}$$

which is the weighted mean of the single mode averages. The non-connected correlation function can be expressed as:

$$\begin{aligned}
\left\langle \mathbf{x}\mathbf{x}^T \right\rangle &= \sum_{\alpha=1}^{K} \rho_\alpha \int \mathrm{d}^N x \, \mathbf{x}\mathbf{x}^T Q_\alpha(\mathbf{x}) \\
&= \sum_{\alpha=1}^{K} \rho_\alpha \left[\int \mathrm{d}^N x \left(\mathbf{x} - \langle \mathbf{x} \rangle_{Q_\alpha} \right) \left(\mathbf{x} - \langle \mathbf{x} \rangle_{Q_\alpha} \right)^T Q_\alpha(\mathbf{x}) \right. \\
&\left. + \int \mathrm{d}^N x \, \mathbf{x} \left\langle \mathbf{x}^T \right\rangle_{Q_\alpha} Q_\alpha(\mathbf{x}) + \int \mathrm{d}^N x \left\langle \mathbf{x}^T \right\rangle_{Q_\alpha} \mathbf{x} Q_\alpha(\mathbf{x}) - \langle \mathbf{x} \rangle_{Q_\alpha} \left\langle \mathbf{x}^T \right\rangle_{Q_\alpha} \right] \\
&= \sum_{\alpha=1}^{K} \rho_\alpha \left[\Sigma^\alpha + \langle \mathbf{x} \rangle_{Q_\alpha} \left\langle \mathbf{x}^T \right\rangle_{Q_\alpha} \right],
\end{aligned} \tag{5.68}$$

which is again the weighted sum of the single modes correlation function. Finally, from Eq. (5.68) we can readily derive the mixture covariance matrix:

$$\Sigma = \left\langle \mathbf{x}\mathbf{x}^T \right\rangle - \langle \mathbf{x} \rangle \left\langle \mathbf{x}^T \right\rangle = \sum_{\alpha=1}^{K} \rho_\alpha \left[\Sigma^\alpha + \left(\langle \mathbf{x} \rangle_{Q_\alpha} - \langle \mathbf{x} \rangle_Q \right) \left(\langle \mathbf{x} \rangle_{Q_\alpha} - \langle \mathbf{x} \rangle_Q \right)^T \right]. \tag{5.69}$$

If one is able to access the statistics of all the modes, then it is in principle still possible to estimate the likelihood gradient employing the global mixture as the approximation of the model probability P. The drawback of this approach is that even if one is able to identify all the mixture components, there is still no simple way to determine the weights of the mixture ρ_α, i.e. how to mix together the single modes.

A possible solution is provided by the iterative algorithm *expectation maximization* (EM) [106], which is often used to cluster data according to a Gaussian mixture. EM is able to simultaneously determine both the weights of the mixture and the mean vector and covariance matrix of the single components. For our specific need, the Gaussian modes are already given by EP and we just need to determine the ρ_α. However, a subtlety arises, for we should in principle perform EM on configurations generated according to the model probability P, requiring the realization of Monte Carlo sampling simulations that we aim to avoid in our method. As a possible alternative, we take inspiration from the CD approach, performing a partial EM on the data themselves. This procedure will certainly introduce biases in the gradient estimate, being more accurate near convergence of the algorithm, where P should be a good approximation of the empirical statistics.

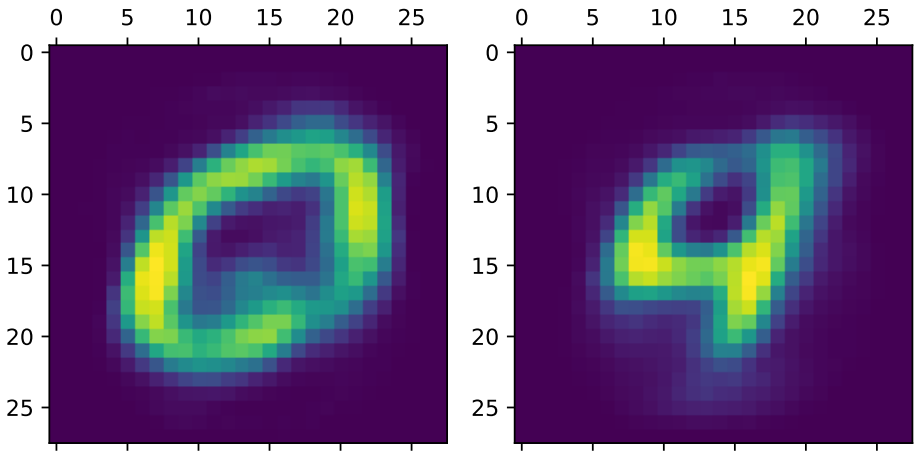
In the following, we report the fundamental steps of the EM strategy to compute the mixture weights.

- Given a mini batch of size b_s and the number of mixture components K, initialize the weights $\rho_\alpha^0 = 1/K$ and define the evidences $r_{i\alpha}$ that data point i ($i = 1, \dots, b_s$) belongs to the α component.
- For each data point and component compute the evidence according to $r_{i\alpha}^t = \frac{\rho_\alpha^t Q_\alpha(\mathbf{x}_i)}{\sum_{\alpha'=1}^K \rho_{\alpha'}^t Q_{\alpha'}(\mathbf{x}_i)}$.
- Update the mixture weights according to $\rho_\alpha^{t+1} = \frac{1}{b_s} \sum_{i=1}^{b_s} r_{i\alpha}^t$.
- If $\max \left[\left| \rho_\alpha^{t+1} - \rho_\alpha^t \right| \right] < \delta$, where δ is an arbitrary tolerance, stop the procedure and return the current ρ_α values.

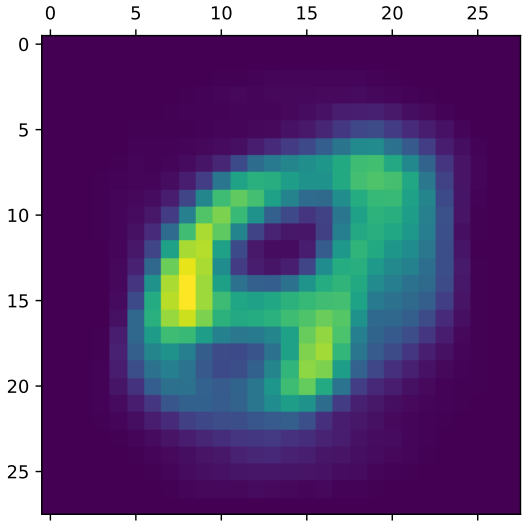
In Fig. 5.3, we report a pair of visible mean vectors related to a two components mixture in a digit like representation, together with the corresponding global visible average of the mixture computed via the EM procedure.

Interestingly, the Gaussian mixture obtained from the single EP components is able to reproduce quite well the statistics of the model distribution, as it can be appreciated from Fig. 5.4, where we report a comparison between averages and correlations (connected and not) of a long equilibrated Monte Carlo. The shown statistics include both visible and hidden units.

Practically speaking, how do we deal with the rise of multiple attractors of the EP algorithm? Our proposal is to realize a replicated EP version, in which we set a certain



(a)



(b)

Figure 5.3: Panel (a): average visible vectors in digit-like representation of the two components of the mixture. Panel (b): visible average in digit-like form of the resulting mixture. The mixture weights are computed with the EM approach at update 71.

number of independent replicas R, whose parameters $\{\mathbf{a}^r, \mathbf{d}^r\}$ $(r = 1, \dots, R)$ are initialized randomly. Then, we check how many unique replicas K are present among the converged set of parameters, and employ these unique replicas to estimate the gradient determining the mixture weights with the partial EM procedure on the minibatch. The advantage of this approach is that, since the replicas are totally independent of each

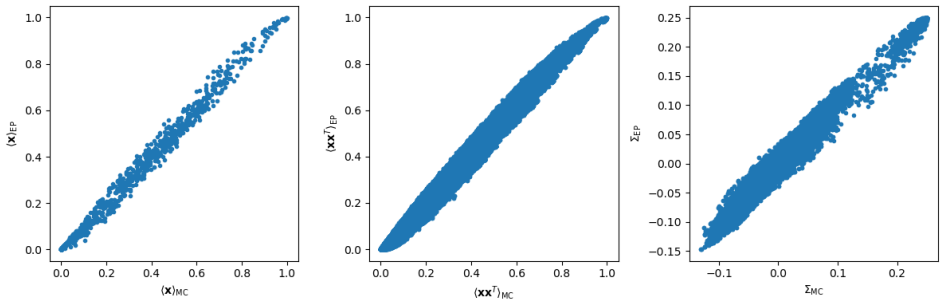


Figure 5.4: Scatter plot between the thermalized Monte Carlo statistics (first and second moment) and the one provided by the EP mixture $Q(\mathbf{x})$. From left to right: Monte Carlo averages $\langle \mathbf{x} \rangle_{\text{MC}}$ versus EP average values $\langle \mathbf{x} \rangle_{\text{EP}}$; Monte Carlo correlations $\langle \mathbf{x}\mathbf{x}^T \rangle_{\text{MC}}$ versus EP correlations $\langle \mathbf{x}\mathbf{x}^T \rangle_{\text{EP}}$; Monte Carlo connected correlations Σ_{MC} versus EP covariance matrix Σ_{EP}.

other, the different EP realizations can be run in parallel.

In Fig. 5.5 we show the trend of the number of unique replicas as a function of the learning time. In particular we show two versions of this trend: in panel (a) we report the number of unique replicas as a function of the learning epoch, whereas in panel (b) we report the average number of replicas, mediated over a certain time window, to obtain a smoother version of the trend. From both plots it emerges how the number of unique replicas increases with time, and the coarse grained version seems to be composed of several linear trends of different slopes. The shifts between the different trends are related to changes of the learning rate, which is progressively diminished along the learning process.

5.3.3 Intermediate learning phase

At the intermediate phase of the learning the likelihood increase rate starts slowing down, whereas the gradient norm, which initially diminishes steadily, increases to an order of magnitude higher values. In this phase, at each learning step the model probability P is characterized by one or few modes that might coincide with a subset of the digits or a mixture of them. Consequently, neither the averages nor the correlation of the empirical data are accurately reproduced, as it is shown in Fig. 5.8b, obtained at learning epoch 143. Depending on the specific individual modes of the model probability, the Pearson correlation coefficient between EP covariance matrix and empirical connected correlations (and equivalently $\langle \mathbf{x}\mathbf{x}^T \rangle_{\text{MC}} - \langle \mathbf{x} \rangle_{\text{MC}} \langle \mathbf{x}^T \rangle_{\text{MC}}$) oscillates between 0.4 and 0.6.

The weights w become increasingly sparse, at the same time broadening towards larger extremal values, as can be appreciated from Fig. 5.6c. Moreover, as it can be seen from 5.6d, they also display an internal structure which sometimes resembles individual

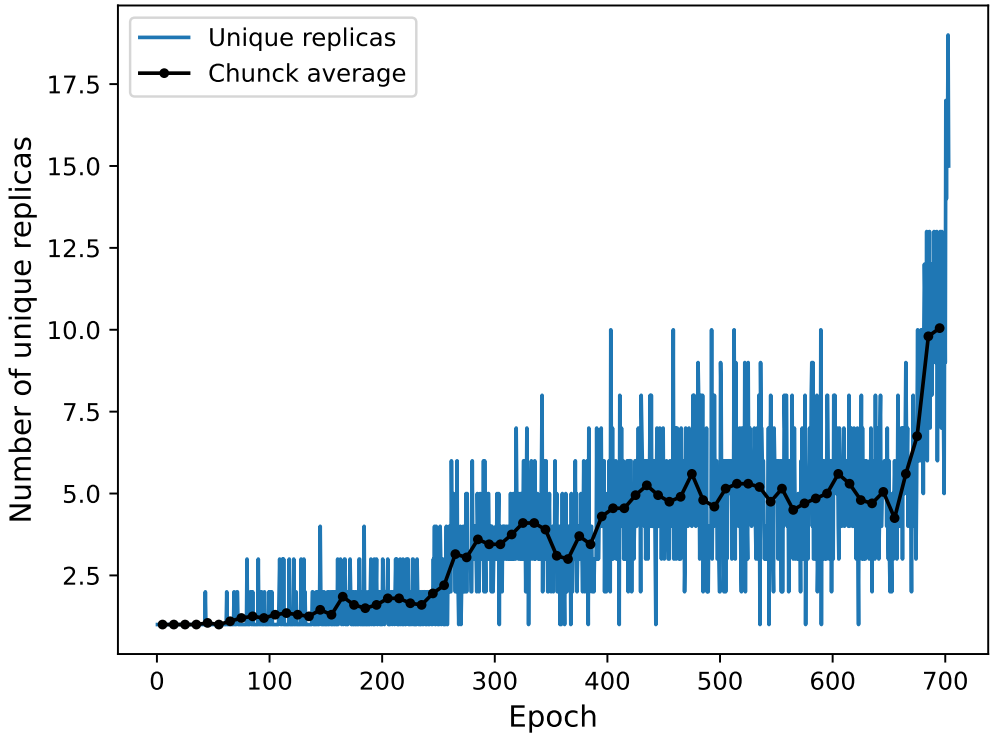


Figure 5.5: Trend of the unique number of EP replicas throughout the learning process. In blue the number of unique replicas at an epoch is reported, whereas in black the average over an epochs chunck is showed.

digits and in other cases appears to represent more complex features.

