Methods for staistical inference on correlated data : application to genomic data

Eleonora De Leonardis

Thèse de Doctorat

En vue de l'obtention du grade de

Docteur de l'École Normale Supérieure

École Doctorale 564 Physique en Ile de France
Discipline : Physique

Présentée et soutenue le 26 Octobre 2015 par
Eleonora De Leonardis

Méthodes pour l'inférence en grande dimension avec des données correlées : application à des données génomiques

Laboratoire de Physique Statistique
et
Laboratoire de Biologie Computationnelle et Quantitative

Thèse dirigée par :

	Simona Cocco	LPS-ENS
	Martin Weigt	LBCQ-UPMC

Membres du jury :

Rapporteurs :	Johannes Berg	Université de Cologne
	Paolo De Los Rios	Ecole Polytechnique de Lausanne
Examinateurs :	Hervé Isambert	Institut Curie
	Thierry Mora	ENS

Numéro identifiant de la Thèse : 92560

Acknowledgements

I'd like to thank firstly my supervisors, Martin and Simona, for having guided and helped me during the last three years. Doing a PhD with you has been to me a great opportunity and a pleasure: I extremely enjoyed, and I will surely miss, working with you. Thanks for having introduced me to research, even if I will leave now I'm sure this experience has changed my way to explore the world and I will always keep what you taught me in mind, whatever my job is.

Many thanks to all the members of the committee for the time they have dedicated to my dissertation, from reading the manuscript to participating to the defence.

A very special thank to the collaborators of this work. Thanks to Alex and his students: firstly Benjamin then Sebastian and Fabian. Many thanks to John for his contribution and help in developing the ACE code. I'd like finally to thank Rémi for incredibly useful discussions and for having been there all along my PhD.

Thanks to all the other PhD students and post-docs (and engineers) of LPS, LBCQ and LPT. Among these ones a special thank to: Dario, Hugo, Matteo, Guido, Francesco, Gaia, Ulisse, Corrado, Alaa, Charles. Many thanks to Alice, I enjoyed a lot working and talking with you. Thanks for having helped me to improve my french!

Thanks to Caterina and Sivia: your Orsay-BBQs brought me back to outdoor life. Thanks again to Silvia, we should have talked more!

Thanks to Massimo for having guided me three years ago during the hard application period.

Of course I cannot forget to thank my family: as you can see, it has been fruitful to move from Turin. Thanks to Marcello: you helped me developing my communication skills and this dissertation is dedicated to you. Since once again, I've explained my work as if you were the only audience.

Finally, thanks Alberto: you're right, an other adventure is about to start!

Methods for statistical inference on correlated data: application to genomic data

Abstract

The availability of huge amounts of data has changed the role of physics with respect to other disciplines. Within this dissertation I will explore the innovations introduced in molecular biology thanks to statistical physics approaches. In the last 20 years the size of genome databases has exponentially increased, therefore the exploitation of raw data, in the scope of extracting information, has become a major topic in statistical physics. After the success in protein structure prediction, surprising results have been finally achieved also in the related field of RNA structure characterisation. However, recent studies have revealed that, even if databases are growing, inference is often performed in the under sampling regime and new computational schemes are needed in order to overcome this intrinsic limitation of real data. This dissertation will discuss inference methods and their application to RNA structure prediction. We will discuss some heuristic approaches that have been successfully applied in the past years, even if poorly theoretically understood. The last part of the work will focus on the development of a tool for the inference of generative models, hoping it will pave the way towards novel applications.

Keywords: inference, RNA, mean-field, Potts model, generative models, regularisation, structure prediction

Résumé

La disponibilité de quantités énormes de données a changé le rôle de la physique par rapport aux autres disciplines. Dans cette thèse, je vais explorer les innovations introduites dans la biologie moléculaire grâce à des approches de physique statistique. Au cours des 20 dernières années, la taille des bases de données sur le génome a augmenté de façon exponentielle : l'exploitation des données brutes, dans le champ d'application de l'extraction d'informations, est donc devenu un sujet majeur dans la physique statistique. Après le succès dans la prédiction de la structure des protéines, des résultats étonnamment bons ont été finalement obtenus aussi pour l'ARN. Cependant, des études récentes ont révélé que, même si les bases de données sont de plus en plus grandes, l'inférence est souvent effectuée dans le régime de sous-échantillonnage et de nouveaux systèmes informatiques sont nécessaires afin de surmonter cette limitation intrinsèque des données réelles. Cette thèse va discuter des méthodes d'inférence et leur application à des prédictions de la structure de l'ARN. Nous allons comprendre certaines approches heuristiques qui ont été appliquées avec succès dans les dernières années, même si théoriquement mal comprises. La dernière partie du travail se concentrera sur le développement d'un outil pour l'inférence de modèles génératifs, en espérant qu'il ouvrira la voie à de nouvelles applications.

Mots-clés: Inférence, ARN, champ moyen, modèl de Potts, modèles génératifs, régularisation, prédiction structurelle

Contents

Motivations

In the last few years, the field of the molecular biology has experienced an almost unbelievable improvement both in the quantity and in the quality of the data available. The number of genome projects has increased as technological improvements continue to lower the cost of sequencing, consequently since 1995, we have assisted at the exponential growth of genome sequence databases. At the same time the profound need for tools able to manage with this huge amount of data and, most importantly, able to extract useful information from sequences analysis has interested scientists with diverse backgrounds. Nowadays computational and quantitative biology are cross-disciplinary fields and more and more innovative works have benefit from this extremely heterogeneous framework.

From the physicist's point of view, the task of exploiting data in order to infer appropriate models is called *inverse problem*. As *direct* problems consist in computing some observables from a known probability distribution, solving an *inverse* problem means to estimate the probability distribution from which the observed data have been drawn. For decades inverse problems have been extensively studied within the theoretical and statistical physics community and a huge and ever growing literature exists. Therefore physicists have played a double role in the exploration of genomic data. On the one hand they have identified biological interesting topics eligible for application of existing statistical physics methods. On the other hand some of these topics have become extremely popular and, since they have been faced for the first time, novel solutions have been developed to the scope of a direct application.

The application of comparative sequence analysis results to protein structure prediction is nowadays a well established framework. Several works, making use of diverse tools, have shown that the correct interpretation of correlations in sequencing data, can help in predicting protein structures. Can we use similar methods for RNA structure prediction? Recent advances in molecular biology have revealed RNA having a crucial role in the cell, thus the structural characterisation of RNAs has become of general interest. Within this dissertation I will address this problem using

Direct-Couplings Analysis (DCA), a mean-field based inference method, proved to give reliable results on protein data.

However, during my thesis, I have worked not only on the application of an existing tool to a novel problem (DCA applied to RNA) but also on the development of a new approach to inverse problems: the Adaptive Cluster Expansion (ACE). Initially developed for binary variables, the generalisation of ACE to sequence-like data promises to provide a powerful tool for comparative sequence analysis.

Chapter 1 of this dissertation will be devoted to the introduction of inverse problems and of the existing approaches to their solution. I will formally define an inverse problem and I will review some interesting works appeared in the field, focusing in particular on the ones concerning application to biological data.

In chapter 2 I will present recent applications of statistical physics method to diverse topics and fields, such as ecology, social science and economics. The aim of this chapter is to stress that not only biology can benefit from advanced statistical analysis: data are nowadays used to describe our everyday life.

In chapter 3 the work on the application of DCA to RNA structure prediction will be exposed. The chapter starts with an exhaustive introduction on the problem from both the biological and the computational point of view. The results are presented within the related paper. Finally, after the reprint, some interesting open problems are shown.

Chapter 4 will report a work we made on the role of regularisation on naive mean-field inference. We tried to deeply understand pros and cons of mean-field approximation and we showed that strong regularisation can only partially correct the mean-field intrinsic errors. Also in this case, after a short introduction, the main results will be included inside the reprinted paper.

In chapter 5, finally, I will expose the work done on ACE. After a review on the original algorithm, the main improvements we introduced will be studied. Note how the knowledge we have about target systems (RNA, proteins, etc...) guided the development of the algorithm. A paper on ACE and its application to biological data is in preparation. The code of the algorithm will be contextually released.

Chapter 1

Review: inverse Ising and Potts

Recent developments in computational sciences have shown the importance of inverse problems. The challenge in this field consists in trying to extract the rule governing a certain system from the statistics of samples of a large number of *microscopic* variables. Often experimental measurements can access a reduced and usually biased sample of the whole possible set of different behaviours of a given system. Advances in statistical physics promise to provide, however, an ever increasing number of useful tools to extract information from experimental data.

Formally, inverse problems try to describe the system estimating an unknown probability distribution $P_{data}(\boldsymbol{\sigma})$, for a high-dimensional feature vector $\boldsymbol{\sigma} = \{\sigma_1, \sigma_2, ...\sigma_N\}$, given a set of M observations of this vector.

The paradigm of inverse problems is the inverse Ising problem also known as Boltzmann machine learning. Born to describe ferromagnetic materials, the Ising model is nowadays applied to the description of a multitude of systems: from neural networks [1] to protein fitness landscapes [2], from protein 3D structures [3] to gene expression networks [4]. Its straightforward generalisation, the Potts model, is the most natural choice for systems with many states variables and it has been proved enhance the system's description.

Inverse and direct problems can be considered under a dual perspective: we can compute averaged quantities, such as magnetisations and correlations (direct problem), given the full set of parameters of the model, meaning fields and couplings, or we can infer the latter ones (inverse problem) such that the data statistics is recovered. A huge amount of diverse approaches exists. Some approaches to the inverse problem have been inspired by this duality and the solution of the inverse problem is faced as the *explicit inversion* of the solution of the direct one (e.g. mean-field [5]). Others methods are based, instead, on the fact that usually direct problems are easier than inverse ones and thus the solution of the former is iteratively used to approximate the

solution of the latter: this was, for instance, the approach for the first Boltzmann machine learning solution [6]. Finally, some of them are rooted in the intrinsic differences between direct and inverse problem (e.g. adaptive cluster expansion [7]). Mentioning all the possible ways to face the problem is beyond the purpose of this thesis, and I will focus on the techniques that have successfully been applied to biological inference problems or that have even been specifically designed for such applications.

In this chapter I will first define the Ising and the Potts model and then show how a very general principle justifies the choice for these models for the description of very complex systems (e.g. protein structures, gene expression or neural networks). The last section of this chapter will be dedicated to some of the most popular methods developed for the solution of the inverse Ising model.

1.1 Ising and Potts models: definition

The Ising and the Potts models describe systems characterised by pairwise interactions among their elements, called spins in the language of statistical physics. While the former is characterised by binary spin variables ($\sigma_i = -1, +1$) the latter presents many *colours* for each spin: σ_i^a where $a = 1, ..., q$, the binary case being recovered when $q = 2$.

$$H_{Ising} = \sum_{i=1}^{N} h_i \sigma_i + \sum_{i<j} J_{ij} \sigma_i \sigma_j$$

$$H_{Potts} = \sum_{i=1}^{N} \sum_{a=1}^{q} h_i(\sigma_i^a) + \sum_{i<j} \sum_{a,b=1}^{q} J_{ij}(\sigma_i^a, \sigma_j^b) \tag{1.1}$$

where the h_i are local fields and the J_{ij} are couplings between pairs of spins. Eq. 1.1 shows the Ising and the Potts Hamiltonians [1] for a system with N variables and, in the Potts case, of q colours.

1.2 Maximum entropy principle

An impressing point about inverse problems is that within the applied problems I will describe in this thesis, Ising and Potts models emerge naturally from the application of a very general tenet: the maximum entropy principle (MEP). According

1. H_{Potts} defined in 1.1 refers actually to the so called generalised Potts model in which couplings and fields also depend on colours. The original Hamiltonian is $H = \sum_{i<j} J_{ij} \delta(\sigma_i, \sigma_j)$ where σ_i and σ_j can take q possible values and δ is the Kronecker delta that is different from zero if and only if $\sigma_i = \sigma_j$. In the following I will always refer to the generalised Potts model as simply Potts model.

to MEP we define the least constrained probability distribution reproducing the observables, i.e. a description of the data variability only in terms of the observables.

Imagine we want to characterise a whatever sample of data. We could extract some information from data computing two simple quantities: the frequency of single variables and the correlation between each pair of variables. Going beyond that is to some extent hard and often useless [8]. Consider now the case of a system made of N Ising variables. Our sample is composed by a set of observations $\boldsymbol{\sigma}^\tau = \{\sigma_1^\tau, \sigma_2^\tau, ..., \sigma_N^\tau\}$ with $\tau = 1, .., M$. Frequencies and correlations are thus defined as in Eq. 1.2.

$$f_i^{data} = \frac{1}{M}\sum_{\tau=1}^{M}\sigma_i^\tau \qquad\qquad f_{ij}^{data} = \frac{1}{M}\sum_{\tau=1}^{M}\sigma_i^\tau\sigma_j^\tau \tag{1.2}$$

However, taking into account f_i^{data} and f_{ij}^{data} to describe the interaction existing among spins gives us only a partial information about the system. A full description is in fact contained in the probabilistic model $P_{data}(\boldsymbol{\sigma})$ from which these samples have been drawn and to which, unfortunately, we do not have access. Thanks to MEP [9] it is possible to compute a probability distribution $P_{mep}(\boldsymbol{\sigma})$ satisfying the following constraints:

$$f_i^{mep} = f_i^{data} \qquad f_{ij}^{mep} = f_{ij}^{data} \tag{1.3}$$

where

$$f_i^{mep} = \sum_{\boldsymbol{\sigma}} P_{mep}(\boldsymbol{\sigma})\sigma_i \qquad f_{ij}^{mep} = \sum_{\boldsymbol{\sigma}} P_{mep}(\boldsymbol{\sigma})\sigma_i\sigma_j \tag{1.4}$$

Constraints in Eq. 1.2 can be introduced into the entropy definition thanks to Lagrangian multipliers and can be in principle generalised to any other observable of P^{data} and P^{mep}. In the specific case of the simplified description we have chosen, we only need two types of Lagrangian multipliers: h_i for frequencies and J_{ij} for correlations .

$$\begin{aligned} S = &-\sum_{\boldsymbol{\sigma}} P(\boldsymbol{\sigma})\ln P(\boldsymbol{\sigma}) + \lambda\sum_{\boldsymbol{\sigma}}\left(P(\boldsymbol{\sigma})-1\right) \\ &+\sum_{i=1}^{N} h_i\left(\sum_{\boldsymbol{\sigma}} P(\boldsymbol{\sigma})\sigma_i - f_i^{data}\right) + \sum_{i<j} J_{ij}\left(\sum_{\boldsymbol{\sigma}} P(\boldsymbol{\sigma})\sigma_i\sigma_j - f_{ij}^{data}\right) \end{aligned} \tag{1.5}$$

P_{mep} is thus defined as the function maximising the entropy S in equation 1.5. The result of this maximisation is the Boltzmann distribution in Eq. 1.6 where the Hamiltonian coincides with H^{Ising} or H^{Potts}. Note that the usual temperature parameter β is fixed to 1.

$$P_{mep}(\boldsymbol{\sigma}) = \frac{1}{Z} e^{-H(\boldsymbol{\sigma})} \tag{1.6}$$

The normalisation Z is referred to as the *partition function* in the language of statistical physics and contains all the information needed about the systems. Its derivatives with respect to the fields and to the couplings coincide indeed to the marginals of the distribution, i.e. the single- and two-site statistics f_i^{mep} and f_{ij}^{mep}.

1.3 Solving inverse problems

Within the described formulation, the first step towards the solution of an inverse problem is the application of MEP, meaning to obtain the analytical form of a model potentially describing the data. The main challenge is to solve the inverse problem and compute a set of parameters fitting the input correlations and magnetisations. Since applications of inverse problems are highly interdisciplinary, several solutions exist having been developed within different communities (e.g. information theory, computer science, physics) and can be formulated under different perspectives. For instance, one can search for the set of parameters minimising the Kullback–Leibler divergence between $P_{data}(\boldsymbol{\sigma})$ and $P_{mep}(\boldsymbol{\sigma})$ in Eq. 1.7.

$$D_{KL}(P_{data} \parallel P_{mep}) = \sum_{\boldsymbol{\sigma}} P_{data}(\boldsymbol{\sigma}) \ln\left(\frac{P_{data}(\boldsymbol{\sigma})}{P_{mep}(\boldsymbol{\sigma})}\right) \tag{1.7}$$

or equivalently, the set of parameters minimising the negative log-likelihood $\mathcal{L}$ (Eq. 1.9) that the model P_{mep} can actually reproduce data.

$$\mathcal{L} = -\frac{1}{M} \sum_{\tau=1}^{M} \ln P_{mep}(\sigma^{\tau}) \tag{1.8}$$

$$= \ln(Z) - \sum_{i=1}^{N} h_i f_i^{data} - \sum_{i<j}^{N} J_{ij} f_{ij}^{data} \tag{1.9}$$

Moreover, thanks to Eqs. 1.10 we know that the obtained distribution reproduces the desired statistics, i.e. Eqs. 1.2 are surely satisfied.

$$\frac{\partial D_{KL}}{\partial h_i} = f_i^{mep} - f_i^{data} \qquad \frac{\partial D_{KL}}{\partial J_{ij}} = f_{ij}^{mep} - f_{ij}^{data} \tag{1.10}$$

Given these general results, many different strategies have been designed. In principle the minimisation of D_{KL} or of the negative log-likelihood can fully solve the problem. The exact computation of these quantities implies a sum over all possible configurations of the system and thus it becomes rapidly infeasible for an increasing number of spins: the number of configurations scales as q^N where N is the number of spins in the system and q is the number of colours. Many approximated solutions have been proposed in the past years in order to find the best equilibrium between result reliability and computational feasibility.

1.3.1 Boltzmann machine learning

The Boltzmann machine learning, as presented in [6], is one of the first approaches to the inverse Ising problem and it was developed within the Computer Science community. The authors' purpose was to define a network able to learn some simple rules and having learning properties similar to those of the Hopfield network [10]. However, differently from the Hopfield model, the stochasticity they have introduced with a Metropolis-like Monte Carlo dynamics lets the system escape from local minima and eventually reach *thermal equilibrium*.

The Boltzmann machine stores learned information in a set of *weights* being the interactions between nodes. The learning process of the machine corresponds to the solution of an inverse Ising problem, the weights between nodes being equivalent to couplings between spins. The strategy introduced by the authors consists in a two-step approach: first solve the direct problem using the Monte Carlo sampling, then solve the inverse problem updating the weights according to Eq. 1.11. These two steps are iterated till convergence is reached.

$$J_{ij}^n = J_{ij}^{n-1} + \eta(f_{ij}^{data} - f_{ij}^{MC}) \tag{1.11}$$

Eq. 1.11 is derived from the minimisation of the Kullback-Leibler distance between the Monte Carlo equilibrium distribution and the data distribution. Being this minimisation a convex optimisation problem, gradient descent is guaranteed to converge to the exact solution. Note that, even if in the original algorithm in [6] no mention was given to fields updating, the generalisation of this algorithm to models with local fields is straightforward and in chapter 5 we will see an example of this kind.

Boltzmann machine learning is a very accurate way to fit parameters however it is extremely expensive in term of computational time. It is still used to analyse diverse types of biological data, from neurons recording to protein sequences [2] usually when strong interactions exist among the variables in the system and fine information need to be extracted from the sample.

1.3.2 Mean-field

Mean-field approximation (MFA) is an extremely widespread topic. The simplest MFA is the so called naive MF entailing the approximation of the model free energy in terms of averaged magnetisations $m_i = \langle \sigma_i \rangle$ over the Gibbs measure in 1.6.

The free energy can be written as:

$$\begin{aligned} \ln Z_{nMF} = &\sum_i \left[\left(\frac{1-m_i}{2} \right) \ln \left(\frac{1-m_i}{2} \right) - \left(\frac{1+m_i}{2} \right) \ln \left(\frac{1+m_i}{2} \right) \right] \\ &+ \sum_i h_i m_i + \sum_{i \neq j} J_{ij} m_i m_j \end{aligned} \tag{1.12}$$

The first two terms in Eq. 1.12 correspond to terms of order zero in the well known Plefka expansion [11] [12] while the last term is the order-one term. Further steps in the expansion can be done. For instance the second-order term (last term in Eq. 1.13) corresponds to TAP approximation [13], derived including also the Onsager reaction term.

$$\begin{aligned} \ln Z_{TAP} = &\sum_i \left[\left(\frac{1-m_i}{2} \right) \ln \left(\frac{1-m_i}{2} \right) - \left(\frac{1+m_i}{2} \right) \ln \left(\frac{1+m_i}{2} \right) \right] \\ &+ \sum_i h_i m_i + \sum_{i \neq j} \left[J_{ij} m_i m_j + \frac{1}{2} J_{ij}^2 (1-m_i^2)(1-m_j^2) \right] \end{aligned} \tag{1.13}$$

Differently from naive MF, where the probability distribution is fully factorised ($P^{mep}(\boldsymbol{\sigma}) \simeq \prod_{i=1}^{N} P_i^{mep}(\sigma_i)$), the so called Bethe approximation considers a model factorised over two-spin interactions only (Eq. 1.14), resulting thus exact on tree graphs.

$$P^{mep}(\boldsymbol{\sigma}) \simeq \prod_{i=1}^{N} P_i^{mep}(\sigma_i) \prod_{ij} \frac{P_{ij}^{mep}(\sigma_i, \sigma_j)}{P_i^{mep}(\sigma_i) P_j^{mep}(\sigma_j)} \tag{1.14}$$

Beside the analytical solutions [5], a very efficient algorithm for the inverse Ising in Bethe approximation is the Susceptibility Propagation introduced by [14], inspired by message-passing procedures.

Given the self-consistency equations (i.e. the relation between magnetisations and parameters found minimising the free-energy with respect to magnetisations) of any of the MF methods described above, non-trivial correlations between distant spins can be derived from linear response theory:

$$C_{ij} = \frac{\partial m_i}{\partial h_i} \qquad (C^{-1})_{ij} = \frac{\partial h_i}{\partial m_i} \tag{1.15}$$

We therefore obtain:

$$(C^{-1}_{nMF})_{ij} = \frac{\delta_{ij}}{1 - m_i^2} - J_{ij} \tag{1.16}$$

$$(C^{-1}_{TAP})_{ij} = \left[\frac{1}{1 - m_i^2} + \sum_k J_{ij}^2 (1 - m_k^2) \right] \delta_{ij} - (J_{ij} + 2 J_{ij}^2 m_i m_j) \tag{1.17}$$

Usually for applications diagonal terms are ignored and thus simplified relations can be easily inverted. Note that for these types of MFA finding a solution for the inverse problem depends our ability to invert the above relations and to find close relations for couplings and fields. As far as nMF and TAP are concerned very simple expressions can be derived:

$$J_{ij}^{nMF} = -(C^{-1})_{ij} \tag{1.18}$$

$$J_{ij}^{TAP} = \frac{\sqrt{1 - 8 m_i m_j (C^{-1})_{ij}} - 1}{4 m_i m_j} \tag{1.19}$$

where C is the empirical correlation matrix. Another simple approximation can be obtained by treating every pair of spins as if they were independent on the rest of the system. This approximation is thus called the Independent pair approximation (IP) [15] and, as you can see from Eq. 1.20 and 1.21 it is related to the small correlation expansion (SCE) developed by Sessak and Monasson [16].

$$J_{ij}^{IP} = \frac{1}{4} \ln \frac{((1 + m_i)(1 + m_j) C_{ij}) \, ((1 - m_i)(1 - m_j) C_{ij})}{((1 + m_i)(1 - m_j) C_{ij}) \, ((1 - m_i)(1 + m_j) C_{ij})} \tag{1.20}$$

$$J_{ij}^{SCE} = -(C^{-1})_{ij} + J_{ij}^{IP} - \frac{C_{ij}}{(1 - m_i^2)(1 - m_j^2)} \tag{1.21}$$

SCE consists in the extension of the approach developed in [12] based on a double Legendre transform of the free energy in order to fix both the magnetisations (already done by [12]) and the correlations. The result is eventually a high-temperature Plefka expansion.

All the diverse MFAs guarantee a very fast implementation, whose time scales in the worst case as $\mathcal{O}(N^3)$ since the connected correlation matrix has to be inverted. However the reliability of results is not always ensured and in particular in the low-temperature (strong-coupling) limit all these approximations fail. Recently new approaches [17] [18] have proposed to correct these effects thanks to clustering of configurations according to thermodynamic states. Both the two solutions rely on the reconstruction of configuration space in the low temperature regime and thus result to be unsuitable for those models with a highly non-trivial set of metastable states. Alternatively the low-temperature regime can be overcome by the introduction of regularisation terms helping to correct inference of strong couplings. As we will see extensively in chapter 4, the introduction of a large regularisation, often necessary to correct finite sample effects, turns out to be crucial also in the case of perfect sampling and enlarges the reliability of MFA.

1.3.3 Pseudo-likelihood

Pseudo-likelihood maximisation (PLM) is nowadays one of the most powerful tools for inverse problems and its application to protein structure prediction [19] has proved to outperform any other existing inference method.

PLM approach to inverse problems was developed within the mathematical statistics community [20] [21]. It consists of an approximation of the maximum-likelihood inference, obtained substituting the probability distribution in Eq. 1.8 with the conditional probability of observing one variable σ_i given the observations of all the other variables $\sigma_{\setminus i}$. The probability distribution of the model is therefore replaced by a large set of conditional probabilities (Eq. 1.22) computed from M different samples

$$P_i(\sigma_i^\tau|\boldsymbol{\sigma}_{\setminus i}^\tau) = \frac{e^{\sigma_i^\tau\left[h_i+\sum_{j=1}^N J_{ij}\sigma_j^\tau\right]}}{2\cosh\left[h_i+\sum_{j=1}^N J_{ij}\sigma_j^\tau\right]} \tag{1.22}$$

$$l_i = -\frac{1}{M}\sum_{\tau=1}^{M}\ln P_i(\sigma_i^\tau|\boldsymbol{\sigma}_{\setminus i}^\tau) \tag{1.23}$$

where $\tau = 1,..,M$. The parameters h_i and J_{ij} can be computed via the minimisation of the local log-likelihood l_i in Eq. 1.23. However this procedure is not

fully consistent and returns two different values for the coupling J_{ij}: $J_{ij}^{*,i}$ and $J_{ij}^{*,j}$, respectively coming from the minimisation of l_i and of l_j. Since both the two values are in agreement with all the other estimated parameters, a simple solution for this issue is to replace J_{ij} with $\overline{J_{ij}} = \frac{1}{2}(J_{ij}^{*,i} + J_{ij}^{*,j})$. It is also possible to force to algorithm to return equal values for these couplings by minimising $L_{pseudo} = \sum_{i=1}^{N} l_i$ [22] under this constraint.

In order to avoid finite samples problems and also to help the minimisation algorithm, a regularisation term is usually added to L_{pseudo}. The most common types of regularisation penalties are $L1$-norm and $L2$-norm, jointly to pseudocounts (cf. chapter 4). As far as pseudo-likelihood is concerned, the L1-norm was originally suggested in [21], since it forces small parameters to zero and reduces effectively the number of parameters to be fit. Within some application [19], also the L2-norm has been successfully used.

Note that, differently from MFAs, pseudo-likelihood maximisation is a statistically consistent method, meaning that the parameters estimated from an infinite i.i.d. sample generated by the same model class are asymptotically exact. This is not the case for MFA which makes significant errors even with perfect sampling.

1.3.4 Adaptive cluster expansion

The Adaptive Cluster Expansion (ACE) [7] [23] consists in a perturbative expansion of the log-likelihood in small *clusters*, meaning sub-systems, built in a recursive way and selected according to their contribution to the log-likelihood of the full model. It has been proved that ACE, as Boltzmann machine learning, provides reliable results also in the low-temperature phase where many other inference methods fail. Moreover, differently from Boltzmann machine learning, ACE does not suffer from computational infeasibility on sparse systems (i.e. when the largest cluster size is *small*) and can be used on reasonable system sizes ($N \sim 100$). The success of such an approach relies on the intrinsic difference between direct and inverse problems. Consider $\boldsymbol{J} = \{h_i, J_{ij}\}$ being the parameters of the model and $\boldsymbol{f^{mep}} = \{f_i^{mep}, f_{ij}^{mep}\}$ the correlations of the model. We define the susceptibility matrix and its inverse as:

$$\boldsymbol{\chi} = \left.\frac{\partial \boldsymbol{f^{mep}}}{\partial \boldsymbol{J}}\right|_{\boldsymbol{J}} \qquad \boldsymbol{\chi}^{-1} = \left.\frac{\partial \boldsymbol{J}}{\partial \boldsymbol{f^{mep}}}\right|_{\boldsymbol{f^{mep}}} \tag{1.24}$$

$\boldsymbol{\chi}$ tells us the response of correlations due to a small change in the parameters and can be thus associated to errors in the direct problem solution. $\boldsymbol{\chi}^{-1}$ measures, instead, the response of parameters due to a small variation of the correlations and

can be associated to the inverse problem. The crucial point here is that these two matrices are far from being similar. $\boldsymbol{\chi}^{-1}$ is usually much sparser and shorter-range than χ, meaning that, even if the system is characterised by many strong long-range correlations, couplings still depend on a small number of correlations. This claim turns out to be true also in the low-temperature regime and thus confirms the applicability of ACE to critical models.

In chapter 5 I will extensively discuss about ACE and its application to biological problems. I therefore postpone the detailed description of the algorithm to that part of my dissertation.

Chapter 2

Inverse models across disciplines

The emergence of a massive collection of data has definitely transformed fields such as physics, informatics and biology. In the last decades we assisted to the explosion of computational and quantitative studies about biological topics. Beside computational biologists, an interesting role is played by several interdisciplinary profile scientists applying usually a background in information theory and theoretical or statistical physics to diverse subjects. Consequently nowadays departments of biophysics or bioinformatics exist in almost every university. Conversely if one looks at the leading disciplinary journals in sociology, economics or even political science only a minimal evidence of the emergence of a computational social science can be observed. For some years now big companies, such as Google, Facebook and Amazon, have been appreciating the power of data collection and analysis. The development of machine learning and inference techniques able to manage with tons of data (generally referred to with the term *Big Data*) has enlarged the possibility of exploit information on people habits till endangering everyone privacy. A question thus spontaneously arises: do we have to expect that the computational revolution we assisted in biology will spread to social science and after that directly to our day-to-day life? I would say yes, but let me remark that the emergence of such a data-driven social science is happening at a rate much slower than the one having been observed in biology. Probably the need for appropriate authority manifested by some people has introduced an inertial term in the process due to institutional reacting times.

In this chapter I will analyse some of the most studied applications of inference methods and statistical physics tools to several different topics, from biology to economics. We will first focus on biomolecular structure prediction, as it is one of the main themes of this dissertation, then we will sketch gene expression analysis and neuroscience. The last biological topic considered will be ecology, focusing on collective behaviour of both micro-organisms and higher-order species communities. In

the second section, I will introduce some social science applications, such as human interactions, diseases spreading and economics.

2.1 Application to biology

2.1.1 Molecular biology

The major interest of computational biologists has been for several decades the structural and functional characterisation of important biomolecules such as DNA, RNA and proteins. For instance knowing the 3D structure of a membrane protein helps understanding its molecular mechanism and accelerates the development of pharmacological agents targeting it. However solving three-dimensional structures is a hard experimental task and the structural characterisation of biomolecules has till now proceeded quite slowly. Sequencing results to be much easier and cheaper, thus we assisted at the exponential increase of available sequences. Given sequencing data, the first step consists in searching for homologous sequences, i.e. phylogenetically related sequences sharing a common ancestor. Then, this set of homologous sequences is rearranged so to create a *Multiple Sequence Alignment* (MSA), meaning a matrix of nucleotides or amino acids having on different lines different homologs and on different columns different sites. Sequence sites must be placed in the correct column according to some *equivalence* rule among species. The best alignments tools existing maximise complex global scores depending on single-site frequency of symbols. When these methods were introduced the number of available sequences was extremely poor, thus it was entirely reasonable to ignore higher-order statistics, since the amount of data was insufficient to estimate joint probabilities. MSAs currently available on databases contain tens of thousands and even hundreds of thousands of sequences. Therefore the deep statistical investigation of MSA is now a common practice and diverse approaches exist [24] [2] [25] [26] [3] [27] [28] [29].

Several MSA analysis tools start from the assumption of the so called *co-evolution*: the function of biomolecules strongly depends on their three-dimensional structure and the structure is stabilised thanks to interacting residues or bases. Since the structure (and function) is often highly conserved across species, while the sequence is not, the existence of crucial interactions among distant sites entails correlations between MSA columns. The huge complexity of cooperative interactions between residues makes this problem highly non-trivial: amino acids are mostly pairwise coupled within three dimensional structures but also many three-way or higher-order couplings have been observed [30]. The result is often a dense and complex network

of interactions and local measures of correlation (e.g. Mutual Information) cannot disentangle direct from indirect contributions. Global inverse models among those we analysed in the previous chapter, have thus been successfully employed [3] [31] [19].

Beside structure analysis, the problem of fitness characterisation of proteins has also been part of interesting joint works, both experimental and computational. A major challenge in the field is the HIV-AIDS epidemic [25] [2]. HIV is characterised by an extreme sequence variability. An accurate description of its fitness landscape, meaning the identification of the network of deleterious, beneficial and compensatory mutations, can inform the design of immunogens and therapies in the scope of targeting the virus in its most vulnerable regions.

Molecular studies help us to understand how proteins, RNA and DNA work. However, the cell activity is carried out through the cooperation of many genes and gene products. The genome is organised in regulatory modules or groups of co-regulated genes contributing to a common function. The identification of such a network of interactions is crucial for understanding cell response to internal and external stimuli. The main assumption, underlying computational studies in this field, is that regulators are themselves transcriptionally regulated, thus their expression profiles carry information about their activity level [32]. Gene expression is measured thanks to sequencing (e.g. RNAseq technologies) analysis, then several inference methods are applied in order to infer gene interaction networks reflecting intracellular communication pathways [33] [4]. The most common approach to this problem focuses on the differences in gene expression and aims to identify of meaningful subgroups of genes with similar expression patterns. However, once again, correlation measures cannot provide insight into the direct interactions among genes underlying the observed expression pattern. Maximum entropy principle has been successfully applied here [34] to infer pairwise interactions able to accurate describe expression data. Moreover some approaches have incorporated both gene expression analysis and structural considerations aiming at a more and more global model for living cell activity [35].

2.1.2 Neuroscience

Populations of sensory neurons encode information about stimuli into sequences of action potentials called spikes [36]. The representation of environment signals depends on correlations among neurons and on their ability to coordinate spike patterns. Spike activity can be measured and has been studied in many different brain areas, however the understanding of the *code* mapping neurons firing and response to stimuli is still challenging and diverse interpretations have been broadly debated.

Correlations among neurons have been proved to govern both the conveyance and the storage of information; moreover several measurements have revealed that correlated patterns exist [37] [38] [39], but their origin and importance for decoding the neural code still remains poorly understood. Several coding strategies have been identified [40]: (i) *independence* where each neuron responds independently to an input, (ii) *decorrelation* where neurons interact in order to produce a decorrelated representation of the input, (iii) *error correction* where many neurons respond to the same stimulus in a redundant way and (iv) *synergistic coding* where instead the cooperation of neurons encodes information that a single neuron cannot manage. Note that the trade-off between redundancy and error correction pervades many different biological information processes and not only neural networking.

As in the field of sequence analysis, the most important revolution in our understanding of these systems follows technical improvements from the experimental side [41]: from first attempts (i.e. single-neuron recording) the number of simultaneously recorded neurons has roughly doubled each year. Nowadays experimentalists can record the activity of many cells (from hundreds to thousands depending on the location in the brain) at the same time, and the spatial and temporal resolution with which these recordings can be done is increasing. The advent of such multi-neuron recordings has paved the way to the development of analytical tools able to model and interpret data, partially unveiling the complexity of the brain. These studies [42] [43] [1] showed that the collective behaviour of neurons in response to complex, naturalistic inputs can be quantitatively described by pairwise-based models assuming no higher-order interactions. Very recently the authors of [44] used an Ising-based analysis to show that functional connections between pairs of grid cells [1] show a peculiar connectivity with neurons with nearby phases exciting and those further apart inhibiting each other. Moreover the statistical model the authors built, allows them to explain some sources of indirect correlations as for instance overlapping fields, that could lead to spurious connections.

2.1.3 Ecology and swarming

Ecological systems are characterised by a stochastic dynamics: random genetic mutations and phenotypic changes, randomness of births and deaths, external forces such as weather or other species migrations. The result is therefore a non-trivial average dynamics and, in principle, several and accurate measurements on replicated

1. Grid cells are neurons in the medial entorhinal cortex, one synapse away from the hippocampus, whose activity lets the organism understand its position in space

systems should be needed in order to recognise a common trend [45] [46]. Such an ideal framework is rarely available when we are dealing with ecological systems. Once again advances in experimental techniques, joint to our ability to extract information from diverse ensemble averages, play a fundamental role. In this work [47] the authors reconstruct the full 3D dynamical trajectory of each bird in a flock of starlings by using a 3-cameras setup and an impressive image analysis tool. The availability of such a detailed dataset let the authors deeply understand the collective behaviour of bird flocking: note that in this case average quantities are computed on the ensemble of the different birds in the flock.

A related field in which the amount of available data enabled the development of computational studies is microbial ecology. Indeed, it is well known that micro-organisms (including viruses, bacteria, archaea and protists) form complex ecological interaction networks. The classification of all possible interactions among micro-organisms is based on a combination of win, loss and neutral outcomes [48]. Quite recently people studying such microbial ecosystems have begun to appreciate the advantages of advanced computational methods in order to predict the network of interactions among species [49]. An important step in exploring species abundance data was the identification of dependencies among the members of the communities obtained with correlations analysis [50]. However, as in many other branches of biology, tools taken from statistical physics helps unveiling interesting critical behaviours [48] promising to open the way towards the definition of new global models for the microbial ecosystem dynamics.

2.2 Application to social science

2.2.1 Sociology

As far as human interactions are concerned, recent years have seen the explosion of computational studies aiming at the understanding of interactions among people from data collected by new technologies, such as e-mail, social networks, smart phones, ad-hoc tracking technologies, etc [51]. For the past decades, network theory has been widely applied to social networks, yielding explanations for social phenomena from individual creativity to profitability [52]: e.g. the unveiling of the underlying network of interactions among people has demonstrated the person-to-person spread of obesity being one of the major factor of the obesity epidemic [53]. Beside many dynamics analysis [54], also inference techniques [55] play a fundamental role in the field providing information about both the structure and the content of relationships.

Moreover outside the academic community, the need for statistical models describing people interactions is increasing: epidemic spreading, viral marketing, default contagion are just some of the already known applications. The science of data promises to concern more and more aspects of our life. Several companies such as banks, transports, health services and public institutions have already changed their business models according to a more accurate observation and analysis of customers habits.

2.2.2 Economics

The application of sophisticated mathematical and physical methods to financial topics is not recent [56] [57]. Quantitative finance is a well established field of applied mathematics, concerning financial markets. The main goal of quantitative finance is to derive mathematical models describing observed market prices in order to predict the best strategy for future activities (e.g. buy/sell something). Beside this approach, leading to establish a link between mathematical modelling and financial theory, the study of correlated price changes of different stocks or the time series analysis has given rise to a novel discipline called econophysics [58] [59]. Econophysics has combined scientific interest and practical relevance in quantifying risks: being well known that an increase in demand should increase prices, while an increase in supply should decrease prices, the author of this paper [60] uses statistical methods in order to reconstruct all large orders on the market, making use of information about single broker transactions.

Quite recently more sophisticated inference theories inspired by stochastic matrix theory have been developed. People and company interactions turn out to be so complex that no simple rule can be established able to reproduce the observed behaviour; inference methods [61] applied to available data have anyway been proved to explain many well known phenomena occurring in financial markets.

Chapter 3

RNA structure prediction: application of an inverse Potts model

Direct-coupling Analysis (DCA) was developed in order to predict contacts between amino acids in the folded structure of proteins [62]. It is based on the assumption of coevolution between structurally related sites and it has been proved to be a powerful tool for protein contact predictions. Given that no structural information about residues is needed to perform DCA, its generalisation to other biomolecules is straightforward and within this chapter I will present a novel application: RNA secondary and tertiary-structure prediction. The secondary structure if RNAs is made by the well known Watson-Crick base-pairs, the same found also in DNA. These pairs strongly co-evolve: only three possible pairings are admitted A-U (or T in the case of DNA), G-C and G-U (called wobble pair). Therefore the covariation signal is in this case much higher than the one of other pairs in contact in the tertiary structure. Standard approaches to coevolution analysis, such as Mutual Information (MI), can predict almost only secondary structure base-pairs and just in a few cases some tertiary-structure contacts. However I will show that, differently from MI, DCA signal is enriched in tertiary contacts and it improves both secondary and tertiary-structure prediction tools.