We can again compare the gradient estimate obtained via the EP approximation and with Rdm-k and CD-k methods. At epoch 143, EP is again able to provide moment estimates that are in good agreement with a long thermalized Monte Carlo simulation, and better than both Rdm and CD for all the selected number of k steps (namely $k =$ 10, 50, 100, 150, 200, 250, 500). In Figs. 5.7c and 5.7d we show a comparison between the different methods for $k = 50$. As the learning proceeds, sampling methods generally need a higher number of sampling steps in order to provide accurate estimates of the model probability statistics. In particular, CD seems to need a smaller minimum number of steps k with respect to Rdm to obtain accurate values of the means and cross correlation, as it can be noticed from the comparison between Fig. 5.7c and Fig. 5.7d.

The mean transfer function also changes significantly from the first stages of the learning, as it can be seen from Fig 5.9b. Two small peaks in 0 and 1 are still visible, but the majority of the histogram *mass* is concentrated at low values of the transfer function.

5.3.4 Final learning stages

The final stages of the learning procedure are characterized by a significant increase in the number of possible attractors of the EP algorithm, resulting in a higher computational burden if one aims to catch all the modes of the model probability. Moreover, EP dynamics slows down because the parameters $\{\mathbf{a}, \mathbf{d}\}$ might oscillate among the different attractors. This makes estimating the likelihood gradients (Eqs. (5.47) and (5.11)) problematic, as shown in Figs. 5.7e and 5.7f, where we compare a thermalized Monte Carlo statistics with the results obtained via EP and both Rdm-k and CD-k methods at epoch 703 of the learning. Since at the final stages of the learning the marginal model probability $P(\mathbf{v})$ is close to the empirical one, Rdm-k struggles significantly more than CD-k to reach equilibrium, as the initial configuration of the latter are likely to be close to thermalization. Consequently, whereas EP still performs better than Rdm-200, CD-10 is already able to surpass EP accuracy. This can be ascribed to the fact that we did not have access to enough unique attractors so to exhaustively cover the modes of the model probability.

On the other hand, the learnt model probability begins to reproduce with a higher accuracy the empirical statistics, as it can be observed in Fig. 5.8c, where we show the scatter plot between the averages and correlations computed from equilibrium samples of the P and the empirical ones, at 703 epochs of the learning. The related Pearson correlations are reported as inserts in the plots. The increased accuracy in reproducing the empirical statistics can also be noted from the plummeting of the $\epsilon^{(2)}$ at the last point of panel (b) in Fig 5.2.

The RBM weights, reported in panel (a) of Fig. 5.6e, continue to display the trend already observed at the intermediate learning phase, as can also be noted from Fig. 5.2 panel (c) and (d), where a steadily decreasing and increasing value of the parameters $\hat{p}$ and W_2 respectively is observed. Moreover, from Fig. 5.6f, we can see that the weights features become progressively more abstract, and it is harder to recognize digits-like shapes among them.

The histogram of the average activation functions Fig. 5.9c, up to the number of epochs we reached, looks very similar to the one obtained in the intermediate learning phase. This can be viewed as an issue of the learning process, since one expects the final set of the RBM parameters to be such that only a small fraction of the hidden units are active. In contrast, we observe an average activation of around $\lesssim M/2$ hidden units. We do not have a clear explanation for this outcome, and it is indeed something we wish to further investigate in the future.

5.4 Conclusions and perspectives

In this chapter we discussed how to use the iterative algorithm EP to learn the constituents parameters of an RBM. The founding idea of the proposed method is to employ

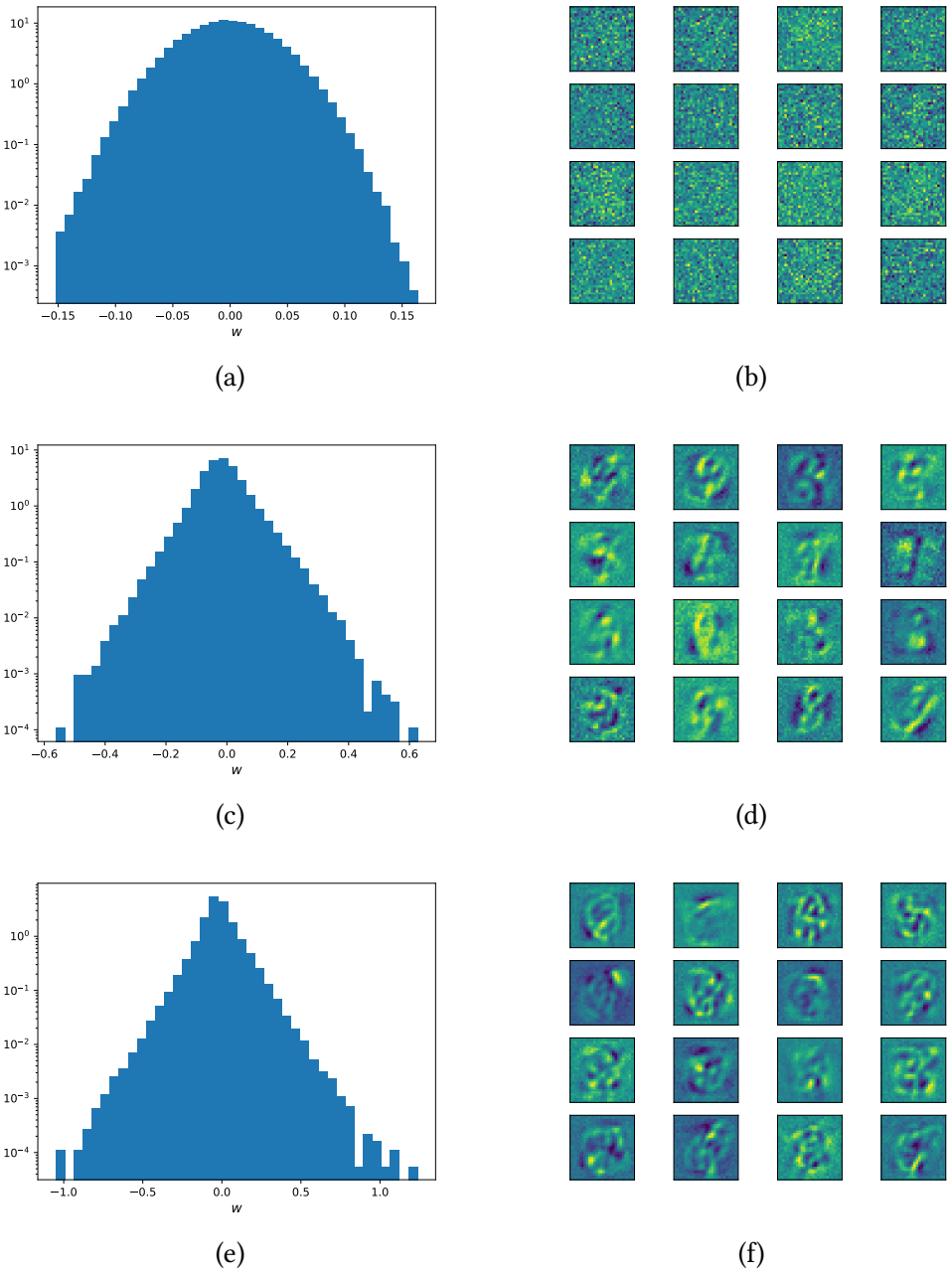
EP in order to obtain a multivariate Gaussian approximation $Q(\mathbf{x})$ of the model probability function $P(\mathbf{x})$ associated to the RBM. Indeed, unsupervised learning of RBM is based on a gradient ascent algorithm for the likelihood function (Eq. (5.10)), and the derivatives of such objective with respect to the model parameters (Eqs. (5.11) and (5.47)) turn out to be a function of some average values of the model probability, which can be approximately computed via the EP probability function.

When testing the method on the MNIST benchmark dataset, we found an interesting behavior: at a certain point of the learning process the EP algorithm stops converging to a unique solution, and a set of possible attractors arise. The specific one to which EP converges depends on the initialization of the EP parameters $\{\mathbf{a}, \mathbf{d}\}$. These attractors can be interpreted as the components of a mixture of multivariate Gaussians, that altogether provide the approximation of the model probability $P(\mathbf{x})$. The drawback of this scenario is that the individual weights of the mixture cannot be computed employing the EP algorithm itself. To workaround this issue, we rely on a partial version of the EM algorithm to be applied as if we were clustering the data within a minibatch among the mixture components. Such a strategy allows to obtain an approximation of the likelihood gradient that is generally superior to both Rdm and CD k-steps approaches for $k \lesssim 100$, up to the final stages of the learning. There, the increased number of possible attractors makes the inference process computationally harder. In this perspective, the major limitation of the method is represented by the slowing down of the EP algorithm when multiple attractors arise. Thus, future efforts will be devoted to try to improve the convergence speed of EP in the advanced phase of the learning process, and to better understand the relation occurring between EP parameters initialization and the corresponding reached attractor.

Furthermore, we would like to apply the EP-based learning strategy to other datasets besides the MNIST one, and in particular to MSA of protein sequence data. To do so, we need to carefully define the prior over the visible units. The natural choice would be to consider a Potts field potential, yielding an RBM energy function:

$$E(\mathbf{v}, \mathbf{h}) = -\sum_{i=1}^{N}\sum_{\mu=1}^{M} w_{i\mu}(v_i)h_\mu + \sum_{i=1}^{N} g_i(v_i) + \sum_{\mu=1}^{M} U_\mu(h_\mu), \tag{5.70}$$

where $g_i(v_i)$ is a Potts field contribution, and the weights become a function of the specific visible units value v_i. Eq. (5.70) is not suited to interpret the interaction term as if it were Gaussian. To overcome this issue we can rely on the one-hot encoding representation of sequence data, which was introduced in Sec. 2.3.2. By doing so, the potential over the visible units becomes binary, as the one employed for the MNIST dataset. However, one-hot encoded data possess a specific block structure such that among the q entries of a block, only a single component can be equal to 1. The EP algorithm can in principle be used to impose that $\sum_{k=1}^{q} v_{k+q(i-1)} = 1$, i.e. the sum of the q entries related to each protein site block $i = 1, \dots, L$ is normalized. Such a constraint can be imposed within the EP framework by using a Gaussian elimination method.



(a) (b) (c) (d) (e) (f)

Figure 5.6: Weight appearance at different stages of the learning. **Early stages** after 14 updates. Panel (a): Gaussian distribution of the weigths. Panel (b): no evident internal structure. **Intermediate stages** after 143 training epochs. Panel (c): the weigths begin to display an exponential, i.e. sparse distribution. Panel (d): emergence of an internal structure, either coinciding with individual digits or with more complex features. **Final stages** after 703 training epochs. Panel (e): the weights remain sparse and become progressively broader. Panel (f): emergence of abstract features.

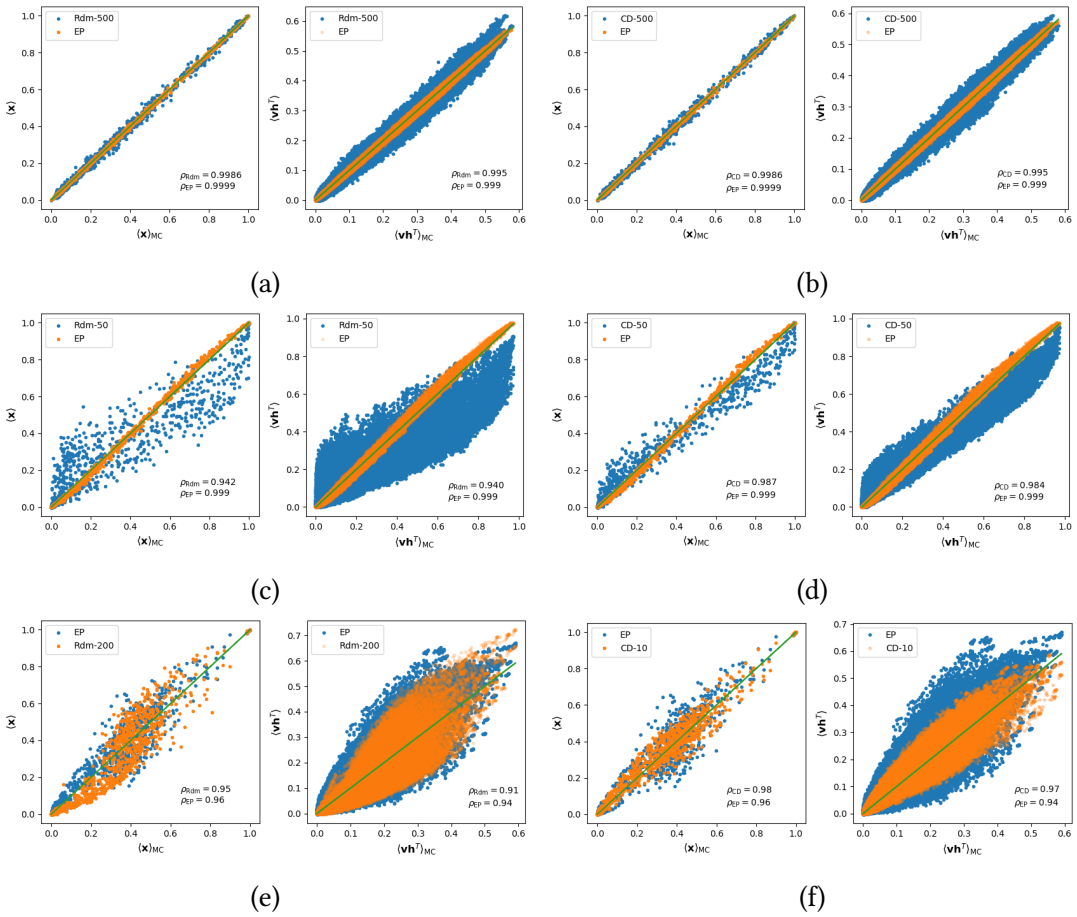


Figure 5.7: Comparison of the likelihood gradient estimate at different learning stages and for different algorithms. The quantities required to estimate Eqs. (5.11) and (5.47) are the average vector $\langle \mathbf{x} \rangle$ and the cross correlation $\langle \mathbf{v}\mathbf{h}^T \rangle$. We consider the estimates provided by EP, CD-k and Rdm-k, and we compare them with a long thermalized Monte Carlo. In the insert of each plot, the corresponding Pearson correlation coefficients are reported. **Early stages**. Panel (a): results provided by Rdm-500 and the EP approximation. Panel (b): results provided by CD-500 and the EP approximation. **Intermediate stages**. Panel (c): results provided by Rdm-50 and the EP approximation. Panel (d): results provided by CD-50 and the EP approximation. **Final stages**. Panel (e): results provided by Rdm-200 and the EP approximation. Panel (f): results provided by CD-10 and the EP approximation.

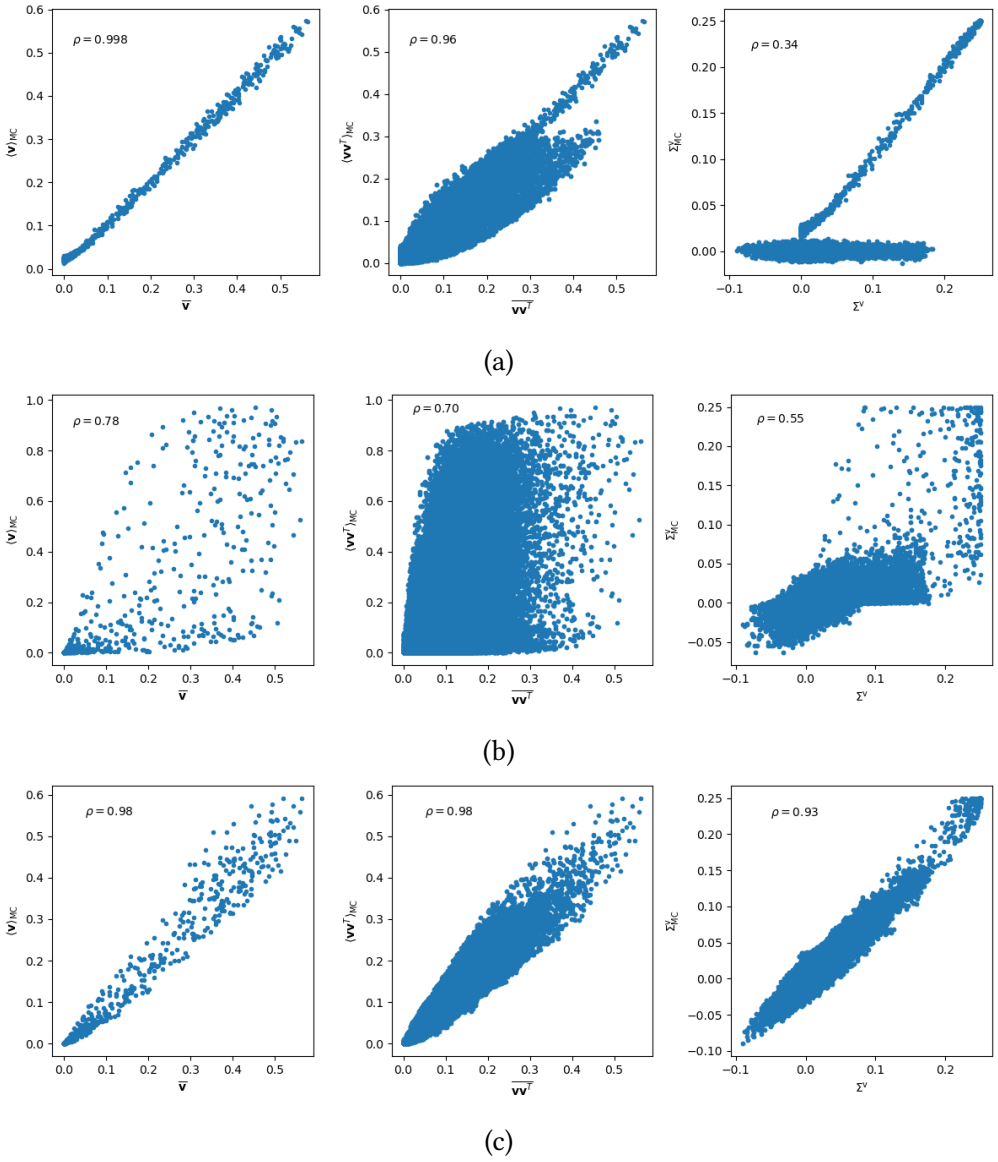


Figure 5.8: Scatter plot between the empirical statistics (first and second moments) and the one provided by a long thermalized Monte Carlo of the model probability $P(\mathbf{v})$, at different learning stages. From left to right: empirical averages $\overline{\mathbf{v}}$ versus model average values $\langle \mathbf{v} \rangle_P$; empirical correlations $\overline{\mathbf{v}\mathbf{v}^T}$ versus model correlations $\langle \mathbf{v}\mathbf{v}^T \rangle_P$; empirical connected correlations Σ^{v} versus visible covariance matrix Σ^{v}_P. **Early stages** in panel (a): the model probability reproduces the empirical averages and the variances. **Intermediate stages** in panel (b): the model probability is concentrated on one or few data features and does not consequently reproduce neither averages or correlations. **Final stages** in panel (c): the model probability embraces globally the dataset, as it is able to reproduce both average values and correlations.

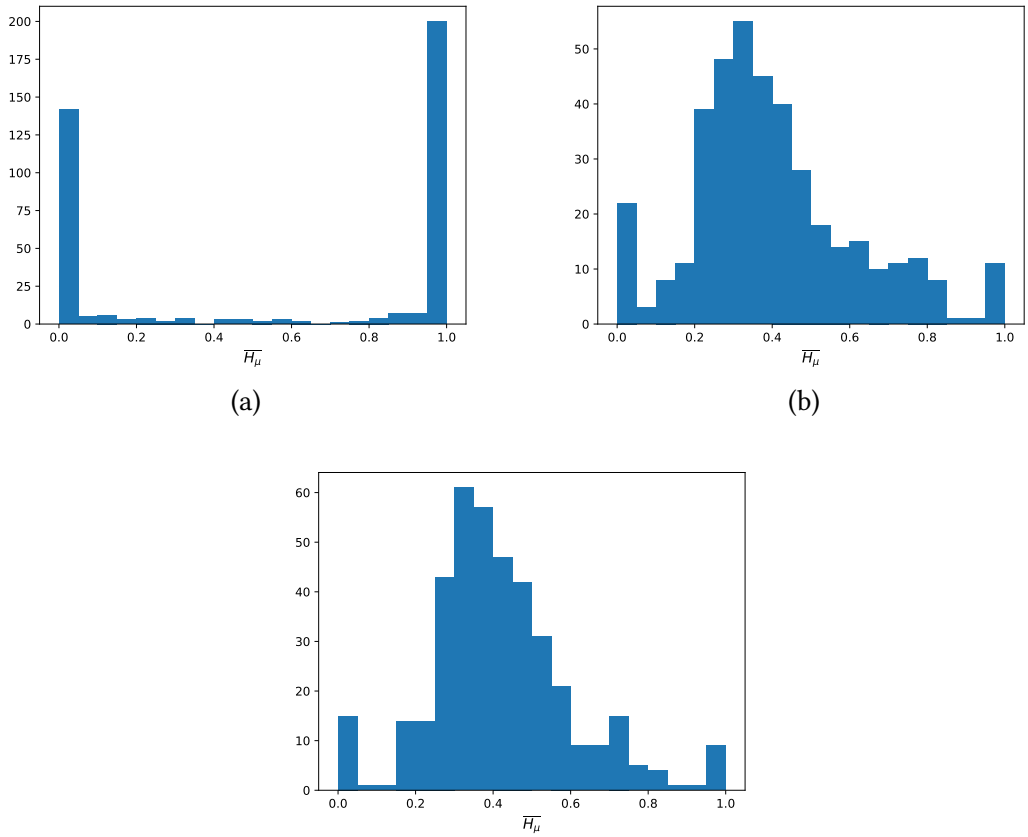


Figure 5.9: Histograms of the average transfer function at different learning stages. The average is computed over the training set. **Early stages** in panel (a): the response is binary as the hidden units are either activated or not. **Intermediate stages** in panel (b): the histogram mass concentrates at relatively low values of $\overline{H_\mu}$, with two small peaks at 0 and 1. **Final stages** in panel (c): similar to the intermediate stages, but the peaks at the extrema decrease in height.

Chapter 6

Overview and conclusions

In this thesis we discussed in Chs. 3 and 4 two inference methods for protein sequence data produced by laboratory experiments, whereas in Ch. 5 we described how to apply the iterative algorithm expectation propagation (EP) to the problem of learning Restricted Boltzmann Machines (RBM).

Protein sequence data are becoming increasingly more available, allowing for the application of statistical physics inspired methods, especially from the perspective of inverse problems, where the constituent parameters of the model are inferred from the available data. In this framework, the generalized Potts model (GPM) has proven to be a valuable tool to describe relevant properties of proteins in a variety of fields: contact prediction and structure determination, protein family assignment, prediction of mutational effects, prediction of protein-protein interactions, generation of functional sequences. Recently, deep supervised methods are gaining considerable attention in the biological domain, and in particular for sequence data. AlphaFold [85] deep architecture has been developed to predict de-novo tertiary structure from protein sequences, practically solving the long standing problem of folding. Deep learning architectures have also been used to predict the effect of mutations [129], to study the relation between sequence and function [63] and for protein engineering and generation [4, 148, 80, 177].

Among the various machine learning methods, transformers architectures and natural language models in general are becoming increasingly popular in the context of protein sequence data [127, 128], for which they proved to be able to encode both structural and phylogenetic information, through unsupervised or self-supervised learning approaches. Moreover, they can be used as a starting point for *fine-tuning* processes, in which the learned parameters are specialized through a supervised learning approach on a limited labeled dataset.

In this perspective, our proposed models provide unsupervised learning methods on local datasets, such as those produced by deep mutational scanning and directed evolution experiments. The models attempt to build a statistical framework which is able, at least effectively, to take into account the dynamics of the underlying experimental

process. We claim that, leveraging this information, it is possible to accurately infer local approximations of the fitness landscape and to provide better structural predictions with respect to related unsupervised models that do not include any dynamical contribution. In particular, we encode the functional properties of protein sequences into a GPM energy function for both the AMaLa and betaDCA models, so that we consider the fitness landscape to be fairly time independent.

For the AMaLa method discussed in Ch. 3, we tested the meaningfulness of the inferred landscape for both prediction of mutational effects and contacts among residues. The functional model energy has been inferred on directed evolution data of [44, 160]. When inferred on [44] data, the selective energy turned out to be highly correlated with independent fitness measurements of antibiotic resistance [51, 81] provided by the TEM-1 β-lactamase protein. Concerning the data of [160], for which such independent fitness measurements are missing, we employed the coupling parameters of the model energy to assess contact prediction for the two studied proteins PSE-1 and AAC6. For the former, the performances are comparable to the standard equilibrium-DCA approach, whereas AMaLa is able to provide a significantly improved outcome for AAC6. By investigating the method performances through in-silico simulation, we conclude that this outcome is compatible with the hypothesis and approximations at the foundation of the method, which define a specific optimal experimental regime for AMaLa application. Such regime is characterized by a high mutation rate and a relatively low selective pressure, or alternatively, the realization of a small number of selective rounds.