This chapter will be structured in 6 sections. The first three will introduce the state of the art in the field: I will first focus on the analysis of known crystal structures of RNA, then I will move to comparative sequence analysis, the most powerful computational tool available to study biomolecules, and finally I will introduce some existing methods for structure prediction. The fourth section will present the pre-processing of data needed for DCA analysis, including both actual data (multiple sequence alignments) and structure for comparison and evaluation of results, and also the DCA algorithm and scoring systems. Within the fifth section, the paper *Direct-Coupling*

Analysis of nucleotide coevolution facilitates RNA secondary and tertiary structure prediction is reprinted. Finally, the last section will include suggestions about some possible improvement based on a deeper analysis of DCA outputs.

3.1 RNA structure analysis

The main source of crystal structures of proteins and other important biological macromolecules is the well known PDB database. It contains information about the known 3D structures of many proteins and nucleic acids involved in the central processes of life. Structures containing the coordinates of each atom belonging to the molecule, are computed experimentally using methods such as X-ray crystallography, NMR spectroscopy and cryo-electron microscopy. Last years have seen an important increase of the number of structures stored on this repository, however technical difficulties have till now penalised nucleic acids with respect to proteins. The amount of information contained in the PDB of a protein or an RNA is quite different: protein-PDBs contain, among the other, information about the secondary structure, while RNA-PDBs do not. Annotations about the secondary structure of an RNA can be achieved using software such as RNAView [63], MC-annotate [64] or Assemble2 [65]. These tools are able, as we will see, also to classify tertiary base-pairs.

In the following I will recall some basic concept about RNA structure and function, then I will review some of the existing methods for extracting structural information from RNA-PDB structures and finally I will talk about the state of the art in RNA structure prediction.

3.1.1 Basic concepts

Ribonucleic acids (RNAs) are the only known polymers able to both bring genetic information and perform chemical catalysis. Even if they are chemically closer to DNA, their ability to fold in complex tertiary structures and thus act as catalysts makes them structurally akin to proteins. Similarly to protein RNA structure can be described at four different levels:

- the primary structure is the sequence and it is made of four basic building blocks called nucleotides. They are made by a ribose sugar ring, a phosphate group and a *purine* or *pyrimidine* base. The most common purine bases found in RNAs are *Guanine* and *Adenine*, while *Cytosine* and *Uracil* are the pyrimidines. However some non-standard bases exist.

- The secondary structure is held together by hydrogen bonds between canonical base-pairs such as A-U and C-G, wobble base-pairs G-U, and base-stacking interactions forming the so called stems. The result is similar to the well known DNA double helix proposed by Watson and Crick in 1953.
- The tertiary structure is characterised by long-range non-canonical interactions. The existing base-pairs have been classified in [66], where the authors define a nomenclature system based on the observation that purines and pyrimidines can be schematically represented as triangles, according to the three available edges for hydrogen bonding interactions: Sugar, Hoogsteen and Watson-Crick. In the following I will call this classification *the Westhof-Leontis classification*. Each edge can interact with any other edge of an other nucleotide giving rise to a total of 12 possible geometries, including two different orientations for the glycosidic bond: *cis* and *trans*. Beside canonical Watson-Crick *cis* base-pairs, forming secondary structure, the other non-canonical pairs are mainly involved in the self-assembly of the molecule and also in RNA-protein interaction.
- The quaternary structure involves bonding with proteins or with other RNAs.

As far as folding is concerned, it has been shown that RNA folds through a hierarchical pathway, in which domains assemble sequentially [67]. First Watson-Crick base-paring and staking interactions form the double helices of the secondary structure and then the resulting molecule is packed in a compact 3D structure through the mediation of tertiary architectural motifs [68]. Also the stability of the two structures is quite different: secondary structure turns out to be highly stable contrary to tertiary structure. This difference is mainly responsible for the difficulties encountered in experimental determination of high-resolution RNA structures, making the structurally characterisation of RNAs challenging.

Besides the well known messenger RNA, bringing genetic information from DNA to protein translation, many other RNAs have been discovered to perform directly their function. These ones are called functional RNAs. The first functional RNAs that have been discovered were the transfer RNA (tRNA) and the ribosomal RNA (rRNA) always involved in protein synthesis. We know that many RNAs are found in complex with proteins (ribonucleoprotein complexes RNP) to perform crucial tasks inside the cell. Moreover there exist catalytic RNAs, called ribosymes, that together with enzymes, boost chemical reactions. Thanks to the study of its atomic structure, it has been proved that the ribosome itself is a ribozyme [69], confirming that structural knowledge is extremely important in order to access functions and to enlarge our comprehension about the cell system.

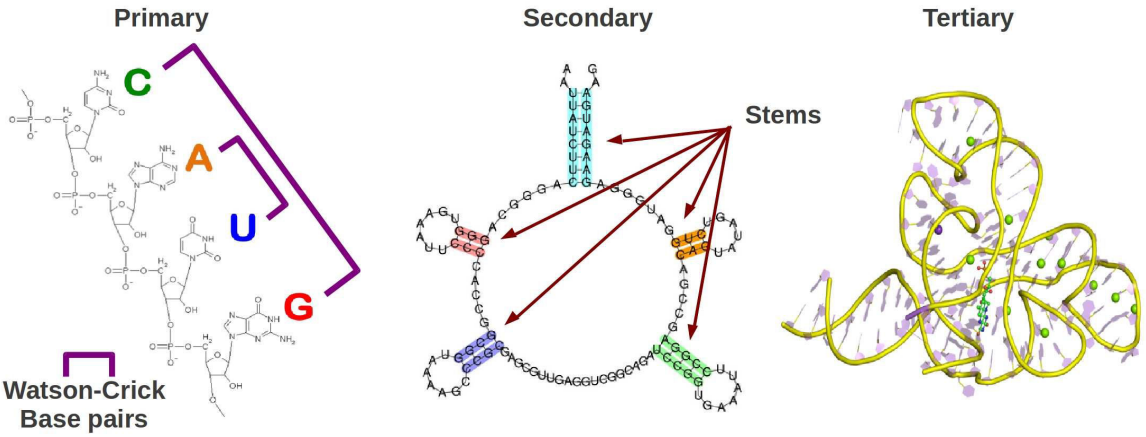


Figure 3.1: Hierarchical structures of RNA.

RNAs can also be involved in gene regulation. For instance, riboswitches are small sequence of RNA that modify their structure to bind particular metabolites. They probably have played an important role in evolution before proteins [70] have been "invented". Finally, in the last years, more and more non-coding [1] RNAs have been discovered and understood thanks to both computational and experimental tools [71] [72] [73] . Function for the most part of these ncRNAs remains unknown. Increase in the number of crystal structures available together with the development of computational structure prediction tools promises to find a map from sequence to function.

Among nc-RNAs some small RNAS, from 20 to 27 nucleotides, have been proved to play essential roles in eukaryote cells: microRNAs (miRNAs) and short interfering RNAs (siRNAs). These small RNAs are involved in a variety of phenomena that are essential for genome stability, development, and adaptive responses to biotic and abiotic stresses. Note that their mode of action does not entail a three dimensional structure but it is mainly based on linear sequence features.

3.1.2 MC-annotate

In order to understand RNA functions, software for the analysis and the visualisation of known structures plays an important role. MC-annotate [64] is a software for the analysis of PDB files. The main aim of this kind of programs is to extract

1. The distinction between non-coding RNA and functional RNA is not universally accepted. Someone refers to the two terms as synonymous, while someone prefers to consider nc-RNAs as a sub-set of functional RNAs. The solution for this nomenclature issues goes beyond the aim of this dissertation, however note that for the rest of the chapter I will refer to functional RNAs generally as RNAs.

information about nucleotides and their interactions (given a native or predicted crystal structure) and decoding PDB language that is mainly made of 3D coordinates. Outputs consist of annotated structural graphs, meaning representations of nucleic acid structures in which nodes correspond to nucleotides and single nucleotides conformation and base-base interactions are classified according to some nomenclature. MC-annotate output-file example is shown in 3.2. The classes of annotations are:

- Residue conformations
- Base-pairs, containing all the secondary and tertiary-structure base-pairs (annotated also in the Westhof-Leontis nomenclature)
- Base Triples
- Adjacent relations
- Helices
- Non-Adjacent stackings
- Strands
- Tertiary base-pairs
- Sequences containing mapping between secondary and tertiary-structure and RNA sequence

3.1.3 RNAview

RNAview [63] is a web server able to recognise and classify, according to the Westhof-Leontis classification, base-pairs given a known crystal structure. The program is designed such that the classification is made through the accurate geometrical characterisation of each nucleotide and of its position with respect to the other nucleotides. Distances, angles and type of bonds are taken into account for base-pairing annotation. Results can easily be managed thanks to graphical and text outputs containing information about both secondary and tertiary-structure.

3.1.4 Assemble2

The last program I will introduce is Assemble2. It is and interactive graphical tool for the analysis of 3D and 2D RNA structures. Given a PDB input it annotates secondary and tertiary-structure base-pairs on a 2D interactive and modifiable picture of the RNA. Also in this case the Westhof-Leontis notation is used. The main advantage of Assemble2 is that very complex structures can be easily manipulated

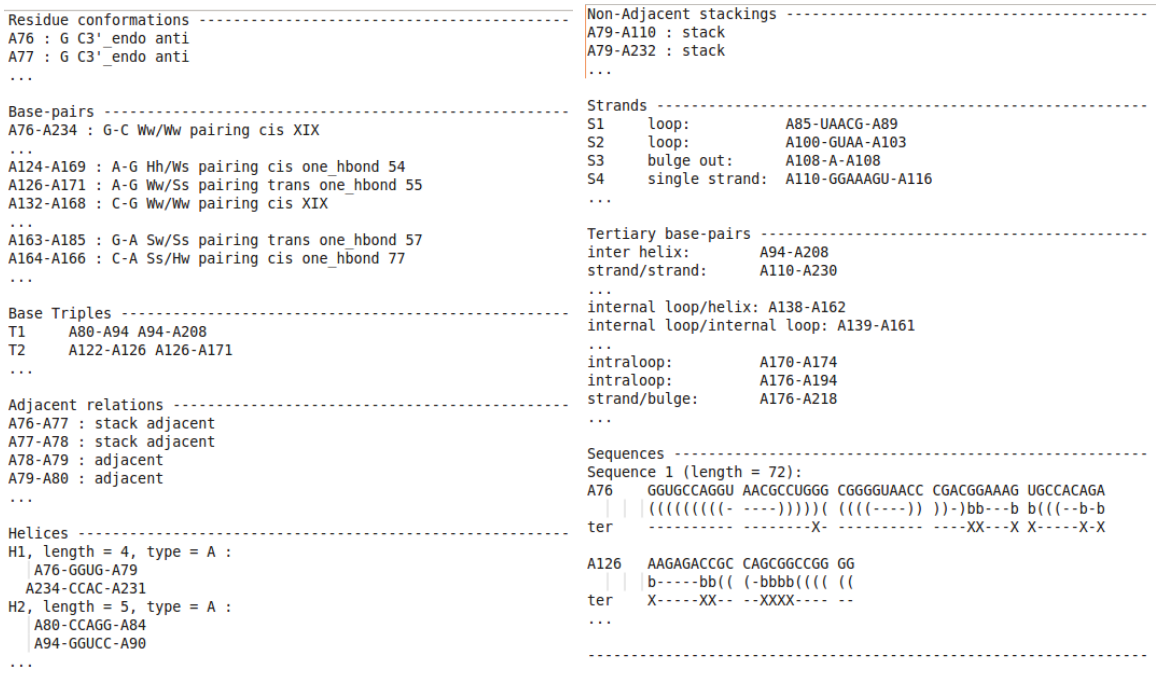


Figure 3.2: MC-annotate output file. In this example only minimal information is shown to fit space. However all the classes of annotations made by the program are shown.

and visualised. Unfortunately the enumeration of the sites is different from the one in PDB and so some difficulties in comparing results arise.

3.1.5 The distances between nucleotides

A more naive way to analyse RNA structures is to look at distances between nucleotides. Among all possible definition for nucleotide-nucleotide distance, depending on the atoms that have been considered for measuring (quite common is the C1'-C1' distance), we choose to look at the distance between the closest heavy atoms. This kind of choice force us to intend the contact between two nucleotides for as proximity relation, differently from the more sophisticate analysis performed by the software I described above, where contact means a real physical bond existing between bases.

However having a precise definition for the distance does not solve the whole problem: which is the distance of two sites in contact? The simple thing to do is to choose a cut-off and define as being contacts those pairs in which nucleotides are closer than the cut-off. However the choice of whatever threshold is definitely not trivial. Protein structure prediction literature [74] proposes two solutions (4Å or 8Å) based on the distribution of distances between amino acids in many protein families. We have performed the same analysis on 20 RNAs whose structure is known with a

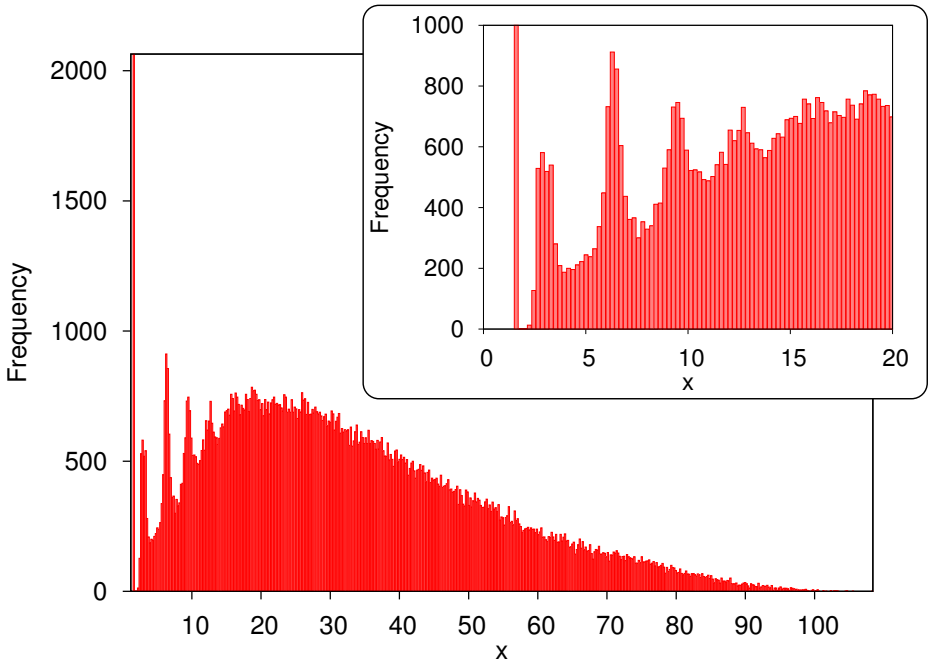


Figure 3.3: Frequency counts of the distances between nucleotides in 20 families whose structure is known. X is the distance between the closest atoms measured in Angstroms. Inset: zoom on the first 20Å.

sufficient (less than 3Å) resolution. As you can see in figure 3.3 the histogram shows some peaks inside the range from 2 to 20 Angstroms. Before that, at around 1.6Å, the very high and sharp peak corresponds to backbone contacts ($|i - j| = 1$). Moreover Fig. 3.4 shows histogram of distances of those pairs found by RNAView and classified according to Westhof-Leontis classification. Then, the first peak located from 2.5Å to 4Å includes both Watson-Crick base-pairs, whose typical distance is 2.7Å, a few stackings and non-canonical base-pairs.

Note that all the characterised interactions are found closer than 4Å and thus they are in agreement with this choice for contact definition. Moreover, among all possible pairs of nucleotides in the analysed dataset, the following percentages have been found:

- 4% of pairs are closer than 4Å
- 10% of pairs are closer than 8Å
- 0.5% of pairs are recognised as non-canonical base-pairs
- 0.5% of pairs are recognised as canonical base-pairs
- 0.1% of pairs are recognised as stacking interactions

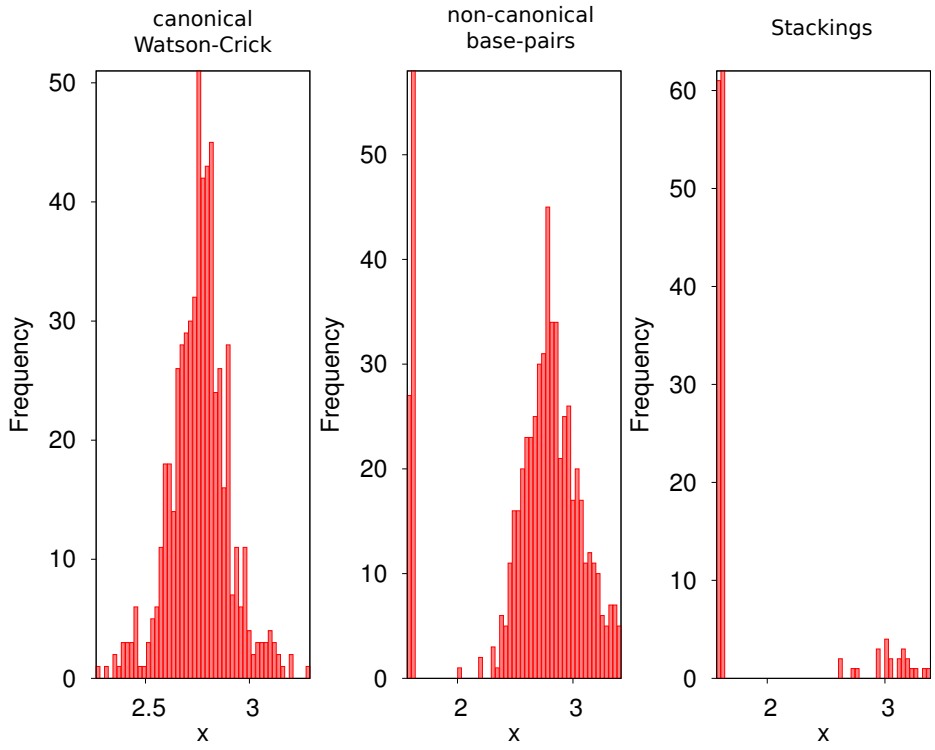


Figure 3.4: Frequency counts of the distances between nucleotides classified according to RNAview software. X is the distance between the closest atoms measured in Angstroms.

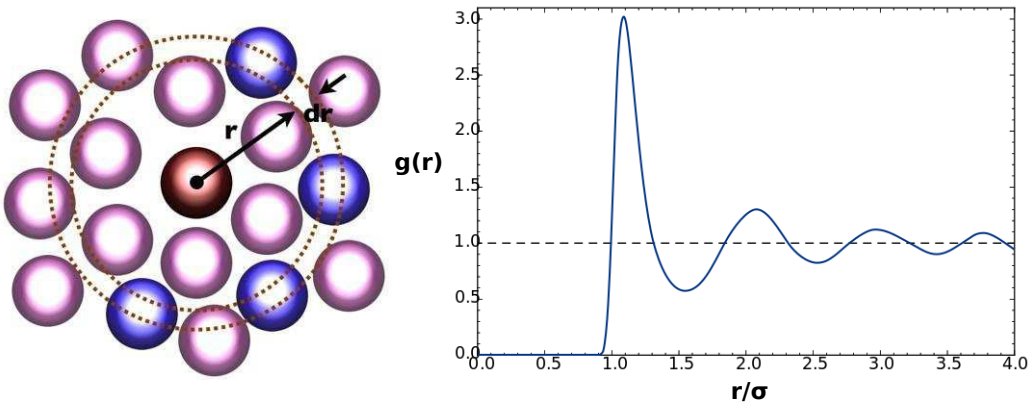


Figure 3.5: Radial distribution function for a liquid g(r), where r is the distance between molecules and σ is the diameter of the diameter of molecules.

So if all the classified interaction are closer than 4Å a question spontaneously arises: what about the other peaks in Fig. 3.3? Statistical mechanics suggests us an answer: the radial distribution function of a dense gas or liquid [75]. This density function measures the probability for molecules in a gas, a liquid or a solid or even a polymer to be at a given distance one from each others. Intuitively, consider a certain molecule, its volume constrains the nearest neighbour particles to stay at least at a distance equal to its diameter. Fig. 3.5 shows on the left a picture explaining such a mechanism and, on the right, an example of the resulting radial distribution function. In this example we consider the simpler case of a liquid, however, even if RNA is a polymer, we would not expect this picture to dramatical change and we can thus explain the peaks seen in the distribution of nucleotide-nucleotide distances with this well known model.

3.2 RNA comparative sequence analysis

Beside structural knowledge, a powerful tool used for understanding biomolecules functions is the search of homology and comparative sequence analysis. Homologous sequences are defined as having a common ancestor in evolution and are characterised by a conserved structure and function. The level of nucleotide conservation varies from RNA to RNA and also from region to region inside the same RNA. Note that the sequence conservation versus functional importance does not hold so well in RNA, since 2D structure is frequently well conserved and plays an important role even if nucleotides have high entropies. Differences we observe between homologs have accumulated since the speciation due to random mutation of nucleotides. Constraints

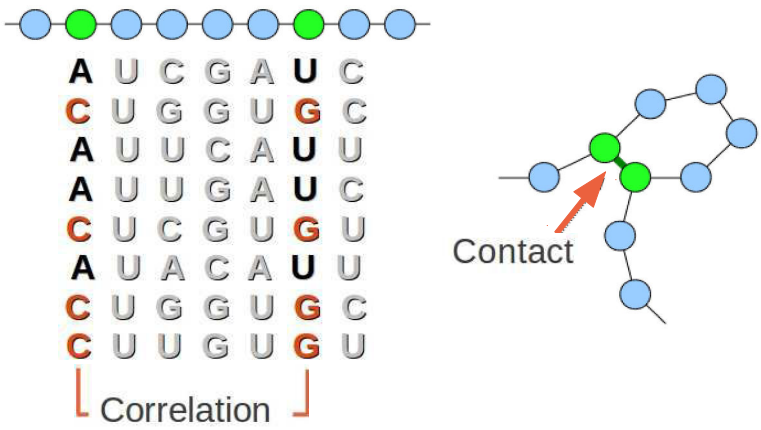


Figure 3.6: Cartoon of a MSA and the underlying conserved structure. The presence of a contact in the three dimensional structure of a protein, or an RNA, gives rise to a correlation between the involved sites.

linked to functionality biased the probability for mutation to occur, because organisms showing mutations that negatively affect RNA behaviour are less likely to survive during evolution. Indeed, remember that sequences of the organisms we observe today have been selected under evolutionary pressure.

In order to compare homologous sequences it is necessary to juxtapose them such that residues descendent from the same ancestor stays in the same column. Gaps are inserted to align sequences whose length is changed during evolution. The most common way to align sequences is based on conservation: we want conserved nucleotides to be aligned. Often compensatory mutations can occur giving rise to the so called coevolution: Fig. 3.6 shows a fake multiple sequence alignment (MSA) and the cartoon of the underlying conserved structure. Consider two nucleotides interacting in the three dimensional structure of the ancestral RNA. During evolution, they may mutate, but only the RNAs in which the mutated sites are still in contact will be functional. This mechanism has been shown to work for Watson-Crick base-pairs [76] and can be used to obtain reliable structural alignments.

The main source of MSA for RNAs is Rfam. Being at the 12.0 version [77], this database contains multiple sequence alignments of RNAs obtained with a software, called *Infernal* [78], based on Covariance Models [79].

	A	C	G	U
A	2	-1	-1	-1
C	-1	2	-1	-1
G	-1	-1	2	-1
U	-1	-1	-1	2

Table 3.1: Example of substitution matrix.

In the following sections I will review first two well known algorithms for the alignment of two sequences (they are not RNA-specific, but are used also for protein sequence alignment), then I will move to MSA and explain the main features of Hidden Markov Models (HMMs) and Covariance Models (CMs) and finally I will sketch Rfam database functionalities focusing on the aspect of interest for RNA structure prediction.

3.2.1 Needleman-Wunsch: global alignment

The alignment of two sequences consists in creating a nucleotide to nucleotide mapping between them inserting, if it is the case, gaps when there is no matching. Needleman-Wunsch (introduced here [80] and improved here [81]) was the first algorithm to be developed for the alignment between two sequences. The basic idea of this algorithm is to build an optimal alignment from optimal alignments of sub-sequences. It consists in two steps: first we compute a L_a by L_b score matrix $\mathbf{F}$, where L is the length of the sequence, and then we trace-back in the matrix, looking for the optimal path.

$\mathbf{F}$ is defined as follows:

$$F_{ij} = \max \begin{cases} F(i-1, j-1) + s(x_i, y_j) \\ F(i-1, j) - d \\ F(i, j-1) - d \end{cases} \tag{3.1}$$

where s is called substitution matrix and assigns a different score depending on the two nucleotides we want to match. d is the cost of a gap. The easiest substitution matrix we can think of is shown in Table 3.1: it forces the algorithm to look for the alignment entailing the highest number of matches. In their work [80] authors used a similar substitution matrix, however a large amount of subsequent works have introduced more accurate scoring systems based on observation of actual rates of mutation or on chemical differences between nucleotides.

3.2.2 Smith-Waterman: local alignment

Needleman-Wunsch algorithm is a *global* alignment method, meaning it aligns sequences considering their full length. For instance, in case the length of two sequences differs a lot, this algorithm may fail and spread out nucleotides of the shorter sequence along the longer one. Therefore if just a subset of the two sequences matches, a different algorithm has to be used: *local* algorithms. The Smith-Waterman [82] algorithm belongs to this class and it returns the optimal alignment of any sub-sequences of the two sequences we want to align.

F of the Smith-Waterman local alignment is defined as follows:

$$F_{ij} = \max \begin{cases} 0 \\ F(i-1, j-1) + s(x_i, y_j) \\ F(i-1, j) - d \\ F(i, j-1) - d \end{cases} \tag{3.2}$$

Note that Smith-Waterman algorithm back-traces from the highest entry in the F matrix until it hits a zero score. Moreover, the zero entries in F let us start the alignment from whatever site and find the highest scored sub-alignment.

3.2.3 Profile Hidden Markov Models

One of the limitation of the algorithms introduced above, is that they use the same scoring systems disregarding the considered position inside the sequences. It is clear that matching the first or the last positions is not as important as matching the core of sequences. Establishing the start and the end point for a certain gene is definitely not trivial and sequencing errors often occur. Therefore, position-specific scoring systems have been introduced. The most powerful among those are profile Hidden Markov Models [83]. Profile HMMs are probabilistic models based on an hidden chain of states that emit the symbols we observe. The full characterisation of an HMM implies the computation of transition probabilities from state to state based on the statistic of the observed symbols. Generally speaking a profile (introduced here [84]) is the statistical description of MSA based on the frequency of symbols in each single if its column. Formally, a HMM is specified by the following two properties:

- the path is Markovian and the chain is represented by transition probabilities a_{kl} between states k and l

$$a_{kl} = P(\pi_i = l | \pi_{i-1} = k) \tag{3.3}$$

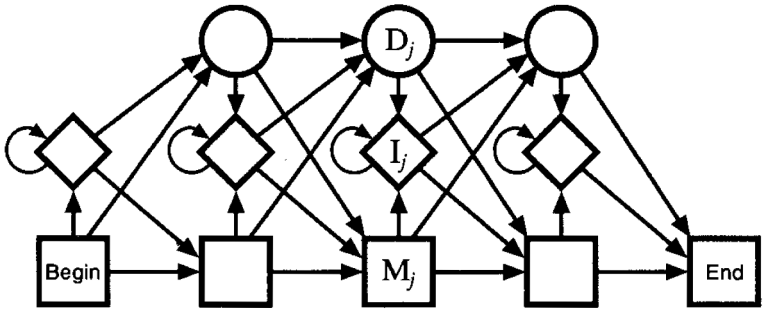


Figure 3.7: Sequences of proteins belonging to the same family can be aligned using a Profile Hidden Markov Model. From a subset of sequences already aligned the parameters of the model are inferred (the transition probabilities, i.e. the probability, in correspondence of each site, of the emission of one particular amino acid, of the opening of a gap, of the deletion of a part of the original sequence etc.). Then for each new sequence the alignment is given by the path maximizing the emission probability for that particular string of symbols (picture taken from [83])

– in each state π_i the visible symbol x_i assumes one of the possible values according to the correspondent emission probability $e_k(b)$

$$e_k(b) = P(x_i = b|\pi_i = k) \tag{3.4}$$

where π is the sequence of the states, π_i is the i^{th} state in the path and x_i is the symbol emitted by the i^{th} state. Is it then clear how a Hidden Markov Model can be used to align a new sequence to the subset already analysed: the parameters of the model (i.e. the probabilities of passing from a state to another one) are estimated from the previously aligned sequences, the residues being seen as the visible outputs (match state M_j) whereas the possibility either of the opening of a gap in the new sequence with respect to the others (insertion I_j) or of the removal of a part of the sequence (deletion D_j) are represented as hidden states (cf. Fig. 3.7). In order to define the probability for a sequence of states to emit a particular sequence of symbols, finally, the emissions of amino acids given the hidden states are assumed to be conditionally independent from each other:

$$P(a_1, ..., a_N|x_1, ..., x_N) = \prod_{i=1}^{N} e_i(a_i|x_i) \tag{3.5}$$

Each sequence is not univocally connected to a path in the HMM as many of them may generate the same sequence. The last part of the alignment consists then

in finding the path maximizing the probability of obtaining the considered sequence given the transition probabilities of the model.

Even if profile HMM have been successfully used for protein MSA they cannot be adapted to RNA modelling because they cannot take into account base-pairing. However from the same family of probabilistic models good substitutes can be found: Covariance Models

3.2.4 Covariance Models

Covariance models are a generalisation of profile HMMs developed for modelling RNA sequences. While profiles HMM are developed on a unidimensional chain, a CM is built on a tree, called *guide tree*, whose nodes closely corresponds to the consensus secondary structure of the aligned RNAs. In Fig. 3.9 a schematic representation of a CM is described. Guide trees are made of 4 different types of nodes (cf. to panel B in the picture):

- a unique ROOT node showing the starting point for the structure
- 3 different nodes for matching: MATP for a matching pair, MATL and MATR for a left or right single-stranded residue matching
- a bifurcation node BIF
- two root nodes for the beginning of a new left (BEGL) or right (BEGR) stem

The emission and transition probabilities of CMs are set the same way as HMMs (see Eq. 3.3 and Eq. 3.4). To build the guide tree and parametrise the model an annotated alignment and its consensus secondary structure are needed. The latter being well-nested, thus not including triples of bases or pseudo-knots [2].

Given a parametrised CM we can use it for homology detection via the equivalent of the Viterbi and Forward algorithms for HMM: the Cocke-Younger-Kasami (CYK) [85] and the Inside algorithm [83]. In addiction to the homology score, CYK algorithm determines also the most probable parse tree for a given sequence assigning to each nucleotide a position within the consensus secondary structure. Therefore the alignment of sequences according to the model consists in aligning the sparse trees and then in converting matches between states on the tree in matches between nucleotides in the same column.

2. A well-nested structure is such if two pairs $i-j$ and $k-l$ with $i < k < j < l$ do not exist. When this rule is violated we say the structure presents a pseudo-knot. A pseudo-knot occurs when there are some base-pairs between a loop and positions outside the enclosing stem as in Fig. 3.8

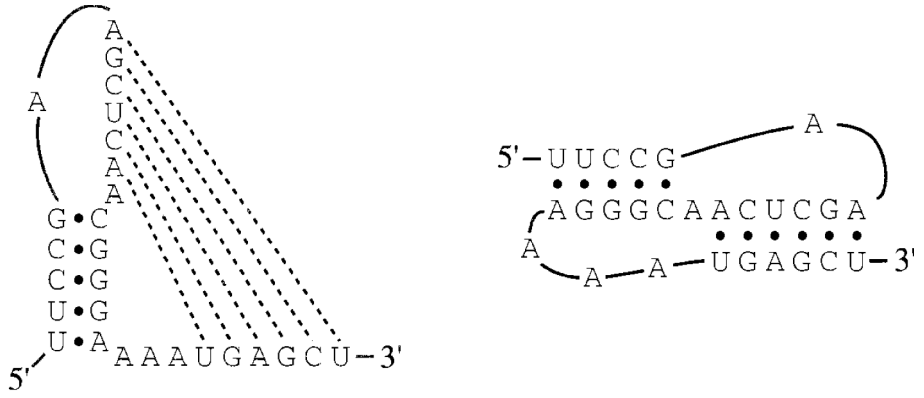


Figure 3.8: An example of secondary structure with pseudo-knots (picture taken from [83]). Cf. with the well nested secondary structure in Fig 3.1

3.2.5 Infernal and Rfam

Infernal [78] is the software used to build the MSAs found in Rfam database. It uses CMs to search nucleic acid sequence databases for homologous RNAs, or to create new secondary-structure based multiple sequence alignments. Before searching for homologous through the CM a first BLAST filter is applied. This step is necessary because CM are computationally expensive and they cannot be run on the full database. Recently the new version of Infernal 1.1 [87] has overcome this difficulty using a new filter pipeline based on accelerated profile Hidden Markov Model methods and HMM-banded CM alignment methods. Results are quite impressive and the software can search for homologi 100-time faster than before.

These recent advances allow the release of a new Rfam 12.0 [77] including more families, more accurate and bigger, than the ones released with Rfam 11.0. Unfortunately, given the huge size of some family, the full Stockholm alignment is no more available for download. Only the seed alignment can be obtained and then users have to run Infernal by themselves. Note that, as many of the analysis I will treat within this dissertation were performed before the release of Rfam 12.0, all the results showed refer to Rfam 11.0 alignment. More reliable MSAs would have improved inference results: in the following we will see, in fact, that inference methods are quite sensible to the alignment quality.

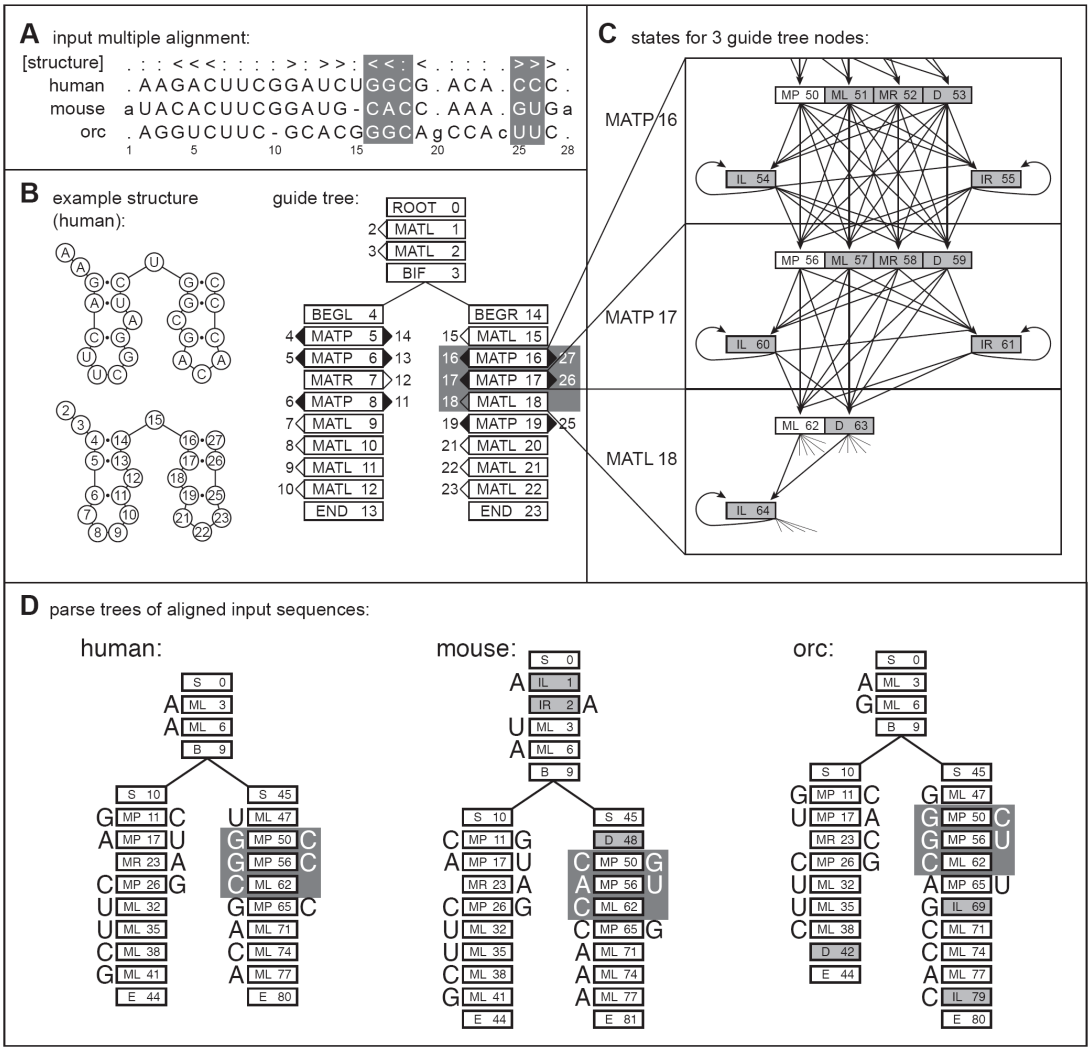


Figure 3.9: Scheme of the steps needed to build a MSA with CMs. (A) An input multiple sequence alignment with annotation about the consensus secondary structure is needed. (B) The guide tree is shown. It is based on the consensus structure. (C) The transition probability network needed for parametrisation. (D) Parse tree of sequences. When the CM is used to align sequences first the sparse trees of each sequence are computed, then they are aligned one to each other and finally nucleotides belonging to the same node are placed in the same column. Note that guide tree is computed from the initial alignment, while parse trees are computed from the model to align sequences in the final MSA. Thus, even if for this toy model sequences in panel A are the same as in panel D, in real-life problems they are supposed to be different. This picture is extracted from [86]

3.3 RNA secondary and tertiary structures prediction

During the last few decades a huge effort has been done for the development of tools able to predict RNA secondary and tertiary structures. Even though experimental methods have improved, the most part of known functional RNAs remains structurally unresolved and often also functionally unresolved. Advances in structure prediction tools have shown it is possible to build reliable computationally determined structures that can be used within probing experiments [88]. Several tools have been developed since now, many of them have been reviewed here [89].

In this section I will explain the problem of RNA structure prediction. Firstly I will describe two secondary structure prediction models, then I will rapidly redraw the picture recently emerged from RNA 3D prediction competition RNA-puzzle round 2 [90].

3.3.1 Secondary structure prediction

Approaches for RNA secondary structure prediction vary widely: the most sophisticated available tools are based on free energy minimization algorithms. They were originally introduced by Zuker [91]. Free energy minimisation algorithms are based on the observation that the best structure would be the one with the lowest equilibrium free energy ΔG. The major limit of these methods is that they need experimental knowledge about the magnitude of the actual interaction between base-pairs [92] and often these data are not precise enough. Also comparative sequence analysis plays a role in this field [93] [94].

We know that even if sequences can change a lot, the secondary structure is often well preserved thanks to compensatory mutations. Sequences, far away from each other in term of evolution, are very difficult to be correctly aligned: the best methods for multiple sequence analysis are based on profiles and the accuracy of such methods decreases with increasing divergence. Structural information helps obtaining better alignments, but good structural predictions often rely on good alignments. This scenario opens to the development of iterative approaches as the one introduced here [79] and inspired by a generalised version of the Nussinov algorithm [95].

Nussinov Nussinov algorithm is a dynamic programming algorithm able to efficiently predict the optimal secondary structure for a RNA sequence. It is a recursive

algorithm based on the idea that, given an optimal sub-structure, there are only 4 possible ways to obtain a longer sub-structure:

- adding a left single site
- adding a right single site
- adding a base pair
- linking two optimal substructures

The predicted structure is optimal in the sense that it maximises a certain score: in the original version of the algorithm, taking as input only one sequence, the number of base-pairs along the structure was maximised. Nevertheless, using information from a MSA it is possible to compute a covariation score for every pair of sites and then use it for maximisation. A very well known score able to estimate compensatory mutation events is Mutual Information (MI).

$$MI_{ij} = \sum_{A_i, A_j} f_{ij}(A_i, A_j) \log \frac{f_{ij}(A_i, A_j)}{f_i(A_i) f_j(A_j)}$$

Coming from information theory, MI tells us the gain in information we have in considering two sites together instead of separately. In Eq. 3.3.1 MI definition is shown, where f_i and f_{ij} are the single site frequency counts and the pair frequency counts computed from MSA (cf. Eq. 1.2). In the generalised Nussinov MI is used. Thus the optimal score of the subMSA of columns from i to j, S_{ij}, is defined as follows:

$$S_{ij} = \max \begin{cases} S_{i+1,j} \\ S_{i,j-1} \\ S_{i+1,j-1} + MI_{ij} \\ \max_{i<k<j} S_{i,k} + S_{k+1,j} \end{cases} \tag{3.6}$$

As other dynamic programming, once matrix S_{ij} is computed, Nussinov algorithm makes use of a trace-back procedure to look for the optimal path giving rise to the best secondary structure for the considered sequence.