The betaDCA method described in Ch. 4 shares some similarities with AMaLa, and specifically for modeling selection as an annealing process defined by a statistical temperature. However, the generality of the statistical model allowed to apply betaDCA to a wider variety of experimental settings, such as deep mutational scanning and antibody repertoire sequencing data, on top of directed evolution experiments as well. The method proved to be particularly effective for datasets characterized by severe undersampling and noisy regimes, as it is the case for antibody repertoire sequences. Indeed, the method displayed excellent discrimination performances between antibodies belonging either to the naïve or to the immunized repertoire [88, 66]. For the considered deep mutational scanning data [18, 175, 56], which provide accurate abundance information, betaDCA displayed worse performance when compared to an alternative method [47] that is able to leverage such population dynamics. Finally, even though the application of betaDCA to directed evolution data is to be considered heuristics, for the model does not explicitly account for the mutation process, it shows very good performances with respect to the contact prediction problem for the data of [160], whereas it falls short of the AMaLa method for the prediction of mutational effects on TEM-1.

The peculiar feature of both the AMaLa and betaDCA methods, is that they can be used in cases where accurate abundance measurements are not accessible, for the statistical modeling does not rely directly on the knowledge of the variants population. If for AMaLa this feature was a necessary requirement, since both directed evolution experiments [44, 160] are realized in a severe undersampling regime, betaDCA was meant to

be applied to experimental setups for which such abundance measurements might not be available for a variety of reasons: very noisy experimental processes, undersampling of the variants population, introduction of mutations alongside selection.

When compared to alternative machine learning approaches, and specifically to deep learning architectures, the advantage of energy based approaches such as the GPM is that their constituent parameters have a direct biological interpretation, as it happens e.g. for the coupling parameters, allowing to take into account epistatic effects and to assess contact prediction by estimating the direct interaction between protein residues.

Among the possible machine learning architectures that recently gathered attention in the protein sequence data community, we can mention the RBM [167, 166, 21, 20, 147], both trained in an unsupervised or semi-supervised manner. In this thesis, we try to answer to the methodological question of whether it was possible to train RBM by employing the EP algorithm. The basic idea, is that EP can be used to approximate the joint model distribution over the visible and hidden units $P(\mathbf{v}, \mathbf{h})$, which is necessary to compute the ensemble averages appearing in the likelihood gradient expressions (see Eqs. (5.47) and (5.11)). We applied this strategy to infer an RBM architecture over the MNIST dataset of handwritten digits. In doing so, we compared the estimates of the model probability statistics provided by both contrastive approaches and naive Monte Carlo sampling methods with the one yielded by EP. The latter is generally better than both Monte Carlo based approaches, when these are characterized by a finite number of sampling steps $k \lesssim 100$. Moreover, we found out an interesting, and to our knowledge unprecedented behavior of the EP algorithm: as the model probability $P(\mathbf{v}, \mathbf{h})$ becomes multimodal along the learning process of the weights, so EP begins to converge separately to the different modes, providing for each one a multivariate Gaussian approximation. This interesting feature generates a slowing down in the convergence of EP parameters, which is the main reason limiting the application of the method, especially in the final stages of the learning when the probability landscape becomes very complex, and several modes appear. Thus, future effort will be devoted to attempt to cure this slowing down, and to study the relation between parameters initialization and the specific reached attractor. Furthermore, we would like to be able to apply such EP-based inference strategy also to protein sequence data.

Appendix A

Elements of Bayesian Inference

Bayes theorem

In this appendix we discuss the probabilistic formalism providing the theoretical framework for the inference methods treated throughout the thesis work, i.e. the Bayes theorem. Such theorem is indeed a formula connecting the conditional probabilities of two events A and B:

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)}. \tag{A.1}$$

Eq. (A.1) can be readily applied to an inference setting. Imagine we have collected a dataset of M samples $X = \{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_M\}$ where $\mathbf{x}_m \in \mathbb{R}^N$, which is the outcome of an experimental process whose underlying rules are either not known or inherently random, so that each sample can be considered as a stochastic realization. Moreover, we consider a set of parameters $\boldsymbol{\theta} \in \mathbb{R}^D$ identifying a model of how the data are generated, i.e. the stochastic rule defining the process. In the Bayesian framework, the model itself is taken as stochastic, for an ensemble of possible models are defined by the different values the parameters can assume. In this perspective, the Bayes formula can be rewritten as:

$$P(\boldsymbol{\theta}|X) = \frac{P(X|\boldsymbol{\theta})P(\boldsymbol{\theta})}{P(X)} = \frac{P(X|\boldsymbol{\theta})P(\boldsymbol{\theta})}{\int \mathrm{d}^D\theta\, P(X|\boldsymbol{\theta})P(\boldsymbol{\theta})}. \tag{A.2}$$

Let's break down the different terms of Eq. (A.2):

- $P(\boldsymbol{\theta}|X)$ is the so called *posterior* probability. It tells us how probable it is for the parameters to assume a specific set of values given the observation made about the data sample X.

- $P(X|\boldsymbol{\theta})$ is the *likelihood* function, embodying the hypothesis related to the experimental mechanism, i.e. how the data are generated given the model parameters $\boldsymbol{\theta}$.
- $P(\boldsymbol{\theta})$ is the *prior* probability, containing our knowledge about how the model parameters should be distributed regardless of the experimental measurements.
- $P(X)$ is considered as a normalization factor, indeed in Eq. (A.2) we expressed it as an integral over the model parameters.

Thus, Eq. (A.2) can be viewed as a flux of information: the a priori knowledge of the model parameters, i.e. the prior, is updated by the collection of observations through the likelihood function to define the posterior probability. An interesting feature of Bayesian inference is that it generally does not provide only a point estimate of the parameters, but rather gives information about their whole distribution. However, it can be employed as well to determine a specific set of values $\hat{\boldsymbol{\theta}}$ of the model parameters. To achieve this goal two possible strategies can be pursued:

- **Maximum a posteriori** (MAP): the model parameters are estimated according to $\hat{\boldsymbol{\theta}} = \text{argmax}_{\boldsymbol{\theta}}[P(\boldsymbol{\theta}|X)] = \text{argmax}_{\boldsymbol{\theta}}[P(X|\boldsymbol{\theta})P(\boldsymbol{\theta})]$, that is, the estimate of the model parameters is provided by the ones maximizing the posterior probability.
- **Maximum likelihood** (ML): the optimal set of parameters is determined via $\hat{\boldsymbol{\theta}} = \text{argmax}_{\boldsymbol{\theta}}[P(X|\boldsymbol{\theta})]$, i.e. through maximization of the likelihood function. This strategy amounts to consider $P(\boldsymbol{\theta}|X) \propto P(X|\boldsymbol{\theta})$ or in other words, as if the parameters prior were uniform. For continuous random variables this implies that $P(\boldsymbol{\theta})$ is actually a pseudo-prior, since it is not properly normalizable.

Though the MAP approach is formally more correct, for the assumption of a uniform prior is not always adequate, there exist some scenarios in which the likelihood function represents the dominant contribution with respect to the prior. This is actually the case in which one has access to a sample of diverging size, i.e. $M \to \infty$. For the sake of simplicity, we can consider the case in which we have M i.i.d. samples, so that the likelihood function can be rewritten as:

$$P(X|\boldsymbol{\theta}) = P(\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_M|\boldsymbol{\theta}) = \prod_{m=1}^{M} P(\mathbf{x}_m|\boldsymbol{\theta}). \tag{A.3}$$

Thus, neglecting the normalization contribution, the posterior can be expressed as:

$$\begin{aligned} P(\boldsymbol{\theta}|X) = P(\boldsymbol{\theta}) \prod_{m=1}^{M} P(\mathbf{x}_m|\boldsymbol{\theta}) &= \exp\left[M \frac{1}{M} \sum_{m=1}^{M} \log P(\mathbf{x}_m|\boldsymbol{\theta}) + \log P(\boldsymbol{\theta}) \right] \\ &= \exp\left\{ M \left[\overline{\log P(\mathbf{x}|\boldsymbol{\theta})} + \frac{1}{M} \log P(\boldsymbol{\theta}) \right] \right\} \approx \exp\left[M \overline{\log P(\mathbf{x}|\boldsymbol{\theta})} \right], \; M \to \infty. \end{aligned} \tag{A.4}$$

Consequently, if $\log P(\boldsymbol{\theta})$ is a smooth function of the parameters its contribution becomes negligible as $M \to \infty$. Moreover, thanks to the law of large number, the empirical average of the log-likelihood converges in probability to the ensemble average given by the *true* set of parameters $\bar{\boldsymbol{\theta}}$:

$$\overline{\log P(\mathbf{x}|\boldsymbol{\theta})} \simeq \int \mathrm{d}^N x \, P(\mathbf{x}|\bar{\boldsymbol{\theta}}) \log P(\mathbf{x}|\boldsymbol{\theta}). \tag{A.5}$$

Eq. (A.5) is the minus *cross-entropy* $S_c\left(\boldsymbol{\theta}, \bar{\boldsymbol{\theta}}\right)$ between $P(\mathbf{x}|\bar{\boldsymbol{\theta}})$ and $P(\mathbf{x}|\boldsymbol{\theta})$, which can be expressed as:

$$S_c\left(\boldsymbol{\theta}, \bar{\boldsymbol{\theta}}\right) = S\left(\bar{\boldsymbol{\theta}}\right) + D_{KL}\left(\bar{\boldsymbol{\theta}} \| \boldsymbol{\theta}\right), \tag{A.6}$$

where $S\left(\bar{\boldsymbol{\theta}}\right) = -\int \mathrm{d}^N x \, P(\mathbf{x}|\bar{\boldsymbol{\theta}}) \log P(\mathbf{x}|\bar{\boldsymbol{\theta}})$, is the entropy of the true parameters likelihood, whereas $D_{KL}\left(\bar{\boldsymbol{\theta}} \| \boldsymbol{\theta}\right) = \int \mathrm{d}^N x \, P(\mathbf{x}|\bar{\boldsymbol{\theta}}) \log \frac{P(\mathbf{x}|\bar{\boldsymbol{\theta}})}{P(\mathbf{x}|\boldsymbol{\theta})}$ is the Kullback-Leibler (KL) divergence between the likelihoods. Since $D_{KL}\left(\bar{\boldsymbol{\theta}} \| \boldsymbol{\theta}\right) \geq 0$, the cross-entropy is bounded from below by the entropy of the likelihood of the true model parameters, and it reaches a minimum for $\boldsymbol{\theta} = \bar{\boldsymbol{\theta}}$, when the KL divergence is zero.

In light of these computations, the posterior can be expressed as:

$$\begin{aligned} P(\boldsymbol{\theta}|X) &= \frac{\mathrm{e}^{-M S_c(\boldsymbol{\theta}, \bar{\boldsymbol{\theta}})}}{\int \mathrm{d}^D \theta \, \mathrm{e}^{-M S_c(\boldsymbol{\theta}, \bar{\boldsymbol{\theta}})}} \\ &\simeq \mathrm{e}^{-M\left[S_c(\boldsymbol{\theta}, \bar{\boldsymbol{\theta}}) - S(\bar{\boldsymbol{\theta}})\right]} \\ &= \mathrm{e}^{-M D_{KL}(\bar{\boldsymbol{\theta}}, \boldsymbol{\theta})}, \end{aligned} \tag{A.7}$$

where in the second passage we approximated the integral over $\boldsymbol{\theta}$ with a saddle point computation, i.e. the integral is substituted by its maximum, which is obtained for $S_c(\bar{\boldsymbol{\theta}}, \boldsymbol{\theta}) = S\left(\bar{\boldsymbol{\theta}}\right)$. Eq. (A.7) tells us that the posterior probability can be asymptotically approximated in a large deviation fashion, and any set of parameters $\hat{\boldsymbol{\theta}} \neq \bar{\boldsymbol{\theta}}$ is exponentially suppressed with the number of observations M, with a rate which is controlled by the KL divergence.

Appendix B

Pseudo-likelihood computations

The pseudo-likelihood (PSL) approximation represents the inference strategy we adopted both in Chs. 3 and 4. In this appendix we analyze some details behind PSL computations. It is for instance interesting to compute the derivative of the objective function in Eq. (3.12) with respect to a $\boldsymbol{\beta}$ component $\beta(t_k) \equiv \beta_k$:

$$\begin{aligned}\frac{\partial g_r(\mathbf{h}_r, \mathbf{J}_r, \boldsymbol{\beta})}{\partial \beta_k} &= -\sum_{m=1}^{M^{(t_k)}} w^{(m,t_k)} \Bigg\{ h_r(\sigma_r^{(m,t_k)}) + \sum_{i \neq r} J_{ri}(\sigma_r^{(m,t_k)}, \sigma_i^{(m,t_k)}) \\ &\quad - \frac{\sum_{a=1}^{q} \left[h_r(a) + \sum_{i \neq r} J_{ri}(a, \sigma_i^{(m,t_k)}) \right] \mathrm{e}^{\beta \left[h_r(a) + \sum_{i \neq r} J_{ri}(a, \sigma_i^{(m,t_k)}) \right] + \nu(t) \delta(a, \sigma_r^{wt})}}{Z_r} \Bigg\} \\ &= \overline{\left\langle E_r(\mathbf{S}^{(m,t_k)}) \right\rangle - E_r(\mathbf{S}^{(m,t_k)})}, \end{aligned} \tag{B.1}$$

which is a function of the difference between the ensemble energy and the energy of the sequence $\mathbf{S}^{(m,t_k)}$ at time t_k, averaged over all the sequences observed at that time. It is then possible to determine the optimal $\boldsymbol{\beta}$ performing a Newton gradient descent (NGD) according to (B.1). NGD optimization employs also the second derivate of the objective function, and has the advantage to be a learning rate free algorithm, for the update equation for our case of interest becomes:

$$\beta_k^{(t+1)} = \beta_k^{(t)} - \frac{\partial g_r / \partial \beta_k}{\partial^2 g_r / \partial \beta_k^2}, \tag{B.2}$$

where $\beta_k^{(t)}$ is the value of the k-th $\boldsymbol{\beta}$ component at iteration t of the optimization algorithm. Eq. (B.2) can be interpreted as if we were using a non-constant learning rate

$\lambda = \frac{1}{\partial^2 g_r / \partial \beta_k^2}$. The subtlety of this strategy is that the objective (3.12) is not contemporarily convex with respect to $\{\mathbf{h}_r, \mathbf{J}_r\}$ and $\boldsymbol{\beta}$, so that if one is to optimze with respect to both set of parameters contemporarily, convergence to a global extremum is not guaranteed. To work around this issue, we decided to adopt the following strategy: we let the energetic parameters converge at fixed $\boldsymbol{\beta}$ at each iteration, performing afterwards a finite number of updates of the fictitious inverse temperature at fixed $\{\mathbf{h}_r, \mathbf{J}_r\}$. The algorithm stops when the variation of the β_k's goes under a chosen tolerance threshold.