RNAalifold RNAalifold is a software for the prediction of the secondary structure of RNA combining free energy minimisation and covariation analysis. It includes a covariation term in the folding energy such that compensatory mutations are taken into account for the evaluation of the energy of any sub-structure. Note that also free energy minimisation based models follow the same recursive procedure of the Nussinov

algorithm: they build a longer optimal sub-sequence given the optimal sub-sequence available. In a latest version of the algorithm [96] the covariation score is computed thanks to a modified version of the statistically defined substitution matrices called RIBOSUM, introduced here [97] in order to improve the homologous research. RIBOSUM matrices give the log-odds ratio for observing a given substitution relative to background nucleotide frequencies and are defined for both single nucleotides and base-pairs.

3.3.2 Tertiary structure prediction

Predicting secondary-structure of RNAs is a crucial issue in the field and many solutions already exist. Even though, the knowledge of the secondary structure give us a blueprint of the RNA molecule, it is often not enough for a fully functional characterisation. Several methods for tertiary structure prediction have been developed. However high quality results are till now restricted to small sequences consisting of simple helices and small loops. When more complex structures are concerned, the reliability of the structure depends on experimental information available about interactions between nucleotides in the molecule. To probe the state of the art a CASP(Critical Assessment of protein Structure Prediction)-like experiment has been performed [98] in 2012 and [90] in 2015. This kind of world-wide experiments, known as RNA-puzzles, let groups developing software and pipelines for 3D RNA predictions compete on hidden known structures. The sequence of the target structure is given to each group, plus some additional experimental information about, and the aim is to predict a tertiary structure as close as possible to the hidden crystal structure.

Last RNA-puzzle competition has involved seven research groups. Three target structures have been proposed and the best results are characterised by root-mean-square deviations (RMSD) of atomic positions range between 6.8 and 11.7 Å[3] and all display predicted structures topologically akin to native ones. If we compare this results to what is nowadays reachable in the related field of proteins, it seems to be quite modest. However, given the size of the target sequences (>160 nucleotides),

3. Results on three different RNAs have been reported.
   - The lariat-capping ribozyme: 24 structures submitted, average RMSD 24.05, standard deviation 4.91
   - The adenosylcobalamin riboswitch: 34 structures submitted, average RMSD 23.09, standard deviation 6.87
   - The T-box–tRNA complex: 26 structures submitted, average RMSD 11.52, standard deviation 2.87

results show a positive trend for RNA structure predictions. The best structures predicted within this collective experiment have been obtained by Das group, who provides also to each group tertiary contact information obtained with a mutate-and-map strategy based on systematic mutagenesis experiments and high-throughput chemical mapping [99].

Rosetta Rosetta is a de novo [4] approach for 3D structure prediction developed by Das and Backer [100]. It was initially introduced in the related field of proteins [101] and then generalised to different macromolecules. Rosetta consists in a fragment assembly of RNA (FARNA) guided by a knowledge-based energy function taking into account experimental knowledge on backbone conformation and side-chain interactions. The fragment library includes fragments made of 3 nucleotides extracted from the large rRNA subunit. Once interesting fragments are selected, a Monte Carlo routine is run to assemble them into a native-like folded structure. The main feature of Rosetta is that it lets us include structural knowledge such as secondary structure or even tertiary interactions. Moreover it has been proved that such information can dramatically increase the quality of the prediction [102].

3.4 A new approach to prediction: DCA

Having in mind the state of the art for RNA structure prediction, the application of DCA to such a problem seems to be straightforward. An urgent need for supplementary information in order to correct 3D folding emerges from RNA-puzzle and opens new scenarios: till now MI has not been able to substantially help 3D prediction and only experimental information have done the job. Can new and more sophisticated approaches to statistical inference of interactions from MSA face this challenge? This is the question my thesis will try to answer and roughly speaking the answer is: "Yes, they can, results are promising but till now modest". I will show in the following that, differently from protein, the DCA signal obtained from RNA MSA shows a multi-scale complexity opening to possible post-processing improvement procedures. The development of these procedures is not mature yet and needs more theoretical efforts for a better interpretation of the signal. I am anyway confident

4. de novo is to be intended in the sense that any information other than the sequence is needed for folding. However in the specific case of RNA information about secondary structure or tertiary interaction can be introduced in the routine and turns out to be crucial to obtain good quality predictions.

that the results I will show in the next part of this chapter will be an interesting starting point for further researches.

In this section I will describe the full prediction pipeline: from pre-processing of input alignments and structure for comparison to post-processing of the output signal. The latter corresponding to an unpublished effort to better understand DCA signal and its complexity with respect to structural knowledge.

3.4.1 The comparison with the alignment

The main interest in structure prediction is homology modelling: finding a homologous sequence with structure, and modelling an unknown sequence using that structure as a template. This is not very successful in RNA due to the low number of families with exemplary structures. Therefore computational approach for structure prediction are needed. A key point in order to test methods for the prediction of three dimensional structure of RNAs is the comparison with known native structures. When the prediction is performed for biological interests, the input of the process is the target sequence, as in the RNA-puzzle competitions. Then, depending on the type of analysis one would perform, other sources of information can be used. Within comparative sequence analysis the first step is to search for homologous sequences, then sequences are aligned and, only when a reasonable MSA is available, the prediction can be performed. The problem I'm facing within this dissertation is slightly different: I want to develop and test the performance of a new tool. At date DCA cannot be used for homology detection nor multiple sequence alignment, so the very input of my work are MSAs obtained with the methods I've described in the previous sections.

As we saw above, alignments and structures come from different databases. Rfam gives, for each family, the PDB id of some available structures, however the latter ones may not be of the same length of the sequences in the alignments. They usually don't cover the full length of the alignment or, otherwise, they include some engineering needed for the realisation of the crystal structure. Moreover some PDBs include proteins in complex with the RNA or multiple chains. Therefore, to avoid all these issues, we analyse available PDB files for the considered family, we take the sequence of the chosen structure and then we compare this sequence and the alignment in order to obtain a map between them.

The most efficient way to find a map between the alignment and the structure is to align them. Usually Rfam tells us the name of the species whom the crystal structure in PDB belongs to, thus it is in principle possible to find the corresponding

sequence and to align it to the PDB file's one. Unfortunately this procedure needs a direct human contribution in reading the names, interpreting them etc. In order to automatise the procedure we align the PDB sequence to every sequence in the MSA and keep the sequence whose alignment has the best score. This one is the sequence with the larger number of matches with the PDB structure sequence.

The score depends on the algorithm used for the pairwise alignment. We tested 2 algorithms: a global (Needleman-Wunsch) and a local (Smith-Waterman) pairwise alignment method. The advantage of a local algorithm instead of a global one depends on the differences between the sequences to be aligned: if the two sequences are very different in size, the local alignment gives better results. Since many observations on our dataset have shown that the local alignment algorithm produces alignments with the smallest number of gaps in the shortest sequence, we have included in our pipeline the Smith-Waterman algorithm.

3.4.2 PDB - RFAM gold standard

We performed our analysis on the release 11.0 of Rfam. All the 51 available families with more than 200 sequences and annotated corresponding structures in PDB have been studied. From the analysis of the structures we have found that only 40 of them were high quality X-ray structures with a resolution smaller than 4Å, thus we discarded the others in order to increase the reliability of our results in term of comparison with the native structure. Tab. 3.2 shows the full list of those families, fitting our minimal quality requirements (> 200 sequences and < 4Å resolution), for which a good mapping between PDB and Rfam is possible.

We have encountered some technical issues regarding input files: some PDB or FASTA files were too big for launching analysis on desktop machines or some Stockholm files were broken. Moreover some families has a very simple structure, made of a single hairpin loop, with no interesting tertiary contacts. Dimers have also been excluded from the final list. Further analyses have revealed that some of these families, even if they seemed showing a good matching between alignment and structure, actually mismatch if we consider the consensus secondary structure. This very special issue can be a sign of bad alignment probably due to the presence of sub-families of sequences. The magnitude of this kind of errors can vary a lot depending on the number of sites involved and also on their position in the native structure. Sometimes we observe the consensus secondary structure to be predicted on sites that are not close in the native structure. Although smaller displacements can also occur within sites being actually close and thus it is impossible to see a priori these errors from the

Family	M_{eff}	PDB ID	Chain(, Residues)	comment
RF00002		1k8a	A	broken gz file (RFAM)
RF00005		2csx	C	broken gz file (RFAM)
RF00001	57991.33	3cc2	9	included in the Gold Standard
RF00028	7922.93	1hr2	A	probably bad quality alignment
RF00029	3339.51	1kxk	A	hairpin loop
RF00163	3009.92	2oeu	A	included in the Gold Standard
RF01118	498.83	2j01	A, 2166-2273	hairpin loop
RF00017	8145.33	1l9a	B	included in the Gold Standard
RF00059	3347.9	2gdi	X	included in the Gold Standard
RF00015	3728.6	2ozb	C	hairpin loop
RF00061	61.92	3t4b	A	hairpin loop
RF00177		2vqe	A, 1-1519	too big (PDB)
RF01959		1vs5	A, 28-1537	too big (PDB)
RF00504	1828.54	3owi	A	included in the Gold Standard
RF00010	2309.67	1u9s	A	included in the Gold Standard
RF00023	2143.3	4abr	Y, 42-89	included in the Gold Standard
RF00169	1318.68	2xxa	F	hairpin loop
RF00162	1165.56	2gis	A	included in the Gold Standard
RF00050	1045.85	3f2q	X	included in the Gold Standard
RF00175	13.64	1nlc	A	hairpin loop
RF00037	224.59	3snp	C	hairpin loop
RF02001	636.43	3bwp	A	included in the Gold Standard
RF00167	588.88	1y26	X	included in the Gold Standard
RF00168	552.38	3dil	A	included in the Gold Standard
RF01051	983.21	3irw	R	included in the Gold Standard
RF01852	340.63	3rg5	A	probably bad quality alignment
RF01998	459.8	4ds6	A	probably bad quality alignment
RF00380	206.7	2qbz	X	included in the Gold Standard
RF00011	215.14	2a64	A	probably bad quality alignment
RF01734	532.03	3vrs	A	included in the Gold Standard
RF00522	106.59	3k1v	A	included in the Gold Standard
RF00234	259.49	2gcs	B	included in the Gold Standard
RF00524	107.66	3u5d	1, 2174-2229	hairpin loop
RF00618	288.94	3siv	C	dimer
RF00164	28.35	1xjr	A	hairpin loop
RF01831	192.15	3suh	X	probably bad quality alignment
RF00094	4.77	1sj3	R	included in the Gold Standard
RF01960	111.82	4a18	1,334-389	too big (FASTA)
RF01857	154.61	1lng	B	included in the Gold Standard
RF01786	108.33	3q3z	A	included in the Gold Standard

Table 3.2: Table showing the list of families for which a good match between Rfam and PDB is found. The upper part of the table contains families with more than 1000 sequences in the alignment, while the bottom part those families with less than 1000 sequences.

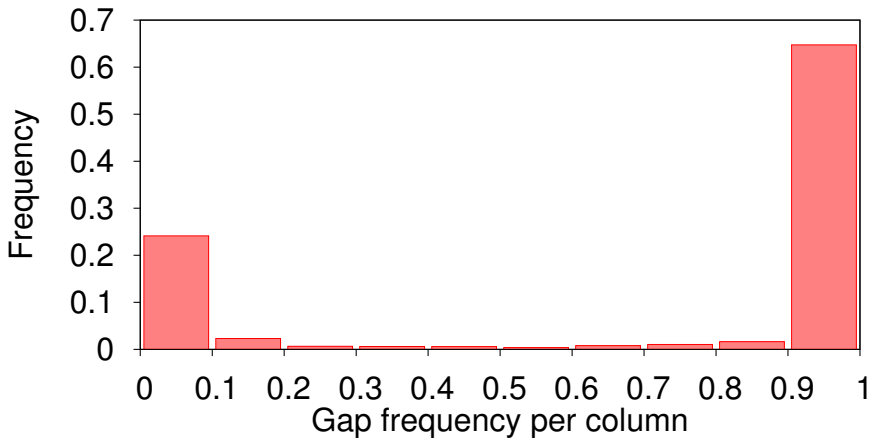


Figure 3.10: Frequency counts of the fraction of gaps in each column of the alignment. Data contain columns from the Gold Standard alignments.

comparison between the native structure and the consensus secondary structure in the alignment. An a posteriori analysis on predictions is indeed needed to understand if false positives can depend on bad alignment of sequences or not. In the next sections I will refer to those families selected according to what said above as the *Gold Standard*. It includes 20 families and all the analysis showed within this chapter has been performed on this restricted list of families.

3.4.3 Pre-processing of the alignment

As we saw above, Rfam alignments are made with Infernal software and, differently from the equivalent software for proteins (Hmmer), it does not give us information about the origin of gapped region. However, if we analyse the number of gaps per column, we obtain the histogram in figure 3.10.

This means that the 90% of the columns is either a site with almost no gaps or it has almost only gaps. Thus we can argue that those sites with a lot of gaps are insertion sites and could be removed from the alignment losing no fundamental information for structure prediction. The choice of a precise threshold is arbitrary: we choose to fix the threshold to 50% of gaps, taking care that no secondary structure sites have been lost.

3.4.4 Removing phylogenetic bias

From MSAs we can compute the frequency per site of each nucleotide. However, MSA sequences set up a biased sample of all possible sequences since the species are

evolutionarily related. In order to reduce this bias we cluster sequences according to similarity and we assign them a *weight* equal to one over the number of sequences in the cluster. We can adjust this re-weighting choosing the percentage of similarity needed to insert two sequences in the same cluster, e.g. if we fix the similarity to 100% we will assign a weight $1/l$ to l identical sequences. The value of the similarity we use for the analysis is 90% and it has been empirically chosen. Note that results are quite robust with respect to the re-weighing threshold.

3.4.5 Direct Coupling Analysis: a brief recall

Mutual Information can measure correlation between two nucleotides, but it cannot distinguish correlations coming from a direct coupling and correlation coming from indirect relations. For instance, two sites in the sequence could be correlated if there exists, for both of them, a coupling with a third site. This *indirect* correlation cannot be distinguished from the direct ones by Mutual Information. In order to disentangle the two effects we need to focus on *couplings* instead of correlations: this is the aim of Direct Coupling Analysis.

We consider the sequences in the MSA as sampled from a global statistical model $P(A_1, ..., A_L)$, where each A_i represents the nucleotide at site i and L is the length of the sequence. We want this model to reproduce the empirical counts f_i and f_{ij}:

$$\sum_{\{A_k|k\neq i\}} P(A_1, ..., A_L) = f_i(A_i) \quad ; \qquad \sum_{\{A_k|k\neq i,j\}} P(A_1, ..., A_L) = f_{ij}(A_i, A_j) \tag{3.7}$$

Eq. 3.7 guarantees coherence of data and model up to the level of pair correlations. Finally, as we have seen in the fist chapter, we apply the maximum entropy principle and we obtain a q-states Potts model, being $e_{ij}(A_i, A_j)$ the coupling between nucleotide A_i in site i with nucleotide A_j in site j and $h_i(A_i)$ the field due to the presence of nucleotide A_i in site i.

$$P(A_1, ..., A_L) = \frac{1}{Z} \exp\left\{ \sum_{i<j} e_{ij}(A_i, A_j) + \sum_i h_i(A_i) \right\} \tag{3.8}$$

In mean-field approximation an immediate relation between the couplings $\mathbf{e}$ and the connected correlation matrix $\mathbf{C}$ can be found:

$$e_{ij}(A_i, A_j) = -((C^{emp})^{-1})_{ij}(A_i, A_j) \tag{3.9}$$

$\mathbf{C}^{emp}$ is the empirical connected correlation matrix and it is defined as follows:

$$C_{ij}^{emp}(A_i, A_j) = f_{ij}^{'}(A_i, A_j) - f_i^{'}(A_i) f_j^{'}(A_j) \tag{3.10}$$

for $i \neq j$, while C_{ii}^{emp} is a diagonal matrix with $C_{ii}^{emp}(A_i, A_i) = f_i^{'}(A_i)$.

Having in mind that real data are not i.i.d. and come from a finite (usually small) size sample, we need a regularisation scheme to correct finite-sampling effects (we will extensively analyse the role of regularisation in the next chapter). The regularisation chosen here is a *pseudocount* regularisation.

$$\begin{cases} f_i^{'}(A_i) = (1-\theta) f_i(A_i) + \frac{\theta}{q} \\ f_{ij}^{'}(A_i, A_j) = (1-\theta) f_{ij}(A_i, A_j) + \frac{\theta}{q^2} \end{cases} \tag{3.11}$$

Equation 3.11 shows the use of pseudocounts as a correction over the single site frequency counts and the pair frequency counts. Parameter θ allows us to set the strength of the correction. According to what we will see in the next chapter, we choose $\theta = 0.5$.

3.4.6 The scores

Direct Coupling Analysis gives us the coupling matrix $\mathbf{e}$, but in order to find base-pairs with the *highest coupling* we define a scalar score for each pair: we compress information that in principle can be useful in order to classify contacts. We use the *Frobenius Norm* (Eq. 3.12) of the matrix $e_{ij}(A_i, A_j)$ with i and j fixed.

$$F_{ij} = \sqrt{\sum_{A_i} \sum_{A_j} |e_{ij}(A_i, A_j)|^2} \tag{3.12}$$

$$F_{ij}^{apc} = F_{ij} - APC_{ij} = F_{ij} - \frac{\langle F_{ij} \rangle_i \langle F_{ij} \rangle_j}{\langle F_{ij} \rangle_{ij}} \tag{3.13}$$

Interesting improvements can be obtained correcting F_{ij} with the so called *average product correction* (Eq. 3.13). APC [103] estimates the background coupling between two sites due to random and phylogenetic reasons and thus can be removed from the score so to obtain a more clear signal coming from coevolving pairs.

Once we have a scalar score for each pair of sites, the simplest thing to do is to sort them: the higher is the score the more reliable is the prediction. Actually, the reliability of the prediction is not fully understood. $TP(n)$ tells us the fraction of true contacts we find if we consider the n pairs with the highest score and it is an useful tool in order to compare the predictive power of different scores. However,

how do we actually compute the best value of n such that our prediction is still reliable? Meaning, how can we estimate till which value of $TP(n)$ our model is still predictive and not random? Trying to answer these questions we studied the *p-value* of true positive rates. The p-value shows the enrichment of true positives considering a certain number X of predictions and is computed using a binomial null model:

- We consider the list of all possible pairs of sites ranked according to a given score
- For each position X in the rank we compute T(X,Y) being the number of TP (True Positives) within a window including the next Y predictions, where Y equals 10% of all elements of the considered list. The size of the window Y is a compromise between local resolution (small Y) and reliability of the p-value (large Y).
- The binomial null model uses a random TP rate, determined from the remaining (from X+1 to the end) list of pairs, taking into account that the native contact we have found within the first X predictions cannot be found again.
- The p-value is determined as the probability that this null model achieves at least T(X,Y) contacts within a random i.i.d. sample of size Y.

The aim of this kind of analysis is to confirm that, even if scores of secondary-structure base-pairs have a much stronger covariation signal than tertiary base-pairs, DCA score contains information about non-canonical base-pairing that MI does not.

3.5 Article: Direct-Coupling Analysis of nucleotide coevolution facilitates RNA secondary and tertiary structure prediction

Results about DCA secondary and tertiary-structure prediction are shown it the following paper. Since for this work we also run some 3D predictions with Rosetta the list of RNA on which we test our tool is shorter than the Gold Standard introduced above. Rosetta segment assembly software needs a big computational effort and thus the length of the sequence to be folded has to be smaller than 100 nucleotides. We thus reduced our target set to 6 Riboswitches.

3.6 Open problems and future improvements

One of the points made in the paper above is that two different scales exist within the DCA score: there is a stronger part, probably due to Watson-Crick coevolution, and a weaker one due to non-canonical base-paring. It as been argue that non-canonical base-pairs show much less correlation [104]. We have shown that this picture changes when DCA is used instead of a measure for correlation as MI. Anyway it seems reasonable, and TP rates in Fig 4 in the paper confirm, that the coevolution signal from tertiary base-pairs is weaker and disentangling it from noise is absolutely non trivial. In the following I will propose two strategies to increase the signal-to-noise ratio based on a deeper analysis of the score.

3.6.1 Filtering matrices

When we use a scalar score, we are forced to waste a large part of the information contained in the coupling matrix **e**. For instance, consider (i,j) being a Watson-Crick base pair in the secondary structure. We know that the substitution of a C with a A in site i has to be probably followed by a substitution of a G with a U in site j. Using this information we can in principle clean up the secondary-structure signal from noise: we can define a weight matrix based on the knowledge of base-pairs that are possible as in the table 3.3.

Finally, we weight each element $e_{ij}(A_i, A_j)$ with the corresponding value of the pair of nucleotides A_i and A_j as in 3.14:

$$S_{ij} = \sum_{A_i} \sum_{A_j} e_{ij}(A_i, A_j) w(A_i, A_j) \tag{3.14}$$

This weighted score is a good alternative to the simple F_{apc}. However it solves the easiest part of the problem: filter out non-canonical interactions. One can argue that a similar approach can also be used the other way round, that is to filter out secondary-structure signal. However results contradict this hypothesis confirming the complexity of the problem or revealing some intrinsic limitation of the mean-field. This particular topic will be treated more in details in Chapter 4.

3.6.2 Local coherence matrix

Even if protein DCA strongly improves the accuracy of residue-contact predictions, we have seen in the previous sections that, when RNAs samples are used, comparing the number of true positive predictions within the n highest DCA scores

	A	C	G	U
A	0	0	0	1
C	0	0	1	0
G	0	1	0	1
U	1	0	1	0

Table 3.3: Matrix $w(A_i, A_j)$ values based on the possible Watson-Crick base-pairs plus the wobble pair

with the number of true positives within the first n mutual informations, only a weak improvement is achieved (Fig. 4 in the paper). Actually, if we remove from the ranking all secondary-structure base-pairs, results are quite different. Focus for instance on a true positive rate equal to 0.6 (i.e. we admit 40% of prediction errors): MI with average product correction predicts on average around 3 true contact if the inter-nucleotide distances is up to 4Å or 4 up to 8Å , while DCA predicts 5 up to 4Å or 10 up to 8Å . Thus, DCA can increase significantly the number of correct predictions within the tertiary structure, in particular when a less stringent threshold for contact definition is chosen.

The clear prevalence of secondary-structure base-pairs among the top predictions shows that tertiary-structure contacts have a weaker coupling scale than secondary-structure contacts. This weak coupling can be partially hidden by the noise generated due to insufficient sampling and the strong secondary-structure couplings. Moreover usually first false positives appear quite close to native contacts. Therefore we compute for each pairs of nucleotides a local coherence score as the average of the score of the considered pair with the scores of the 8 nearest neighbours in the contact map. Being F_{apc} the coevolution score, we define the local coherence score $\boldsymbol{C}$ as:

$$C_{ij} = \sum_{l \in i, i\pm 1} \sum_{k \in j, j\pm 1} F_{kl}^{apc} \tag{3.15}$$

i.e., for each pair (i, j), we include also $(i, j \pm 1)$, $(i \pm 1, j)$ and $(i \pm 1, j \pm 1)$ into the average. As an immediate consequence, background noise is almost homogeneous, while around existing contacts, some compact clusters of pairs with higher score arise. In Fig. 3.11 we show results of the averaged procedure for the 6 Riboswitches analysed in the paper.

In principle it would be possible to reinforce local coherence by an averaging procedure over larger neighbourhoods of each pair (i,j), however we observed that the proposed environment is a good compromise between the noise reduction due to local signal coherence, and the loss of specificity of the scores due to averaging.

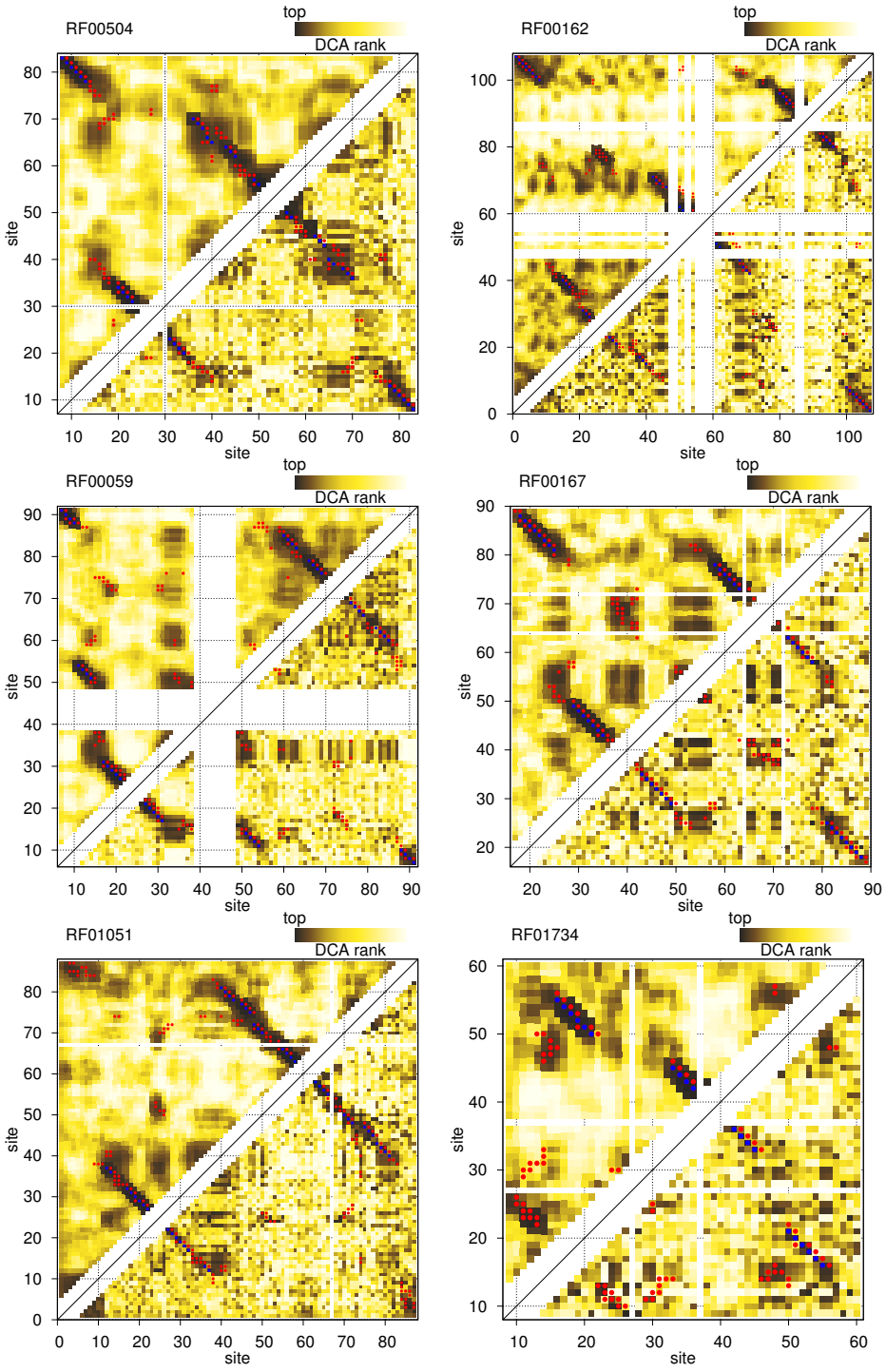


Figure 3.11: Contact maps: Yellow scale shows both the DCA (F_{apc}) and its local coherence ranking. In the bottom-right triangle of each map DCA score is shown. In the top-left triangle instead we show local coherence score. Red and blue dots indicate true contacts and secondary-structure base-pairs. The same set of six Riboswitches analysed in the paper is shown.

3.6.3 Clustering procedure

Given the local coherence scores, we need to define a procedure to separate potential coevolutionary signal from background noise. Average scores are divided into four classes using a simple K-means clustering, cf. insets in Fig. 3.12. Empirically we find that the two classes of highest scores correspond to residue pairs inside or close to the secondary structure. The fourth class of lowest scores contains background noise, it will be discarded from further analysis. Potential tertiary-structure contacts are mainly restricted to the third class, more precisely to clusters of third-class position pairs which are isolated from the secondary structure.

Since we know the secondary structure a priori, we can select, among pairs belonging to the third class, tertiary structure predictions. Indeed, being the secondary-structure signal so strong, one can expect the averaged procedure producing high score pairs nearby all predicted Watson-Crick base-pairs. We observe this effect being propagated till the second nearest neighbour of each base-pair. Thus, being (i, j) a secondary-structure base-pair, we remove from the ranking pair itself and all possible pair combination among sites $(i, j, i \pm 1, j \pm 1, i \pm 2, j \pm 2)$. The effects of this removal are shown in Fig. 3.13. Finally, we rank the remaining pairs using the original score Fapc (i.e. the average score is used to discard locally incoherent predictions, the original score for the final ranking of maintained position pairs).

With the above described procedure we produce a clustered DCA score ready to be used in structural predictions and extracting the best part of information from the original DCA: the local coherence method produces a signal that is locally homogeneous and clustering filters out noise.

Unfortunately, even if the comparison between the two scores in Fig 3.11 suggests that a post-processing of the signal is possible, final results obtained with Rosetta were generally inconclusive and very similar to simple DCA ones: some of the families benefits of the post-processing, some others not. The absence of a systematic improvement entails the impossibility of an automatic pipeline including the local coherence analysis. For instance the choice of the number of classes, defined within the K-means clustering, seems to be suboptimal for RF00162 and RF01734 (cf. Fig. 3.12) or even the definition of the local coherence score can be probably improved including more neighbours in the average or weighted in a smooth way. However in the field of RNA structure prediction a human intervention and optimisation of techniques is still common and cannot be a priori discarded.

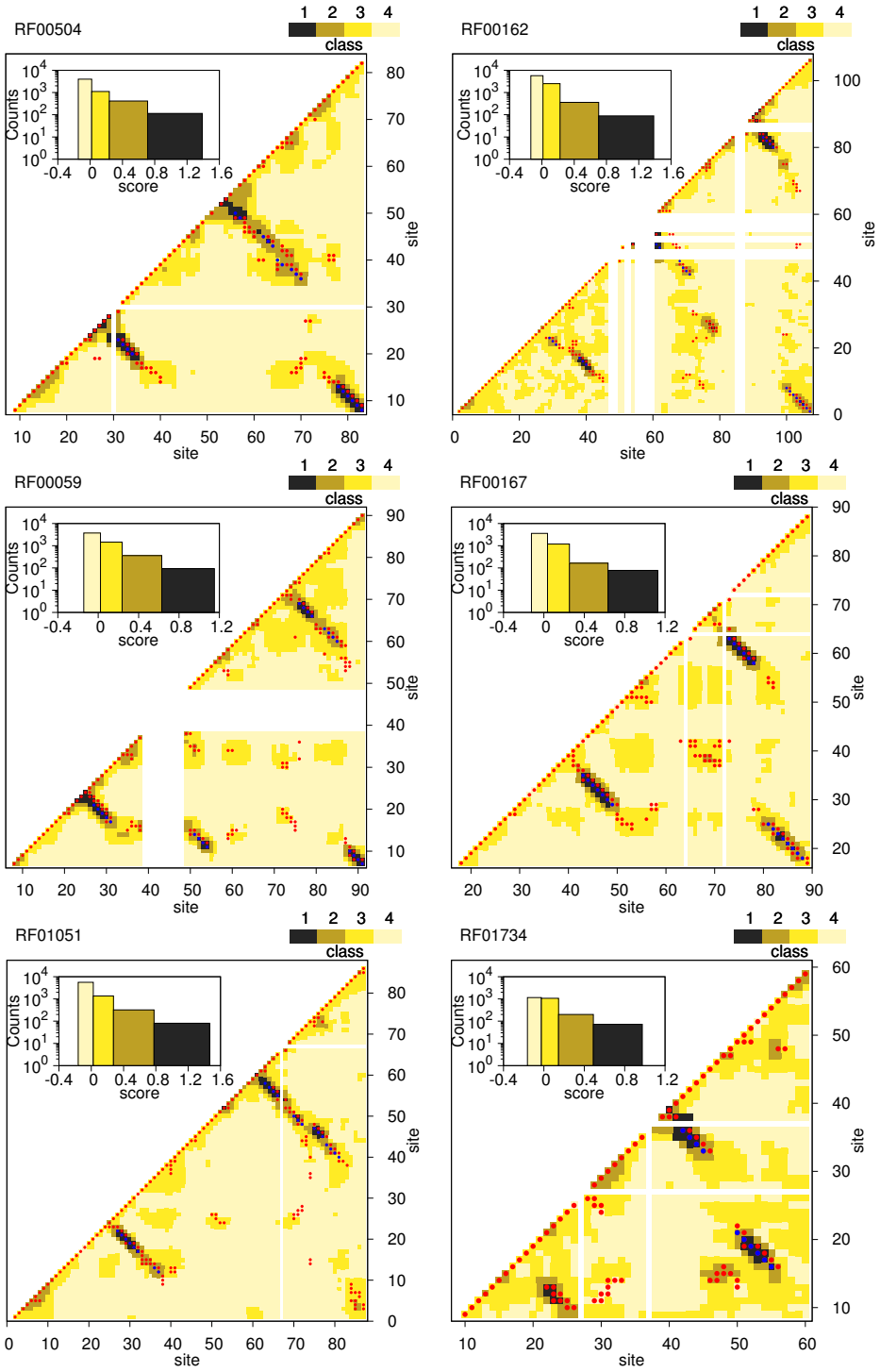


Figure 3.12: K-means clustering results. Pair of sites belonging to different score classes are shown with different colours in the maps. Darker colours (black and green) correspond to secondary structure and its neighbours plus, in some cases, some tertiary contacts. The most part of tertiary true predictions belong to the yellow class. The fourth class represents background. Insets: K-means clustering classification is shown with respect to the score frequency counts.

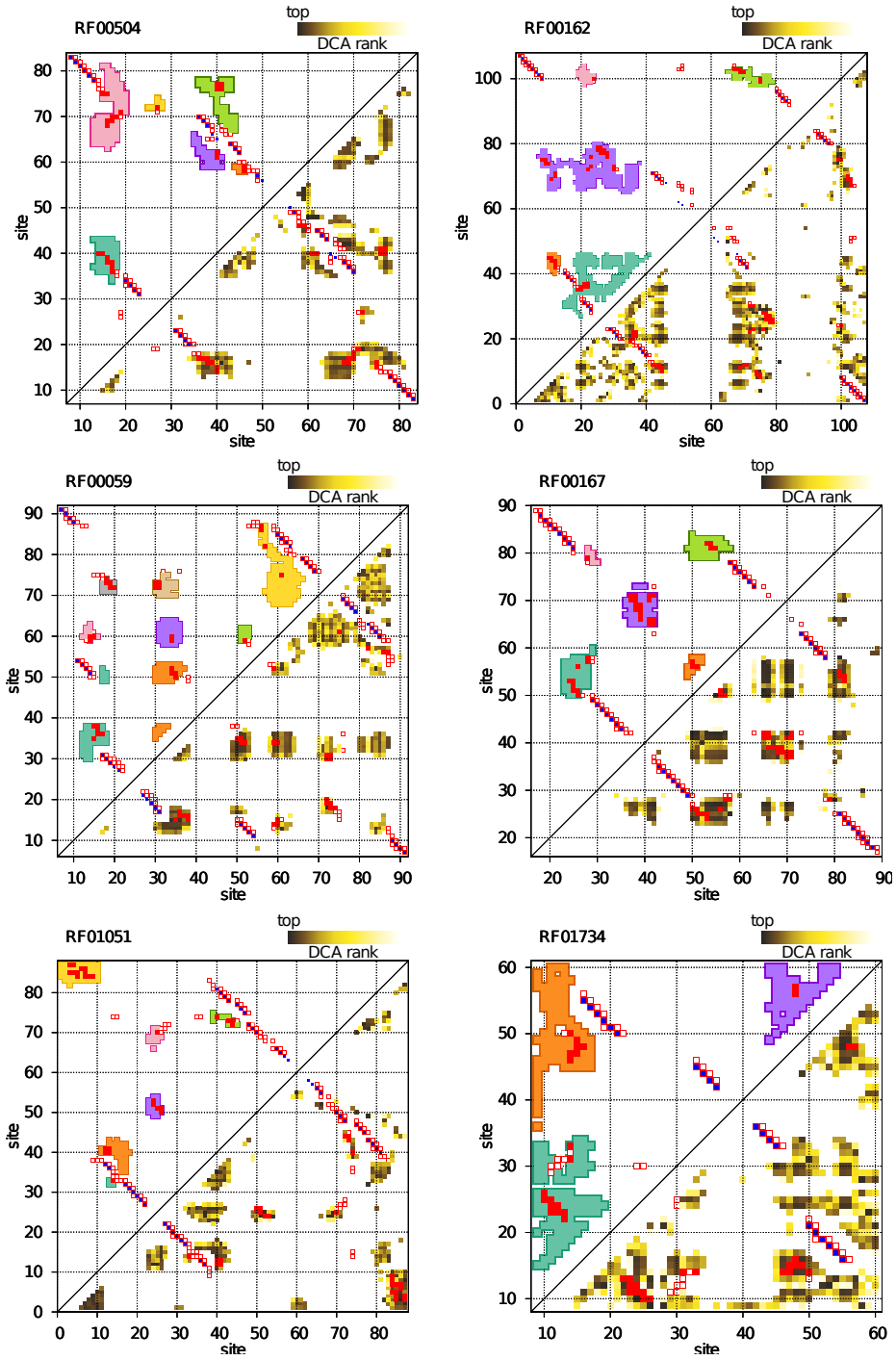


Figure 3.13: Contact maps: Yellow scale shows DCA scores for each pair of sites included in the clustering selection (both true and false positives) after removing the first and the second nearest neighbours of secondary structure base-pairs. The first n pairs with the highest score have been included in the structure prediction. In the bottom-right triangle true positive predictions are shown with closed red squares while all the other contacts in the crystal structure with open red squares, the consensus secondary structure is shown with blue squares. In the top-left triangle clusters containing true positives are shown.

3.7 Conclusions

Within this chapter we have faced the problem of functional RNA structure prediction. Given its intrinsic flexibility, the experimental determination of reliable RNA structures is still challenging and computational approaches have been developed to solve the problem. However the quality of 3D predictions still depends on the experimental knowledge of tertiary interactions. Outperforming MI and its adjusted version MIapc, DCA is proposed as a novel method for comparative sequence analysis. It has been proved that combined to standard and very well known tools as Nussinov algorithms and Rosetta, DCA systematically improves predictions on a 6-RNA benchmark. Results shown in this chapter open to further application of DCA to a diverse range of software already including covariational scoring systems or not.

Chapter 4

Limits of mean-field inference and the role of regularisation

In the previous chapter we have seen how mean-field (MF) inference can be successfully applied to RNA covariational sequence analysis. Moreover results on RNAs poorly represent the real potential of DCA, whose performances are considerably better on proteins [3] [31]. However, these results still depend on the use of a particular type of regularisation scheme, called pseudocounts, needed to ensure that the inverse problem is always well defined. As we have seen in the previous chapter, the empirical covariance matrix C_{ij}^{emp} in Eq. 3.10 is defined on the transformed counts $f_i^{'}$ and $f_{ij}^{'}$:

$$\begin{cases} f_i^{'}(A_i) = (1-\theta) f_i(A_i) + \frac{\theta}{q} \\ f_{ij}^{'}(A_i, A_j) = (1-\theta) f_{ij}(A_i, A_j) + \frac{\theta}{q^2} \end{cases} \tag{4.1}$$

being θ the strength of the pseudocount regularisation and q the number of colours on the Potts model. From a Bayesian point of view the optimal value for θ should depend on the level of noise in the sample (i.e. $\theta \sim \frac{1}{B}$ for a sample of size B) and should vanish for perfect a sampling. However several empirical studies [26] [3] [105] [106] have shown that it is not the case for MF inference. As large ($\theta \gg \frac{1}{B}$) pseudocounts are in this case used, no dependence of θ on the sample size is observed.

In the following paper we have analytically studied MF inference performance on diverse systems in the perfect sampling case. We observe that large regularisation terms help correcting the bias introduced by MF approximation: MF approximation over-estimates large couplings, while with a strong regularisation we under-estimate them. The result is that for medium-range values of couplings the quality of the inference is dramatically improved with $\theta \gg \frac{1}{B}$ compared to $\theta \sim \frac{1}{B}$. We show that both large pseudocounts and $L2$-norm regularisations yield couplings which correlate better with the true ones.

Moreover we have claimed that the strength of the regularisation depends on the number of colours in the model and even substantial differences exist between the Ising and the Potts case: thanks to some toy-models made of 2 spins, we have shown that on Potts models the inference is poorer than on the Ising model case because terms in the coupling matrix J_{ij} are differently biased by the MF approximation. In particular we have observed that the hardness of the inference depends on how couplings matrices within the Potts model are defined. Note that once we have decided that an interaction exists among site i and size j of the graph we need to specify the form of matrix J_{ij}. The size of this matrix is $q \times q$ and thus for each interacting pair q^2 parameters have to be drawn from a certain distribution. We have distinguished two variants:

- *Homogeneous variant:* For each interacting pair (i, j), we randomly draw a number, J_0, and we define J_{ij} as follows:

$$J_{ij} = \begin{pmatrix} J_A & J_B & J_B & \dots & J_B \\ J_B & J_A & J_B & \dots & J_B \\ J_B & J_B & J_A & \dots & J_B \\ \dots & \dots & \dots & \dots & J_B \\ J_B & J_B & J_B & J_B & J_A \end{pmatrix} \quad \text{with} \quad J_A = \frac{q-1}{q} J_0 \,, \quad J_B = -\frac{J_0}{q} \,. \tag{4.2}$$

  This model is such that the q Potts colours have equal frequencies $f_i(a) = \frac{1}{q}$.

- *Heterogeneous variants:* The simpler extension of this model to non-equal frequencies can be easily obtained by adding some local fields $h_i(a)$ on each site and for each colour. Fields introduce a bias on some particular colour on each site and entail a heterogeneous distribution for frequencies. In the following we will call this model *heterogenous-A model*. Moreover, one can induce an even more heterogeneous distribution for frequencies with the addiction of randomly chosen elements in J_{ij}. This is the most general case of a random characterisation of a Potts model, since no constraints exist among parameters. We refer to this model as the *heterogeneous-B model*.

We have noticed that significant differences exist between these two classes of models: we have analytically computed the relation between true couplings and MF inferred couplings for a toy-model containing only 2 spins. However analytical results have been numerically confirmed on larger systems of size $N = 50$ and results on both $q = 5$ and $q = 21$ will be shown below. Parameters are drawn from a uniform distribution between $-L$ and L. These results (Figs. 4.1 4.2 and 4.3) are obtained in

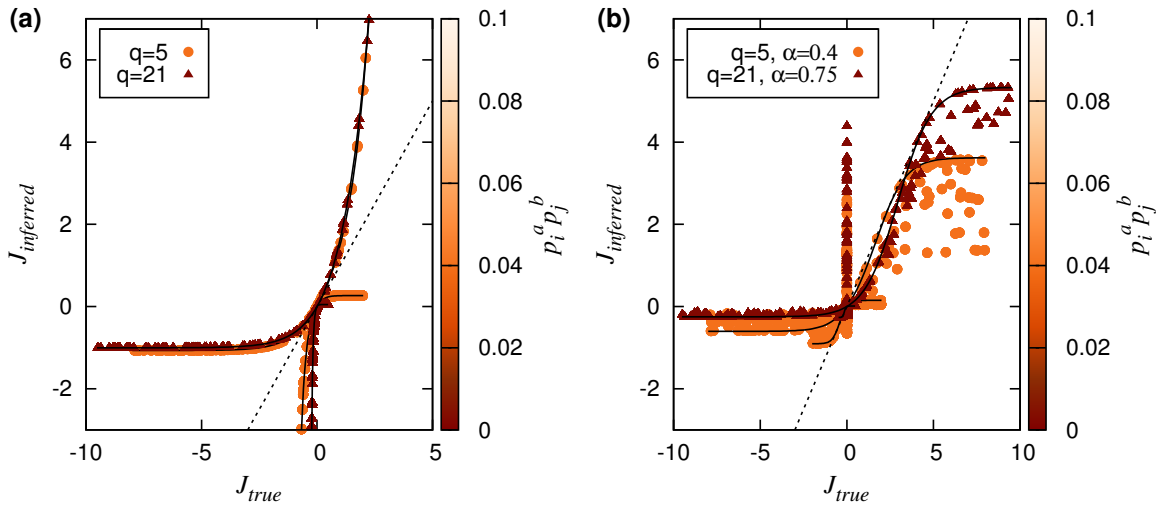


Figure 4.1: Scatter plot of the couplings $J_{ij}(a,b)$ for the homogenous Potts model for $q = 5$ (filled circles) and for $q = 21$ (triangles) colours; perfect sampling. **(a)** No pseudocount. **(b)** With pseudocount (in the figure θ correspond to the pseudocount strength called θ in the text). Each panel shows results from three realizations with different sets of couplings ($L = 10$). Black solid lines correspond to the analytical predictions made with the 2-spin toy model. Colours show values of $f_i(a)f_j(a)$ (called in the figure p_i^a and p_j^a), here equal to q^{-2} for all interacting sites and for all symbols, see right scale.

the perfect sampling regime ($B = \infty$) on a system with nearest-neighbour interactions on a $1D$ lattice. The exact solution for frequencies and correlations is obtained through a transfer matrix calculation.

Fig. 4.1 shows the relation found between true and inferred couplings for the homogeneous model. Note that, differently from what is found in the Ising case, the two curves exist as two different values of couplings (J_A and J_B) are found in the coupling matrix 4.2. The quality of the inference is extremely improved with pseudocount as couplings in the range from nearly zero to five are well estimated. However some significant mistakes are made: the pick in zero found in panel B of Fig. 4.1 is due to the fact that pseudocount regularisation entails a rescaling of colour frequencies inducing some fake interaction among couples that are not in contact in the real model. The points found under the right side of curves are a compensation of this effect.

When the heterogeneous variant is considered (Figs. 4.2 and 4.3) analytical computed curves are no more distinguishable, even if in Fig. 4.2 the overall trend is still visible. This is no more true for Fig. 4.3. The point is that a curve of the type seen in 4.1 exists for any different entry of matrix J_{ij} and follows a slightly different path.

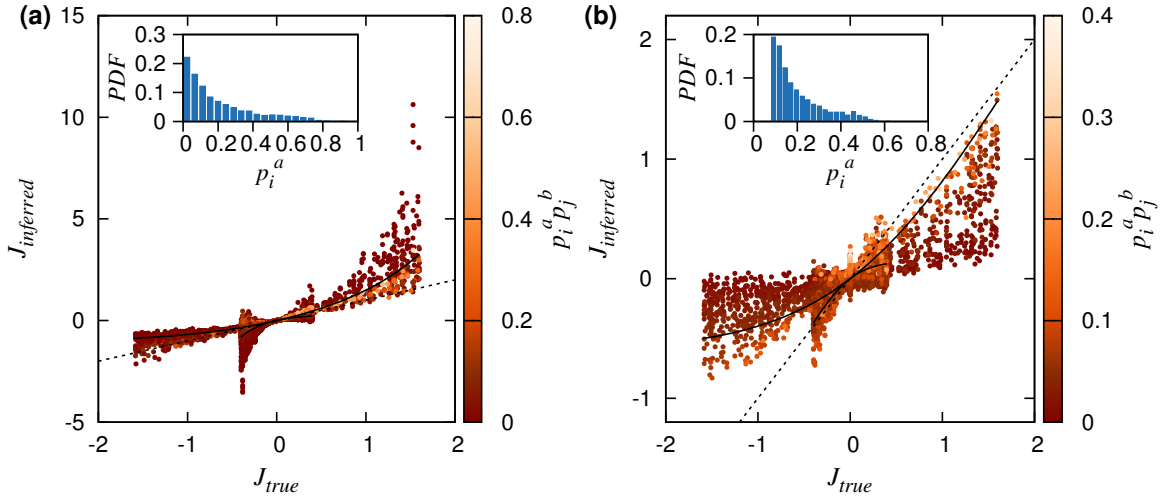


Figure 4.2: Scatter plot of the couplings $J_{ij}(a,b)$ for the heterogenous-A Potts model for $q = 5$ symbols, with perfect sampling. **(a)** No pseudocount. **(b)** With pseudocount ($\theta = 0.4$). Each panel shows results from five realizations with different sets of couplings and fields ($L = 2$). Insets: distributions of the frequencies $f_i(a)$. Black solid lines correspond to the analytical predictions made with the 2-spin toy model. Colors show values of $f_i(a)f_j(a)$ (called in the figure p_i^a and p_j^a), see right scale.

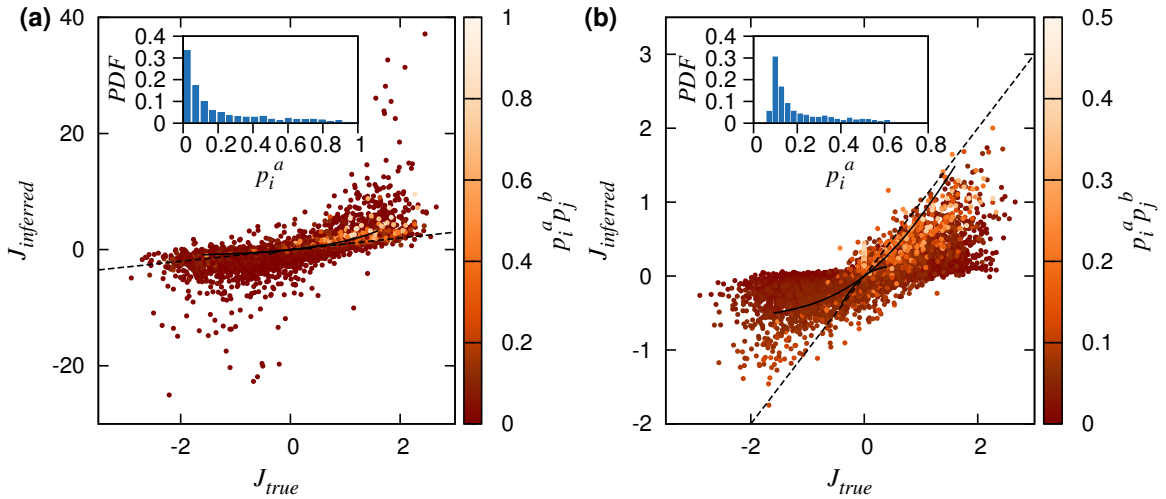


Figure 4.3: Scatter plot of the couplings $J_{ij}(a,b)$ for the heterogenous-B Potts model for $q = 5$ symbols, with perfect sampling. **(a)** No pseudocount. **(b)** With pseudocount ($\theta = 0.4$). Each panel shows results from five realizations with different sets of couplings and fields ($L = 2$). Insets: distributions of the frequencies $f_i(a)$. Black solid lines correspond to the analytical predictions made with the 2-spin toy model. Colors show values of $f_i(a)f_j(a)$ (called in the figure p_i^a and p_j^a), see right scale.

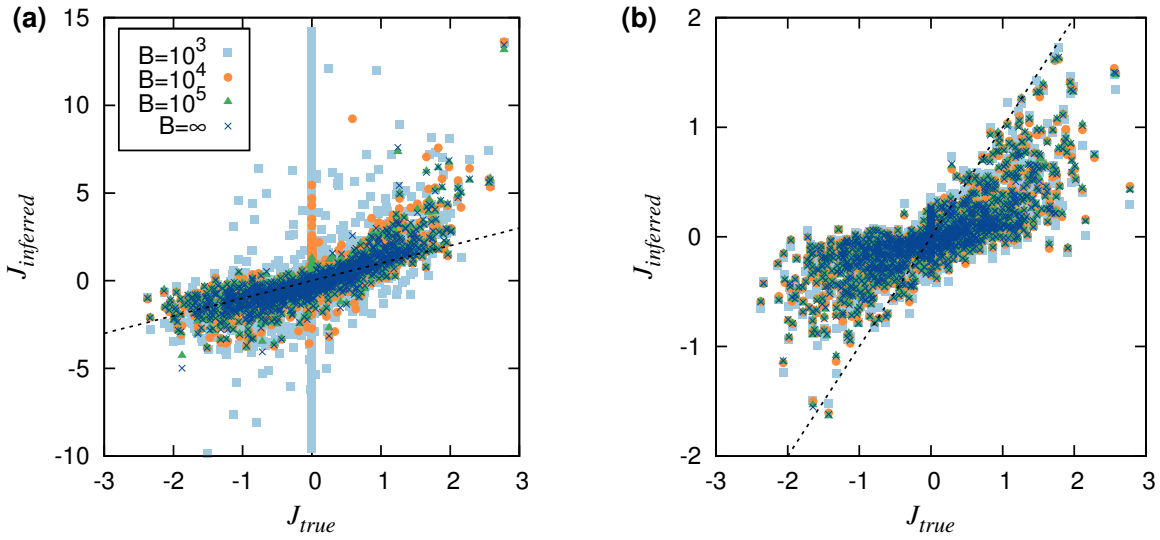


Figure 4.4: Heterogenous-B Potts model for $q = 5$ symbols, for various depths of sampling. **(a)** Small pseudocount $\theta = \theta^B = \frac{1}{B}$. **(b)** Large (optimal) pseudocount $\theta = \theta^{MF} = 0.4$. Each panel shows results from one realization of the Potts model with random couplings and fields ($L = 2$), and three sets of B sampled configurations (for finite B).

Consider now the case of finite sampling. We know that in principle the main role of regularisation is to correct from finite-sampling effects. This is true, and confirmed by our analysis, when the regularisation is tuned according to bayesian considerations: panel A of Fig. 4.4 shows how finite samples entail diverging inferred couplings: the smaller the sample is, the larger finite-sampling errors are. In this case we use a small pseudocount, $\theta = 1/B$, in order to correct these effects, and therefore a certain dependence on the sample size is visible. Conversely, when large pseudocounts are used, no dependence on the sample can be observed (cf. with panel B in Fig 4.4). Finally note that in Fig. 4.4 a Potts model on an Erdos-Reniy random graph is shown. No significant differences with the heterogeneous B model shown above exist.

As a final, but significant, remark let us observe in Fig. 4.5 the behaviour of the Frobenius norm computed with the couplings shown in Fig. 4.4. Even if the inference of the coupling values seemed to be confused, the inference of the interaction network is definitely ensured by the use of large pseudocounts. Probably the success of methods such as DCA mainly depends on the surprising synergy among pseudocounts, MF and Frobenius norm.

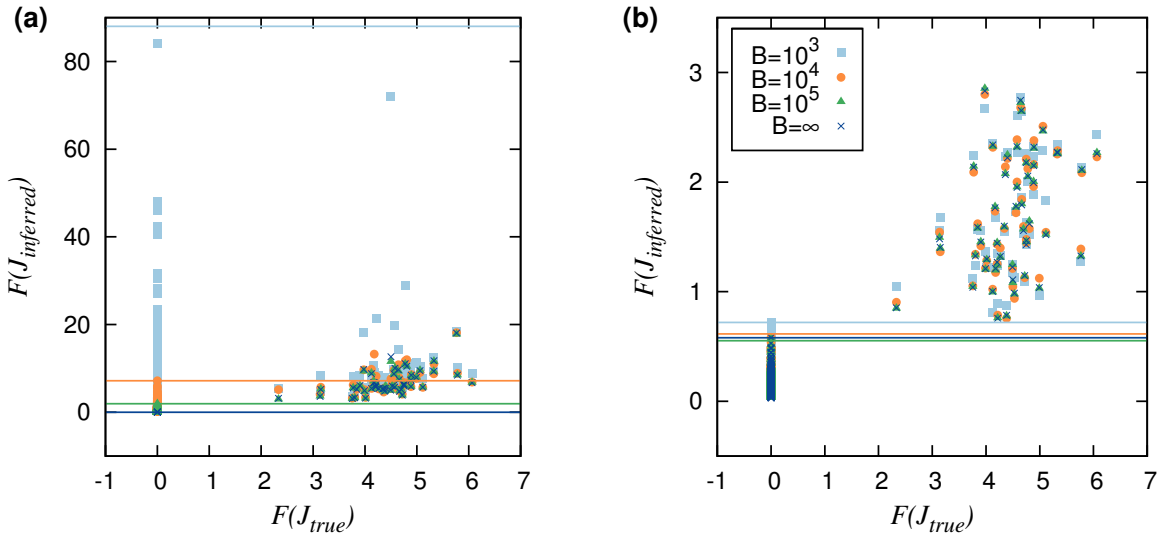


Figure 4.5: Scatter plot of the Frobenius norms of the inferred couplings vs. their true values for the pseudocount strengths $\theta = \frac{1}{B}$ **(a)** and $\theta = 0.4$ **(b)**. Same heterogeneous-B model and same conditions as in Fig. 4.4. Lines locate the largest Frobenius norm corresponding to a pair of sites (i, j) which are not neighbours on the one-dimensional graph, *i.e.* which have zero true coupling.

4.1 Article: Large pseudocounts and L2-norm penalties are necessary for the mean-field inference of Ising and Potts models

The full work on MF inference and regularisation schemes is reported in the following paper. After a short reminder on the main topics of the paper, we compare the effect of pseudocounts and $L2$-norm regularisation on small toy-model systems computing the analytical form for MF couplings given the value of the true ones. Both the Ising and the Potts cases are treated. Note that in the Potts case the relation between MF and true couplings is made of at least two different curves: each term in the coupling matrix J has a different dependency on the true couplings. We then show that on bigger systems ($N = 50$) the proliferation of these contributes makes Potts inference systematically worse than Ising one. In the last section we use the $\mathcal{O}(m)$ model, a generalisation of the Ising model, to estimate the error on the inferred couplings due to the MF approximation and how this error can be corrected for with an appropriate regularisation.

4.2 Conclusions

With this paper we have shown that MF inference introduces strong errors on large couplings. These errors can be partially corrected thanks to strong regularisation terms and in particular a large pseudocount turns out to be the best approach. Moreover we better understand why DCA needs pseudocount regularisation for predicting the network of long-range interactions in proteins: firstly the use of large pseudocounts dramatically reduces the dependence of the inference quality on the sample size. Secondly, thanks to the large pseudocount correction on inference, the Frobenius norm of couplings matrices averages out differences between diverse couplings keeping separated in ranking interacting and non-interacting pairs. Anyway note that results in the paper are shown for artificial models, where only the pairs of sites actually interacting have non-zero coupling matrices, but this is absolutely not true for real data. The relations we compute here are true if we assume the inferred model to be a Potts model, that is obviously not the case for sequence analysis. Therefore additional errors depending on the choice of the model have to be taken into account.

This last point opens a huge debate on how could be possibly assess a certain inference method to be better than another. The first question to ask is probably which is the task we want to accomplish. For instance, since we know the Potts model is just our interpretation and simplification of the evolution process, we could argue that having a method able to perfectly fit Potts parameters does not imply that it will also be the best in predicting the contact map. Furthermore, we saw that a MF approximated solution for the Potts model gives modest results, while DCA is a very powerful tool for contact predictions. Therefore inference methods are commonly tested on real data taken from diverse biological topics, in order to compare performances on real-world cases. Contact map prediction is one of the task commonly used for comparison, as several different tools exist and standard results are known. However a more complex task having interested a large part of the scientific community is the possibility of building a model able to reproduce data statistics. Experimental studies on artificial sequence folding [28] have stressed the importance of computationally predicting whether a given sequence will fold or not depending on the amino acids of the sequence itself.

MF is not statistically consistent, meaning that even in case of perfect sampling MF parameters cannot produce a model able to generate a sample whose statistics recalls the one of the sample from which the parameters have been inferred. As we

saw in the first chapter, two examples of statistically consistent methods are pseudo-likelihood and ACE. In the following chapter I will introduce ACE and how it has been adapted to Potts inference. We will see that, keeping the quality of contact maps almost at the same level, ACE infers parameters reproducing data statistics with a precision compatible with the size of finite-sampling errors, differently from pseudo-likelihood. We will compare DCA, plmDCA (pseudo-likelihood) and ACE according to both the criteria of contact map prediction and statistical of consistence. We will first focus on artificial models and then we will consider the same set of riboswitches we analysed in the first chapter in order to test inference methods on real data.

Chapter 5

An inference tool for generative models: The adaptive cluster expansion

In chapter 1 we have analysed some of the existing algorithms for the solution of the inverse Ising and Potts model. The last tool I described was the Adaptive Cluster Expansion [7] [23]. This algorithm has been introduced and tested for the Ising model. Its extension to the Potts model is straightforward from the analytical point of view, while entails some computational issues that need to be handled more carefully. Recall that, even in the MF case, Potts inference turns out to be harder than Ising one.

In this chapter a many colours Potts model (e.g. 21 colours as it is the number of protein amino acids plus the gap symbol) will be treated. The first part of the chapter will be dedicated to a short review of the original algorithm and to its adaptation to the formalism of Potts model. Then, I will introduce the numerical procedures we designed in order to improve results and to reduce computational efforts. Finally some interesting results on artificial models and RNA data will be reported.

5.1 The ACE algorithm

As we have seen in chapter 1, the solution of inverse models entails the minimisation of the negative log-likelihood $\mathcal{L}$, introduced in Eq. 1.8, equivalent to the cross-entropy S between the data and the model, that for the Potts model is defined as follows:

$$S \equiv \mathcal{L} = \log Z - \sum_{i=1}^{N}\sum_{a=1}^{q} h_i(a) f_i(a) - \sum_{i<j}^{N}\sum_{a=1}^{q}\sum_{b=1}^{q} J_{ij}(a,b) f_{ij}(a,b) \tag{5.1}$$

where from now on $f_i \equiv f_i^{data}$ and $f_{ij} \equiv f_{ij}^{data}$ in order to simplify notation. The inclusion of a prior distribution can be helpful for avoiding over-fitting. A Gaussian prior distribution for the parameters is a typical choice:

$$S^{L2} = S - \gamma' \sum_{i=1}^{N}\sum_{a=1}^{q_i} h_i(a)^2 - \gamma \sum_{i=1}^{N-1}\sum_{j=i+1}^{N}\sum_{a=1}^{q_i}\sum_{b=1}^{q_j} J_{ij}(a,b)^2 \tag{5.2}$$

where $\gamma' = 0.01\gamma$ as we expect fields need a smaller regularisation compared to couplings.

Monasson and Cocco [7] have proposed an expansion of S based on a graphical subdivision of the target network of interactions in small sub-systems called *clusters* and defined as $\Gamma = \{i_1, \ldots, i_k\}, k \leq N$,

$$S = \sum_{\Gamma} \Delta S_{\Gamma}, \tag{5.3}$$

where the summation is made over all possible sub-systems of the N-spin system. ΔS_Γ is the cluster entropy and can be recursively computed thanks to the following relation:

$$\Delta S_\Gamma = S_\Gamma - \sum_{\Gamma' \subset \Gamma} \Delta S_{\Gamma'}\,. \tag{5.4}$$

The term S_Γ represents the minimum of Eq. 5.1 restricted to those spins included in the cluster Γ. As we will see, the authors have claimed that this sum can be truncated, with a minor loss of information, to a restricted number of selected clusters for computational feasibility. Groups of strongly interacting spins in the system contribute more to the overall cross-entropy, than weakly interacting ones. Therefore the selection of clusters is possible on the basis of cluster absolute contribution to S and a truncated sum can be defined. The convergence of the series is ensured by the fact that contributes from clusters of spins within the same integration path partially cancelled each other out [23]. However, the numerical minimisation of the cluster cross-entropy entails a sum over q^k terms, where q is the number of colours of the model and k is the size of the sub-system. The great advantage of this algorithm is

that the exponential complexity of this computation is moved from the size of the system to the size of the considered cluster, ensuring reasonable execution times also for large (i.e. $N \sim 100$) systems, as long as the size of the clusters having been included in the sum remains small (i.e. $k \sim 10$). Note that the summation on the full set of clusters bring to the exact computation of the log-likelihood.

Being recursive, Eq. 5.4 ensures that the minimisation of the log-likelihood of a given cluster depends only on the frequencies and the correlations of the variables in the sub-system. For instance, the cluster of size one $\Gamma = \{i_1\}$ depends only on the frequencies of observations of each colour on site i_1 and its entropy is defined as:

$$S_\Gamma = -\sum_{a=1}^{q} f_{i_1}(a) \log(f_{i_1}(a)) \tag{5.5}$$

Clusters of size two, as $\Gamma = \{i_1, i_2\}$, depend on $f_{i_1}(\sigma_{i_1})$, $f_{i_2}(\sigma_{i_2})$ and $f_{i_1 i_2}(\sigma_{i_1}, \sigma_{i_2})$ and their cluster entropy ΔS_Γ corresponds to the mutual information between sites i_1 and i_2. More generally the cluster entropy of a cluster of size k represents the gain in information when the k variables are considered to be mutually interacting. When the two variables in a 2-variable cluster are independent their ΔS_Γ vanishes.

The full algorithm is described below:

1. We define a threshold t on the overall cross-entropy. We will use it in order to discriminate clusters significantly contributing to the log-likelihood from those which can be neglected.
2. We compute analytically the entropy and the parameters of all clusters of size 1
3. We define a list L_k of clusters of size $k = 2$
4. For each cluster $\Gamma \in L_k$
   (a) We compute S_Γ by the numerical minimisation of 5.1 restricted to Γ.
   (b) We record the parameters (fields and couplings) minimising 5.1.
   (c) We compute ΔS_Γ using 5.4.
5. We select significant clusters among $\Gamma \in L_k$ with $|\Delta S_\Gamma| > t$.
6. We construct a list L_{k+1} of clusters of size $k + 1$ from overlapping clusters selected during the previous step.
7. We lower t and iterate from step 4.

The construction of lists L_{k+1}, given selected clusters in L_k, can be performed according to two different rules: the so called *lax rule* implies a new cluster Γ to be added to the list L_{k+1} if there is at least a pair of clusters Γ_1, Γ_2 in L_k of size k such that $\Gamma_1 \cup \Gamma_2 = \Gamma$; the *strict rule* implies, instead, only the sub-clusters of Γ to be included in L_k.

At each step not only the contribution to the entropy is computed, but also the approximated value of the parameters minimising the cross-entropy:

$$\mathbf{J}(t) = \sum_k \sum_{\Gamma \in L_k(t)} \Delta\mathbf{J}_\Gamma, \quad \Delta\mathbf{J}_\Gamma = \mathbf{J}_\Gamma - \sum_{\Gamma' \subset \Gamma} \Delta\mathbf{J}_{\Gamma'} \,. \tag{5.6}$$

where $\mathbf{J}$ is an array representing both field and couplings. As in 5.3 and 5.4 $\mathbf{J}_\Gamma$ is obtained via the numerical optimisation of the log-likelihood, while $\Delta\mathbf{J}_{\Gamma'}$ is the contribution due to smaller clusters. At each step the sparsity of the obtained graph is thus guaranteed by the fact that the sum in Eq. 5.6 is restricted over those clusters with $|\Delta S_\Gamma| > t$.

Given inferred parameters, we test them computing corresponding frequencies and correlations with the Monte Carlo sampling. To avoid over-fitting we stop the algorithm when the difference between the observed and the Monte Carlo correlations stays in the error bars due to finite-sampling approximation (cf. Eq. 5.7). When the observed correlations are not yet well reproduced, the algorithm is iterated decreasing the threshold t in order to include more clusters in the computation of parameters.

The typical uncertainties for frequencies and correlations can be determined simply from the susceptibility matrix χ (i.e. the hessian of the cross-entropy, also known as Fisher information matrix)

$$\begin{aligned} \delta f_i &= \sqrt{\frac{1}{B}\chi_{i,i}} = \sqrt{\frac{f_i(\sigma_i)(1-f_i(\sigma_i))}{B}} \,, \\ \delta f_{ij} &= \sqrt{\frac{1}{B}\chi_{ij,ij}} = \sqrt{\frac{f_{ij}(\sigma_i,\sigma_j)(1-f_{ij}(\sigma_i,\sigma_j))}{B}} \,. \end{aligned} \tag{5.7}$$

The estimation of the quality of the inference is made recording at each value of t the average error ϵ_P on frequencies, the average error ϵ_{P2} on correlations (alternatively also the error on connected correlations ϵ_C) and the maximum error ϵ_{max} among all terms included in ϵ_P and $\epsilon_P 2$ as shown in Eqs. 5.8, where f^{MC} represent the statistics

of the Monte Carlo sampling run with inferred parameters.

$$\epsilon_{P2} = \sqrt{\frac{1}{\frac{N(N-1)}{2}q^2}\sum_{i<j}\sum_{\sigma_i,\sigma_j}\frac{(f_{ij}(\sigma_i,\sigma_j) - f_{ij}^{MC}(\sigma_i,\sigma_j) - (2\gamma J_{ij}(\sigma_i,\sigma_j)))^2}{\delta f_{ij}^2(\sigma_i,\sigma_j)}} \tag{5.8a}$$

$$\epsilon_{P} = \sqrt{\frac{1}{Nq}\sum_{i}\sum_{\sigma_i}\frac{(f_{i}(\sigma_i) - f_{i}^{MC}(\sigma_i) - (2\gamma' h_i(\sigma_i)))^2}{\delta f_{i}^2(\sigma_i)}} \tag{5.8b}$$

$$\epsilon_{max} = \sqrt{\frac{\max_{ij}\left(\frac{(f_{ij}^{MC}(\sigma_i,\sigma_j) - f_{ij}(\sigma_i,\sigma_j))^2}{\delta f_{ij}^2(\sigma_i,\sigma_j)}, \frac{(f_{i}^{MC}(\sigma_i) - f_{i}(\sigma_i))^2}{\delta f_{i}^2(\sigma_i)}\right)}{2 * \ln(\frac{N(N-1)}{2}q^2 + Nq)}} \tag{5.8c}$$

These errors represent finite-sampling errors. The term depending on γ (or $\gamma' = 0.01\gamma$) at the numerator was introduced to prevent from over-fitting small frequencies and correlations. Note that δf_i and δf_{ij} are found in Eq. 5.8 at the denominator, thus they need to be treated carefully in case of zero correlations: taking into account definitions 5.7, we fix a lower bound for both f_i and f_{ij} equal to $\frac{1}{B}$. Another possible correction consists in computing χ from the $L2$-regularised cross-entropy adding thus a regularisation term 2γ at the numerator of Eqs. 5.7. However, we noticed that differences between the two approaches are negligible for typical values of the regularisation strength and we prefer the first solution from a practical point of view: errors are well defined also when the algorithm is run with $\gamma = 0$.

Practically the algorithm starts with a large threshold ($t = 1$) and only 1-spin clusters are taken into account, then the threshold is lowered $t \to t/1.05$ until a set of parameters fitting the 1- and 2-point statistics is found. The computation of ϵ_P, ϵ_C and ϵ_{max} is performed at each iteration of the algorithm. The convergence point is reached when ϵ_P, ϵ_C and ϵ_{max} are lower than 1. The quality of the inference can be also tested on non-fit statistics as 3-point correlations and mutational probability. The latter is a biological interesting observable, since it represents the probability of mutation[1] of a given number of sites per sequence. Note that, sequence similarity, as we discussed in chapter 3, plays a fundamental role in the classification and in the modelling of sequence homology and a good characterisation of related quantities, such as the mutational probability, remains challenging in the field. In next sections

1. Consider a MSA. The sequence composed by the most frequent symbol on each column is called *consensus sequence*. A mutation occurs when on a certain site a sequence express a symbol different from the consensus. Within an experimental framework mutations are usually computed with respect to the *wild-type sequence*.

I will often consider the so called *generative test*, meaning the comparison between the statistics computed from the data and the statistics computed with the Monte Carlo sampling. Quantities considered within the generative test are:

- 2-point connected correlations [2]
- 3-point connected correlations
- $P(k)$, is the probability of observe k mutation per sequence with respect to a reference sequence, called *consensus sequence*, in which on each site the most probable a.a. for that site is taken.

Lastly, also the ability to reproduce the network of interactions is verified and therefore true positive rates and contact maps, similar to those we analysed in chapter 3, will be considered.

5.2 Computational refinements for the Potts case

As discussed before, the complexity of ACE algorithm depends exponentially on the number of colours in the model, therefore the extension of this method to the Potts model requires some adjustments of existing routines and the introduction of new intermediate steps between the extraction of data and the exploitation of results.

In the following section I will present the improvements we have introduced in the last version of the ACE algorithm in order to easily manage many colour Potts data. I will first focus on some possible refinements of input data in the scope of reducing computational costs, then I will introduce the concept of *reference structure* and the different ways it can be used to guide the inference. Moreover, we will analyse two improvements of the code acting on the exact computation of the cluster log-likelihood, being the latter the bottleneck of the algorithm. Finally, I will explain how we have combined our algorithm with Boltzmann Machine Learning when the convergence of the ACE results to be too slow.

5.2.1 Colour compression

A first procedure we have introduced consists in fitting the minimal number of parameters per site. Observing real data, such as MSAs of RNAs and proteins, several sites contain much less than 5 or 21 symbols. Functional constraints, joint to

2. This quantity is actually fit by our algorithm, thus, to some extent, it can be considered as a lax generative test. We introduce it in the analysis to remark the difference with other inference methods (e.g. mean-field approaches) not able to reproduce these observables, even if they are used to fit the model.

the finiteness of samples, prevent us from observing all the possible amino acids or nucleotides at least one time per column. Being the number of colours so crucial in terms of computational time, we have decided to force the algorithm to fit a *restricted* Potts model, where the number of colours per site q_i depends on the considered site and it corresponds to the number of effectively observed symbols.

Discarding non-observed colours, meaning colours with frequency equal to zero, does not entail any loss of information. However, frequency thresholds larger than zero can be considered and we have demonstrated they give rise to reasonable approximations: the need for computational feasibility can be fruitfully paid in term of information about the system. We compress all those colours whose frequency is smaller than a threshold p in a single grouped colour $\tilde{q}_i$. We leave all the other colours unchanged. The threshold p can be fixed choosing for instance a minimum number of required observations within the sample, or such that a certain fraction of the overall entropy of the site has been reproduced:

$$\begin{aligned} S_{q_i} &= -\sum_{a=1}^{q_i-1} f_i(a)\log f_i(a) - \left(1-\sum_{a=1}^{q_i-1} f_i(a)\right)\log\left(1-\sum_{a=1}^{q_i-1} f_i(a)\right) \\ &\geq f S_q\,. \end{aligned} \tag{5.9}$$

Contributions to S_{q_i} are progressively added according to decreasing frequency. We will see in the results section that this colour compression scheme barely decreases the quality of the inference, entailing instead a huge gain in computational tractability of diverse systems. Moreover we have observed on protein data that the use of colour compression helps avoiding over-fitting and improves inference quality.

5.2.2 Reference structure

The colour compression reduces the amount of information about the system for speed, conversely a similar gain in term of computational feasibility can be achieved adding more information, when available. Within many applications of interest, indeed, some partial information on the system are sometimes available: RNA alignments usually contain a consensus secondary structure, experimental knowledges about the interaction network can be found in the literature and even faster inference methods can be used for contact map prediction.

Unveiling the network of interactions underlying a certain system is often the first aim of inference on biological data [3]. However also the strength of interactions, the configuration probabilities or in general a more detailed characterization of the

distribution of symbols turns out to be of interest in many cases, for instance when experimental measurements about sequence folding probability are available [28]. In those situations the ACE can be used to refine the description of the system, given the interaction network as reference structure. We consider a restricted selection of sites for the construction of clusters based on the known interaction graph: we firstly include all the clusters of size two whose sites are directly interacting and then we build larger clusters using only those sites included in the initial list.

Similarly, one can use ACE with different levels of data compression: first we run the ACE with an high compression threshold for colours: in the extreme case, an Ising model, where $\sigma = 1$ corresponds to the most frequent colour and $\sigma = 0$ to the compression of all the other ones, can be inferred. At convergence the final list of selected clusters is recorded and then submitted to a second run of the algorithm, whose target is a non-compressed (or less-compressed) model. The gain in term of time depends on the fact that only the clusters in the list will be computed and no selection will be performed: all the computed clusters will contribute to the overall cross-entropy.

Finally, in the original Ising version of the ACE [23] an expansion of the cross-entropy around the mean-field solution has been introduced in order to help the numerical optimisation of the log-likelihood. Note that the result of this procedure is that ACE provides an expansion around a reference Gaussian model instead of around a reference structure.

5.2.3 Analytical computation of 2-site clusters

When q is of order 20, the computation of clusters of size two still requires a long computational time. However, the exact solution for the q-state Potts model inference when $N = 2$ is known: the probability of a configuration (σ_1, σ_2) for the two variables is expressed as

$$P_{12}(\sigma_1, \sigma_2) = e^{h_1(\sigma_1)+h_2(\sigma_2)+J_{12}(\sigma_1,\sigma_2)} \tag{5.10}$$

The conditional probability of having σ_2 in position 2 given σ_1 in position 1 is instead $P(2, \sigma_2|1, \sigma_1) = e^{h_2(\sigma_2)+J_{12}(\sigma_1,\sigma_2)}$; by rewriting $P_{12}(\sigma_1, \sigma_2) = f_1(\sigma_1)P(2, \sigma_2|1, \sigma_1) = f_1(\sigma_1)e^{h_2(\sigma_2)+J_{12}(\sigma_1,\sigma_2)}$ and comparing with Eq. (5.10) we obtain $f_1(\sigma_1) = e^{h_1(\sigma_1)}$ thus:

$$h_1(\sigma_1) = \log f_1(\sigma_1) \tag{5.11}$$

an analogous expression is obtained for $h_2(\sigma_2)$. Substituting expression (5.11) for $h_1(\sigma_1)$ and $h_2(\sigma_2)$ in (5.10) we obtain

$$J_{12}(\sigma_1, \sigma_2) = \log \frac{f_{12}(\sigma_1, \sigma_2)}{f_1(\sigma_1)\, f_2(\sigma_2)} \tag{5.12}$$

It is easy to verify that the conservations of probabilities $\sum_{a=1}^{q} f_i(a) = 1$, $\sum_{a=1}^{q} f_{ij}(a,b) = f_j(b)$, and $\sum_{a=1}^{q}\sum_{b=1}^{q} f_{ij}(a,b) = 1$ are satisfied by the above choice of parameters $h_1(\sigma_1), h_2(\sigma_2)$ and $J_{12}(\sigma_1, b)$.

Note that the above equations for the couplings and fields are also obtained by deriving the minimal cross-entropy, for a system of 2 spins with respect to the fields and couplings, which can be rewritten as

$$\begin{aligned} S_\Gamma^{An} = &\sum_{a,b} f_{12}(a,b) \log \frac{f_{12}(a,b)}{f_1(a) f_2(b)} \\ &+ \sum_a f_1(a) \log f_1(a) + \sum_b f_2(b) \log f_2(b) \end{aligned} \tag{5.13}$$

where $\Gamma = \{1, 2\}$ is the considered sub-system of size two. Thus in order to speed the algorithm we introduce the analytical computation of 2-site clusters. However, as we have already seen, we introduce in the computation of the cross-entropy a $L2$-norm regularisation term, as in Eq. 5.2, to compensate finite-sampling errors. Therefore the analytical solution S^{An} turns out to be different from the regularised cluster entropy S_Γ. Anyway, in principle, parameters in Eqs. 5.11 and 5.12 can be used as an initial guess for minimisation of the regularised S_Γ. The optimisation of the cluster cross-entropy is performed using both the gradient descent and the Newton method, depending on the value of the gradient [3] and several tests we have performed on artificial data have shown that this non-zero guess does not reduce significantly the computational time: within the very first steps the gradient descent reaches an approximation of the final results that is extremely close to the one computed from S_Γ^{An}.

The use of a pseudocount regularisation instead of the $L2$-norm can be useful in this framework. Pseudocounts consist in a rescaling of correlations and frequencies able to correct finite-sampling effects and do not need further regularisations (in Eq. 5.2 $\gamma = 0$ and $\gamma' = 0$). Being $S_\Gamma^{An} = S_\Gamma$ 2-site clusters can actually be computed analytically dramatically increasing the speed of the algorithm.

3. The advantage of using gradient descent or Newton method depends on the value of the gradient of the target function: when the gradient is too small gradient descent steps become negligible and the algorithm gets stuck, so in those cases we prefer to compute also the hessian term and perform Newton steps in order to speed the optimisation.

5.2.4 Sparse regularisation

A computational refinement, suggested by J.P Barton, is to perform an efficient expansion of the partition function in order to use sparsity of couplings in the scope of decreasing computational costs. We firstly observe that, given the Potts Hamiltonian in 1.1, the partition function can be written in a trivial form in case of independent spins:

$$Z = \prod_{i=1}^{N} \left(\sum_{a=1}^{q_i} e^{h_i(a)} \right) \tag{5.14}$$

Since fields $h_i(a)$ make independent contributions to the energy, the sum over all configurations can be rewritten as a product of terms from each site. The gain in term of complexity is evident: 5.14 requires only $\sum_{i=1}^{N} q_i$ operations rather than $\prod_{i=1}^{N} q_i$.

If we assume the sparsity of the interaction graph, we can expand the partition function ignoring loops in a tree-like expansion.