In Sec. 2.2.1, we discussed the gauge invariance property of the GPM, and in Sec. 2.3.3 we mentioned how the PSL approximation allows to impose automatically a gauge at the end of the learning process. To be more precise, the gauge is imposed by the presence of the regularization term (see Sec. 2.3.6), which for our case of interest is an l_2 contribution. In the following, we perform the computation for the asymmetric PSL approach applied to the objective function of Eq. (3.12). Firstly we need to rewrite Eqs. (3.13) and (3.14) as:

$$\frac{\partial g_r(\mathbf{h}_r, \mathbf{J}_r, \boldsymbol{\beta}, \boldsymbol{\nu})}{\partial h_r(c)} = -\sum_{t=\{t_1,\dots,t_f\}} \sum_{m=1}^{M^{(t)}} w^{(m,t)} \beta(t) \left[\delta(c, \sigma_r^{(m,t)}) - P(\sigma_r = c | \boldsymbol{\sigma}_{\setminus r} = \boldsymbol{\sigma}_{\setminus r}^{(m,t)})\right] + 2\lambda_h h_r(c) = 0, \tag{B.3}$$

$$\frac{\partial g_r(\mathbf{h}_r, \mathbf{J}_r, \boldsymbol{\beta}, \boldsymbol{\nu})}{\partial J_{rj}(c,d)} = -\sum_{t=\{t_1,\dots,t_f\}} \sum_{m=1}^{M^{(t)}} w^{(m,t)} \beta(t) \delta(d, \sigma_r^{(m,t)}) \left\{\delta(c, \sigma_j^{(m,t)}) - P(\sigma_r = c | \boldsymbol{\sigma}_{\setminus r} = \boldsymbol{\sigma}_{\setminus r}^{(m,t)})\right\} + 2\lambda_J J_{rj}(c,d) = 0. \tag{B.4}$$

The equality to zero holds at convergence of the algorithm. Summing Eq. (B.3) with respect to all possible amino acids, one gets the gauge condition over the fields:

$$\sum_{c=1}^{q} h_r(c) = 0, \tag{B.5}$$

since for every $t \in \{t_1, \dots, t_f\}$ and $m \in \{1, \dots, M^{(t)}\}$ it holds $\sum_{c=1}^{q} \delta(c, \sigma_r^{(m,t)}) = \sum_{c=1}^{q} P(\sigma_r = c | \boldsymbol{\sigma}_{\setminus r} = \boldsymbol{\sigma}_{\setminus r}^{(m,t)}) = 1$. Moreover, Eq. (B.3) can be used to express the fields at convergence:

$$h_r(c) = \frac{1}{2\lambda_h} \sum_{t=\{t_1,\dots,t_f\}} \sum_{m=1}^{M^{(t)}} w^{(m,t)} \beta(t) \left[\delta(c, \sigma_r^{(m,t)}) - P(\sigma_r = c | \boldsymbol{\sigma}_{\setminus r} = \boldsymbol{\sigma}_{\setminus r}^{(m,t)})\right]. \tag{B.6}$$

The gauge conditions for the couplings are respectively obtained by summing Eq. (B.4) with respect to the first and the second amino acid entry:

$$\begin{cases} \sum_{c=1}^{q} J_{ri}(c,d) = 0, \\ \sum_{d=1}^{q} J_{ri}(c,d) = \frac{\lambda_h}{\lambda_J} h_r(c). \end{cases} \tag{B.7}$$

where in the second equality we employed Eq. (B.6) to express the result as a function of the fields. These gauge conditions have been verified for the asymmetric PSL approach both on real and synthetic data. One can also readily derive the gauge conditions for the symmetrized coupling parameters, obtained when combining J_{ri} and J_{ir}. Indeed, one has:

$$\begin{cases} \sum_{a=1}^{q} J_{ij}(a,b) = \frac{1}{2}\left[\sum_{a=1}^{q} J_{ij}^{i}(a,b) + \sum_{a=1}^{q} J_{ji}^{j}(b,a)\right] = \frac{\lambda_h}{2\lambda_J} h_j(b), \\ \sum_{b=1}^{q} J_{ij}(a,b) = \frac{1}{2}\left[\sum_{b=1}^{q} J_{ij}^{i}(a,b) + \sum_{b=1}^{q} J_{ji}^{j}(b,a)\right] = \frac{\lambda_h}{2\lambda_J} h_i(a), \end{cases} \tag{B.8}$$

in which we added the superscript J_{rk}^{r} to specify that such coupling parameter was determined by optimization of the objective g_r (Eq. (3.12)), related to site r.

Appendix C

Jukes Cantor mutational model

In this appendix we go deeper into the Jukes-Cantor (JC) model of neutral evolution, i.e. in which the dynamics in sequence space is solely determined by the mutation process. In Sec. 3.2.2 we treated the continuous time version of the JC model. Here, we would like to give some additional results about the continuous time formalism, highlighting how it can be related to the discrete time dynamics.

We already stated that the fundamental quantities defining the JC model are, for the continuous time case, the number of symbols q and the single site mutation rate μ, i.e. how many mutations are expected to take place in the time unit. For the discrete time case, the equivalent of the mutation rate is the single site mutation probability p, i.e. the probability for a single site to be mutated over a round. Indeed, rather than writing down a master equation as in Eq. (3.6), the stochastic dynamics will be modeled as a Markov chain. The state of the chain is probabilistically described by a $q \times L$ matrix P^t, such that P^t_{ai} is the probability that at time t site i is found in amino acid a. At $t = 0$, all sequences must coincide with the wild-type, so that the matrix elements read $P^0_{ai} = \delta(a, \sigma_i^{wt})$. For a Markov chain process, the relation between the probabilities at neighboring times is defined as:

$$P^{t+1} = W P^t, \tag{C.1}$$

where W is the $q \times q$ transition matrix, that for the discrete JC model reads:

$$W = \begin{pmatrix} 1-p & \frac{p}{q-1} & \frac{p}{q-1} & \dots & \frac{p}{q-1} \\ \frac{p}{q-1} & 1-p & \frac{p}{q-1} & \dots & \frac{p}{q-1} \\ \vdots & & \ddots & & \vdots \\ \frac{p}{q-1} & \frac{p}{q-1} & \dots & \dots & 1-p \end{pmatrix}. \tag{C.2}$$

Diagonal elements in Eq. (C.2) account for the probability of a site not to mutate,

whereas out of diagonal elements account for the probability that a site does mutate over the cycle. Such event has to be normalized to $q-1$, which is the number of available symbols. Interestingly, the W matrix is doubly stochastic, i.e. the sum over both rows and columns is equal to 1. This automatically guarantees that the asymptotic distribution exists and is uniform, so that $P^{\infty}_{ai} = 1/q$ for every $a = 1, \dots, q$ and $i = 1, \dots, L$. Since the JC model is site-independent, the probability of a specific sequence $\mathbf{S} = (\sigma_1, \sigma_2, \dots, \sigma_L)$ (where we consider $\sigma \in \{1, \dots, q\}$) is obtained by multiplying the proper P^t elements: $P^{(t)}(\mathbf{S}) = \prod_{i=1}^{L} P^t_{\sigma_i, i}$. Consequently, the asymptotic probability becomes $P^{(\infty)}(\mathbf{S}) = 1/q^L$ for any sequence, as expected.

Eq. (C.1) can also be expressed as $P^t = (W)^t P^0$, where $(W)^t$ is the t-th power of the transition matrix. In this way, it is possible to obtain the probability of a sequence to be observed at time t as an expansion in powers of p, as the single site mutation probability is supposed to be generally small $p \ll 1$. In particular, we get a different expansion depending on the Hamming distance between the considered sequence and the wild-type. Namely:

$$\begin{cases} P^{(t)}\left(\mathbf{S} | \mathrm{h_D}(\mathbf{S}, \mathbf{S}^{wt}) = 0\right) = 1 - (tL)p + C_2(t, L, q)p^2 + O(p^3), \\ P^{(t)}\left(\mathbf{S} | \mathrm{h_D}(\mathbf{S}, \mathbf{S}^{wt}) = d\right) = \left(\frac{pt}{q-1}\right)^d + O(p^{d+1}). \end{cases} \tag{C.3}$$

where $C_2(t, L, q) = \frac{tL}{2(q-1)} \{t[(q-1)L - t] + t(t+1) - q\}$. Even if p is small, we notice that the first of Eq. (C.3) breaks down as soon as $tL \sim p$. However, the second equation might suggest an ansatz for the time dependent probability: $P^{(t)}\left(\mathbf{S} | \mathrm{h_D}(\mathbf{S}, \mathbf{S}^{wt}) = d\right) \propto \exp\left\{-d \ln\left(\frac{q-1}{pt}\right)\right\}$, which closely resembles Eq. (3.9). From this computation, the advantage of the continuous time formalism should emerge, as it is able to automatically re-sum all the perturbative contributions in Eq. (C.3).

At this point, it is worth deriving the relation between p and μ. In particular, from the second of Eq. (3.8) it follows:

$$p(\mu, \tau) = \frac{q-1}{q}\left(1 - \mathrm{e}^{-\mu\tau}\right) = \frac{q-1}{q}\mu\tau + \mathcal{O}(\mu\tau)^2, \tag{C.4}$$

expressing p as a function of μ and a time interval τ. Eq. (C.4) states how we could switch from the continuous to the discrete time formalism, interpreting the interval of duration τ as a single cycle of mutagenesis.

In the following, we will go back to the continuous time formalism with the aim to survey some further features of the purely mutational process. We recall that the probability of observing a sequence at time t given a certain mutation rate μ reads:

$$P^{(t)}(\mathbf{S}) = \frac{\mathrm{e}^{-\nu(t)\mathrm{h_D}(\mathbf{S}, \mathbf{S}^{wt})}}{Z^{(t)}}, \tag{C.5}$$

where $Z^{(t)}$ is the normalization factor which can be explicitly computed as:

$$
\begin{aligned}
Z^{(t)} &= \sum_{\{\mathbf{S}\}} \mathrm{e}^{-\nu(t)\mathrm{h_D}(\mathbf{S},\mathbf{S}^{wt})} \\
&= \sum_{d=0}^{L} \binom{L}{d} (q-1)^d \mathrm{e}^{-\nu d} \\
&= \mathrm{e}^{-\nu L}\left[(q-1)+\mathrm{e}^{\nu}\right]^L \\
&= \left[\frac{q}{1+(q-1)\mathrm{e}^{-\mu t}}\right]^L ,
\end{aligned}
\tag{C.6}
$$

where at the second line we changed the summing variables from the sequence space to the possible distances from 0 to L (with the proper Jacobian), and we pointed out two possible expressions of $Z^{(t)}$. Eq. (C.5) can be readily employed to compute the moments of the Hamming distance for the neutral evolution process:

$$
\langle d(t)\rangle = \frac{1}{Z^{(t)}} \sum_{d=0}^{L} d\binom{L}{d}(q-1)^d \mathrm{e}^{-\nu(t)d} = \frac{(q-1)L}{q-1+\mathrm{e}^{\nu(t)}},
\tag{C.7}
$$

$$
\left\langle d^2(t)\right\rangle - \langle d(t)\rangle^2 = \frac{(q-1)L\mathrm{e}^{\nu t}}{\left[q-1+\mathrm{e}^{\nu(t)}\right]^2}.
\tag{C.8}
$$

These observables can be used as a benchmark to verify how much an evolution process which includes selection deviates from a neutral one.

Appendix D

Further results on DE experiments

In this appendix we report further results obtained on DE experiments. Firstly, we study how the performances of the standard DCA approach change when all the sequenced rounds of the experiment are taken into account contemporarily, as opposite to considering only the last one. Generally, we find a worsening of the performances, both with respect to the prediction of mutational effects and residue-residue contacts. In Fig. D.1 we show the analogous of Fig. 3.2, i.e. the correlation between the model energies and the independent fitness measurement of TEM-1, for two DCA models: (i) one is inferred over the last round only; (ii) the other uses sequences from all sequenced rounds.