Finally is it also possible to use a $L0$-norm regularization, which can be very useful in the case of a large number of effective colours per site. This regularisation, applied on coupling only, enforces the sparsity of the inferred model:

$$\Delta\ell = -\gamma_0 \sum_{i=1}^{N-1} \sum_{j=i+1}^{N} \sum_{a=1}^{q_i} \sum_{b=1}^{q_j} \|J_{ij}(a,b)\|_0 \,. \tag{5.15}$$

$L0$-regularization force those couplings that do not increase the log-likelihood of the model by at least γ_0 to be exactly zero. The form of the regularization was implemented following the adaptive forward-backward algorithm of [107].

5.2.5 MC-learning refinement

The ACE procedure is extremely fast as long as it does not reach cluster sizes for which the computational cost for the calculation of the partition function becomes prohibitive. As I have already discussed, computational time grows exponentially on average as q_{eff}^k, where $q_{eff} = \frac{1}{N}\sum_{i=1}^{N} q_i$. Typically the calculation of the partition function requires a sum over 10^5 configurations for clusters of about size 16, 8, 5 for $q = 2$, 5, 10, respectively. When the ACE enters this regime it is better to stop the algorithm, even if ϵ_{max} is not yet of order one, and to use the output fields and couplings as initial guess for a MC-learning procedure. This values of the parameters are usually good initial guesses and the MC-learning rapidly converges.

The learning algorithm we implemented is a Potts-adapted version of RPROP algorithm for neural network learning [108]. Given an input set of fields and couplings, we first compute the model correlations $f_i^{MC}(a), f_{ij}^{MC}(a,b)$ through Monte Carlo simulations, implemented in the code by J.P. Barton. The couplings and fields are then updated according to the gradient of the log-likelihood, multiplied by a parameter-specific weight factor

$$\begin{aligned} h_i(a) &\to h_i(a) - \left(f_i^{MC}(a) - f_i(a)\right) w_i(a)\,, \\ J_{ij}(a,b) &\to J_{ij}(a,b) - \left(f_{ij}^{MC}(a,b) - f_{ij}(a,b)\right) w_{ij}(a,b) \end{aligned} \tag{5.16}$$

Regularization can also be incorporated by adding $2\gamma J_{ij}(a,b)$, or the analogous term for fields, to the gradient.

The use of the MC-learning refinement goes beyond those cases when convergence cannot be found. Sometimes, indeed, a quite good solution is found, then, lowering t, the error rises and we have to wait long time before a new set of parameters providing a smaller error has been found. It has been proved in [23] that the error is not a monotonous function of t and several local minima of the error exist: remember that the entropy summation needs the cancellation of many cluster contributions to converge. Therefore, we not only record parameters at convergence, but also in correspondence of some local minima of the error. This intermediate t values can be used to run the MC-learning refinement and thus to obtain reliable parameters in a shorter time.

5.3 Gauge choice

As many other models in theoretical physics the Potts model is invariant under so called gauge transformations, meaning for any K the following transformations

$$\begin{aligned} J_{ij}(a,b) &\to J_{ij}(a,b) + K_{ij}(b) \\ h_i(a) &\to h_i(a) - \sum_{j \neq i} K_i(a) \end{aligned} \tag{5.17}$$

entail no changes on the probability distribution 1.6 and thus on all those quantities that are related to it. The gauge invariance comes from the conservation laws of probabilities we have cited in section 5.2.3, responsible for removing some degrees of freedom from the system. Thus, the number of independent fields at each site i is $(q_i - 1)$ instead of q_i, and the number of independent couplings for each pair of sites

is $(q_i - 1)(q_j - 1)$. Given a set of couplings $J_{ij}(a, b)$ and fields $h_i(a)$ and chosen a certain colour c_i we can fix the gauge such that

$$J_{ij}(\sigma_i, c_j) = J_{ij}(c_i, \sigma_j) = J_{ij}(c_i, c_j) = h_i(c_i) = 0 \tag{5.18}$$

To implement these constraints we define the transformed couplings $\tilde{J}_{ij}(a, b)$ as follows

$$\tilde{J}_{ij}(\sigma_i, \sigma_j) = J_{ij}(\sigma_i, \sigma_j) - J_{ij}(c_i, \sigma_j) - J_{ij}(\sigma_i, c_j) + J_{ij}(c_i, c_j) \tag{5.19}$$

Thus, we are forced to modify fields according to equation 5.19 such that for each configuration the energy is unchanged, unless for constant terms. The original hamiltonian is $H(\boldsymbol{\sigma}) = \sum_{j>i} J_{ij}(\sigma_i, \sigma_j) + \sum_i h_i(\sigma_i)$, while after the gauge fixation of the couplings we obtain:

$$\tilde{H}(\boldsymbol{\sigma}) = \sum_{j>i} J_{ij}(\sigma_i, \sigma_j) - \sum_{j>i} J_{ij}(c_i, \sigma_j) - \sum_{j>i} J_{ij}(\sigma_i, c_j) + \sum_{j>i} J_{ij}(c_i, c_j) + \text{field terms} \tag{5.20}$$

discarding the last J term that does not depend on a or b, there are still two terms to be cancelled out with a suitable transformation of fields:

$$\tilde{h}_i(\sigma_i) = h_i(\sigma_i) - h_i(c_i) + \sum_{j>i} [J_{ij}(\sigma_i, c_j) - J_{ij}(c_i, c_j)] + \sum_{j<i} [J_{ji}(c_j, \sigma_i) - J_{ji}(c_j, c_i)] \tag{5.21}$$

Note that, since any transformation of the type 5.17 is permitted, it is important to first choose a particular gauge before comparing inferred parameters, e.g. when we consider artificial data and we plot true parameters versus inferred ones or when we compare results from two different inference methods. The choice of the gauge can be different between the true and the inferred model (typically in artificial model we draw the parameters from some random distribution for the complete q-state Potts model and then we infer a model in a certain gauge with $q - 1$ colours).

When we infer couplings and fields with ACE, we are forced to regularise the cross-entropy to solve finite-sampling issues and also to help the gradient descent optimisation to find rapidly the maximum of the log-likelihood. Even if the inference itself is gauge invariant, and thus the result does not depend on the choice for the gauge, the regularisation term is not. Thus, depending on the studied model, an appropriately choice for gauge can help or not convergence. In this section we will analyse some results about the role of the gauge in ACE inference. We will analytically

compute finite-sampling errors for small models (i.e. $N = 15$) and we will average results on 100 different and randomly chosen realisations with similar characteristics in term of number of sites, number of colours and sample size.

In order to understand better the point of this analysis, consider the case of an artificial model whose inference has been performed with the colour compression approximation: we have obtained a model that is slightly different from the original one and we need to specify a certain gauge in order to understand results. Usually, we perform the inference in gauge of Eq. 5.18 where the colour fixed to zero, c_i, corresponds to the grouped colour $\tilde{q}_i$ on each site. This is the most natural choice because it does not need any rearrangement of magnetisations and correlations. However, since we do not have for parameters a closed relation linking the grouped colour and the colours inside the group, we cannot easily convert the true couplings and fields to this gauge. We have to convert both the inferred and the true parameters into another gauge, hoping it will not spread too much errors (we will see extensively in the next section how errors propagate from one gauge to the others). When no compression is performed there is no need, in principle, for chosing two different gauges for the inference and the comparison. Anyway, we will show that some advantages can derive from the choice of an appropriate gauge, different from the usual zero-sum gauge, before computing the Frobenius.

In the following we will consider a set of simple toy models on which we can analytically compute finite-sampling errors and we will compare results on different choices for the gauge, both for the inference and for the comparison between true and inferred parameters.

5.3.1 Finite-sampling error on parameters and its propagation

The information about finite-sampling errors made on inferred parameters, is contained, equivalently to errors on frequencies and correlations, in the so called *Fisher Information Matrix* χ, corresponding to the susceptibility matrix of the system. When the model is small ($N \sim 10$ and $q_i \sim 5$) and the sample too ($B \sim 10^4$) the susceptibility matrix can be easily inverted and thus errors over couplings and fields can be analytically computed

$$\delta J_{ij}(\sigma_i, \sigma_j) = \sqrt{\frac{1}{B}(\chi^{-1})_{ij,ij}} \tag{5.22}$$

$$\delta H_i(\sigma_i) = \sqrt{\frac{1}{B}(\chi^{-1})_{i,i}} \tag{5.23}$$

The above defined errors are gauge invariant quantities, but, given the finiteness of the sampling, some regularisation of χ is needed. Thus, before inversion, the following term (resulting from a $L2$ regularisation of the cross-entropy with $\gamma = \frac{1}{B}$ for couplings and $\gamma' = 0.01\gamma$ for fields) is added to the diagonal elements of χ:

$$\chi_{ij,ij} \to \chi_{ij,ij} + \gamma \tag{5.24}$$

$$\chi_{i,i} \to \chi_{i,i} + \gamma' \tag{5.25}$$

These regularised errors are no more gauge invariant, thus also in this case the choice of the gauge can modify results. Moreover, in the scope of comparing results, we change the gauge of parameters also after the inference, thus we have to propagate errors computed in the inference gauge into the gauge chosen for comparison. The gauge transformations 5.19 and 5.21 can be rewritten in matrix form, where **JH** is the vector containing the list of all fields and couplings, as:

$$\tilde{\mathbf{JH}} = \mathbf{A} * \mathbf{JH} \tag{5.26}$$

where $\mathbf{A}$ is a $\left(\frac{N(N-1)}{2}q^2 + Nq\right) \times \left(\frac{N(N-1)}{2}q^2 + Nq\right)$ binary matrix selecting terms in the vector **JH** according to 5.19 and 5.21.

Given matrix χ^{-1} we can select the elements in the **JH** list (in case some compression of colours has been performed) and, finally, propagate errors according to usual rule:

$$\tilde{\chi}^{-1} = \mathbf{A} * \chi^{-1} * \mathbf{A^t} \tag{5.27}$$

5.3.2 Small systems analysis

We consider 100 different toy models with $N = 15$, $Q = 5$ and Erdos-Renyi interaction network with parameters similar to the ones showed in the following picture:

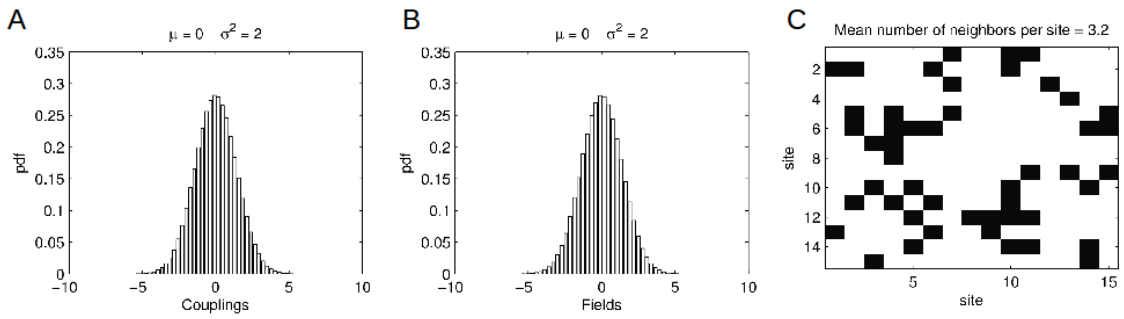


Figure 5.1: (A) and (B): Gaussian distributions from which true parameters have been drawn. We fix $\mu = 0$ and $\sigma^2 = 2$ for both couplings and fields. (C) The interaction network is a Erdos-Renyi random graph with $p = 0.05$, where p is the probability to have a link between two sites. In the following we will call the interaction network *contact map* in analogy to biological cases.

To compute correlations we use Monte Carlo sampling collecting 10^4 different configurations. Given the small size of these systems we easily compute analytical errors on couplings and fields and thus we use them to estimate the relative average errors on the inference, in analogy to what we do with correlations and magnetisation within the ACE.

$$\epsilon_J = \sqrt{\frac{1}{\frac{N(N-1)}{2}q^2}\sum_{i<j}\sum_{\sigma_i,\sigma_j}\frac{(J_{ij}^{inf}(\sigma_i,\sigma_j) - J_{ij}^{true}(\sigma_i,\sigma_j))^2}{\delta J_{ij}^2(\sigma_i,\sigma_j)}} \tag{5.28}$$

$$\epsilon_H = \sqrt{\frac{1}{Nq}\sum_{i}\sum_{\sigma_i}\frac{(H_{i}^{inf}(\sigma_i) - H_{i}^{true}(\sigma_i))^2}{\delta H_i^2(\sigma_i)}} \tag{5.29}$$

Then, being in real-life cases interested in the inference of the *contact map* of the system, we compress the information inside couplings with a scalar score of interaction: the Frobenius norm of the matrix $J_{ij}(\sigma_i, \sigma_j)$. We add the so called *Average Product Correction* to reduce entropic contribution to the score. Therefore we obtain the score F_{ij}^{apc} defined as follows:

$$F_{ij} = \sqrt{\sum_{\sigma_i,\sigma_j} J_{ij}(\sigma_i,\sigma_j)^2} \qquad\qquad F_{ij}^{apc} = F_{ij} - \frac{\langle F_{ij}\rangle_i \langle F_{ij}\rangle_j}{\langle F_{ij}\rangle_{ij}} \tag{5.30}$$

Before computing F_{ij} we usually fix couplings in the the zero-sum gauge: the latter ensures the minimum value for F_{ij} (see chapter 3 for details).

Given the list of all F_{ij}^{apc} we consider the rank correlation between F_{true}^{apc} and F_{inf}^{apc}. When such simple artificial models are concerned, the true positive rate is indeed poorly informative because it reaches rapidly its maximum value. A finest observable is then needed to measure inference performance. We define ρ as follows:

$$\rho = \frac{1}{\sigma_r(F_{true}^{apc})\sigma_r(F_{inf}^{apc})nz}\sum_{k=1}^{nz}\left(k - \frac{nz+1}{2}\right)\left(r((F_{true}^{apc})_{i_k,j_k}) - \bar{r}^{true}\right) \tag{5.31}$$

where nz is the number of non-zero coupled pairs and r is the ranking of the pair according to true couplings (cf. with paper in chapter 4).

Finally we consider also the performance of the inferred parameters in reproducing the input statistics computing their *Root Mean Square Deviation* with respect to input magnetisations, 2-point connected correlations and 3-point connected correlations.

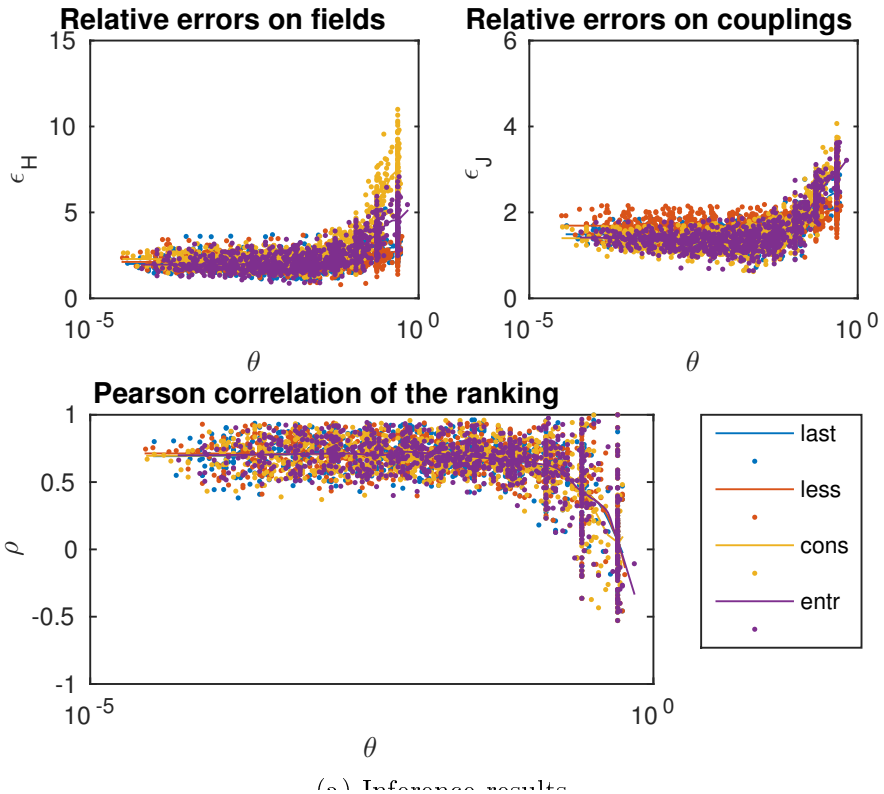
All the above described quantities are computed on 4 different sets of parameters coming from the same model, but obtained running the ACE with a different gauge fixing of the type in Eq. 5.18:

- Last: c_i is chosen to be the last colour (it correspond to a random choice)
- Less: c_i is chosen to be the least frequent colour
- Cons: c_i is chosen to be the most frequent (consensus) colour
- Entr: c_i is chosen to be the maximum entropy colour

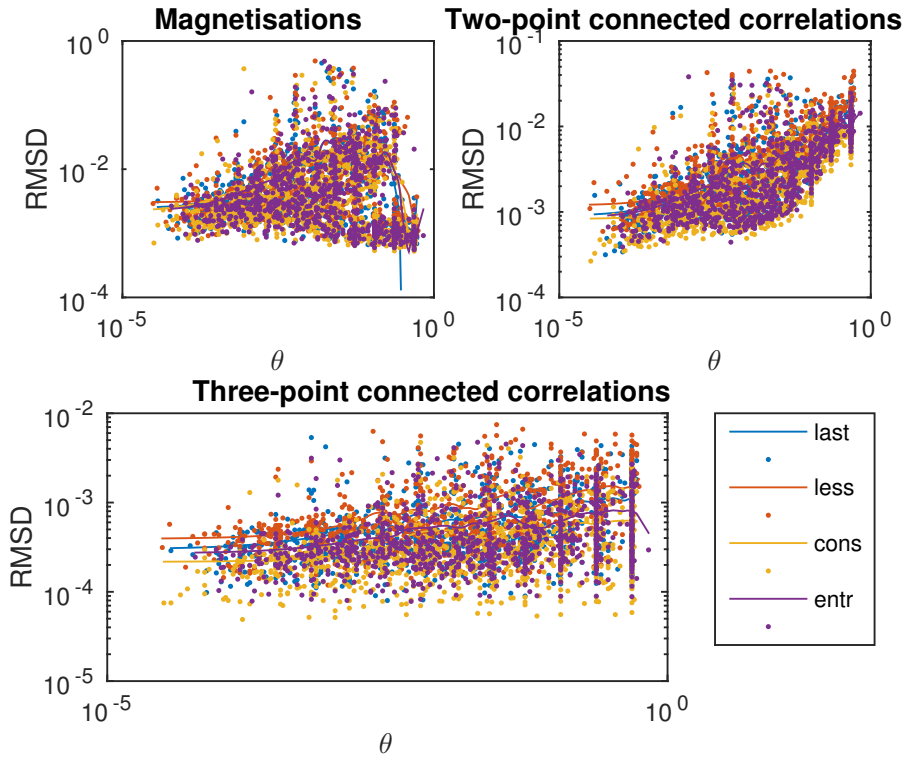
Different inference gauges In figure 5.2 we show results obtained using different gauges within the inference.

As you can see the 4 gauges perform similarly on average on the datasets, both within the inference (Fig. 5.2a) and within the generative test (Fig. 5.2b). Note that for small θ (Fig. 5.2a top left) the consensus gauge performs worse than the others for fields inference: the consensus gauge, indeed, forces true fields to be negative, while the inferred ones are less negative than expected thanks to regularisation. Regarding couplings (Fig. 5.2a top right), the least frequent gauge seems to be significantly worse than the others, since finite-sampling effects are more pronounced in this particular gauge. However ranking (Fig. 5.2a bottom) is well inferred in any case, even if there is a huge variation (from 0.6 to 0.9) depending on the model. Finally, as expected, neither the ranking nor the statistics depends on the choice for the inference gauge.

Different comparison gauges Chosen a certain gauge for the inference we can then change the gauge before comparing results with true parameters and before computing the F^{apc}_{inf}, F^{apc}_{true} and the ρ (see Fig. 5.3 and 5.4). However with this study we show that the choice of comparison gauge is not crucial: there are some small differences among different gauges but we cannot recognise a significant trend in our results. Comparing ρ curves in Figs. 5.3, 5.4 with in Fig. 5.2 we can anyway say that the consensus gauge can actually produce better results in term of ranking than the zero-sum gauge. This evidence has suggested that contact map predictions on biological data can be improved thanks to the use of the consensus gauge. We have tested the consensus gauge on protein data and we have obtained better contact predictions with respect to the usual zero-sum gauge, the same used also within DCA.

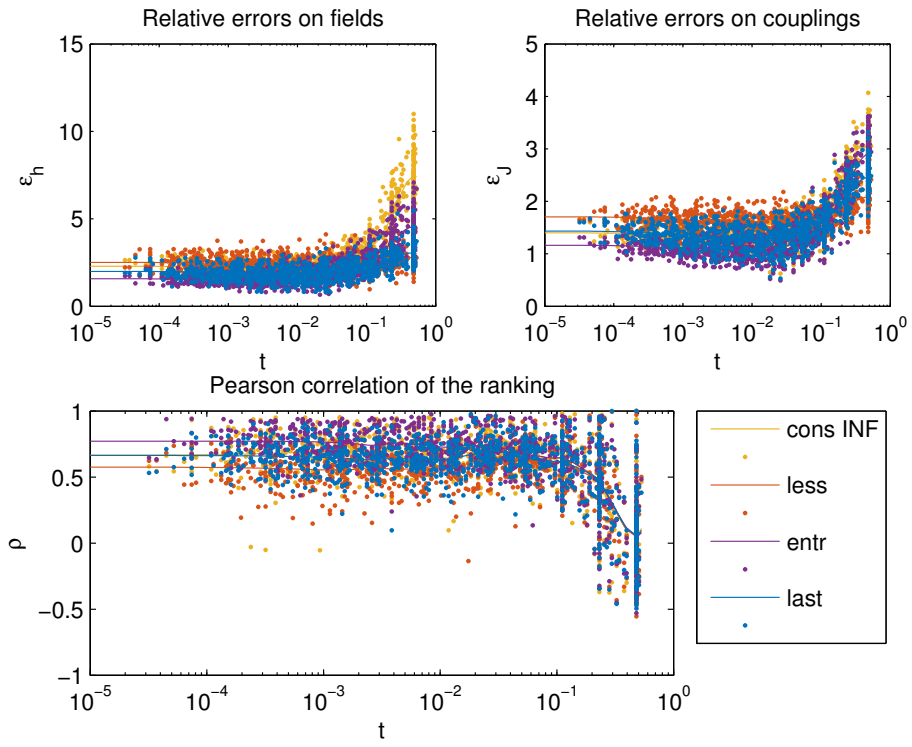


(a) Inference results

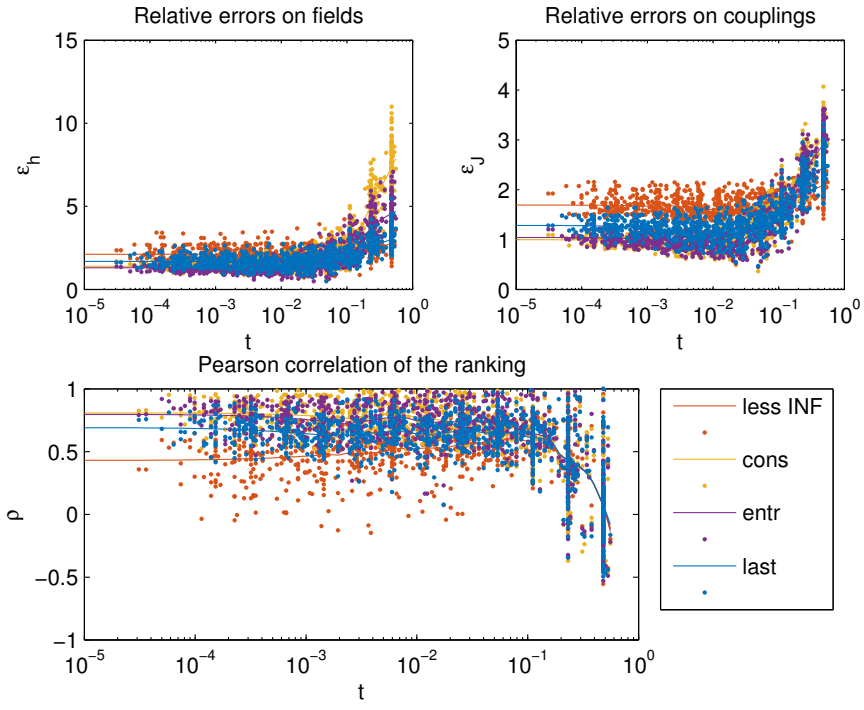


(b) Generative test

Figure 5.2: Points: results obtained on 100 models. Lines: smooth averages of points to guide eye. Couplings and fields are compared in the gauge used for the inference. To compute ρ couplings have been moved to the zero-sum gauge.

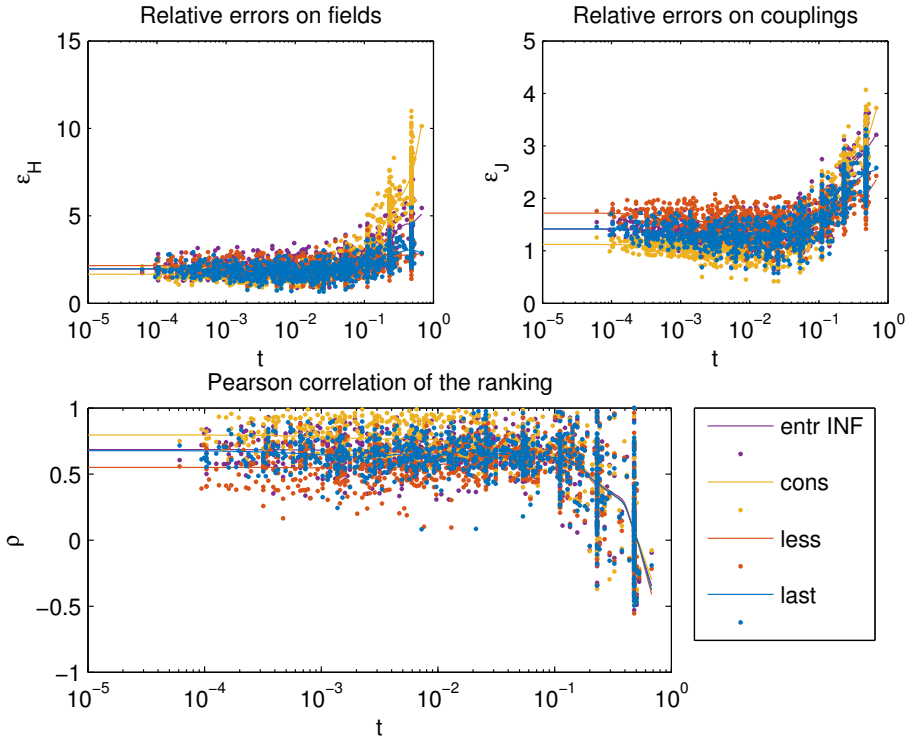


(a) Cons for inference

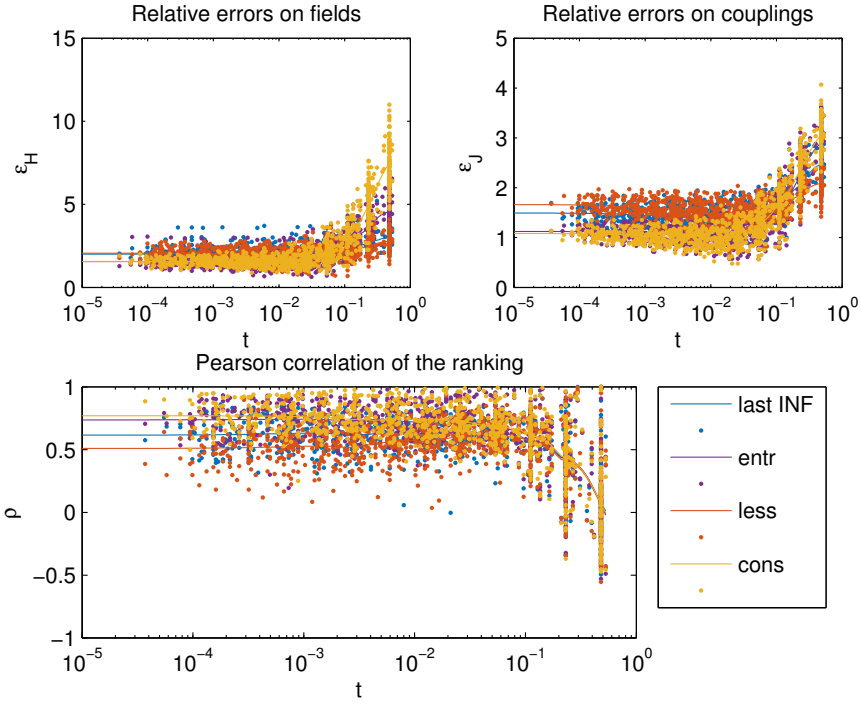


(b) Less for inference

Figure 5.3: Points: results obtained on 100 models. Lines: smooth averages of points. Inference has been made in a certain gauge ((a) most frequent colour, (b) least frequent colour) and then parameter have been moved to the other gauges before computing of ϵ_J, ϵ_H and F^{apc}.



(a) Entr for inference



(b) Last for inference

Figure 5.4: Points: results obtained on 100 models. Lines: smooth averages of points. Inference has been made in a certain gauge ((a) maximum entropy colour, (b) last colour) and then parameter have been moved to the other gauges before computing of ϵ_J, ϵ_H and F^{apc}.

5.3.3 Gauge invariant regularization of the couplings

As we have seen above, the non-gauge-invariance of the $L2$-norm regularisation and consequently the arbitrary choice of the inference and the comparison gauge can modify results, in particular when a strong regularisation is used. Therefore we have included in the code a gauge-invariant modification of the $L2$-norm so to ensure gauge-invariant results.

Instead of an $L2$-norm penalty on $J_{ij}(a,b)$, we introduce the regularisation on a transformed coupling value

$$\begin{aligned} K_{ij}(a,b) = J_{ij}(a,b) - \frac{1}{q_j}\sum_{c=1}^{q_j} J_{ij}(a,c) - \frac{1}{q_i}\sum_{c=1}^{q_i} J_{ij}(c,b) \\ + \frac{1}{q_i q_j}\sum_{c=1}^{q_i}\sum_{d=1}^{q_j} J_{ij}(c,d)\,. \end{aligned} \tag{5.32}$$

One can then verify that $K_{ij}(a,b)$ is invariant under gauge transformations, and thus an $L2$-norm regularization of the form

$$\gamma \sum_{i=1}^{N-1}\sum_{j=i+1}^{N}\sum_{a=1}^{q_i}\sum_{b=1}^{q_j} K_{ij}(a,b)^2 \tag{5.33}$$

does not depend on the choice of gauge. As far as fields are concerned, the regularisation parameter γ' has to be set to zero to ensure the gauge-invariance of the final model. Unfortunately we empirically observe that the gauge-invariant regularisation has a negative impact on the routine optimisation of the log-likelihood: optimisation results to be slightly slower than the standard $L2$-norm.

5.3.4 Approximated error on the inferred parameters

We have seen at beginning of this section that the covariance matrix χ can be used to estimate finite-sampling errors on both correlations and parameters. However the inversion of χ is computationally infeasible for long sequences and for large q, since it has size $\left(qN + q^2(\frac{N(N-1)}{2})\right) \times \left(qN + q^2(\frac{N(N-1)}{2})\right)$. Some approximate values for errors are needed in most biologically interesting cases (protein sequences typically have $N \sim 100$ and $q \gtrsim 10$). Having observed that a strong contribution to the variances comes from out-diagonal terms of χ, meaning couplings and fields are far from being independent variables, we exclude from putative candidates the approximation $\chi^{-1}_{iajb,iajb} = \dfrac{1}{\chi_{iajb,iajb}}$. Considering interactions among colours more relevant than interactions among different pairs of sites, we define for each pair ij a reduced

$\hat{\chi}_{ij}$ corresponding to the hessian of a two-spin model. From the inversion of all the possible $\hat{\chi}_{ij}$, whose size is $(2q+q^2)\times(2q+q^2)$, we obtain, indeed, a reliable approximation for the $\delta J_{ij}(a,b)$. As regard to fields, this approximation does not give us a unique value: the same term $\delta H_1(a)$ can be found inverting $\hat{\chi}_{1i}$ for whatever $i>1$, thus we can, for instance, consider the average value among all the possible ones. Alternatively we can choose to propagate the errors on magnetisations $\delta f_i(a)$ and correlations $\delta f_{ij}(a,b)$ using the 2-variable approximation seen before:

$$J_{ij}(a,b) = \log\left(\frac{f_{ij}(a,b)}{f_i(a)f_j(b)}\right), \qquad h_i(a) = \log f_i(a)$$

Note that because of the finiteness of the sample a regularisation term is needed:

$$f_i(a) \to f_i(a) + \frac{1}{B}, \qquad f_{ij}(a,b) \ \to f_{ij}(a,b) + \frac{1}{B}$$

We then propagate errors on the gauge transformations described in 5.17 and obtain an approximate formula for inferred fields and couplings in the comparison gauge on colour c_i:

$$\begin{aligned} h_i(a) &= \log f_i(a) - \log f_i(c_i) + \sum_{j=1}^{N}\left(\log\left(\frac{f_{ij}(a,c_j)}{f_i(a)f_j(c_j)}\right) - \log\left(\frac{f_{ij}(c_i,c_j)}{f_i(c_i)f_j(c_j)}\right)\right) \\ J_{ij}(a,b) &= \log f_{ij}(a,b) - \log f_{ij}(c_i,b) - \log f_{ij}(a,c_j) + \log f_{ij}(c_i,c_j) \end{aligned} \tag{5.34}$$

Finally, the corresponding error terms for the fields and couplings due to finite-sampling are given by

$$\begin{aligned} \delta h_i(a) = (N-2)\sqrt{\frac{1-f_i(a)}{B\,f_i(a)}} + (N-2)\sqrt{\frac{1-f_i(c_i)}{B\,f_i(c_i)}} \\ \sum_{j\neq i}\left(\sqrt{\frac{1-f_{ij}(a,c_j)}{B\,f_{ij}(a,c_j)}} + \sqrt{\frac{1-f_{ij}(c_i,c_j)}{B\,f_{ij}(c_i,c_j)}}\right), \end{aligned} \tag{5.35}$$

$$\begin{aligned} \delta J_{ij}(a,b) = \sqrt{\frac{1-f_{ij}(a,b)}{B\,f_{ij}(a,b)}} + \sqrt{\frac{1-f_{ij}(c_i,b)}{B\,f_{ij}(c_i,b)}} \\ + \sqrt{\frac{1-f_{ij}(a,c_j)}{B\,f_{ij}(a,c_j)}} + \sqrt{\frac{1-f_{ij}(c_i,c_j)}{B\,f_{ij}(c_i,c_j)}}. \end{aligned} \tag{5.36}$$

Type of approximation	fields	couplings
Analytical errors (independent variables propagation)	0.583 ± 0.007	0.784 ± 0.004
2-site inversion of χ	0.665 ± 0.006	0.938 ± 0.002
2-site inversion of χ (independent variables propagation)	0.600 ± 0.007	0.788 ± 0.004
2-variable approximation	0.919 ± 0.004	0.953 ± 0.001

Table 5.1: Table showing the Pearson correlation between the different error approximations and the analytical errors. Averages are made on a sample of 100 different Erdos-Renyi models whose parameters are defined in Fig. 5.1

In table 5.1 we compare the Pearson correlation between the approximated error estimations and the analytical one. Note that approximations are made at two different levels: we approximate the inversion of the χ in Eq. 5.23 and 5.22 (2-site inversion of χ) and we approximate the propagation of errors on the comparison gauge shown in Eq. 5.27. We use the independent variables approximation, i.e. only variances are considered and any covariance is discarded. Remember, indeed, that the gauge we use within the inference is usually different from the gauge we use to compare true and inferred parameters, thus errors on the inferred parameters have to be propagated to the final gauge. As far as the 2-variable approximation is concerned the propagation on the gauge is computed analytically in 5.36 and 5.35.

Results shown here confirm our choice: in the following we will use the 2-variable approximated errors. As you can see from table 5.1 the 2-site inversion of χ and the 2-variable approximation are, as may be expected, almost equivalent for couplings, but the latter outperforms the former for fields. Note, finally, that the independent variable approximation for the error propagation on the gauge highly deteriorates results even if the analytical errors are considered.

5.4 ACE applications

5.4.1 Artificial data

In order to test the quality of the inference made by the ACE, we study some artificial models whose parameters (couplings and fields) are randomly chosen from Gaussian distributions; we fixed $\mu = 0$ and $\sigma^2 = 5$ for fields and $\mu = 0$ and $\sigma^2 = 1$ for couplings, according to what we has been observed on protein data inference. The networks of interactions are Erdos-Renyi random graphs with 50 nodes generated with $p = 0.05$ and $p = 0.1$, respectively called ER005 and ER010, where p is the probability

to have a link between two sites. Regarding colours, no preferential scheme is imposed, i.e. if i and j interact then J_{ij} is a 21×21 matrix whose elements are chosen according to the above defined Gaussian distribution. The models we obtained have a maximum connection equal to 7 for ER005 and 12 for ER010; the number of interacting sites is 61 for ER005 and 121 for ER010.

Given the set of couplings and fields, a Monte Carlo routine is used to generate data in the form of a *Multiple Sequence Alignment* with $B = 10^2$, $B = 10^3$, $B = 10^4$ and $B = 10^5$ unbiased sequences. Then, two different colour compressions have been applied to the dataset: for each site, colours with magnetisation $f_i(a_i) < 0.05$ and $f_i(a_i) < 0.01$, are compressed in a single *gauge* colour c_i, fixed such that $J_{ij}(a_i, c_j) = J_{ji}(a_j, c_i) = h_i(c_i) = 0$. Consequently an effective number of colours ($q_i^{eff} \leq 21$) for each site is defined and only $q_i^{eff} - 1$ colours will be used in the inference of this *colour-compressed model*. Finally a small ($\gamma = \frac{1}{B}$) $L2$-regularisation is included in the computation of the cross-entropy.

Results for model ER005 with $f_i(a_i) > 0.05$ In the following I will summarise the results obtained running ACE on the ER005 model with colour reduction $f_i(a_i) > 0.05$.

The behaviour of the inference depending on threshold for the cluster selection can be appreciated in Fig.5.5 and Fig.5.6. We show how the cross-entropy of the inferred model, errors on the statistics, the number and the size of the selected clusters have changed depending on t within the 4 different sample sizes analysed. Firstly note that the intermediate plateau in the cross-entropy (we can easily see it for $B = 10^3$, but it is still there also for the other samples) corresponds to a similar plateau in the number of selected 2-site clusters. The interpretation of this effect is linked to the fact that in this case we are inferring a real Potts model with reasonably high couplings, differently from what we do on biological data. Indeed in this simple case any 3- or more site interaction exists and the presence of this plateau confirms the algorithm is effectively selecting the interacting 2-site clusters first. To reach the lower plateau of the entropy, meaning the point after that no more significant contribution to the entropy can be added, bigger clusters have to be selected so to correct network effects. However it is important to stress that the full network of interactions is generally recovered before the convergence of the algorithm, for the example shown here, at the end of the intermediary plateau. This effect depends on the strength of interactions and it could be more pronounced here than in other models depending on the choice for the variance σ^2 of the Gaussian distribution from which the parameters

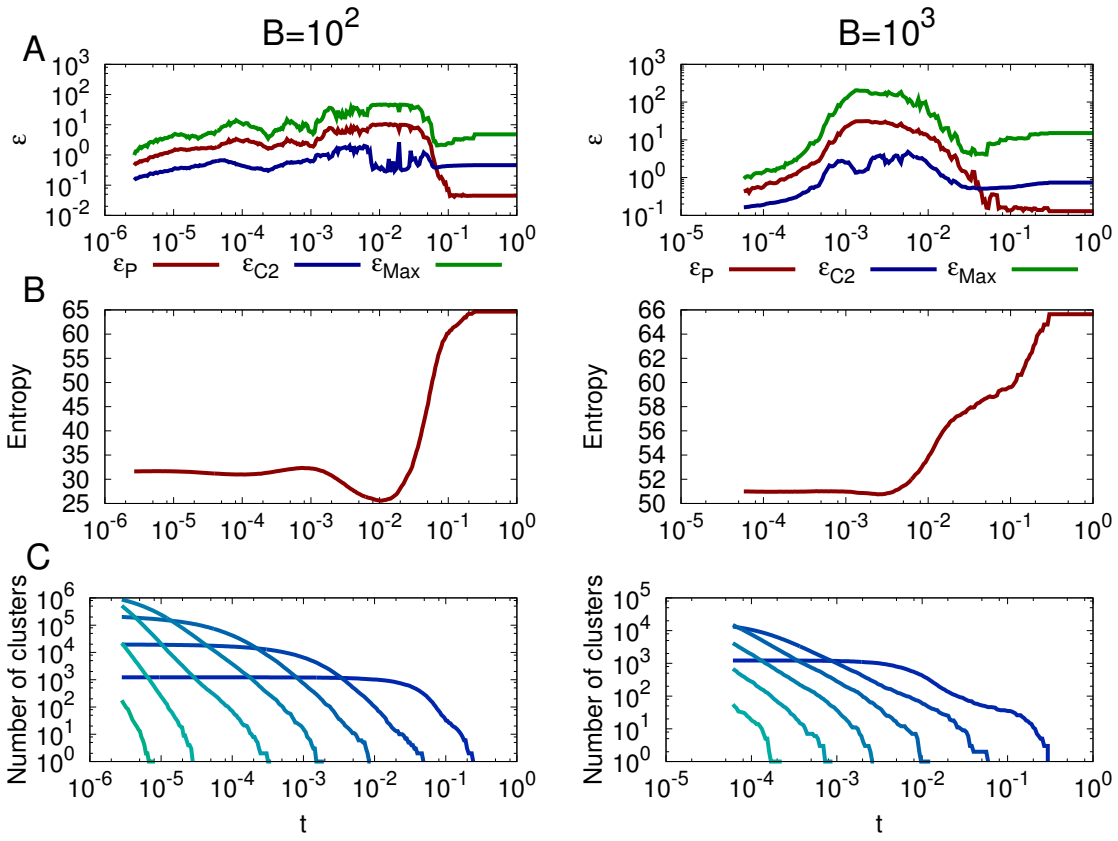


Figure 5.5: **ER005 $\mathbf{f_i(a_i) > 0.05}$** ACE inference. Row A: errors on the statistics obtained with the inferred model. Errors on magnetisations (ϵ_P) are plotted in red, errors on 2-point connected correlations (ϵ_{C2}) in blue and the maximum overall error (ϵ_{max}) in green. Row B: the red line represents the value of the overall cross-entropy. Row C: lines represent the number of computed clusters. The darker the colour the smaller the cluster size starting from 2-site clusters. Results for $B = 10^2$ and $B = 10^3$ are shown.