In Fig. D.2, we show the results obtained from [160] data for the contact prediction problem of proteins PSE1 and AAC6. Specifically, we report the sensitivity plots analogous to Fig. 3.5 in the main text, with the addition of the DCA model inferred on all sequenced rounds. For both the considered proteins, such a strategy yields a worsening with respect to when DCA is inferred only on the last available round.

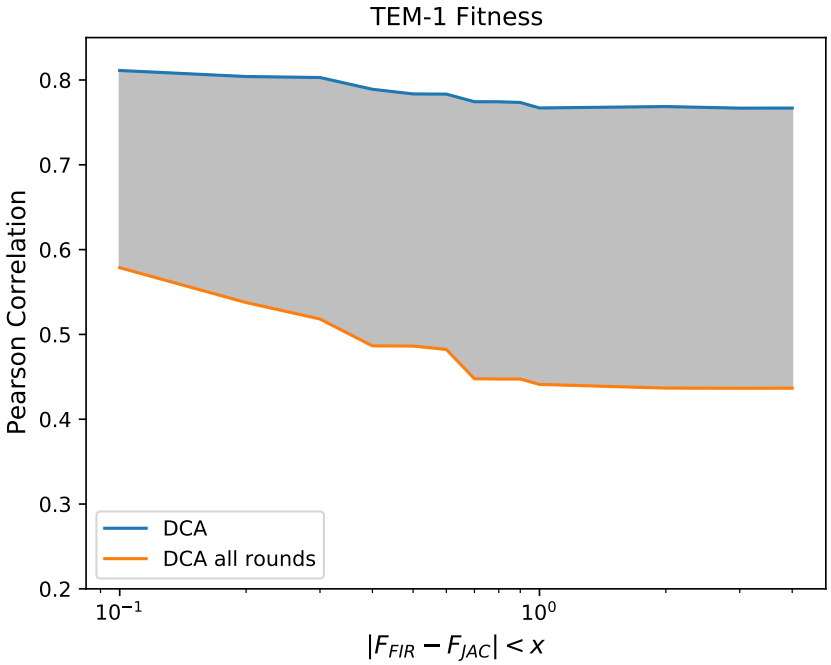
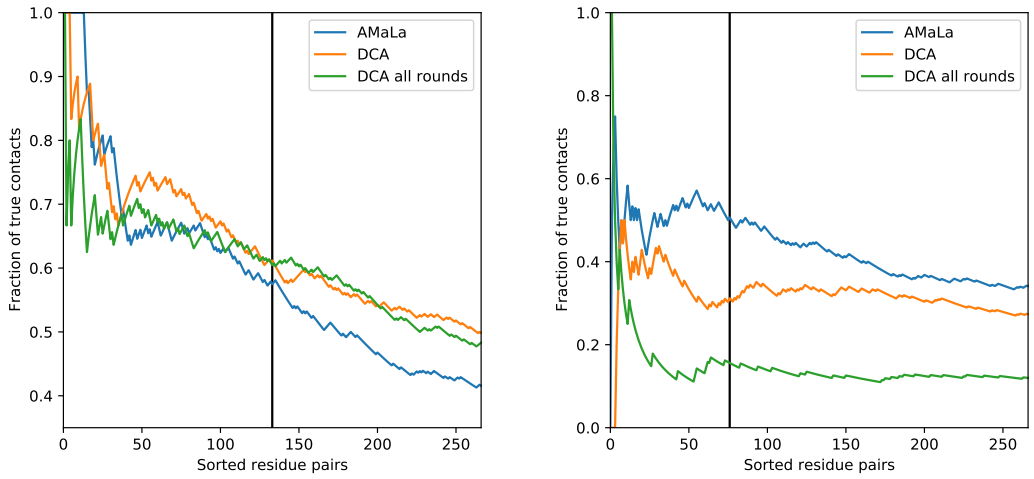


Figure D.1: Comparison between two DCA models: one is inferred only on the last round of [44] data, the other is inferred from a global alignment containing the sequences produced from all the sequenced rounds. Strikingly, even if the second model is inferred over a larger number of sequences, its performances in terms of prediction of mutational effects are considerably worse than when DCA is determined over the last round only.



(a) (b)

Figure D.2: Sensitivity plots for the contact prediction of PSE1 (panel (a)) and AAC6 (panel (b)) proteins, obtained from the AMaLa and DCA methods applied to the data of [160]. For the DCA approach, we both performed the inference on the last and on all the available sequenced rounds.

Bibliography

[1] Christopher D Aakre et al. "Evolving new protein-protein interaction specificity through promiscuous intermediates." In: *Cell* 163.3 (2015), pp. 594–606.

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

[3] Fedaa Ali, Amal Kasry, and Muhamed Amin. "The new SARS-CoV-2 strain shows a stronger binding affinity to ACE2 due to N501Y mutant." In: *Medicine in drug discovery* 10 (2021), p. 100086.

[4] Ethan C Alley et al. "Unified rational protein engineering with sequence-based deep representation learning." In: *Nature methods* 16.12 (2019), pp. 1315–1322.

[5] Carlos L Araya and Douglas M Fowler. "Deep mutational scanning: assessing protein function on a massive scale." In: *Trends in biotechnology* 29.9 (2011), pp. 435–442.

[6] Carlos L Araya et al. "A fundamental protein property, thermodynamic stability, revealed solely from large-scale measurements of protein function." In: *Proceedings of the National Academy of Sciences* 109.42 (2012), pp. 16858–16863.

[7] Lorenzo Asti et al. "Maximum-entropy models of sequenced immune repertoires predict antigen-antibody affinity." In: *PLoS computational biology* 12.4 (2016), e1004870.

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

[9] Carlo Baldassi et al. "Fast and accurate multivariate Gaussian modeling of protein families: predicting residue contacts and protein-interaction partners." In: *PloS one* 9.3 (2014), e92721.

[10] Pierre Barrat-Charlaix, Matteo Figliuzzi, and Martin Weigt. "Improving landscape inference by integrating heterogeneous data in the inverse Ising problem." In: *Scientific Reports* 6.1 (2016), pp. 1–9.

[11] Pierre Barrat-Charlaix et al. "Sparse generative modeling via parameter reduction of Boltzmann machines: application to protein-sequence families." In: *Physical Review E* 104.2 (2021), p. 024407.

[12] John P Barton et al. "ACE: adaptive cluster expansion for maximum entropy graphical model inference." In: *Bioinformatics* 32.20 (2016), pp. 3089–3097.

[13] Jennifer Benichou et al. "Rep-Seq: uncovering the immunological repertoire through next-generation sequencing." In: *Immunology* 135.3 (2012), pp. 183–191.

[14] William Bialek et al. "Statistical mechanics for natural flocks of birds." In: *Proceedings of the National Academy of Sciences* 109.13 (2012), pp. 4786–4791.

[15] Matteo Bisardi et al. "Modeling sequence-space exploration and emergence of epistatic signals in protein evolution." In: *Molecular biology and evolution* 39.1 (2022), msab321.

[16] Anne-Florence Bitbol et al. "Inferring interaction partners from protein sequences." In: *Proceedings of the National Academy of Sciences* 113.43 (2016), pp. 12180–12185.

[17] Jesse D. Bloom et al. "Protein stability promotes evolvability." In: *Proceedings of the National Academy of Sciences* 103.15 (2006), pp. 5869–5874. DOI: 10.1073/pnas.0510098103. eprint: https://www.pnas.org/content/103/15/5869.full.pdf. URL: https://www.pnas.org/content/103/15/5869.

[18] Sébastien Boyer et al. "Hierarchy and extremes in selections from pools of randomized proteins." In: *Proceedings of the National Academy of Sciences* 113.13 (2016), pp. 3482–3487. ISSN: 0027-8424. DOI: 10.1073/pnas.1517813113. eprint: https://www.pnas.org/content/113/13/3482.full.pdf. URL: https://www.pnas.org/content/113/13/3482.

[19] Barbara Bravi et al. "Learning the differences: a transfer-learning approach to predict antigen immunogenicity and T-cell receptor specificity." In: *bioRxiv* (2022), pp. 2022–12.

[20] Barbara Bravi et al. "Probing T-cell response by sequence-based probabilistic modeling." In: *PLoS Computational Biology* 17.9 (2021), e1009297.

[21] Barbara Bravi et al. "RBM-MHC: a semi-supervised machine-learning method for sample-specific prediction of antigen presentation by HLA-I alleles." In: *Cell systems* 12.2 (2021), pp. 195–202.

[22] Lukas Burger and Erik Van Nimwegen. "Disentangling direct from indirect co-evolution of residues in protein alignments." In: *PLoS computational biology* 6.1 (2010), e1000633.

[23] Thomas C Butler et al. "Identification of drug resistance mutations in HIV from constraints on natural evolution." In: *Physical Review E* 93.2 (2016), p. 022412.

[24] Frédéric Cadet et al. "A machine learning approach for reliable prediction of amino acid interactions and its application in the directed evolution of enantioselective enzymes." In: *Scientific reports* 8.1 (2018), pp. 1–15.

[25] Miguel A Carreira-Perpinan and Geoffrey Hinton. "On contrastive divergence learning." In: *International workshop on artificial intelligence and statistics*. PMLR. 2005, pp. 33–40.

[26] Ryan R Cheng et al. "Connecting the sequence-space of bacterial signaling proteins to phenotypes using coevolutionary landscapes." In: *Molecular biology and evolution* 33.12 (2016), pp. 3054–3064.

[27] KyungHyun Cho, Tapani Raiko, and Alexander Ilin. "Parallel tempering is efficient for learning restricted Boltzmann machines." In: *The 2010 international joint conference on neural networks (ijcnn)*. IEEE. 2010, pp. 1–8.

[28] Patrick C Cirino, Kimberly M Mayer, and Daisuke Umeno. "Generating mutant libraries using error-prone PCR." In: *Directed evolution library creation*. Springer, 2003, pp. 3–9.

[29] Simona Cocco and Rémi Monasson. "Adaptive cluster expansion for the inverse Ising problem: convergence, algorithm and tests." In: *Journal of Statistical Physics* 147.2 (2012), pp. 252–314.

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

[31] Charles Darwin. *On the origin of species, 1859*. Routledge, 2004.

[32] Nicholas G Davies et al. "Estimated transmissibility and impact of SARS-CoV-2 lineage B. 1.1. 7 in England." In: *Science* 372.6538 (2021), eabg3055.

[33] Aurélien Decelle, Cyril Furtlehner, and Beatriz Seoane. "Equilibrium and non-equilibrium regimes in the learning of restricted Boltzmann machines." In: *Advances in Neural Information Processing Systems* 34 (2021), pp. 5345–5359.

[34] Zhifeng Deng et al. "Deep sequencing of systematic combinatorial libraries reveals β-lactamase sequence constraints at high resolution." In: *Journal of molecular biology* 424.3-4 (2012), pp. 150–167.

[35] Guillaume Desjardins et al. "Tempered Markov chain Monte Carlo for training of restricted Boltzmann machines." In: *Proceedings of the thirteenth international conference on artificial intelligence and statistics*. JMLR Workshop and Conference Proceedings. 2010, pp. 145–152.

[36] Andrea Di Gioacchino et al. "Generative and interpretable machine learning for aptamer design and analysis of in vitro sequence selection." In: *bioRxiv* (2022).

[37] Javier M Di Noia and Michael S Neuberger. "Molecular mechanisms of antibody somatic hypermutation." In: *Annu. Rev. Biochem.* 76 (2007), pp. 1–22.

[38] Michael B Doud and Jesse D Bloom. "Accurate measurement of the effects of all amino-acid mutations on influenza hemagglutinin." In: *Viruses* 8.6 (2016), p. 155.

[39] Stanley D Dunn, Lindi M Wahl, and Gregory B Gloor. "Mutual information without the influence of phylogeny or entropy dramatically improves residue contact prediction." In: *Bioinformatics* 24.3 (2008), pp. 333–340.

[40] Richard Durbin et al. *Biological sequence analysis: probabilistic models of proteins and nucleic acids*. Cambridge university press, 1998.

[41] Sean R. Eddy. "Profile hidden Markov models." In: *Bioinformatics (Oxford, England)* 14.9 (1998), pp. 755–763.

[42] Magnus Ekeberg, Tuomo Hartonen, and Erik Aurell. "Fast pseudolikelihood maximization for direct-coupling analysis of protein structure from many homologous amino-acid sequences." In: *Journal of Computational Physics* 276 (2014), pp. 341–356.

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

[44] Marco Fantini et al. "Protein Structural Information and Evolutionary Landscape by In Vitro Evolution." In: *Molecular Biology and Evolution* 37.4 (Oct. 2019), pp. 1179–1192. ISSN: 0737-4038. DOI: 10.1093/molbev/msz256. eprint: https://academic.oup.com/mbe/article-pdf/37/4/1179/32960043/msz256.pdf. URL: https://doi.org/10.1093/molbev/msz256.