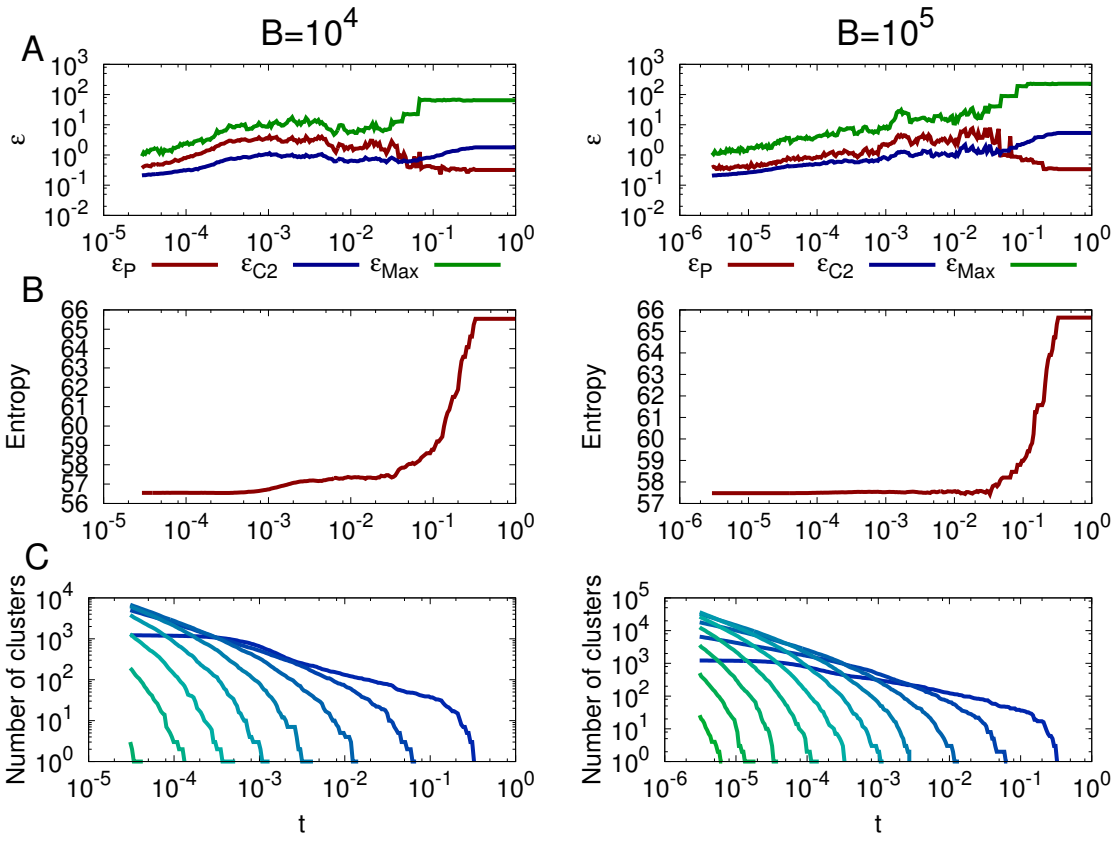


Figure 5.6: **ER005 $\mathbf{f_i(a_i) > 0.05}$** ACE inference. Row A: errors on the statistics obtained with the inferred model. Errors on magnetisations (ϵ_P) are plotted in red, errors on 2-point connected correlations (ϵ_{C2}) in blue and the maximum overall error (ϵ_{max}) in green. Row B: the red line represents the value of the overall cross-entropy. Row C: lines represent the number of computed clusters. The darker the colour the smaller the cluster size starting from 2-site clusters. Results for $B = 10^4$ and $B = 10^5$ are shown.

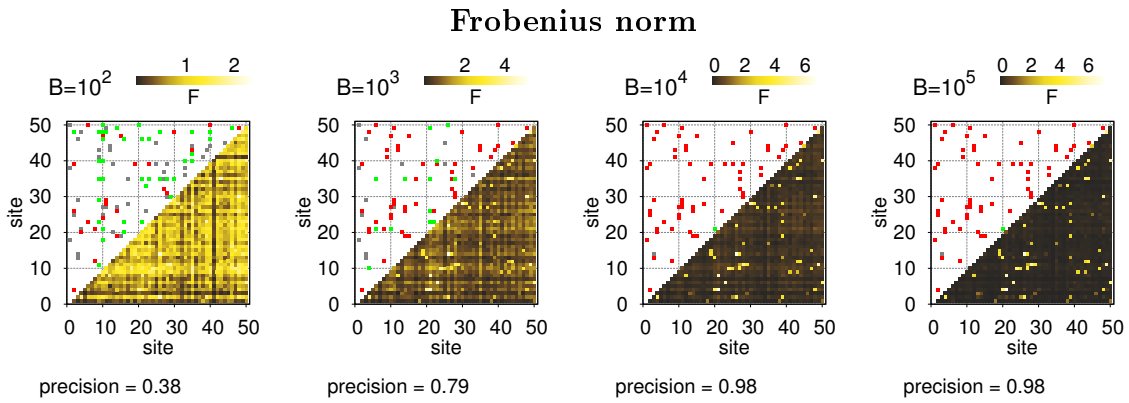


Figure 5.7: **ER005 $\mathbf{f_i(a_i) > 0.05}$** Contact maps obtained with inferred couplings. In the top-left triangle red squares represent true positives, green squares false positives and grey squares false negatives. In the bottom-right triangle the full Frobenius Norm matrix is shown.

are extracted. Secondly, looking again at the cross-entropy behaviour, note that the under-sampled case ($B = 100$) shows a peculiar minimum value corresponding to very high errors on the statistics. This behaviour is quite similar to what we have observed many times when performing analysis on real data and it probably means that the regularisation strength is too large.

Given the inferred model the first observation we make regards contact prediction. As we have already stressed in the previous chapters this is a major topic in statistical physics and many other efficient inference methods have been used to this scope. Here we show that also ACE can be successfully used to infer the network of interactions of the given model: Fig. 5.7 shows the contact maps obtained with the inferred couplings compressing information on different colours with the Frobenius Norm. Compare Fig. 5.7 with Fig. 5.8 where the Frobenius norms are corrected with the Average Product Correction. Good samplings do not need the correction to recover the whole interaction network, while smaller samples take great advantages from APC: for $B = 100$ the precision is almost doubled from 0.38 to 0.62.

Being this analysis made on artificial data we can quantify the goodness of the inference performed by ACE in comparing inferred parameters with the true ones. Fig. 5.9 shows the reconstruction of the true couplings and fields. In this picture the inferred and the true parameters have been converted to the consensus gauge and the grouped colour is not shown, because it simply does not exist within the true parameters. It corresponds, to same extent, to an effective coupling or field for all those colours that have been grouped together. As it is clear from Fig. 5.9, the other parameters are not influenced by the inference of such an effective colour and they

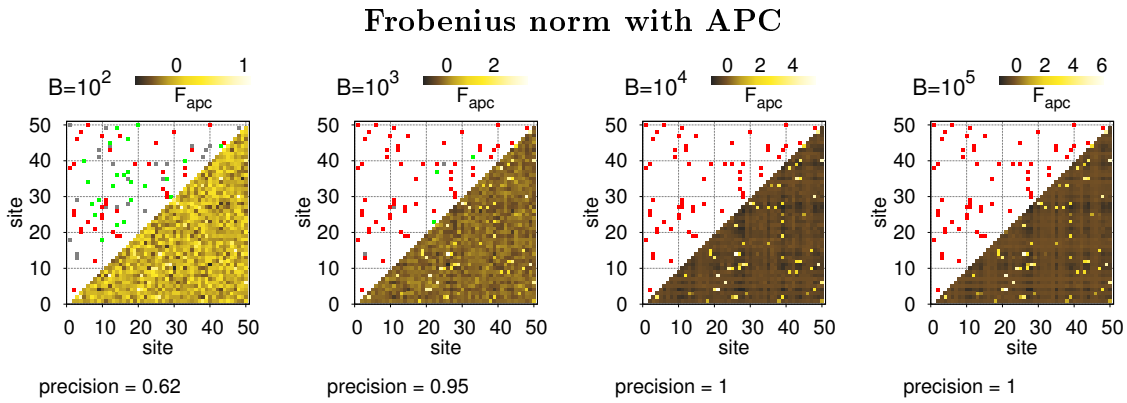


Figure 5.8: **ER005 $\mathbf{f_i(a_i) > 0.05}$** Contact maps obtained with inferred couplings. In the top-left triangle red squares represent true positives, green squares false positives and grey squares false negatives. In the bottom-right triangle the full Frobenius Norm with APC matrix is shown.

can systematically reproduce true values within the predicted error bars. We claim that, in the sense explained above, the colour compression does not effect the quality of the inference. Note that for $B = 100$ the inferred couplings are smaller than the true ones because of the regularization strength which is, as already remarked, too large.

Finally the main result we obtained concern the fact that ACE inferred model are generative, meaning they can be used to produce new samples reproducing the statistics of input data (Fig.5.10). Here we consider the $2-$ and the $3-$point connected correlations and the mutational probability of sequences. Note that in any case correlations obtained with the inferred model stay in the error bars. As far as 3-point correlations are concerned, a good sampling (at least $B = 10^4$) is needed in order to have good results: for $B < 10^4$ the 3-point correlations are small with respect to error bars and therefore the reconstruction cannot be good.

In order to better understand ACE pros and cons let us focus now on the small-sample case in Fig 5.5 first column. In this particular case the algorithm has run (on a standard desktop) for more than 2 days before converging. However, in order to reduce time consuming we can stop the algorithm long before the convergence point and run the MC-learning refinement of parameters. The same procedure can be used also in case the algorithm gets stuck far away from convergence and a reliable model is required. Here, we have chosen the threshold value $t = 0.00108$ as it is the first local minimum of the ϵ_{max} (cf. with Fig. 5.5) in a region where the cross-entropy is already flat. We have launched the MC-learning algorithm with this parameters as initial conditions. Fig. 5.11 shows the comparison in term of generative test

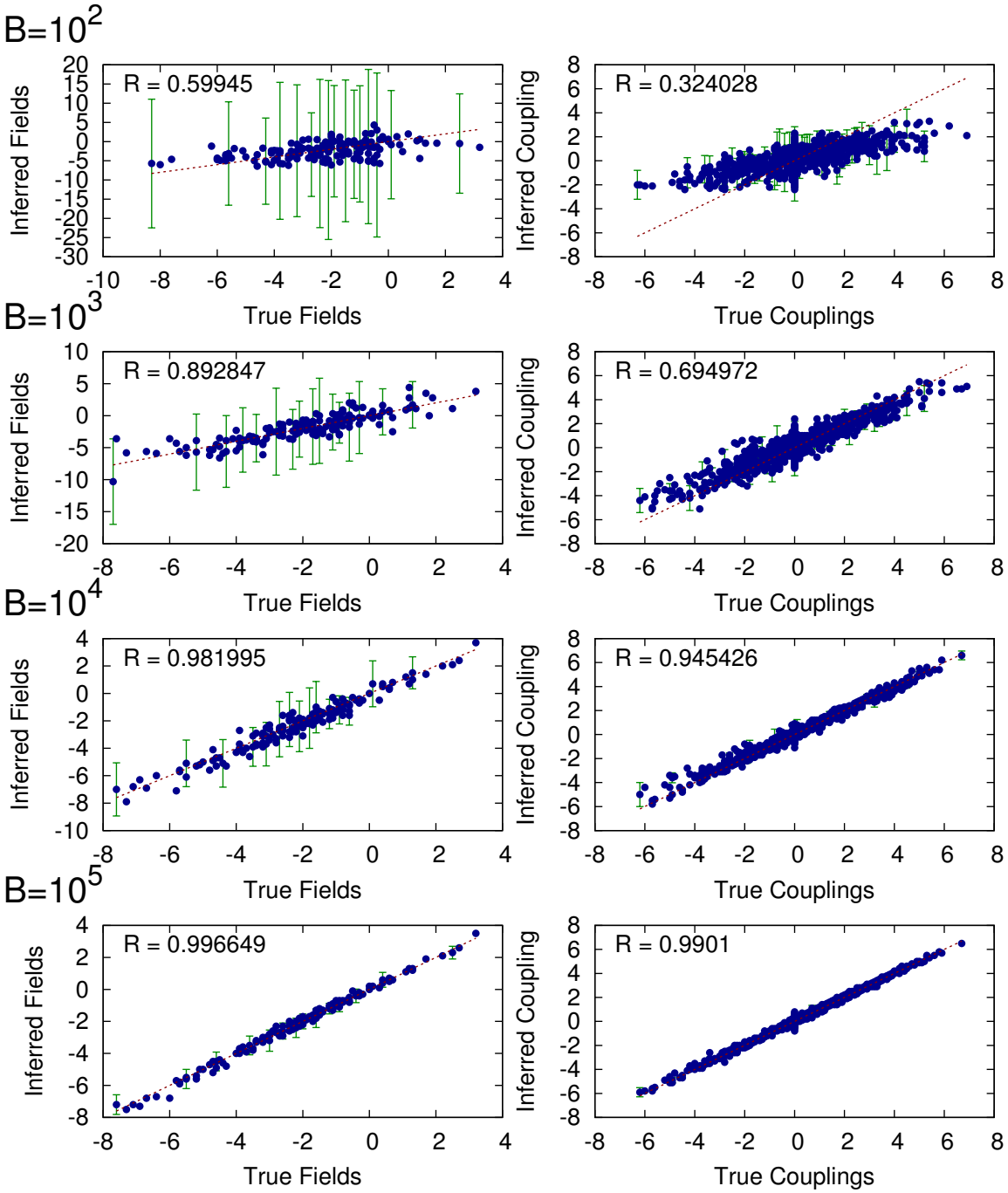


Figure 5.9: **ER005 $\mathbf{f_i(a_i) > 0.05}$** Left column: inferred fields are shown against true fields. Right column: inferred couplings are shown against true couplings. Rows show different sample results. Errors are computed propagating errors on magnetisation and correlations through the approximated formulas: $h_i = \log(f_i)$ and $J_{ij} = \log(\frac{f_{ij}}{f_i f_j})$ as shown in 5.36 and 5.35. True and inferred parameters are compared in the consensus gauge, and the grouped colour is neglected.

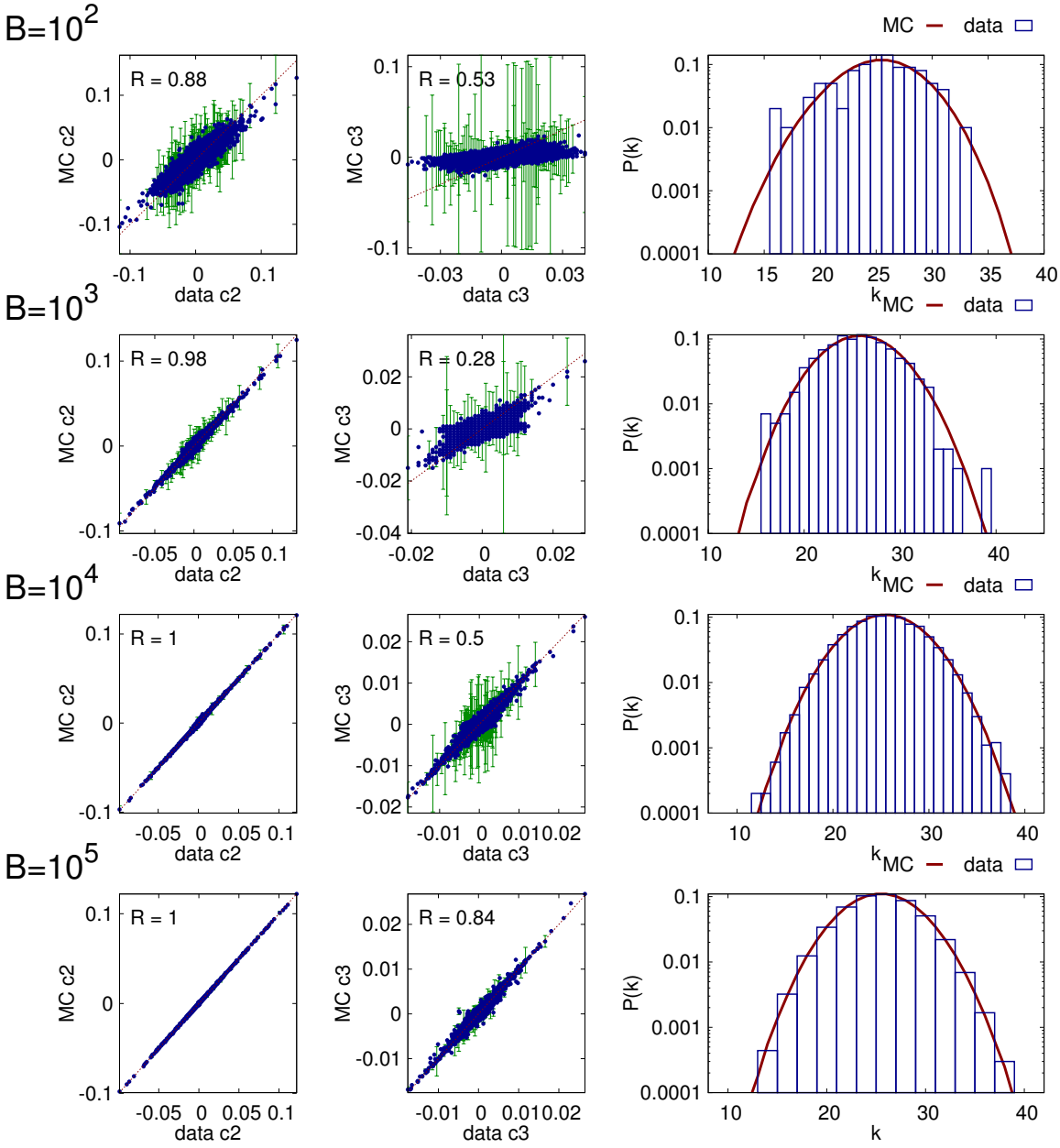


Figure 5.10: **ER005 $\mathbf{f_i(a_i) > 0.05}$** First column: 2-point connected correlations. Second column: 3-point connected correlations. Third column: probability to see a given number of mutated sites with respect to the consensus sequence. Four different sample-sizes are shown. Error bars represent the finite-sampling error in 5.7

among (A) the final converged parameters at $t = 3 \cdot 10^{-6}$, (B) the ACE parameters at $t = 0.00108$ and (C) the same parameters after MC-learning. As you can see ACE plus the MC-learning refinement shows results quite similar to ACE at convergence, even if the cost in term of computational time is significantly smaller. Note that ACE and ACE plus MC-learning recover the statistics highly better than the other intermediate case, where parameters are far from convergence. We can argue that, thanks to appropriate initial conditions, MC-learning gives us a generative model as good as ACE in a more reasonable time. Finally if we compare the inferred couplings and fields with the true ones, we do not see any substantial difference from the first row of Fig. 5.9, meaning that indeed the MC-learning cannot improve the inference of the interaction network.

Results for model ER005 with $f_i(a_i) > 0.01$ In the following section we will show results for the ER005 model when a weaker ($f_i(a_i) > 0.01$) colour-compression is applied. What we observed in this case is that ACE cannot easily converge as in the previous case: none of the four samples has converged in a reasonable time (about a day) since the lower reduction entails a bigger computational time, however final results (cf. Fig. 5.12) are quite similar to those obtained with reduction $f_i(a_i) > 0.05$. Note that the inference from the smallest sample ($B = 100$) results in this case quite hard: the errors on statistics remains very large disregarding the value of t and only a MC-learning refinement can in this case produce a meaningful generative model. The other three samples show instead a behaviour similar to the respective curves in Fig 5.5 and 5.6, but without reaching convergence. It is important to stress that within this artificial model analysis we have not selected the model so to give the best results, or fine-tuned parameters for the best inference. Our aim here is in fact to show how ACE works with standard options and on a random, and as general as possible, model.

Results for model ER010 with $f_i(a_i) > 0.05$ We analyse in this part a more connected model. Note that ACE is based on the idea to infer the sparser network compatible with the data: it is not useful to infer couplings different from zero when their error bars are extremely large. Consider in fact that fully connected inference methods still exist and have proved to produce good results [3] [22]. However the choice of the best coupling threshold in order to consider two sites as interacting is quite heuristic and often challenging. With the ACE we aim to solve this problem considering the errors on the statistics. We have tested the algorithm to infer a

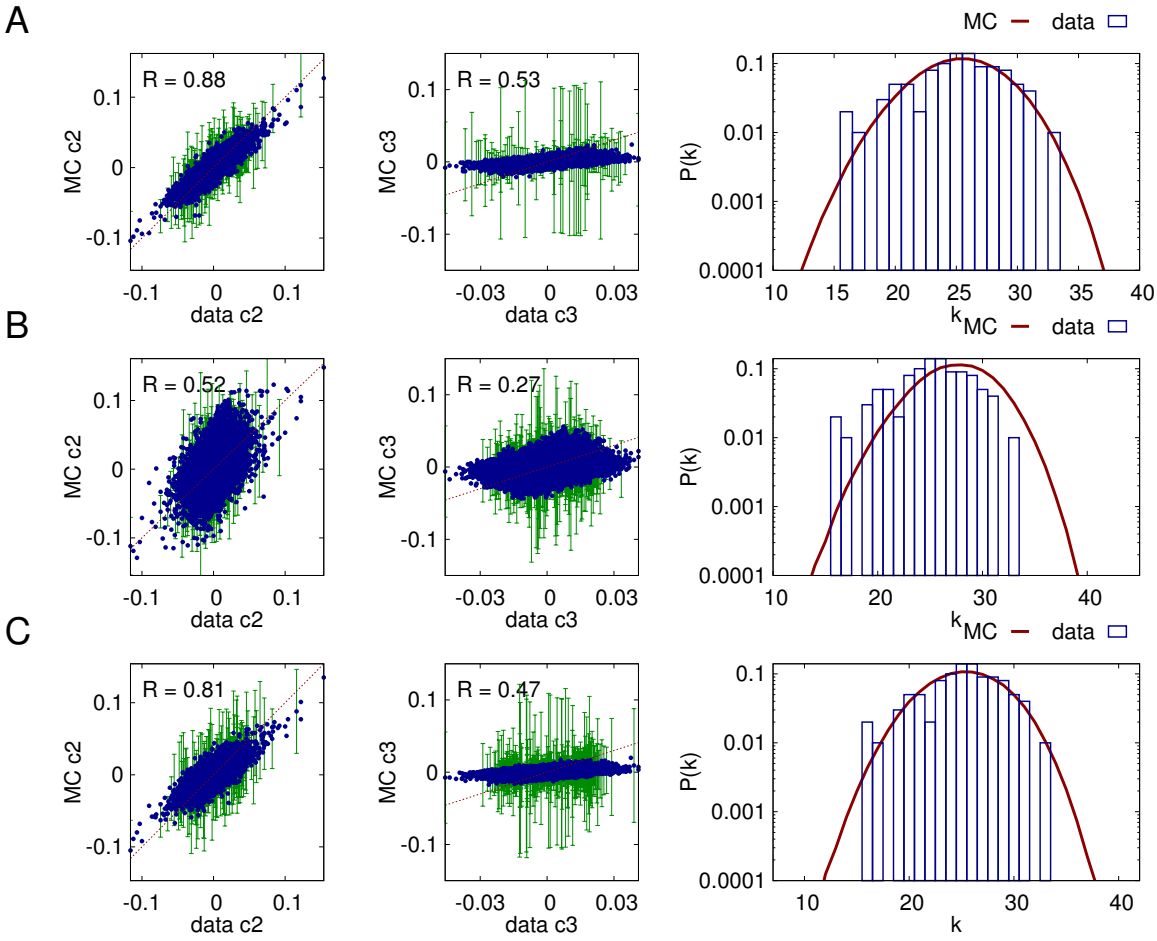


Figure 5.11: **ER005 $\mathbf{f_i(a_i) > 0.05}$ $\mathbf{B = 100}$** First column: 2-point connected correlations. Second column: 3-point connected correlations. Third column: probability to see a given number of mutated sites with respect to the consensus sequence. Rows show different threshold results: (A): results for the convergence threshold $t = 3{\cdot}10^{-6}$, (B): results for $t = 0.00108$, (C): results for $t = 0.00108$ plus MC-learning. Error bars represent the finite-sampling error in 5.7

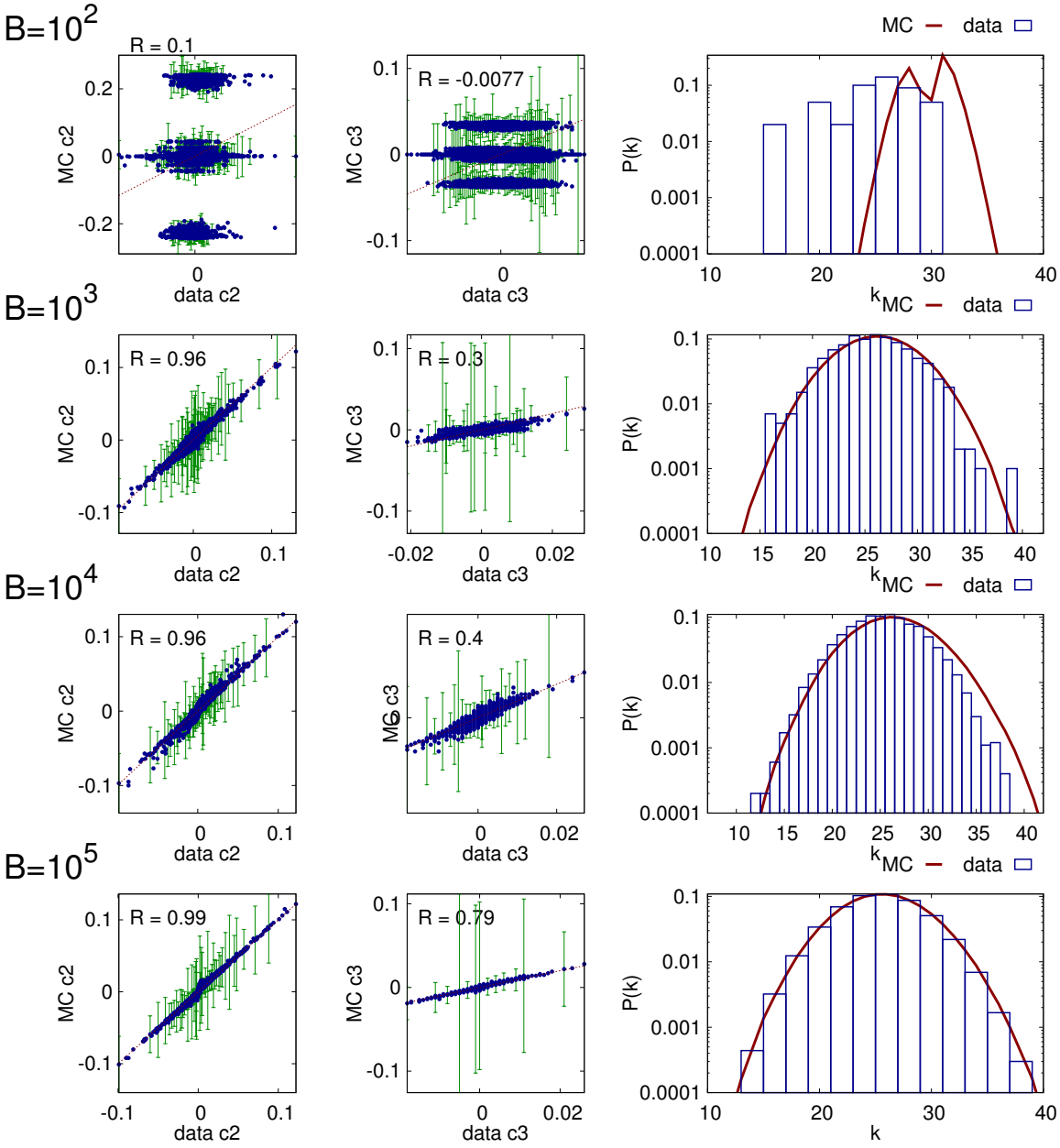


Figure 5.12: **ER005 $\mathbf{f_i(a_i) > 0.01}$** First column: 2-point connected correlations. Second column: 3-point connected correlations. Third column: probability to see a given number of mutated sites with respect to the consensus sequence. Four different samples are shown. Error bars represent the finite-sampling error in 5.7

more connected network with maximum connection of 12, meaning that in principle clusters of size 12 are needed to correctly infer the model. What we observed is that very large clusters have indeed been selected, meaning the the computational time needed for convergence is infeasible: also in this case any sample has reached convergence. Anyway, even if the convergence is not reached without MC-learning for such a connected model, results shown in Fig. 5.13 are in general quite good and comparable to those obtained for ER005.

Results for model ER010 with $f_i(a_i) > 0.05$: refinement with MC-learning Results shown in the last paragraph stressed the computational limitations of ACE: even if the system size of ER010 is the same of ER005, here a more connected network entails bigger clusters and thus ACE enters its computational infeasible regime. As discussed in the introduction, we have developed a MC-learning routine in order to find convergence parameters also when the ACE gets stuck. The output parameters available result to be good input parameters for the MC-learning refinement. In particular we force the algorithm to save parameter within some fixed interval, when ϵ_{max} is minimum. Note, indeed, that the adaptive nature of ACE entails errors not to be monotonic functions of t, thus it is possible to choose a posteriori the best t and use the corresponding parameters within the MC-learning. Here we use t corresponding to the last local minimum of ϵ_{max}, being sure that at that value the entropy has already found the final plateau. Comparing Fig. 5.14 with Fig. 5.13 we note that MC-learning produces a reliable generative model, substantially improved with respect to the one ACE has inferred.

Comparison with DCA and plmDCA This paragraph is devoted to the comparison of ACE with two existing methods: DCA [3] and plmDCA [22]. Note that DCA and plmDCA are run on the full alphabet while ACE has been run with reduction $f_i(a_i) > 0.05$.

Compare Fig. 5.15 and Fig. 5.16 with Fig. 5.10. As expected, DCA cannot reasonably reproduce even the 2-point connected correlations used to fit the model, while plmDCA and ACE can. Anyway ACE outperforms plmDCA both on 2-point correlations and on 3-point correlations. Also the P(k) is very well reproduced by ACE, while it is not reproduced by DCA and poorly reproduced by plmDCA. The latter performs in any case considerably better than DCA in inferring generative models. As far as contact map prediction the three methods are quite similar (cf.

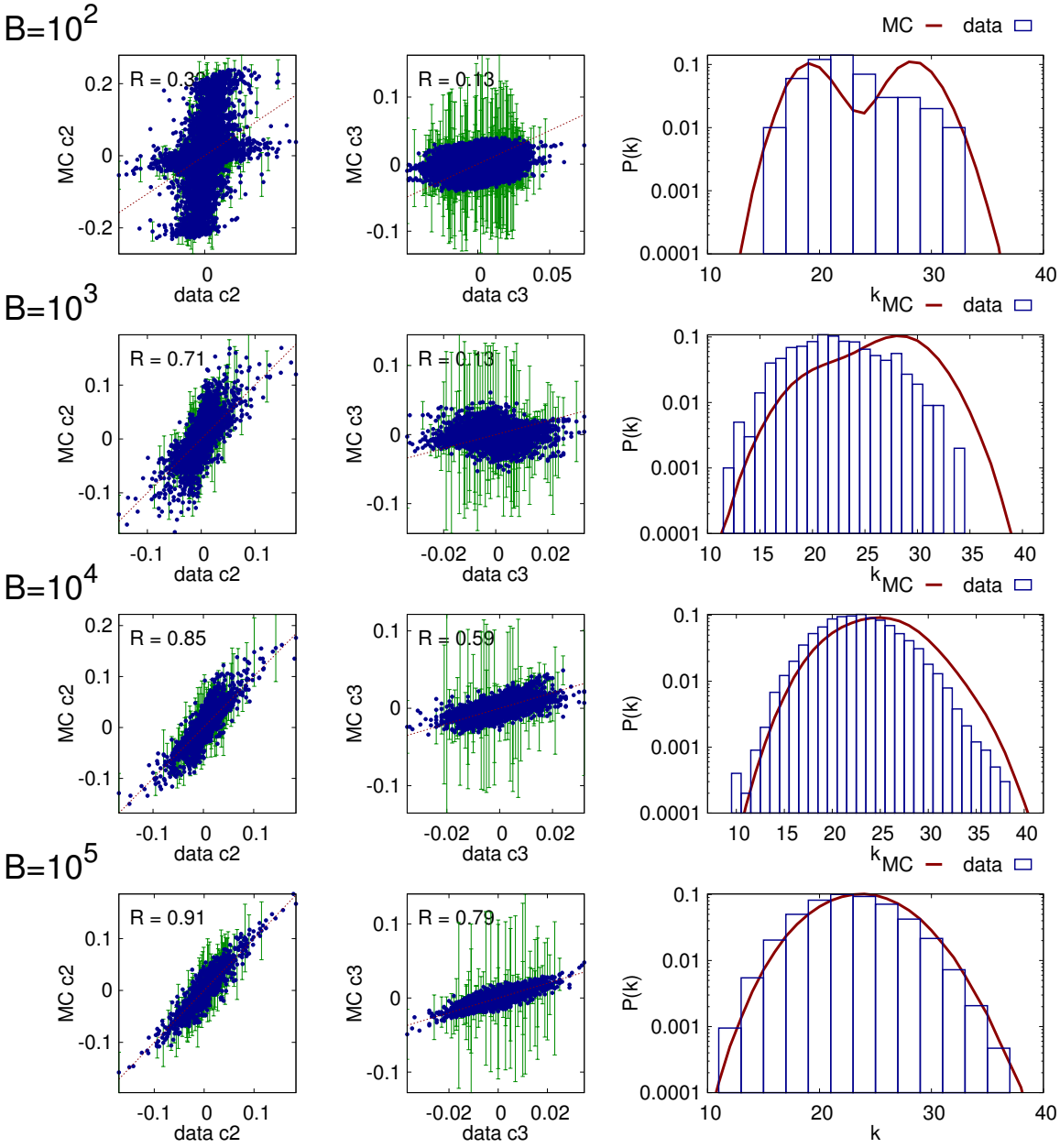


Figure 5.13: **ER010 $\mathbf{f_i(a_i) > 0.05}$** First column: 2-point connected correlations. Second column: 3-point connected correlations. Third column: probability to see a given number of mutated sites with respect to the consensus sequence. Four different samples are shown. Error bars represent the finite-sampling error in 5.7

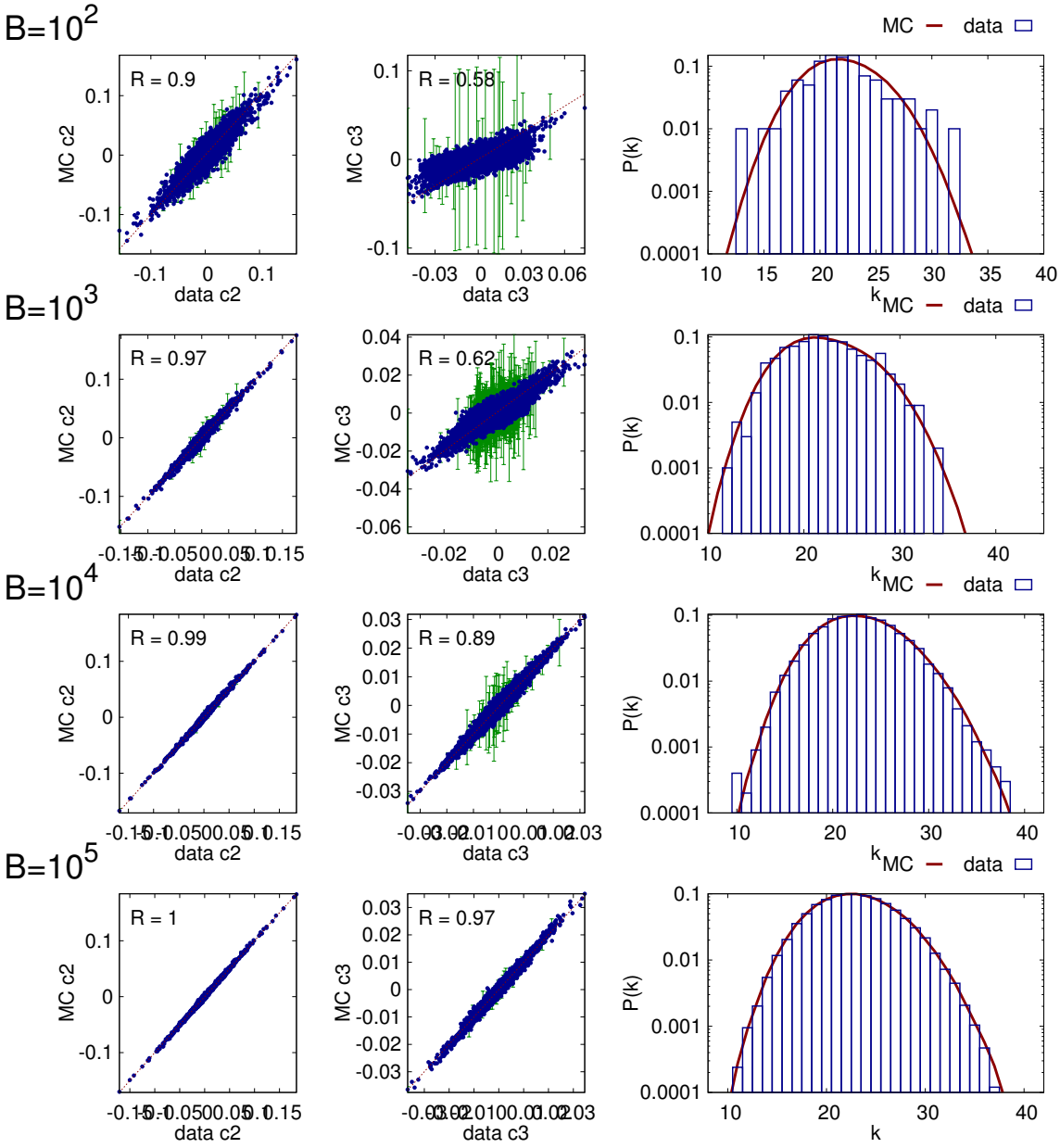


Figure 5.14: **ER010 $f_i(a_i) > 0.05$ with MC-learning** First column: 2-point connected correlations. Second column: 3-point connected correlations. Third column: probability to see a given number of mutated sites with respect to the consensus sequence. Four different samples are shown. Error bars represent the finite-sampling error in 5.7. Differently from Fig. 5.14, here ACE output parameters are refined with MC-learning before the generative test is performed.