[45] Christoph Feinauer and Martin Weigt. "Context-aware prediction of pathogenicity of missense mutations involved in human disease." In: *arXiv preprint arXiv:1701.07246* (2017).

[46] Christoph Feinauer et al. "Inter-protein sequence co-evolution predicts known physical interactions in bacterial ribosomes and the Trp operon." In: *PloS one* 11.2 (2016), e0149166.

[47] Jorge Fernandez-de-Cossio-Diaz, Guido Uguzzoni, and Andrea Pagnani. "Unsupervised Inference of Protein Fitness Landscape from Deep Mutational Scan." In: *Molecular Biology and Evolution* (Aug. 2020). msaa204. ISSN: 0737-4038. DOI: 10.1093/molbev/msaa204. eprint: https://academic.oup.com/mbe/advance-article-pdf/doi/10.1093/molbev/msaa204/33862547/msaa204.pdf. URL: https://doi.org/10.1093/molbev/msaa204.

[48] Matteo Figliuzzi, Pierre Barrat-Charlaix, and Martin Weigt. "How pairwise coevolutionary models capture the collective residue variability in proteins?" In: *Molecular biology and evolution* 35.4 (2018), pp. 1018–1027.

[49] Matteo Figliuzzi et al. "Coevolutionary landscape inference and the context-dependence of mutations in beta-lactamase TEM-1." In: *Molecular biology and evolution* 33.1 (2016), pp. 268–280.

[50] Robert D Finn, Jody Clements, and Sean R Eddy. "HMMER web server: interactive sequence similarity searching." In: *Nucleic acids research* 39.suppl_2 (2011), W29–W37.

[51] Elad Firnberg et al. "A comprehensive, high-resolution map of a gene's fitness landscape." In: *Molecular biology and evolution* 31.6 (2014), pp. 1581–1592.

[52] Asja Fischer and Christian Igel. "Empirical Analysis of the Divergence of Gibbs Sampling Based Learning Algorithms for Restricted Boltzmann Machines." In: *ICANN (3)* 6354 (2010), pp. 208–217.

[53] Anthony A Fodor and Richard W Aldrich. "Influence of conservation on calculations of amino acid covariance in multiple sequence alignments." In: *Proteins: Structure, Function, and Bioinformatics* 56.2 (2004), pp. 211–221.

[54] Douglas M Fowler and Stanley Fields. "Deep mutational scanning: a new style of protein science." In: *Nature methods* 11.8 (2014), pp. 801–807.

[55] Douglas M Fowler, Jason J Stephany, and Stanley Fields. "Measuring the activity of protein variants on a large scale using deep mutational scanning." In: *Nature protocols* 9.9 (2014), pp. 2267–2284.

[56] Douglas M Fowler et al. "High-resolution mapping of protein sequence-function relationships." In: *Nature methods* 7.9 (2010), p. 741.

[57] Steven A Frank and Montgomery Slatkin. "Fisher's fundamental theorem of natural selection." In: *Trends in Ecology & Evolution* 7.3 (1992), pp. 92–95.

[58] Trevor S Frisby and Christopher James Langmead. "Bayesian optimization with evolutionary and structure-based regularization for directed protein evolution." In: *Algorithms for Molecular Biology* 16.1 (2021), pp. 1–15.

[59] Marylou Gabrié, Eric W Tramel, and Florent Krzakala. "Training Restricted Boltzmann Machine via the Thouless-Anderson-Palmer free energy." In: *Advances in neural information processing systems* 28 (2015).

[60] Carlos A Gandarilla-Perez et al. "Combining phylogeny and coevolution improves the inference of interaction partners among paralogous proteins." In: *bioRxiv* (2022).

[61] Wilfredo F Garcia-Beltran et al. "Multiple SARS-CoV-2 variants escape neutralization by vaccine-induced humoral immunity." In: *Cell* 184.9 (2021), pp. 2372–2383.

[62] Molly Gasperini, Lea Starita, and Jay Shendure. "The power of multiplexed functional analysis of genetic variants." In: *Nature protocols* 11.10 (2016), pp. 1782–1787.

[63] Sam Gelman et al. "Neural networks to learn protein sequence–function relationships from deep mutational scanning data." In: *Proceedings of the National Academy of Sciences* 118.48 (2021), e2104878118.

[64] Stuart Geman and Donald Geman. "Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images." In: *IEEE Transactions on pattern analysis and machine intelligence* 6 (1984), pp. 721–741.

[65] Antoine Georges and Jonathan S Yedidia. "How to expand around mean-field theory using high-temperature expansions." In: *Journal of Physics A: Mathematical and General* 24.9 (1991), p. 2173.

[66] Annabelle Gérard et al. "High-throughput single-cell activity-based screening and sequencing of antibodies using droplet microfluidics." In: *Nature biotechnology* 38.6 (2020), pp. 715–721.

[67] Andonis Gerardos, Nicola Dietler, and Anne-Florence Bitbol. "Correlations from structure and phylogeny combine constructively in the inference of protein partners from sequences." In: *PLOS Computational Biology* 18.5 (2022), e1010147.

[68] Thomas Gueudré et al. "Simultaneous identification of specifically interacting paralogs and interprotein contacts by direct coupling analysis." In: *Proceedings of the National Academy of Sciences* 113.43 (2016), pp. 12186–12191.

[69] Allan Haldane et al. "Structural propensities of kinase family proteins from a Potts model of residue co-variation." In: *Protein Science* 25.8 (2016), pp. 1378–1384.

[70] Ulrich HE Hansmann. "Parallel tempering algorithm for conformational studies of biological molecules." In: *Chemical Physics Letters* 281.1-3 (1997), pp. 140–150.

[71] Ryan T Hietpas, Jeffrey D Jensen, and Daniel NA Bolon. "Experimental illumination of a fitness landscape." In: *Proceedings of the National Academy of Sciences* 108.19 (2011), pp. 7896–7901.

[72] Geoffrey E Hinton. "Training products of experts by minimizing contrastive divergence." In: *Neural computation* 14.8 (2002), pp. 1771–1800.

[73] Geoffrey E Hinton, Simon Osindero, and Yee-Whye Teh. "A fast learning algorithm for deep belief nets." In: *Neural computation* 18.7 (2006), pp. 1527–1554.

[74] Markus Hoffmann et al. "SARS-CoV-2 variants B. 1.351 and P. 1 escape from neutralizing antibodies." In: *Cell* 184.9 (2021), pp. 2384–2393.

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

[76] Thomas A Hopf et al. "Sequence co-evolution gives 3D contacts and structures of protein complexes." In: *elife* 3 (2014), e03430.

[77] John J Hopfield. "Neural networks and physical systems with emergent collective computational abilities." In: *Proceedings of the national academy of sciences* 79.8 (1982), pp. 2554–2558.

[78] Edwin Rodríguez Horta et al. "Global multivariate model learning from hierarchically correlated data." In: *Journal of Statistical Mechanics: Theory and Experiment* 2021.7 (2021), p. 073501.

[79] Nobumichi Hozumi and Susumu Tonegawa. "Evidence for somatic rearrangement of immunoglobulin genes coding for variable and constant regions." In: *Proceedings of the National Academy of Sciences* 73.10 (1976), pp. 3628–3632.

[80] John Ingraham et al. "Generative models for graph-based protein design." In: *Advances in neural information processing systems* 32 (2019).

[81] Hervé Jacquier et al. "Capturing the mutational landscape of the beta-lactamase TEM-1." In: *Proceedings of the National Academy of Sciences* 110.32 (2013), pp. 13067–13072.

[82] Edwin T Jaynes. "Information theory and statistical mechanics." In: *Physical review* 106.4 (1957), p. 620.

[83] Steven G Johnson and Julien Schueller. "NLopt: Nonlinear optimization library." In: *Astrophysics Source Code Library* (2021), ascl–2111.

[84] Philippe Julien et al. "The complete local genotype–phenotype landscape for the alternative splicing of a human exon." In: *Nature communications* 7.1 (2016), pp. 1–8.

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

[86] Hetunandan Kamisetty, Sergey Ovchinnikov, and David Baker. "Assessing the utility of coevolution-based residue–residue contact predictions in a sequence- and structure-rich era." In: *Proceedings of the National Academy of Sciences* 110.39 (2013), pp. 15674–15679.

[87] Harry Kemble, Philippe Nghe, and Olivier Tenaillon. "Recent insights into the genotype–phenotype relationship from massively parallel genetic assays." In: *Evolutionary applications* (2019).

[88] Tarik A Khan et al. "Accurate and predictive antibody repertoire profiling by molecular amplification fingerprinting." In: *Science advances* 2.3 (2016), e1501371.

[89] Justin B Kinney and David M McCandlish. "Massively parallel assays and quantitative sequence–function relationships." In: *Annual review of genomics and human genetics* 20 (2019).

[90] Jacob O Kitzman et al. "Massively parallel single-amino-acid mutagenesis." In: *Nature methods* 12.3 (2015), pp. 203–206.

[91] Aleksandr Kovaltsuk et al. "Observed antibody space: a resource for data mining next-generation sequencing of antibody repertoires." In: *The Journal of Immunology* 201.8 (2018), pp. 2502–2509.

[92] Alan Lapedes, Bertrand Giraud, and Christopher Jarzynski. "Using sequence alignments to predict protein structure and stability with high accuracy." In: *arXiv preprint arXiv:1207.2484* (2012).

[93] Brian Lee et al. "Inferring effects of mutations on SARS-CoV-2 transmission from genomic surveillance data." In: *medRxiv* (2022), pp. 2021–12.

[94] Chuan Li et al. "The fitness landscape of a tRNA gene." In: *Science* 352.6287 (2016), pp. 837–840.

[95] Binquan Luan, Haoran Wang, and Tien Huynh. "Enhanced binding of the N501Y-mutated SARS-CoV-2 spike protein to the human ACE2 receptor: insights from molecular dynamics simulations." In: *FEBS letters* 595.10 (2021), pp. 1454–1461.

[96] Jaclyn K Mann et al. "The fitness landscape of HIV-1 gag: advanced modeling approaches and validation of model predictions by in vitro testing." In: *PLoS computational biology* 10.8 (2014), e1003776.

[97] Guillaume Marmier, Martin Weigt, and Anne-Florence Bitbol. "Phylogenetic correlations can suffice to infer protein partners from sequences." In: *PLoS computational biology* 15.10 (2019), e1007179.

[98] David Mavor et al. "Determination of ubiquitin fitness landscapes under different chemical stresses in a classroom setting." In: *Elife* 5 (2016), e15802.

[99] Richard N McLaughlin Jr et al. "The spatial architecture of protein function and adaptation." In: *Nature* 491.7422 (2012), pp. 138–142.

[100] Matthijs Meijers et al. "Vaccination shapes evolutionary trajectories of SARS-CoV-2." In: *bioRxiv* (2022).

[101] Daniel Melamed et al. "Deep mutational scanning of an RRM domain of the Saccharomyces cerevisiae poly(A)-binding protein." In: *RNA* 19.11 (2013), pp. 1537–1551. DOI: 10.1261/rna.040709.113. eprint: http://rnajournal.cshlp.org/content/19/11/1537.full.pdf+html. URL: http://rnajournal.cshlp.org/content/19/11/1537.abstract.

[102] Alexandre Melnikov et al. "Comprehensive mutational scanning of a kinase in vivo reveals substrate-dependent fitness landscapes." In: *Nucleic acids research* 42.14 (2014), e112–e112.

[103] Francesca Mignacco, Pierfrancesco Urbani, and Lenka Zdeborová. "Stochasticity helps to navigate rough landscapes: comparing gradient-descent-based algorithms in the phase retrieval problem." In: *Machine Learning: Science and Technology* 2.3 (2021), p. 035029.

[104] Thomas P Minka. "Expectation propagation for approximate Bayesian inference." In: *arXiv preprint arXiv:1301.2294* (2013).

[105] Charlotte M Miton and Nobuhiko Tokuriki. "How mutational epistasis impairs predictability in protein evolution and design." In: *Protein Science* 25.7 (2016), pp. 1260–1272.

[106] Todd K Moon. "The expectation-maximization algorithm." In: *IEEE Signal processing magazine* 13.6 (1996), pp. 47–60.

[107] Thierry Mora et al. "Maximum entropy models for antibody diversity." In: *Proceedings of the National Academy of Sciences* 107.12 (2010), pp. 5405–5410.

[108] Faruck Morcos et al. "Direct-coupling analysis of residue coevolution captures native contacts across many protein fami lies." In: *Proceedings of the National Academy of Sciences* 108.49 (2011), E1293–E1301.

[109] Jamie A Moroco et al. "Differential sensitivity of Src-family kinases to activation by SH3 domain displacement." In: *PloS one* 9.8 (2014), e105629.

[110] Hamish Nisbet Munro. *Mammalian protein metabolism*. Vol. 4. Elsevier, 2012.

[111] Anna Paola Muntoni et al. "Aligning biological sequences by exploiting residue conservation and coevolution." In: *Physical Review E* 102.6 (2020), p. 062409.

[112] Kenneth Murphy, Paul Travers, Mark Walport, et al. "Janeway's immunobiology. Garland science." In: *Taylor & Francis Group, LLC* 15 (2008), pp. 655–708.