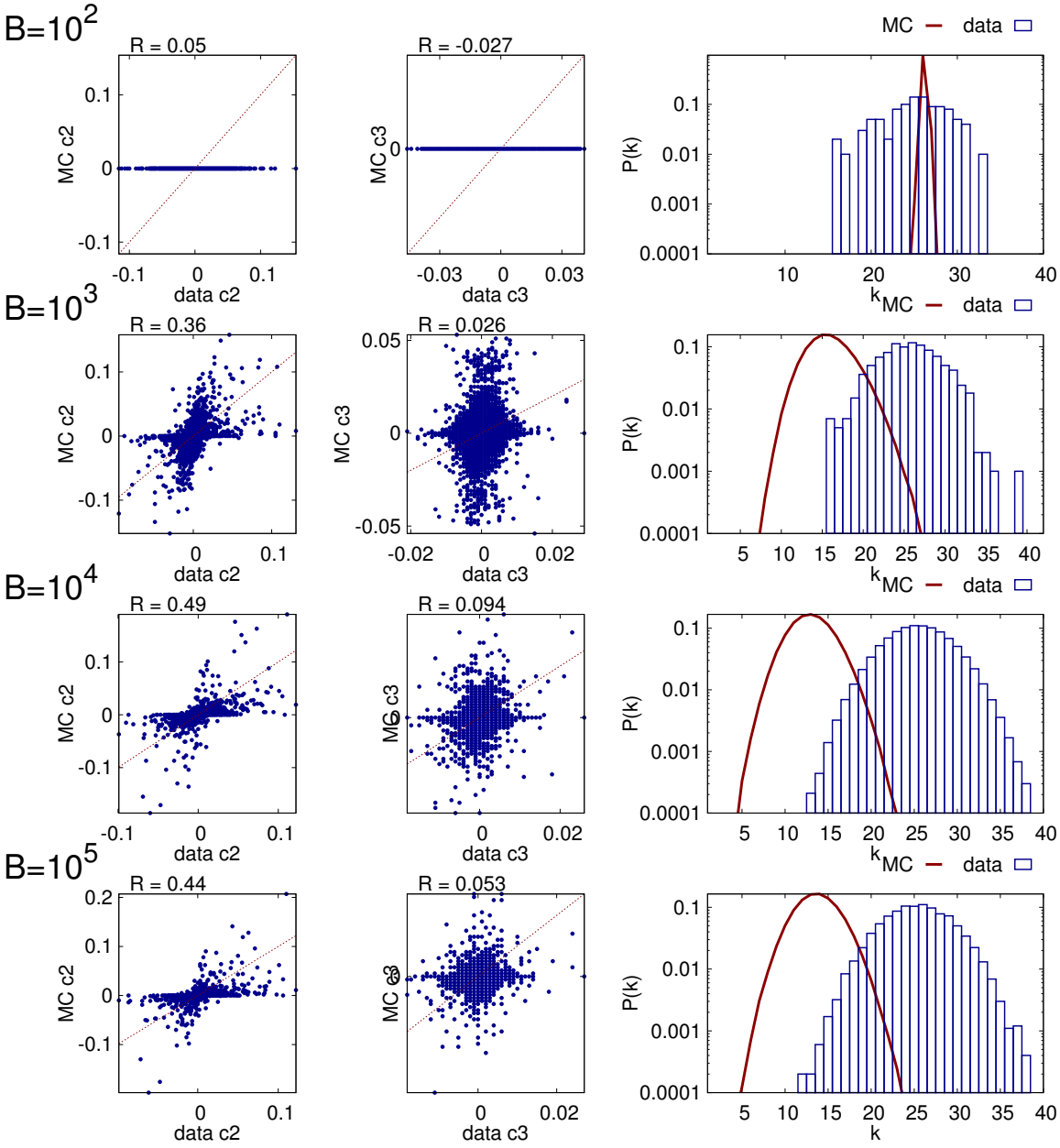


Figure 5.15: **ER005 DCA** First column 2-point connected correlations. Second column 3-point connected correlations. Third column probability to see a given number of mutated sites with respect to the consensus sequence. Four different samples are shown.

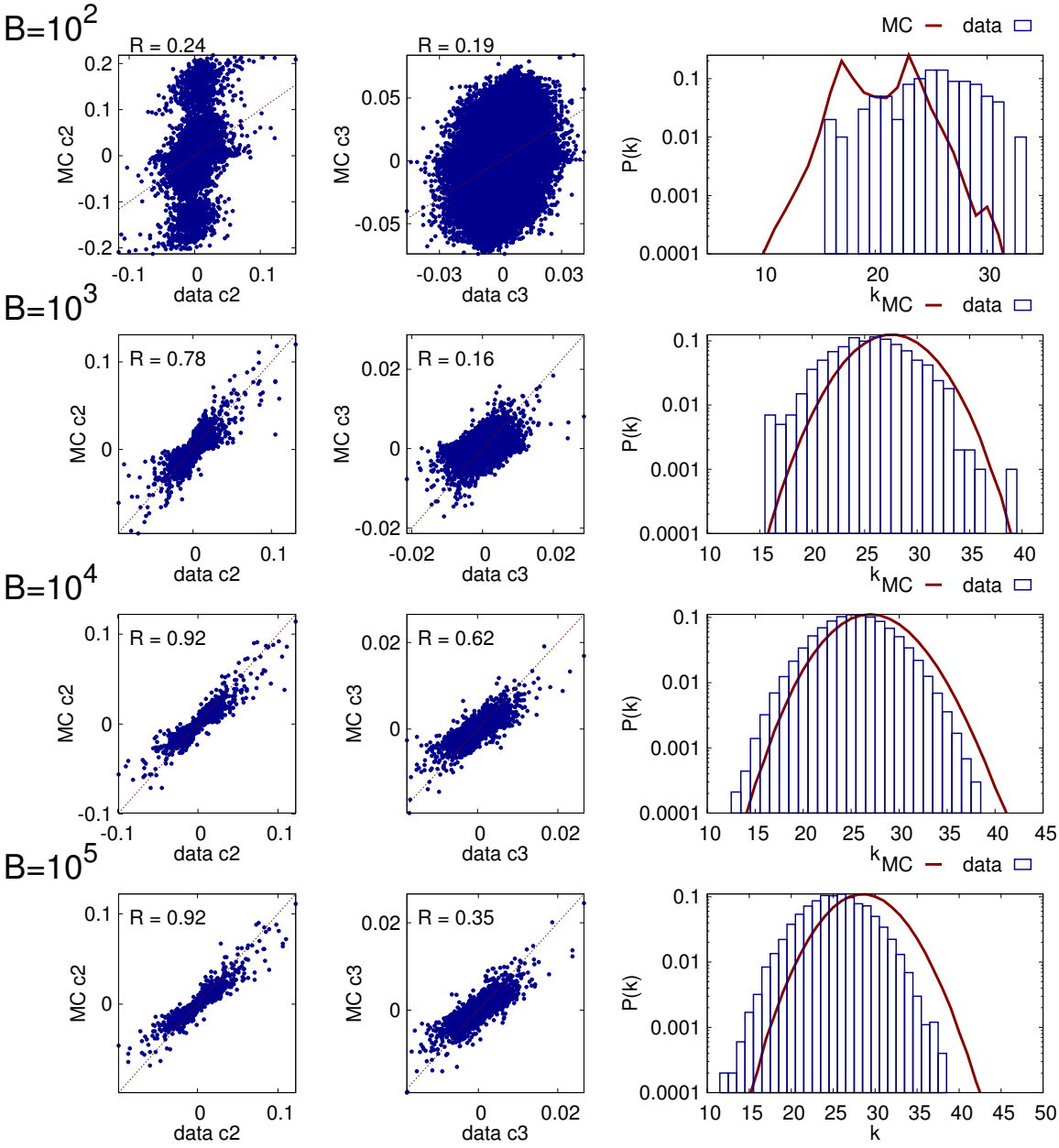


Figure 5.16: **ER005 plmDCA** First column 2-point connected correlations. Second column 3-point connected correlations. Third column probability to see a given number of mutated sites with respect to the consensus sequence. Four different samples are shown.

Inference methods	$B = 10^2$	$B = 10^3$	$B = 10^4$	$B = 10^5$
ACE	0.38	0.79	0.98	0.98
ACE + APC	0.62	0.95	1	1
DCA	0.36	0.8	0.98	0.98
DCA + APC	0.56	0.95	0.98	1
plmDCA	0.39	0.69	1	1
plmDCA + APC	0.59	0.97	1	1

Table 5.2: Table showing the precision of the contact prediction of the three inference methods analysed on the model ER005. Scalar scores for interactions ranking are computed with the Frobenius norm of the inferred coupling and then also the effect of average product corrections (APC) is shown.

Table 5.2) and slightly improvement are reached for small samples adding the average product correction.

The following analysis wants to show that, a part from the inference of the contact map, DCA and plmDCA are outperformed by ACE. Of course from the computational point of view both DCA and plmDCA are much faster than our algorithm. However we claim that, when more sophisticated results have to be achieved, as for generative models, ACE ensures a very good inference in many sample and connectivity regimes, while DCA and plmDCA almost fail.

Finally, as we have seen that MC-learning refinement remarkably improves generative test results, one can use DCA and plmDCA inferred parameter as MC-learning input and obtain, also in these cases, generative models. However, for the model ER005 considered above, when we use the parameters inferred with DCA and plmDCA as input parameters for the MC-learning we do not reach convergence and errors on the statistics saturates to values higher than one. The saturation value for ϵ_{max} depends on the considered model and it varies from 10 to 50 for DCA and from 4 to 12 for plmDCA. These difficulties encountered in reaching convergence with DCA and plmDCA input parameters have also been observed on biological data.

5.4.2 Biological data application: RNA

As promising results on artificial data have been shown, I will present in this section the application of ACE to the same selection of six riboswiches studied in chapter 3. The following ones have to be considered as preliminary tests, since RNA peculiarities stressed in the dedicated chapter have not been taken into account to improve results. Application to proteins and neuronal data will be exploited in the article (in preparation) which this chapter refers to.

This section will report some tests done in order to estimate the correct strength of the gamma regularisation parameter for RNA data. Tests are motivated by the observation that typical value for the gamma $\gamma \sim \frac{1}{B}$ used for artificial data produces over-estimated couplings in the first steps of the algorithm and then, it takes long time before the correct value for parameters is recovered. Intuitively we can understand this effect as an erroneous estimation of the sample noise: bayesian analysis of i.i.d. samples estimate in $\frac{1}{B}$ the best regularisation strength, anyway MSA are far from being i.i.d. samples, even when the re-weighting correction is applied. This empirical refinement, we successfully applied in the case of DCA, does not provide here a reliable correction of the alignment and we need an accurate choice of the regularisation strength.

Once some *good* values for γ have been chosen, we run ACE on our RNA dataset and compare results on contact predictions, in the form of true positive rates, and generative tests.

Gamma selection We have observed on artificial data that usually a very good fit of the parameters is found when the convergence of cross-entropy is reached, even if errors, in particular ϵ_{max} are still quite far from 1. Precisely we have showed that parameters computed at this point of the iteration usually rapidly converge, if improved with MC-learning. This effect is due to the fact that the biggest contributions to the log-likelihood have, at this point, already been included in the calculation. The very last iterations of the algorithm are needed in order to correct intrinsic errors in the cluster expansion. Remember that the selection of a particular cluster does not depend on the other selected clusters. Therefore, as explained in [23], after that a cluster is selected, errors on statistics usually rise and a sort of *cascade* of sub-clusters needs to be selected before seeing errors dropping down again. The selection of this *cascade* is hard because contributions can be small and it takes usually a long time. However observing the emergence of a plateau in the cross-entropy can help detecting this situations.

Being interested in define a pipeline, as fast as possible, in order to understand the best value for the regularisation strength γ we decided to launch the algorithm with many different values of γ for a reasonable amount of time (within this particular case ACE was left running for 20 minutes on a standard desktop). The aim of this procedure is to understand which is the regularisation strength favouring the most the convergence of the algorithm. Outputs of these procedure let us make some observations:

- The most time-consuming routine is the computation of the cluster cross-entropy and it consists of two main parts: the computation of the partition function via the sum over all the q^k configurations and the numerical evaluation of the cluster parameters optimising the cross-entropy. Monitoring the number and the size of selected clusters gives us a good estimate about the first part. Keeping fixed these quantities, the speed of the algorithm turns out to tell us, to some extent, whether the optimisation routine is well regularised or not. The regularisation helps the optimisation routine, thus a fast convergence of the optimisation algorithm needs an appropriate regularisation. However, this considerations usually lead to high γ as gradient descent significantly benefit from large $L2$-norm regularisations.
- If the cross-entropy has reached or it seems close to a plateau, often parameters are still quite good regardless of errors. Otherwise, when the cross-entropy is still oscillating important changes in the model are happening: the algorithm is selecting highly significant clusters and, even for similar value of t, output parameters change a lot.
- For very large γ the cross-entropy rapidly saturates as too many clusters are selected thanks to the contribution of the regularisation term. Since in this regime the regularisation term in Eq. 5.2 is larger than the other contributions, the algorithm cannot reasonably fit the data.

Given this considerations it is clear that the a priori estimation of the best value for γ is definitely not trivial. For the concerned RNA Fig. 5.17, showing the cross-entropy curves for different γ, cannot suggest us an optimal value: for $t > 0.012$ no more oscillations appear, moreover till $t = 0.1$ the algorithm is running relatively fast. Consider now the number and the size of selected clusters 5.18: decreasing γ means computing greater clusters already with large t. Therefore the algorithm gets stuck quite soon when a few clusters have been selected, preventing from an accurate inference of parameters. Otherwise large γ values allow the selection of a huge number

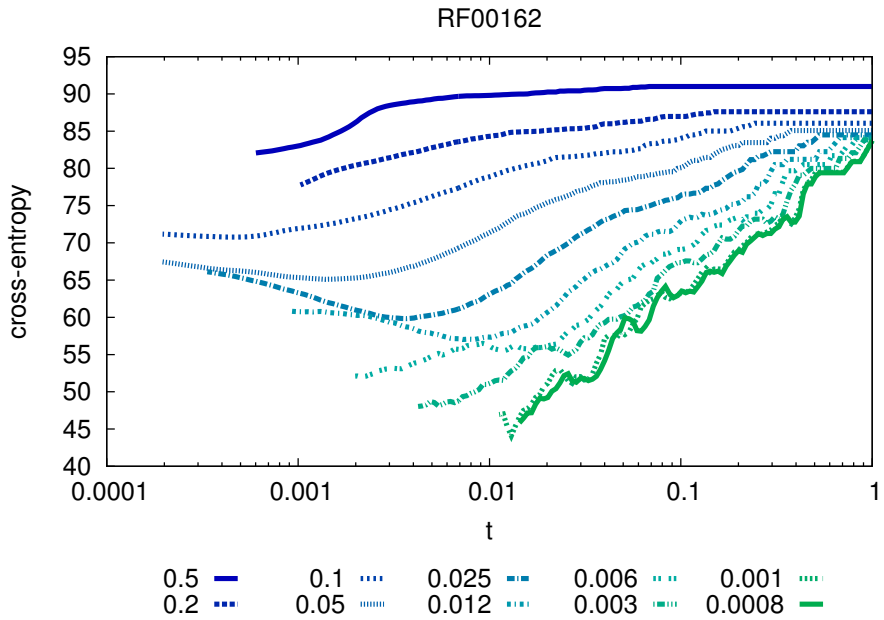


Figure 5.17: The final cross-entropy of one among the six riboswiches, RF00162, is showed. Different colours refer to different values of γ. The alignment for this RNA family is made of 4757 sequences and the B_{eff} obtained with a re-weighting threshold equal to 0.1 is 1165.98. For the run of ACE no colour compression is performed, however only observed colours are included in the inferred Potts model.

of small clusters; to some extend in this case almost no selection is performed: the huge regularisation term in the cross-entropy prevents the algorithm from distinguish useful and useless contributions.

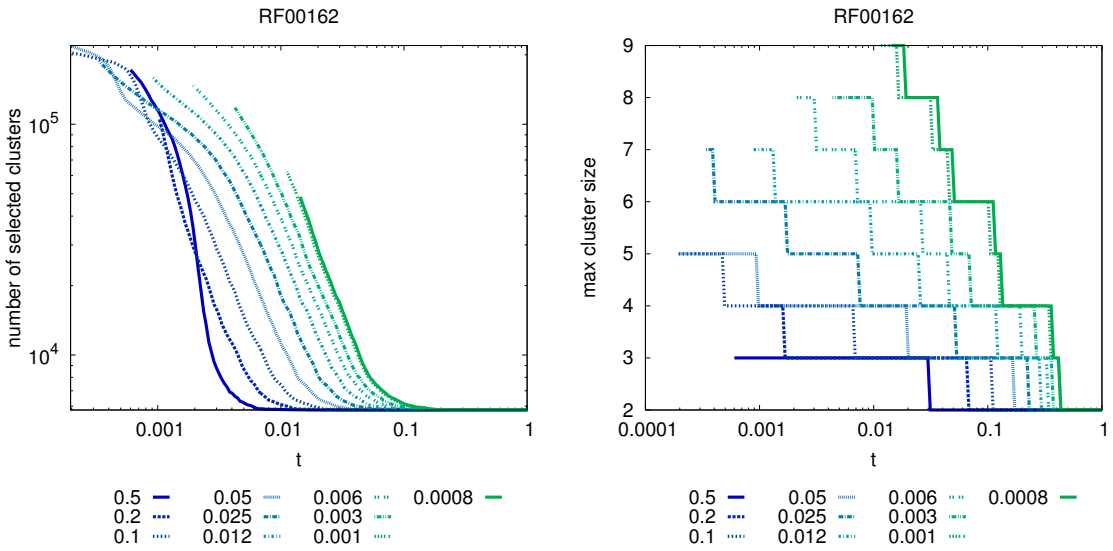


Figure 5.18: Left: the number of selected clusters. Right: the maximum size of selected clusters. Again different values for γ are showed.

Finally consider Figure 5.19. If gamma is too large (dark blue curves) the algorithm produces small errors but it is too slow to converge till a reasonable t. Otherwise if the γ is too small (dark green curves) errors diverge for small γ and, also in this case, we cannot observe the convergence of the algorithm.

Let me stress that this heuristic method for finding the best γ is absolutely not proposed as a definitive solution for the problem of correctly estimate how *good* (i.e. how akin to a i.i.d. sample) an alignment of sequences is. The hardness of the problem is well known and phylogenetic-based methods [109] promise to improve results. However, till now, the most sophisticated tools require a huge computational effort and are usually infeasible for reasonable size sequences. Our approach is instead extremely practical. We actually test our algorithm for any γ and look for the best γ depending on the observed performance of ACE. Obviously the result is not pretended to be of any generality, but it is restricted to the use within ACE itself. Moreover even for ACE we often run the algorithm for more than one value of γ before inferring a promising generative model.

Riboswiches results The same analysis showed for RF00162 was performed also on the other five RNAs and produced similar results: the choice of the optimal value

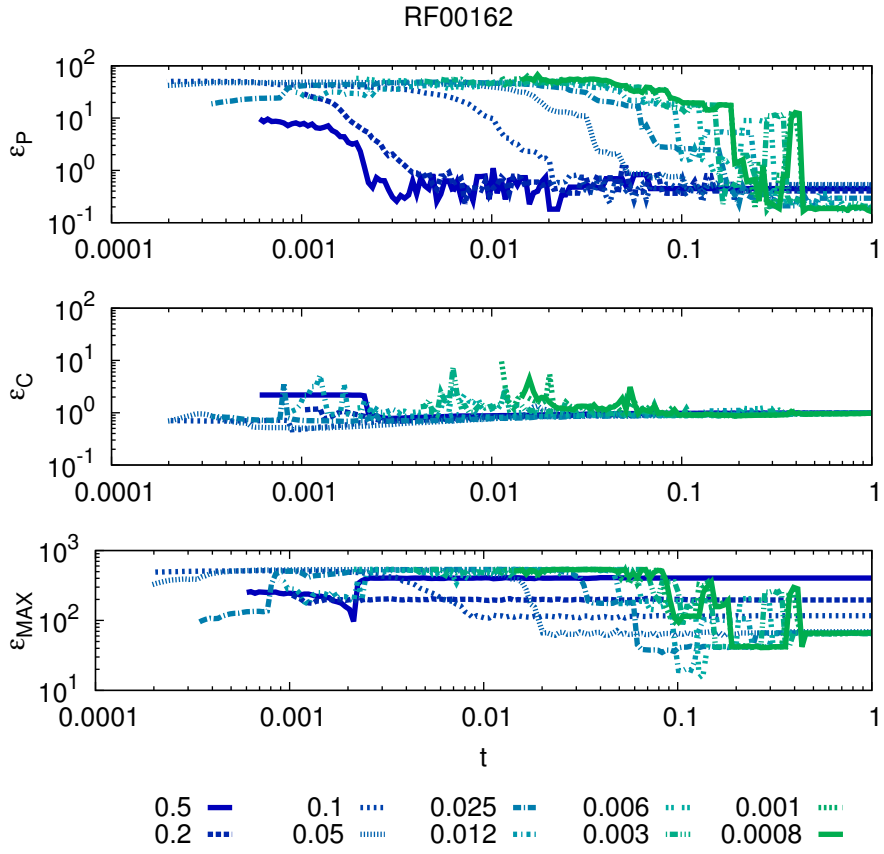


Figure 5.19: Errors on the statistics. From top: ϵ_P, ϵ_C and ϵ_{max}. Different colours represent different values of γ

Family	γ_1	γ_2	$\frac{1}{B}$	$\frac{1}{B_{eff}}$
RF00162	0.05	0.006	$2 \cdot 10^{-4}$	$8 \cdot 10^{-4}$
RF00167	0.05	0.003	$4 \cdot 10^{-4}$	0.002
RF01051	0.1	0.05	$5 \cdot 10^{-4}$	0.001
RF01734	0.1	0.003	$8 \cdot 10^{-4}$	0.002
RF00504	0.2	0.012	$1 \cdot 10^{-4}$	$5 \cdot 10^{-4}$
RF00059	0.1	0.025	$9 \cdot 10^{-5}$	$3 \cdot 10^{-4}$

Table 5.3: Table showing the chosen strengths for the regularisation (with $\gamma_1 > \gamma_2$) for the six riboswiches. B is the number of sequences in the alignment, while B_{eff} is the effective number after re-weighting.

for γ is hard and painful. In the following we show results for two different value of γ, called γ_1 and γ_2, shown in Table 5.3. DCA and plmDCA were also used to infer parameters.

We confirm what has been observed on artificial models: the choice of the best algorithm depends on the type of information one need to extract from alignments. Contact map prediction clearly does not need either ACE analysis or plmDCA, the naive-MF solution gives almost always the best results in the shortest time, generative tests showed, instead, that ACE is the only algorithm giving reasonable results. As far as contact predictions are concerned, we show only results for the higher value of the regularisation γ_1, since differences with results obtained with γ_2 are negligible. MC-learning, as showed before, do not change significantly the inference of the network of interactions, therefore, in Figs. 5.20 and 5.21, couplings used for predictions belong to the last set of parameters recorded by the ACE. In these two figures we show that lowering t, and thus including more terms in the cross-entropy series, we improve the true positive rate with respect to native structures. Anyway DCA and plmDCA represent, almost in any case, the upper bound of this progressive improvement.

Comparing Fig. 5.22 and Fig. 5.23 the role of regularisation in the inference is evident: when we use γ_1 the way the inferred parameters reproduce the 2-point and 3-point connected correlations is strongly biased towards smaller values. Large regularisations force the inferred model into an high temperature regime characterised by small interactions (cf. chapter 4). However small regularisations entail longer computational time and, therefore, we cannot decrease the value of γ till $\frac{1}{B}$.

The complexity of RNA data inference is again well represented here: results from one riboswich to the others significantly change.

– **RF00162**: DCA, plmDCA and low t ACE true positive rates are comparable. DCA and plmDCA perform slightly better when the secondary structure is

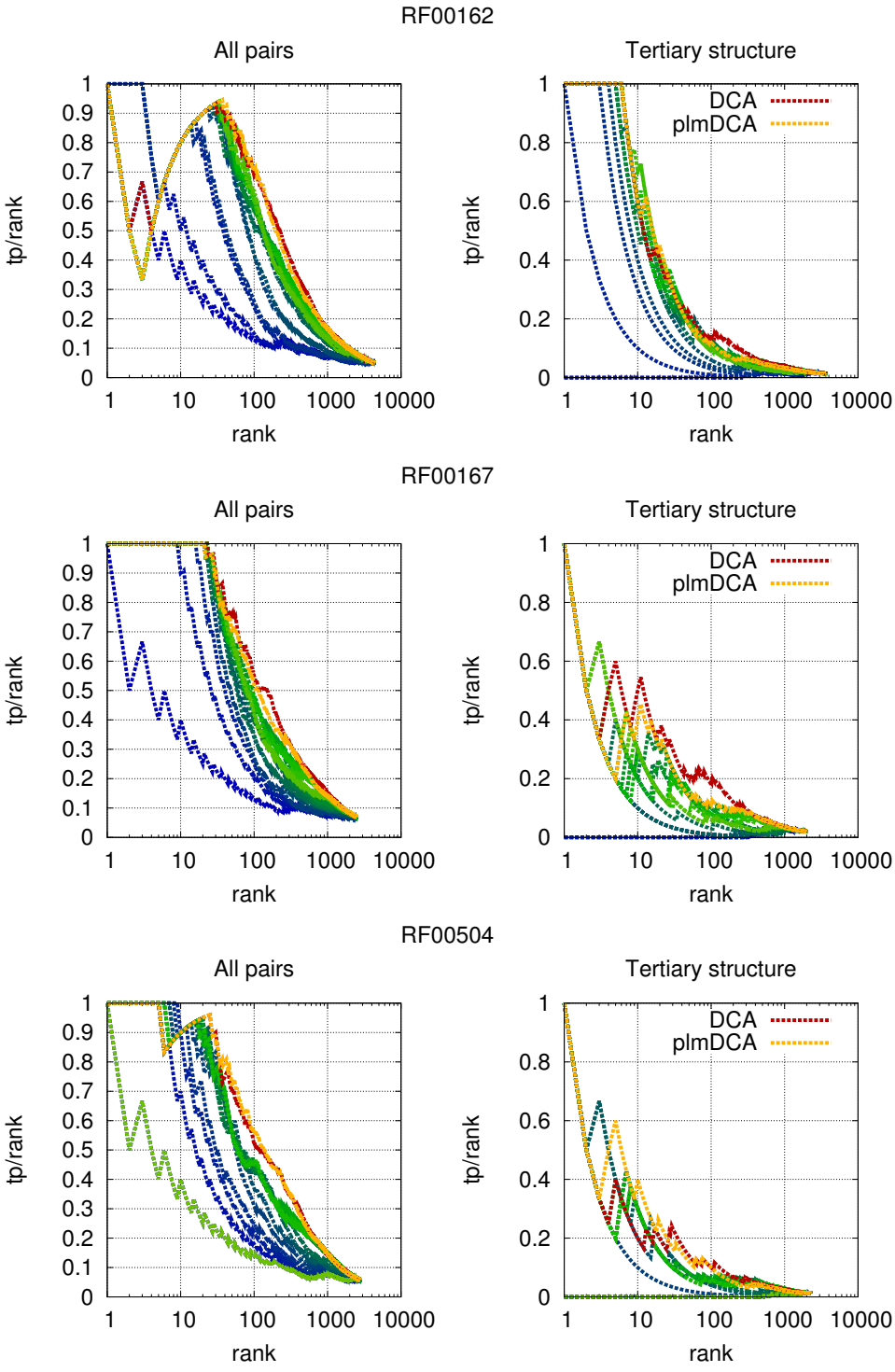


Figure 5.20: True positive rates for riboswiches RF00162, RF00167 and RF00504. Lines coloured from blue to green represent ACE results for decreasing threshold t. DCA and plmDCA are performed on the full alphabet model, while ACE includes only observed colours. 4 Å threshold is used for contacts definition (cf. chapter 3)

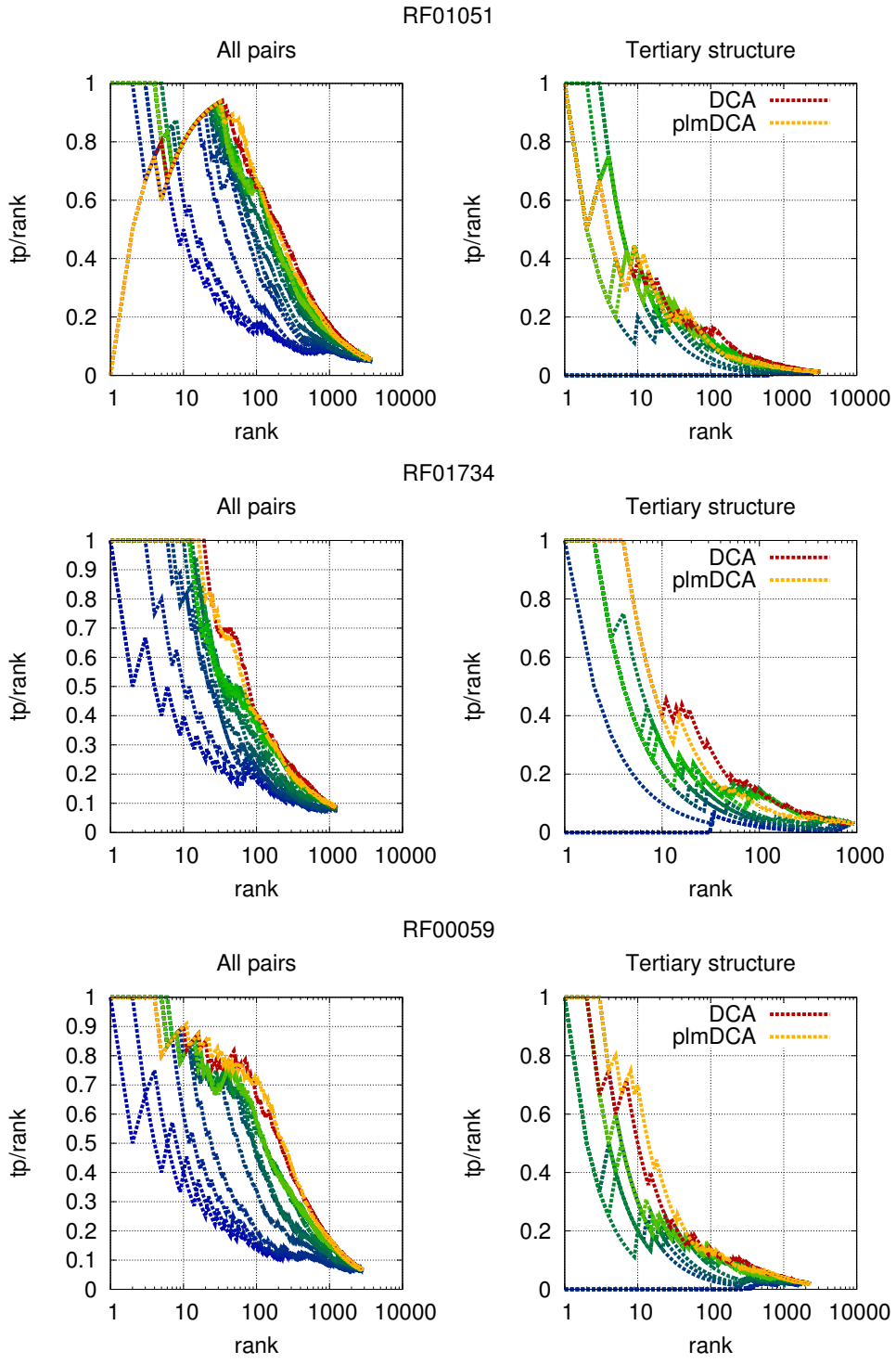


Figure 5.21: True positive rates for riboswiches RF01051, RF01734 and RF00059. Lines coloured from blue to green represent ACE results for decreasing threshold t. DCA and plmDCA are performed on the full alphabet model, while ACE includes only observed colours. 4 Å threshold is used for contacts definition (cf. chapter 3)

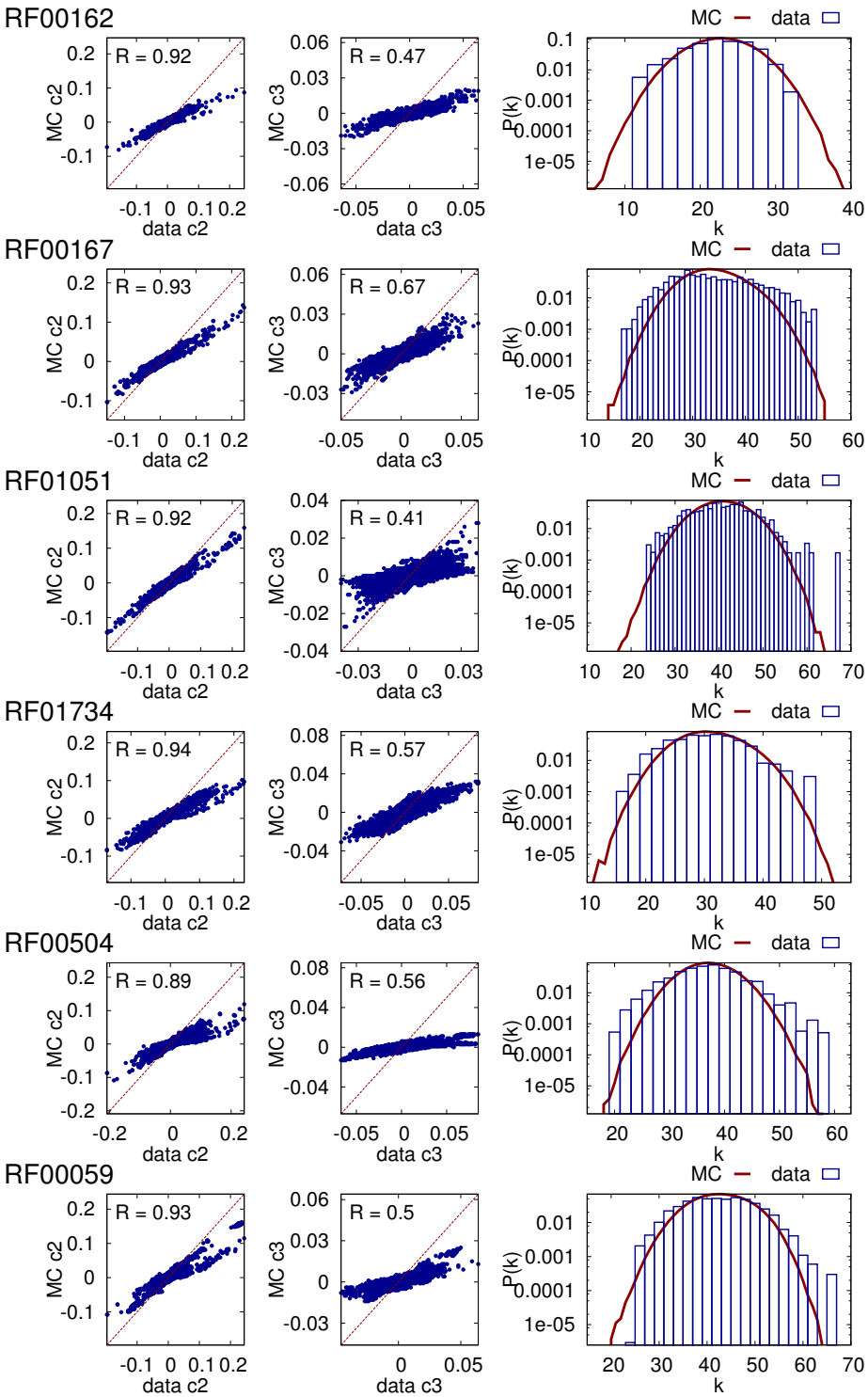


Figure 5.22: **Results for γ_1 ACE + MC-learning**. First column 2-point connected correlations. Second column 3-point connected correlations. Third column probability to see a given number of mutated sites with respect to the consensus sequence.

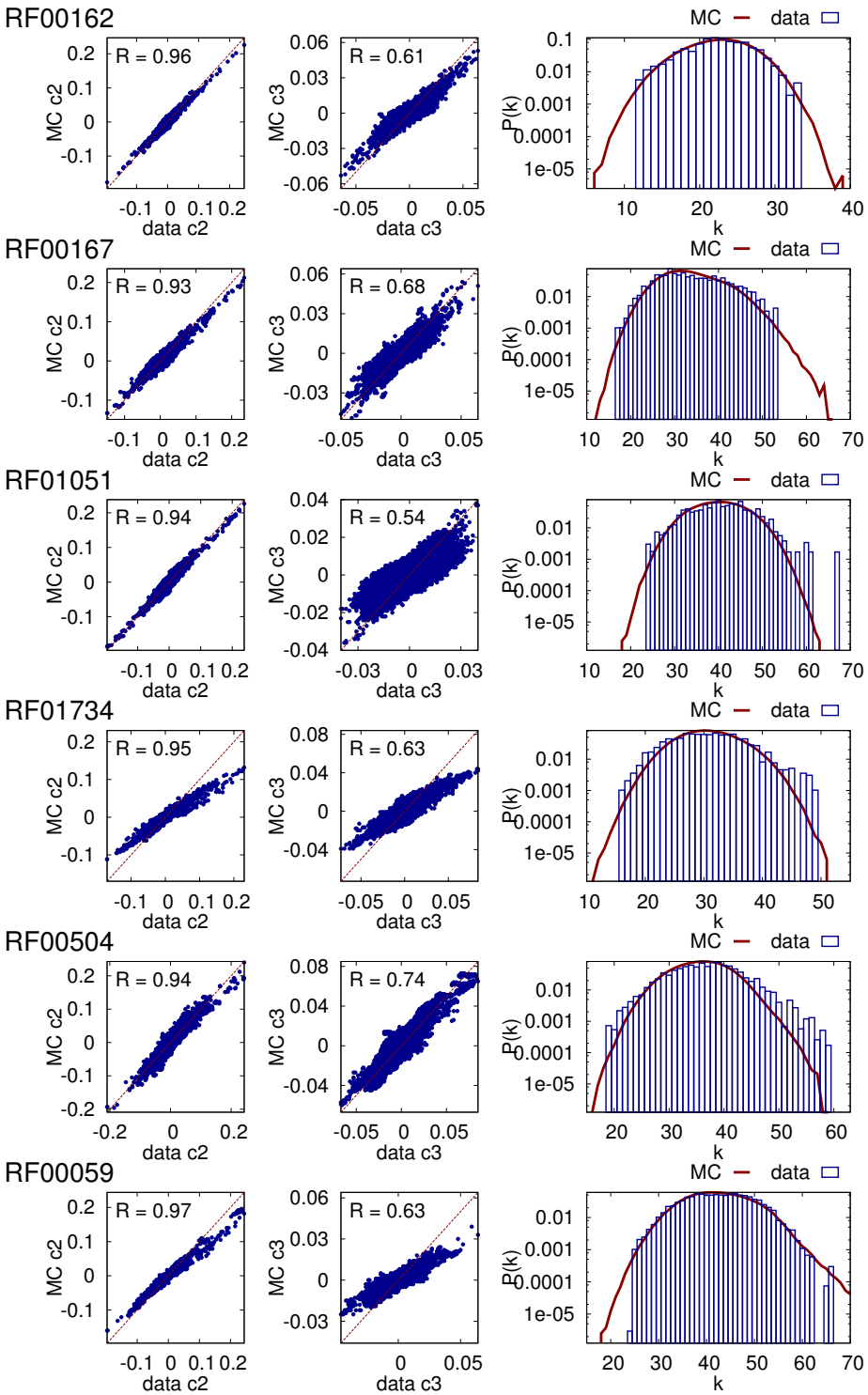


Figure 5.23: **Results for γ_2 ACE + MC-learning**. First column 2-point connected correlations. Second column 3-point connected correlations. Third column probability to see a given number of mutated sites with respect to the consensus sequence.

included while ACE is preferable for tertiary structure predictions only (ACE results are worse for γ_2. Data not shown here). The model inferred with ACE with γ_2 is extremely good: both 2- and 3-point connected correlations are well inferred and the match between the predicted and the observed $P(k)$ is almost perfect. In this case Fig. 5.22 clearly shows that γ_1 over-estimates the optimal regularisation.