[113] Richard A Neher and Boris I Shraiman. "Statistical genetics and evolution of quantitative traits." In: *Reviews of Modern Physics* 83.4 (2011), p. 1283.

[114] Peter D Nieuwkoop and Lien A Sutasurya. *Primordial germ cells in the chordates: embryogenesis and phylogenesis*. Vol. 7. CUP Archive, 1979.

[115] C Anders Olson, Nicholas C Wu, and Ren Sun. "A comprehensive biophysical description of pairwise epistasis throughout an entire protein domain." In: *Current Biology* 24.22 (2014), pp. 2643–2651.

[116] Manfred Opper and David Saad. *Advanced mean field methods: Theory and practice*. MIT press, 2001.

[117] Manfred Opper and Ole Winther. "Adaptive and self-averaging Thouless-Anderson-Palmer mean-field theory for probabilistic modeling." In: *Physical Review E* 64.5 (2001), p. 056131.

[118] Manfred Opper and Ole Winther. "Gaussian processes for classification: Mean-field algorithms." In: *Neural computation* 12.11 (2000), pp. 2655–2684.

[119] Angel R Ortiz et al. "Ab initio folding of proteins using restraints derived from evolutionary information." In: *Proteins: Structure, Function, and Bioinformatics* 37.S3 (1999), pp. 177–185.

[120] Jakub Otwinowski. "Biophysical Inference of Epistasis and the Effects of Mutations on Protein Stability and Function." In: *Molecular Biology and Evolution* 35.10 (Aug. 2018), pp. 2345–2354. ISSN: 0737-4038. DOI: 10.1093/molbev/msy141. eprint: https://academic.oup.com/mbe/article-pdf/35/10/2345/27171842/msy141.pdf. URL: https://doi.org/10.1093/molbev/msy141.

[121] Jakub Otwinowski, David M McCandlish, and Joshua B Plotkin. "Inferring the shape of global epistasis." In: *Proceedings of the National Academy of Sciences* 115.32 (2018), E7550–E7558.

[122] Sergey Ovchinnikov, Hetunandan Kamisetty, and David Baker. "Robust and accurate prediction of residue–residue interactions across protein interfaces using evolutionary information." In: *elife* 3 (2014), e02030.

[123] Delphine Planas et al. "Reduced sensitivity of SARS-CoV-2 variant Delta to antibody neutralization." In: *Nature* 596.7871 (2021), pp. 276–280.

[124] Timm Plefka. "Convergence condition of the TAP equation for the infinite-ranged Ising spin glass model." In: *Journal of Physics A: Mathematical and general* 15.6 (1982), p. 1971.

[125] Lorenzo Posani et al. "Infer global, predict local: quantity-quality trade-off in protein fitness predictions from sequence data." In: *bioRxiv* (2022).

[126] Whitney E Purtha et al. "Memory B cells, but not long-lived plasma cells, possess antigen specificities for viral escape mutants." In: *Journal of Experimental Medicine* 208.13 (2011), pp. 2599–2606.

[127] Roshan Rao et al. "Transformer protein language models are unsupervised structure learners." In: *Biorxiv* (2020), pp. 2020–12.

[128] Roshan M Rao et al. "MSA transformer." In: *International Conference on Machine Learning*. PMLR. 2021, pp. 8844–8856.

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

[130] Olivier Rivoire. "Parsimonious evolutionary scenario for the origin of allostery and coevolution patterns in proteins." In: *Phys. Rev. E* 100 (3 Sept. 2019), p. 032411. DOI: 10.1103/PhysRevE.100.032411. URL: https://link.aps.org/doi/10.1103/PhysRevE.100.032411.

[131] Edwin Rodriguez Horta, Pierre Barrat-Charlaix, and Martin Weigt. "Toward inferring Potts models for phylogenetically correlated sequence data." In: *Entropy* 21.11 (2019), p. 1090.

[132] Edwin Rodriguez Horta and Martin Weigt. "On the effect of phylogenetic correlations in coevolution-based contact prediction in proteins." In: *PLOS Computational Biology* 17.5 (2021), e1008957.

[133] Juan Rodriguez-Rivas et al. "Epistatic models predict mutable sites in SARS-CoV-2 proteins and epitopes." In: *Proceedings of the National Academy of Sciences* 119.4 (2022), e2113118119.

[134] Nathan J Rollins et al. "Inferring protein 3D structure from deep mutation scans." In: *Nature genetics* 51.7 (2019), p. 1170.

[135] Philip A Romero and Frances H Arnold. "Exploring protein fitness landscapes by directed evolution." In: *Nature reviews Molecular cell biology* 10.12 (2009), p. 866.

[136] Philip A. Romero, Tuan M. Tran, and Adam R. Abate. "Dissecting enzyme function with microfluidic-based deep mutational scanning." In: *Proceedings of the National Academy of Sciences* 112.23 (2015), pp. 7159–7164. ISSN: 0027-8424. DOI: 10.1073/pnas.1422285112. eprint: https://www.pnas.org/content/112/23/7159.full.pdf. URL: https://www.pnas.org/content/112/23/7159.

[137] Benjamin P. Roscoe and Daniel N.A. Bolon. "Systematic Exploration of Ubiquitin Sequence, E1 Activation Efficiency, and Experimental Fitness in Yeast." In: *Journal of Molecular Biology* 426.15 (2014), pp. 2854–2870. ISSN: 0022-2836. DOI: https://doi.org/10.1016/j.jmb.2014.05.019. URL: http://www.sciencedirect.com/science/article/pii/S0022283614002587.

[138] Alan F Rubin et al. "A statistical framework for analyzing deep mutational scanning data." In: *Genome biology* 18.1 (2017), p. 150.

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

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

[141] Karen S Sarkisyan et al. "Local fitness landscape of the green fluorescent protein." In: *Nature* 533.7603 (2016), pp. 397–401.

[142] Jórn M Schmiedel and Ben Lehner. "Determining protein structures using deep mutagenesis." In: *Nature genetics* 57 (2019), pp. 1177–1186.

[143] Elad Schneidman et al. "Weak pairwise correlations imply strongly correlated network states in a neural population." In: *Nature* 440.7087 (2006), pp. 1007–1012.

[144] Alexander Schug et al. "High-resolution protein complexes from integrating genomic information with molecular simulation." In: *Proceedings of the National Academy of Sciences* 106.52 (2009), pp. 22124–22129.

[145] Luca Sesta et al. "Amala: Analysis of directed evolution experiments via annealed mutational approximated landscape." In: *International journal of molecular sciences* 22.20 (2021), p. 10908.

[146] Jack Sherman and Winifred J Morrison. "Adjustment of an inverse matrix corresponding to a change in one element of a given matrix." In: *The Annals of Mathematical Statistics* 21.1 (1950), pp. 124–127.

[147] Kai Shimagaki and Martin Weigt. "Selection of sequence motifs and generative Hopfield-Potts models for protein families." In: *Physical Review E* 100.3 (2019), p. 032128.

[148] Jung-Eun Shin et al. "Protein design and variant prediction using autoregressive generative models." In: *Nature communications* 12.1 (2021), p. 2403.

[149] Paul Smolensky. *Information processing in dynamical systems: Foundations of harmony theory*. Tech. rep. Colorado Univ at Boulder Dept of Computer Science, 1986.

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

[151] Muhammad S Sohail et al. "Resolving genetic linkage reveals patterns of selection in HIV-1 evolution." In: *bioRxiv* (2019), p. 711861.

[152] Muhammad Saqib Sohail et al. "MPL resolves genetic linkage in fitness inference from complex evolutionary histories." In: *Nature Biotechnology* 39.4 (2021), pp. 472–479.

[153] Lea M Starita et al. "Activity-enhancing mutations in an E3 ubiquitin ligase identified by high-throughput mutagenesis." In: *Proceedings of the National Academy of Sciences* 110.14 (2013), E1263–E1272.

[154] Lea M Starita et al. "Massively parallel functional analysis of BRCA1 RING domain variants." In: *Genetics* 200.2 (2015), pp. 413–422.

[155] Tyler N Starr, Lora K Picton, and Joseph W Thornton. "Alternative evolutionary histories in the sequence space of an ancient protein." In: *Nature* 549.7672 (2017), pp. 409–413.

[156] Tyler N Starr and Joseph W Thornton. "Epistasis in protein evolution." In: *Protein Science* 25.7 (2016), pp. 1204–1218.

[157] Tyler N Starr et al. "Deep mutational scanning of SARS-CoV-2 receptor binding domain reveals constraints on folding and ACE2 binding." In: *cell* 182.5 (2020), pp. 1295–1310.

[158] DK Steward. "Essential Amino Acids: Chart, Abbreviations and Structure." In: *Applied Sciences from Technology Networks. http://www. technologynetworks. com/appliedsciences/articles/essential-amino-acids-chart-abbreviations-and-structure-324357* (2019).

[159] Michael A Stiffler, Doeke R Hekstra, and Rama Ranganathan. "Evolvability as a function of purifying selection in TEM-1 β-lactamase." In: *Cell* 160.5 (2015), pp. 882–892.

[160] Michael A Stiffler et al. "Protein structure from experimental evolution." In: *Cell Systems* 10.1 (2020), pp. 15–24.

[161] Ludovico Sutto et al. "From residue coevolution to protein conformational ensembles and functional dynamics." In: *Proceedings of the National Academy of Sciences* 112.44 (2015), pp. 13567–13572.

[162] Toshiyuki Tanaka. "Mean-field theory of Boltzmann machine learning." In: *Physical Review E* 58.2 (1998), p. 2302.

[163] Tijmen Tieleman. "Training restricted Boltzmann machines using approximations to the likelihood gradient." In: *Proceedings of the 25th international conference on Machine learning*. 2008, pp. 1064–1071.

[164] Jeanne Trinquier et al. "Efficient generative modeling of protein sequences using simple autoregressive models." In: *Nature communications* 12.1 (2021), pp. 1–11.

[165] Jérôme Tubiana. "Restricted Boltzmann machines: from compositional representations to protein sequence analysis." PhD thesis. Paris Sciences et Lettres (ComUE), 2018.

[166] Jérôme Tubiana, Simona Cocco, and Rémi Monasson. "Learning compositional representations of interacting systems with restricted boltzmann machines: Comparative study of lattice proteins." In: *Neural computation* 31.8 (2019), pp. 1671–1717.

[167] Jérôme Tubiana, Simona Cocco, and Rémi Monasson. "Learning protein constitutive motifs from sequence data." In: *Elife* 8 (2019), e39397.

[168] Jérôme Tubiana and Rémi Monasson. "Emergence of compositional representations in restricted Boltzmann machines." In: *Physical review letters* 118.13 (2017), p. 138301.

[169] Gabriel D Victora and Michel C Nussenzweig. "Germinal centers." In: *Annu Rev Immunol* 30.1 (2012), pp. 429–457.

[170] Timothy R Wagenaar et al. "Resistance to vemurafenib resulting from a novel mutation in the BRAFV 600 E kinase domain." In: *Pigment cell & melanoma research* 27.1 (2014), pp. 124–133.

[171] Sheng Wang et al. "Accurate de novo prediction of protein contact map by ultra-deep learning model." In: *PLoS computational biology* 13.1 (2017), e1005324.

[172] Martin Weigt et al. "Identification of direct residue contacts in protein–protein interaction by message passing." In: *Proceedings of the National Academy of Sciences* 106.1 (2009), pp. 67–72.

[173] Madeline C Weiss et al. "The physiology and habitat of the last universal common ancestor." In: *Nature microbiology* 1.9 (2016), pp. 1–8.

[174] Timothy A Whitehead et al. "Optimization of affinity, specificity and function of designed influenza inhibitors using deep sequencing." In: *Nature biotechnology* 30.6 (2012), pp. 543–548.

[175] Nicholas C Wu et al. "Adaptation in protein fitness landscapes is facilitated by indirect paths." In: *Elife* 5 (2016), e16965.

[176] Zachary Wu et al. "Machine learning-assisted directed protein evolution with combinatorial libraries." In: *Proceedings of the National Academy of Sciences* 116.18 (2019), pp. 8852–8858.

[177] Kevin K Yang, Zachary Wu, and Frances H Arnold. "Machine-learning-guided directed evolution for protein engineering." In: *Nature methods* (2019), p. 1.

[178] Lizhou Zhang et al. "SARS-CoV-2 spike-protein D614G mutation increases virion spike density and infectivity." In: *Nature communications* 11.1 (2020), pp. 1–9.

[179] Jia Zheng, Ning Guo, and Andreas Wagner. "Selection enhances protein evolvability by increasing mutational robustness and foldability." In: *Science* 370.6521 (2020).

[180] Wei Zheng et al. "Deep-learning contact-map guided protein structure prediction in CASP13." In: *Proteins: Structure, Function, and Bioinformatics* 87.12 (2019), pp. 1149–1164.

This Ph.D. thesis has been typeset by means of the TEX-system facilities. The typesetting engine was LuaLATEX. The document class was `toptesi`, by Claudio Beccari, with option `tipotesi=scudo`. This class is available in every up-to-date and complete TEX-system installation.