- **RF00167**: same as RF00162 a part from the fact that DCA and plmDCA give better results in term of contact predictions also when only the tertiary structure is considered.
- **RF01051**: this is the only case in which ACE works considerably better than DCA and plmDCA for tertiary-structure predictions. Conversely 3-point connected correlations show the worst correlation coefficient $R = 0.54$ and also the tails of the $P(k)$ have not been recognised by the model. Probably the value of γ_2 is still high.
- **RF01734**: here ACE performs worse than the other methods for contact prediction in general. Moreover, the choice of a smaller regularisation seems not to improve generative test results: both Fig. 5.22 and Fig. 5.23 show strongly biased 2- and 3-point connected correlations and the $P(k)$ is not correctly reproduced. Probably a lower γ has to be tested.
- **RF00504**: DCA and plmDCA preform better than ACE for contact predictions. In Fig. 5.23, predicted 2- and 3-point connected correlations are extremely similar to those observed within the MSA, however the model cannot recover the $P(k)$: large part of tails is missing.
- **RF00059**: for contact prediction similar to RF00504, while the generative test gives different results. In this case with γ_2, indeed, the $P(k)$ is surprisingly well reproduced, even if the 2- and 3-point statistics are not: probably, also in this case, we can decrease more the value of γ.

5.5 Conclusions

Within this chapter we have explored the ACE algorithm and tested its performance on both artificial and real data. We claim that ACE is a good alternative to mean-field method when a fine information about the system is needed. Moreover ACE guarantees the inference of a sparse graph and it is reasonably robust to the choice of the stopping threshold. We showed that, even when errors on correlations

are large, the output parameters can be successfully used as input for a MC-learning able to reach convergence. Applications on artificial data have proved ACE producing generative models whose quality is highly better than mean-field and even pseudo-likelihood models. However the larger computational cost of ACE is not justified for the inference of the interaction network only: both mean-field and pseudo-likelihood can equivalently infer interactions in a much shorter time.

Application to RNA confirm artificial model results. ACE is a powerful tool and provide generative models able to extremely well reproduce the statistics of biological data. Competitive results reached by DCA in the framework of RNA structure prediction promise to open a novel exploration of RNA-related topics. ACE has been developed in the scope of application on a wide range of problems and RNA will probably be one of them.

The analysis I have shown here represents the general set of studies that can be performed thanks to ACE. As it is extensively explained in Appendix A, the ACE package is made of three core programs: ACE, MC-learning and the generative test routine. The ACE package code will be released in the next days with the relative paper containing test on different datasets (neurons, proteins, artificial models).

Conclusions

All the projects introduced within this dissertation, related to inverse statistical methods used for interesting applications on biological subjects, are different from each other with respect to some fundamental aspects. As discussed in the Motivations, indeed, two different ways to approach multidisciplinary topics exist: one can apply an existing tool to a new topic or develop a new tool in order to solve a known problem. The former case has been explored within this thesis in chapter 3, where the application of DCA to RNA structure prediction has been discussed.

In the last years DCA has been applied in the field of protein structure characterisation and amazing results have been achieved. However, we have shown that DCA can be successfully applied on RNA data in the scope of computing a coevolution score more reliable than the existing ones. In the related paper we have employed DCA score to improve performance of known algorithms for RNA secondary and tertiary structure prediction. The procedure we designed has been proved to be convincing and competitive. However, the results we have achieved, far from being conclusive, have to be intended as a preliminary evidence of the fact that the use of DCA within existing software for structure prediction would bring significant improvements. Enlarging the use of DCA within the molecular biology community is our future challenge. These days we are implementing a web-server for RNA structure prediction. In the first version it will implement the same software we employed in the analysis performed in the paper, but releases based on the application of DCA within diverse algorithms will follow.

From the theoretical point of view, the paper of chapter 4 has confirmed that DCA is a reliable tool for the inference of interactions in networks, but at the same time it has stressed the weaknesses of mean-field inference. Going beyond contact prediction towards more general and generative models for biomolecules would require a reliable inference of interactions that mean-fields cannot achieve. The development of new tools to face these problems remains challenging. In the last chapter I introduced the ACE algorithm and its generalisation to the Potts model. The analysis both on

artificial and on RNA data confirmed that ACE provides full generative models and computational limitations can be overcome thanks to MC-learning refinements. Also in this case the work on the ACE has not to be considered concluded. The algorithm, even if reliable and refined from different perspectives, is not yet competitive with respect to other more naive approximations due to the huge computational effort needed. Despite the important improvements introduced, discussed within this manuscript, there is still a great deal of work to be done. In particular a theoretical understanding of some of the procedures introduced is still missing. For instance we have empirically proved that the colour compression, fundamental for reducing the otherwise limiting computational cost of Potts implementation, does not entail a significant deterioration of the inference performance. However, the consequences on the cross-entropy expansion of using effective colours substituting some of the original ones and compressing the information stored in these latter are still poorly understood. Knowledge on such theoretical aspects of the algorithm is the first step towards novel improvements or even novel methods.

Appendix A

ACE: short user manual

Beside analytical efforts for the development of an efficient new inference algorithm, my major interest was that this algorithm was performed by an intuitive software, ensuring people coming from diverse backgrounds can easily run it. Thus, we hardly work at some side-codes performing analysis on ACE output or preparing inputs starting from the most common formats used in the community of molecular biology or neuroscience. This appendix schematically lists all the analysis our code performs, all the different parameters one can set and all the output files produced.

A.1 The full analysis script

In order to perform full analysis, of the type showed in the previous section, on both real and artificial data, we built a bash script *RunScript_2.0.sh* running the programs contained in the ACE package and some interfacing Matlab scripts. To launch the analysis both Matlab and a c++ compiler are requested. Main options can be specified to the bash script or directly to one of the programs in the package.

A.1.1 ACE package software

The ACE package contains the three main programs for performing the inference, run the MC-learning and run the generative test analysis.

- **sce** , the ACE algorithm performing inference on data
- **qls** , the MC-learning algorithm used to improve parameters till convergence in case ACE only does not succeed
- **qgt** , the algorithm performing the generative tests on output parameters

All these programs are written in c++. To compile and install them a standard Makefile is used, thus to run the program from within the ACE folder type:

```
$ ./configure
$ make
```

Or equivalently to run ACE from whatever folder type:

```
$ ./configure
$ make install
```

After installation programs have to be individually run. Command line instructions tell the program where to look for input files and where to send output, as well as the setting of various parameters (gamma, theta, etc) and flags (useSparse, etc). Note that numerical parameters may be entered in either scientific (recommended) or standard decimal notation. True/false switches are off by default - if entered in the command line, the corresponding option is set to true. The conventions are given below:

sce The most part of the implementation of sce code has been done by J.P. Barton.

- -(flag name): (type of input expected)
- -d: string
  Default: "." (current directory)
  Path to the directory where the data file is located, and where output will be written.
- -i: string
  Default: "input"
  The location of the file containing a set of correlations from which to infer Ising model parameters.
- -o: string
  Default: "output"
  The location of the file where output is to be sent. Each different type of output file will have a different file type, e.g. .j for couplings.
- -cmap: string
  Default: none
  When the network of interactions (e.g. contact map) is known, a list of preselected 2-site clusters can be given. "string" represent the name and the location of the file from which clusters are read. The extension of the file has to be .cl (indexing from 0)

– -inputcl: string
Default: none
When a list of interesting clusters (e.g. from previously runs) is known, this list of preselected n-site clusters can be given and used for inference. "string" represent the name and the location of the file from which clusters are read. The extension of the file has to be .cl (indexing from 0)

– -cl: none
Print the list of selected clusters in a file .cl in the output folder

– -b: real number
Default: $1.0e+4$
Number of samples used to compute the correlations. Used to determine the inference error.

– -kmin: integer
Default: 1
Minimum cluster size, useful for avoiding the inference of models that are too sparse. The algorithm will continue to lower the threshold until clusters of at least this size are selected.

– -kmax: integer
Default: N (system size)
Maximum cluster size. The algorithm will halt when clusters of this size are selected.

– -cmax: integer
Default: $10e+8$
Maximum number of configurations per cluster. The algorithm will halt when clusters of size k (where k is defined such that $\langle q_{eff} \rangle^k = 10e+8$) are selected. This command is redundant with kmax, but it is helps users to better estimate a time limit for runs.

– -t: real number
Default: none
Run the algorithm at the input value of t, in scientific or standard decimal notation. This line is intended to be used when inference is to be done only for a single value of t, and will be overridden if thetaMax and thetaMin are set different from t.

– -tmin: real number
  Default: $1.0e-10$
  The minimum value of t. See description of -ts below for more information.
– -tmax: real number
  Default: $1.0e+0$
  The maximum value of t. See description of -ts below for more information.
– -ts: real number
  Default: 1.05
  The logarithmic step size to use for successive updates of t. When the program loops over different values of t, it begins by running the algorithm at the largest value of the cutoff and stores the cluster information. The algorithm is then re-run for successively smaller values of the cutoff, $t_{i+1} = \frac{t_i}{thetastep}$, until $t < thetaMin$. These re-runs use the previously stored cluster information, so they take considerably less time to run.
– -trec: real number
  Default: 3.0 (set 0 to avoid recording)
  The logarithmic step size for t to record the inferred parameters. Given this interval the chosen value corresponds the t producing the minimum error on correlations.
– -mcb: integer
  Default: $4.0e+4$
  Number of Monte Carlo samples to take to check inference error.
– -mcr: integer
  Default: 1
  Number of independent Monte Carlo runs to perform.
– -g0: real number
  Default: $1.0e-4$
  The L0 regularization strength. Using this flag also turns on L0 regularization.
– -g2: real number
  Default: 0.0
  The L2 regularization strength. L2 regularization is enabled by setting the regularization strength to a nonzero value using this flag, or by using the -ag flag below.
– -gi: none
  Use gauge invariant L2 regularization for couplings.

- -ag: none
  Attempt to set the L0 and L2 regularization strengths to their optimal values, based on the number of samples (input) in the data.
- -l0: none
  If selected, L0-norm (sparse) regularization is used.
- -lax: none
  If selected, use a laxer cluster construction rule.
- -v: none
  Enable verbose output.

qls The implementation of qls code has been done by J.P. Barton.

- -(flag name): (type of input expected)
- -d: string
  Default: "." (current directory)
  Path to the directory where the data file is located, and where output will be written.
- -i: string
  Default: "input"
  The location of the file containing a set of couplings, the starting values for the Monte Carlo learning algorithm.
- -o: string
  Default: "output"
  The location of the file where output is to be sent. Each different type of output file will have a different file type, e.g. .j for couplings.
- -c: string
  Default: "input"
  The location of the file containing the set of correlations to reproduce (i.e. the correlations obtained from the data).
- -s: string
  Default: none
  Starting configuration for MC simulations. (File extension requested .dat)
- -g2: real number
  Default: 0.0
  The L2 regularization strength. L2 regularization is enabled by setting the

regularization strength to a nonzero value using this flag, or by using the -ag flag below.

- -gi: none
  Use gauge invariant L2 regularization for couplings.
- -ag: none
  Attempt to set the L2 regularization strengths to its optimal value, based on the number of samples (input) in the data.
- -b: real number
  Default: $1.0e+4$
  Number of samples used to compute the correlations. Used to determine the inference error.
- -mcb: integer
  Default: $8.0e+5$
  Number of Monte Carlo samples to take to check inference error.
- -mcr: integer
  Default: 1
  Number of independent Monte Carlo runs to perform.
- -e: real number
  Default: 1.0
  Maximum tolerable error threshold. The MC learning algorithm will continue to run until the error on the one- and two-point correlations falls below this level.
- -v: none
  Enable verbose output.

qgt

- -(flag name): (type of input expected)
- -d: string
  Default: "." (current directory)
  Path to the directory where the data file is located, and where output will be written.
- -i: string
  Default: "input"
  The location of the file containing a set of couplings for the Monte Carlo sampling.

- -o: string
  Default: "output"
  The location of the file where output is to be sent.
- -c or -cons: string
  Default: "input" The location of the file containing the reference sequence for mutation probability. e.g. consensus or wildtype sequence
- -m or -msa: string
  Default: "input"
  The location of the file containing the compressed alignment .cmsa
- -w: string
  Default: "input"
  The location of the file containing the re-weighting vector
- -s: string
  Default: none
  Starting configuration for MC simulations.
- -b: real number
  Default: $1.0e+4$
  Number of samples used to compute the correlations. (MSA size)
- -g2: real number
  Default: 0.0
  The L2 regularization strength. L2 regularization is enabled by setting the regularization strength to a nonzero value using this flag, or by using the -ag flag below.
- -ag: none
  Attempt to set the L2 regularization strengths to its optimal value, based on the number of samples (input) in the data.
- -mcb: integer
  Default: $8.0e+5$
  Number of Monte Carlo samples to take to check inference error.
- -mcr: integer
  Default: 1
  Number of independent Monte Carlo runs to perform.
- -msaout: none
  If selected, print Monte Carlo alignment in outputfile.msa and energies of sequences in outputfile.e

- -p3: none
  Default: false (needed for VERY large systems)
  Compute also 3-point correlations and print all of them in an output files
- -p3red: none
  Default: false (needed for large systems)
  Compute also 3-point correlations and print all of those that are larger than $\langle p \rangle^3$. Among the other ones print only 1 over 50.
- -err: none
  Compute and write connected correlation errors on statistics. (Written in the corresponding file)
- -v: none
  Enable verbose output.

Input and output files Files are distinguished thanks to their extension referring to a particular type or formatting of data found inside. Standard extension are the following:

- *.p
  Contains the input frequencies and correlations. Frequencies are listed before, colours belonging to the same site stays on the same line. Then correlations for pair of sited ij are listed. Again the same line contains all colour combinations existing for the considered pair of site. Colours are ordered according to site i first and then site j. Only $j > i$ pairs are included and ordered according to site i first and then to site j.
- *.j
  Contains output parameters. Fields and couplings are listed in the same format as frequencies and correlations.
- *.sce
  Contains supplementary information about ace iterations and convergence. Columns contain in the order: t, $epsilon_P$, $epsilon_C$, $epsilon_{max}$, final cross-entropy, maximum cluster size, total number of computed clusters, total number of significant clusters, $L2$-norm regularisation term for both J and h.
- *.cl
  Contains a list of clusters. Each cluster has to contain more than one site. Sites belonging to the same cluster are written on the same line.

– *.fit

  Contains MC-learning iteration outputs. Columns contain in the order: iteration , $epsilon_P$, $epsilon_P 2$, $epsilon_{max}$,$L\infty$-norm of weights.

– *.cmsa

  Contains the MSA in a compressed format made of a single column. Only symbols seen at least once are reported. -1 flags divide different sites and colours. After the flag the full set of sequences containing that colours on than site is listed using different numbers to represent different sequences. Sites are listed first and then colours.

– *.cons

  Contains the consensus sequence needed for the computation of $P(k)$.

– *.wgt

  Contains the re-weighting vector assigning a weight to each sequence according to sequence similarity.

– *.m

  Contains both input and output magnetisations. Input on the first column output on the second one.

– *.p2

  Contains both input and output 2-point correlations. Input on the first column output on the second one.

– *.c2

  Contains both input and output 2-point connected correlations. Input on the first column output on the second one.

– *.p3

  Contains both input and output 3-point correlations. Input on the first column output on the second one.

– *.c3

  Contains both input and output 3-point connected correlations. Input on the first column output on the second one.

– *.pk

  Contains the $P(k)$ distribution. In the first column k is listed, in the second column the input $P(k)$ and in the third one the output $P(k)$.

– *.msa

  Contains the output MSA made of MC sequences.

– *.e
  Contains the energies of the sequences in *.msaout

– *.msae
  Contains the energies of the sequences in *.cmsa according to the inferred model.

A.1.2 RunScript_2.0.sh package software

The bash script *RunScript_2.0.sh* runs both ACE programs and some Matlab scripts for pre- and post-processing of data. The major advantage given by the use of this scripts is that it parallelises the algorithm and output analysis. After that all input files are ready and also DCA and plmDCA have run (if required), the ACE is launched on input data. If the option for t recording is chosen each time a new set of parameters is recorded at a certain t the script runs first the MC-learning and then the generative tests on both learned and non-learned parameters. The minimum number of cores required is thus two. This parallelisation of inference and analysis is particularly interesting in order to monitor the convergence of the algorithm: comparing the generative tests of learned and non-learned parameter one can easily understand if it is the case to stop the algorithm or if a better approximation of the cross-entropy is needed to obtain reasonable results. For artificial models also the contact map and the parameter comparison is performed at each recorded t. *RunScript_2.0.sh* contains and manages runs for the following programs:

CreateModel this Matlab function build artificial models. It extracts random parameters from Gaussian or Uniform distributions and assign them to different types of graphs such as 1D chains, Erdos-Renyi random graphs or specific RNA-based graphs.

qDataMC It is a c++ program preforming a MC-sampling of a given q-state Potts model.

rnaDCA Again a Matlab function for the DCA mean-field inference starting from a MSA.

plmDCA_symmetric This program belongs to the *plmDCA_symmetric_v2* package for the plmDCA. This software have not been developed by the author of this dissertation but the corresponding reference is [19].

ComputeErrors This Matlab function takes as input a MSA and compute all the statistical observable needed by the ACE algorithm. All ACE input files are written in the correct format from this function. It performs the colour compression and compute exact or approximated (depending on system size and user input) errors on inferred parameters. The artificial model version map compressed correlations to the full original model to prepare comparison among inferred parameters.

GaugeFixing It is a Matlab function to move both inferred parameters and input ones, for artificial models, to a given lattice-gas gauge.

PredictMap A Matlab function for the computation of scalar scores from coupling and true positive rates based on the input network of interactions. When real data are used a contact map has to be specified.

ACE package All the programs in the ACE package can be run within *RunScript_2.0.sh*. Only the most common option have anyway been included. For a more personal use we recommend to launch ACE programs individually.

A.1.3 RunScript_2.0.sh input options

The following list contains all the input options one can give to *RunScript_2.0.sh*:

- -g: real number
  Regularisation strength
- -i: string
  Input directory
- -o: string
  Output directory
- -p: real number
  Colour compression threshold
- -m: string
  Colour compression method: one can both specify a threshold on the frequency "pmin“ or on the entropy contribution "entr”
- -r: real number
  Re-weighting threshold
- -f: real number
  Frequency for t recording of parameters within ACE

- -M: string
  Matlab path
- -S: string
  ACE path
- -N: integer
  Number of sites in the model
- -q: integer
  Max number of colours per site. For artificial model specifies the number of colours of the input Potts model.
- -B: integer
  Sample size
- -b: integer
  Change the number of samples computed by the MC for generative tests
- -J: real number
  Couplings variance. For artificial models only.
- -H: real number
  Field variance. For artificial model only.
- -h: real number
  Fields extra-mean for bimodal distribution. For artificial models only.
- -t: string
  Type of model.
  - '1D' = 1D Potts ring
  - 'ER00' = Erdos Renyi graph
    The two numbers that follow ER represent the p probability to create a link between two nodes. Examples: ER20 , p=0.2, ER59, p=0.59, etc.
  - 'SS' = Hairpin loop graph with Watson-Crick base pairs
    Base pairs start from 1 - N and come up (2 - (N-1), 3 - (N-2), etc.). The number of W-C base pais has to be expressed in the two numbers that follows the type. Also a certain numbers of tertiary contacts can be added, use other two numbers in the name.

  Adding an S to the end of the type name means "solve analytically the model". For 1D model this is quite fast (transfer matrix method), but for the other models is computationally very expensive. Adding an U means "do not solve

analytically the model". Between the type and the SU it is possible to put a whatever name. Some examples: ER40_test1_U = Erdos Renyi p=0.4 unsolved called "test1", SS1708bS = secondary structure with 17 W-C base pairs and 8 tertiary.

- -x: string
  Use a select list of clusters as input. Specify the file containing the list
- -c: 1
  Create model. For artificial models only.
- -n: 1
  Run MC-sampling form input parameters. For artificial models only.
- -e: 1
  Run ComputeErrors
- -d: 1
  Run mean-field analysis on the colour-compressed model
- -s: 1
  Run ACE
- -a: 1
  Run the full analysis on output of ACE algorithm. It includes qls and qgt for all recorded value of t
- -O: 1
  Run other algorithms (DCA and plmDCA) on the non-compressed model in the original alignment.
- -R: 1
  Run analysis with real parameters to check thermalisation of MC routines.

References

[1] Simona Cocco, Stanislas Leibler, and Rémi Monasson. Neuronal couplings between retinal ganglion cells inferred by efficient inverse statistical physics methods. *Proceedings of the National Academy of Sciences*, 106(33):14058–14062, 2009.

[2] Jaclyn K Mann, John P Barton, Andrew L Ferguson, Saleha Omarjee, Bruce D Walker, Arup Chakraborty, and Thumbi Ndung'u. The fitness landscape of HIV-1 gag: advanced modeling approaches and validation of model predictions by in vitro testing. 2014.

[3] Faruck Morcos, Andrea Pagnani, Bryan Lunt, Arianna Bertolino, Debora S Marks, Chris Sander, Riccardo Zecchina, José N Onuchic, Terence Hwa, and Martin Weigt. Direct-coupling analysis of residue coevolution captures native contacts across many protein families. *Proceedings of the National Academy of Sciences*, 108(49):E1293–E1301, 2011.

[4] Marc Bailly-Bechet, Alfredo Braunstein, Andrea Pagnani, Martin Weigt, and Riccardo Zecchina. Inference of sparse combinatorial-control networks from gene-expression data: a message passing approach. *BMC Bioinformatics*, 11(1):355, 2010.

[5] Federico Ricci-Tersenghi. The Bethe approximation for solving the inverse Ising problem: a comparison with other inference methods. *Journal of Statistical Mechanics: Theory and Experiment*, 2012(08):P08015, 2012.

[6] David H Ackley, Geoffrey E Hinton, and Terrence J Sejnowski. A learning algorithm for Boltzmann machines*. *Cognitive Science*, 9(1):147–169, 1985.

[7] Simona Cocco and Rémi Monasson. Adaptive cluster expansion for inferring Boltzmann machines with noisy data. *Physical Review Letters*, 106(9):090601, 2011.

[8] William Bialek and Rama Ranganathan. Rediscovering the power of pairwise interactions. *arXiv preprint arXiv:0712.4397*, 2007.

[9] Edwin T Jaynes. Information theory and statistical mechanics. *Physical Review*, 106(4):620, 1957.

[10] John J Hopfield. Neural networks and physical systems with emergent collective computational abilities. *Proceedings of the National Academy of Sciences*, 79(8):2554–2558, 1982.

[11] T Plefka. Convergence condition of the TAP equation for the infinite-ranged Ising spin glass model. *Journal of Physics A: Mathematical and general*, 15(6):1971, 1982.

[12] Antoine Georges and Jonathan S Yedidia. How to expand around mean-field theory using high-temperature expansions. *Journal of Physics A: Mathematical and General*, 24(9):2173, 1991.

[13] David J Thouless, Philip W Anderson, and Robert G Palmer. Solution of'solvable model of a spin glass'. *Philosophical Magazine*, 35(3):593–601, 1977.

[14] Marc Mezard and Thierry Mora. Constraint satisfaction problems and neural networks: a statistical physics perspective. *Journal of Physiology-Paris*, 103(1):107–113, 2009.

[15] Yasser Roudi, Joanna Tyrcha, and John Hertz. Ising model for neural data: model quality and approximate methods for extracting functional connectivity. *Physical Review E*, 79(5):051915, 2009.

[16] Vitor Sessak and Rémi Monasson. Small-correlation expansions for the inverse Ising problem. *Journal of Physics A: Mathematical and Theoretical*, 42(5):055001, 2009.

[17] H Chau Nguyen and Johannes Berg. Mean-field theory for the inverse Ising problem at low temperatures. *Physical Review Letters*, 109(5):050602, 2012.

[18] Aurélien Decelle and Federico Ricci-Tersenghi. Solving the inverse Ising problem by mean-field methods in a clustered phase space with many states. *arXiv preprint arXiv:1501.03034*, 2015.

[19] Magnus Ekeberg, Cecilia Lövkvist, Yueheng Lan, Martin Weigt, and Erik Aurell. Improved contact prediction in proteins: using pseudolikelihoods to infer Potts models. *Physical Review E*, 87(1):012707, 2013.

[20] Julian Besag. Statistical analysis of non-lattice data. *The Statistician*, pages 179–195, 1975.

[21] Pradeep Ravikumar, Martin J Wainwright, John D Lafferty, et al. High-dimensional Ising model selection using L1-regularized logistic regression. *The Annals of Statistics*, 38(3):1287–1319, 2010.

[22] Erik Aurell and Magnus Ekeberg. Inverse Ising inference using all the data. *Physical Review Letters*, 108(9):090201, 2012.

[23] Simona Cocco and Rémi Monasson. Adaptive cluster expansion for the inverse Ising problem: convergence, algorithm and tests. *Journal of Statistical Physics*, 147(2):252–314, 2012.

[24] Duccio Malinverni, Simone Marsili, Alessandro Barducci, and Paolo De Los Rios. Large-scale conformational transitions and dimerization are encoded in the amino-acid sequences of Hsp70 chaperones. *PLoS Comput Biol*, 11(6):e1004262, 2015.

[25] Andrew L Ferguson, Jaclyn K Mann, Saleha Omarjee, Thumbi Ndung'u, Bruce D Walker, and Arup K Chakraborty. Translating HIV sequences into quantitative fitness landscapes predicts viral vulnerabilities for rational immunogen design. *Immunity*, 38(3):606–617, 2013.

[26] Thomas A Hopf, Lucy J Colwell, Robert Sheridan, Burkhard Rost, Chris Sander, and Debora S Marks. Three-dimensional structures of membrane proteins from genomic sequencing. *Cell*, 149(7):1607–1621, 2012.

[27] Lukas Burger and Erik Van Nimwegen. Disentangling direct from indirect co-evolution of residues in protein alignments. 2010.

[28] Michael Socolich, Steve W Lockless, William P Russ, Heather Lee, Kevin H Gardner, and Rama Ranganathan. Evolutionary information for specifying a protein fold. *Nature*, 437(7058):512–518, 2005.

[29] Steve W Lockless and Rama Ranganathan. Evolutionarily conserved pathways of energetic connectivity in protein families. *Science*, 286(5438):295–299, 1999.

[30] Junmei Chen and Wesley E Stites. Higher-order packing interactions in triple and quadruple mutants of staphylococcal nuclease. *Biochemistry*, 40(46):14012–14019, 2001.

[31] Alexander Schug, Martin Weigt, José N Onuchic, Terence Hwa, and Hendrik Szurmant. High-resolution protein complexes from integrating genomic information with molecular simulation. *Proceedings of the National Academy of Sciences*, 106(52):22124–22129, 2009.

[32] Dana Pe'er, Aviv Regev, and Amos Tanay. Minreg: inferring an active regulator set. *Bioinformatics*, 18(suppl 1):S258–S267, 2002.

[33] Eran Segal, Michael Shapira, Aviv Regev, Dana Pe'er, David Botstein, Daphne Koller, and Nir Friedman. Module networks: identifying regulatory modules and their condition-specific regulators from gene expression data. *Nature Genetics*, 34(2):166–176, 2003.

[34] Timothy R Lezon, Jayanth R Banavar, Marek Cieplak, Amos Maritan, and Nina V Fedoroff. Using the principle of entropy maximization to infer genetic interaction networks from gene expression patterns. *Proceedings of the National Academy of Sciences*, 103(50):19033–19038, 2006.

[35] Richard Bonneau, Marc T Facciotti, David J Reiss, Amy K Schmid, Min Pan, Amardeep Kaur, Vesteinn Thorsson, Paul Shannon, Michael H Johnson, J Christopher Bare, et al. A predictive model for transcriptional control of physiology in a free living cell. *Cell*, 131(7):1354–1365, 2007.

[36] Fred Rieke, David Karsten Warland, and William S. Bialek. *Spikes : exploring the neural code*. Cambridge, Mass. : MIT Press, c1997, 1997.

[37] Jonathan W Pillow, Jonathon Shlens, Liam Paninski, Alexander Sher, Alan M Litke, EJ Chichilnisky, and Eero P Simoncelli. Spatio-temporal correlations and visual signalling in a complete neuronal population. *Nature*, 454(7207):995–999, 2008.

[38] Apostolos P Georgopoulos, Andrew B Schwartz, and Ronald E Kettner. Neuronal population coding of movement direction. *Science*, 233(4771):1416–1419, 1986.

[39] Leland H Hartwell, John J Hopfield, Stanislas Leibler, and Andrew W Murray. From molecular to modular cell biology. *Nature*, 402:C47–C52, 1999.

[40] Gašper Tkačik, Jason S Prentice, Vijay Balasubramanian, and Elad Schneidman. Optimal population coding by noisy spiking neurons. *Proceedings of the National Academy of Sciences*, 107(32):14419–14424, 2010.

[41] Jonathon Shlens, Greg D Field, Jeffrey L Gauthier, Matthew I Grivich, Dumitru Petrusca, Alexander Sher, Alan M Litke, and EJ Chichilnisky. The structure of multi-neuron firing patterns in primate retina. *The Journal of Neuroscience*, 26(32):8254–8266, 2006.

[42] Elad Schneidman, Michael J Berry, Ronen Segev, and William Bialek. Weak pairwise correlations imply strongly correlated network states in a neural population. *Nature*, 440(7087):1007–1012, 2006.

[43] Gaia Tavoni, Ulisse Ferrari, Francesco P Battaglia, Simona Cocco, and Rémi Monasson. Inferred network from prefrontal cortex activity of rats unveils cell assemblies. *BMC Neuroscience*, 14(Suppl 1):O20, 2013.

[44] Benjamin Dunn, Maria Mørreaunet, and Yasser Roudi. Correlations and functional connections in a population of grid cells. *PLoS Computational Biology*, 11(2):e1004052–e1004052, 2015.

[45] Doeke R Hekstra, Simona Cocco, Remi Monasson, and Stanislas Leibler. Trend and fluctuations: analysis and design of population dynamics measurements in replicate ecosystems. *Physical Review E*, 88(6):062714, 2013.

[46] William Bialek, Andrea Cavagna, Irene Giardina, Thierry Mora, Edmondo Silvestri, Massimiliano Viale, and Aleksandra M Walczak. Statistical mechanics for natural flocks of birds. *Proceedings of the National Academy of Sciences*, 109(13):4786–4791, 2012.

[47] Alessandro Attanasi, Andrea Cavagna, Lorenzo Del Castello, Irene Giardina, Tomas S Grigera, Asja Jelić, Stefania Melillo, Leonardo Parisi, Oliver Pohl, Edward Shen, et al. Information transfer and behavioural inertia in starling flocks. *Nature Physics*, 10(9):691–696, 2014.

[48] Charles K Fisher and Pankaj Mehta. The transition between the niche and neutral regimes in ecology. *Proceedings of the National Academy of Sciences*, 111(36):13111–13116, 2014.

[49] Karoline Faust, J Fah Sathirapongsasuti, Jacques Izard, Nicola Segata, Dirk Gevers, Jeroen Raes, and Curtis Huttenhower. Microbial co-occurrence relationships in the human microbiome. *PLoS Comput Biol*, 8(7):e1002606–e1002606, 2012.

[50] Jonathan Friedman and Eric J Alm. Inferring correlation networks from genomic survey data. *PLoS Computational Biology*, 2012.

[51] David Lazer, Alex Sandy Pentland, Lada Adamic, Sinan Aral, Albert Laszlo Barabasi, Devon Brewer, Nicholas Christakis, Noshir Contractor, James Fowler, Myron Gutmann, et al. Life in the network: the coming age of computational social science. *Science (New York, NY)*, 323(5915):721, 2009.

[52] Stephen P Borgatti, Ajay Mehra, Daniel J Brass, and Giuseppe Labianca. Network analysis in the social sciences. *Science*, 323(5916):892–895, 2009.

[53] Nicholas A Christakis and James H Fowler. The spread of obesity in a large social network over 32 years. *New England Journal of Medicine*, 357(4):370–379, 2007.

[54] Ciro Cattuto, Wouter Van den Broeck, Alain Barrat, Vittoria Colizza, Jean-François Pinton, and Alessandro Vespignani. Dynamics of person-to-person interactions from distributed RFID sensor networks. *PloS One*, 5(7):e11596, 2010.

[55] Michal Kosinski, David Stillwell, and Thore Graepel. Private traits and attributes are predictable from digital records of human behavior. *Proceedings of the National Academy of Sciences*, 110(15):5802–5805, 2013.

[56] Jerome B Cohen, Fischer Black, and Myron Scholes. The valuation of option contracts and a test of market efficiency. *The Journal of Finance*, 27(2):399–417, 1972.

[57] Robert C Merton. Option pricing when underlying stock returns are discontinuous. *Journal of Financial Economics*, 3(1):125–144, 1976.

[58] Hideki Takayasu. *Empirical science of financial fluctuations: the advent of econophysics*. Springer Science & Business Media, 2013.

[59] Vasiliki Plerou, Parameswaran Gopikrishnan, Bernd Rosenow, Luís A Nunes Amaral, and H Eugene Stanley. Universal and nonuniversal properties of cross correlations in financial time series. *Physical Review Letters*, 83(7):1471, 1999.

[60] Esteban Moro, Javier Vicente, Luis G Moyano, Austin Gerig, J Doyne Farmer, Gabriella Vaglica, Fabrizio Lillo, and Rosario N Mantegna. Market impact and trading profile of hidden orders in stock markets. *Physical Review E*, 80(6):066102, 2009.

[61] Jean-Philippe Bouchaud. Crises and collective socio-economic phenomena: simple models and challenges. *Journal of Statistical Physics*, 151(3-4):567–606, 2013.

[62] Martin Weigt, Robert A White, Hendrik Szurmant, James A Hoch, and Terence Hwa. Identification of direct residue contacts in protein–protein interaction by message passing. *Proceedings of the National Academy of Sciences*, 106(1):67–72, 2009.

[63] Huanwang Yang, Fabrice Jossinet, Neocles Leontis, Li Chen, John Westbrook, Helen Berman, and Eric Westhof. Tools for the automatic identification and classification of RNA base pairs. *Nucleic Acids Research*, 31(13):3450–3460, 2003.

[64] Patrick Gendron, Sébastien Lemieux, and François Major. Quantitative analysis of nucleic acid three-dimensional structures. *Journal of Molecular Biology*, 308(5):919–936, 2001.

[65] Fabrice Jossinet and Eric Westhof. S2S-Assemble2: a semi-automatic bioinformatics framework to study and model RNA 3D architectures. *Handbook of RNA Biochemistry: Second, Completely Revised and Enlarged Edition*, pages 667–686, 2014.

[66] Neocles B Leontis and Eric Westhof. Geometric nomenclature and classification of RNA base pairs. *RNA*, 7(04):499–512, 2001.

[67] Robert T Batey, Robert P Rambo, Jennifer A Doudna, et al. Tertiary motifs in RNA structure and folding. *Angewandte Chemie International Edition*, 38(16):2326–2343, 1999.

[68] Philippe Brion and Eric Westhof. Hierarchy and dynamics of RNA folding. *Annual Review of Biophysics and Biomolecular Structure*, 26(1):113–137, 1997.

[69] Poul Nissen, Jeffrey Hansen, Nenad Ban, Peter B Moore, and Thomas A Steitz. The structural basis of ribosome activity in peptide bond synthesis. *Science*, 289(5481):920–930, 2000.

[70] Walter Gilbert. Origin of life: the RNA world. *Nature*, 319(6055), 1986.

[71] Alexandre Dawid, Bastien Cayrol, and Hervé Isambert. RNA synthetic biology inspired from bacteria: construction of transcription attenuators under antisense regulation. *Physical Biology*, 6(2):025007, 2009.

[72] Igor Ulitsky, Alena Shkumatava, Calvin H Jan, Hazel Sive, and David P Bartel. Conserved function of lincRNAs in vertebrate embryonic development despite rapid sequence evolution. *Cell*, 147(7):1537–1550, 2011.

[73] Jakob Skou Pedersen, Gill Bejerano, Adam Siepel, Kate Rosenbloom, Kerstin Lindblad-Toh, Eric S Lander, Jim Kent, Webb Miller, and David Haussler. Identification and classification of conserved RNA secondary structures in the human genome. *PLoS Comput Biol*, 2(4):e33, 2006.

[74] Nobuhiro Go and Hiroshi Taketomi. Respective roles of short-and long-range interactions in protein folding. *Proceedings of the National Academy of Sciences*, 75(2):559–563, 1978.

[75] David Chandler and Jerome K Percus. Introduction to modern statistical mechanics. *Physics Today*, 41(12):114–118, 2008.

[76] Aurelie Lescoute, Neocles B Leontis, Christian Massire, and Eric Westhof. Recurrent structural RNA motifs, isostericity matrices and sequence alignments. *Nucleic Acids Research*, 33(8):2395–2409, 2005.

[77] Eric P Nawrocki, Sarah W Burge, Alex Bateman, Jennifer Daub, Ruth Y Eberhardt, Sean R Eddy, Evan W Floden, Paul P Gardner, Thomas A Jones, John Tate, et al. Rfam 12.0: updates to the RNA families database. *Nucleic Acids Research*, page gku1063, 2014.

[78] Eric P Nawrocki, Diana L Kolbe, and Sean R Eddy. Infernal 1.0: inference of RNA alignments. *Bioinformatics*, 25(10):1335–1337, 2009.

[79] Sean R Eddy and Richard Durbin. RNA sequence analysis using covariance models. *Nucleic Acids Research*, 22(11):2079–2088, 1994.

[80] Saul B Needleman and Christian D Wunsch. A general method applicable to the search for similarities in the amino acid sequence of two proteins. *Journal of Molecular Biology*, 48(3):443–453, 1970.

[81] Peter H Sellers. On the theory and computation of evolutionary distances. *SIAM Journal on Applied Mathematics*, 26(4):787–793, 1974.

[82] Temple F Smith and Michael S Waterman. Identification of common molecular subsequences. *Journal of Molecular Biology*, 147(1):195–197, 1981.

[83] Richard Durbin, Sean R Eddy, Anders Krogh, and Graeme Mitchison. *Biological sequence analysis: probabilistic models of proteins and nucleic acids*. Cambridge University Press, 1998.

[84] Michael Gribskov, Andrew D McLachlan, and David Eisenberg. Profile analysis: detection of distantly related proteins. *Proceedings of the National Academy of Sciences*, 84(13):4355–4358, 1987.

[85] Daniel H Younger. Recognition and parsing of context-free languages in time n 3. *Information and control*, 10(2):189–208, 1967.

[86] Eric Paul Nawrocki. *Structural RNA homology search and alignment using covariance models*. WASHINGTON UNIVERSITY IN ST. LOUIS, 2009.

[87] Eric P Nawrocki and Sean R Eddy. Infernal 1.1: 100-fold faster RNA homology searches. *Bioinformatics*, 29(22):2933–2935, 2013.

[88] Wipapat Kladwang, Fang-Chieh Chou, and Rhiju Das. Automated RNA structure prediction uncovers a missing link in double glycine riboswitches. *arXiv preprint arXiv:1110.0800*, 2011.

[89] Christian Laing and Tamar Schlick. Computational approaches to 3D modeling of RNA. *Journal of Physics: Condensed Matter*, 22(28):283101, 2010.

[90] Zhichao Miao, Ryszard W Adamiak, Marc-Frédérick Blanchet, Michal Boniecki, Janusz M Bujnicki, Shi-Jie Chen, Clarence Cheng, Grzegorz Chojnowski, Fang-Chieh Chou, Pablo Cordero, et al. RNA-Puzzles Round II: assessment of RNA structure prediction programs applied to three large RNA structures. *RNA*, 2015.

[91] Michael Zuker and Patrick Stiegler. Optimal computer folding of large RNA sequences using thermodynamics and auxiliary information. *Nucleic Acids Research*, 9(1):133–148, 1981.

[92] Ruxandra I Dima, Changbong Hyeon, and D Thirumalai. Extracting stacking interaction parameters for RNA from the data set of native structures. *Journal of Molecular Biology*, 347(1):53–69, 2005.

[93] RR Gutell, A Power, GZ Hertz, EJ Putz, and GD Stormo. Identifying constraints on the higher-order structure of RNA: continued development and application of comparative sequence analysis methods. *Nucleic Acids Research*, 20(21):5785–5795, 1992.

[94] David KY Chiu and Ted Kolodziejczak. Inferring consensus structure from nucleic acid sequences. *Computer Applications in the Biosciences: CABIOS*, 7(3):347–352, 1991.

[95] Ruth Nussinov and Ann B Jacobson. Fast algorithm for predicting the secondary structure of single-stranded RNA. *Proceedings of the National Academy of Sciences*, 77(11):6309–6313, 1980.

[96] Stephan H Bernhart, Ivo L Hofacker, Sebastian Will, Andreas R Gruber, and Peter F Stadler. RNAalifold: improved consensus structure prediction for RNA alignments. *BMC Bioinformatics*, 9(1):474, 2008.

[97] Robert J Klein and Sean R Eddy. RSEARCH: finding homologs of single structured RNA sequences. *BMC Bioinformatics*, 4(1):44, 2003.

[98] José Almeida Cruz, Marc-Frédérick Blanchet, Michal Boniecki, Janusz M Bujnicki, Shi-Jie Chen, Song Cao, Rhiju Das, Feng Ding, Nikolay V Dokholyan, Samuel Coulbourn Flores, et al. RNA-Puzzles: a CASP-like evaluation of RNA three-dimensional structure prediction. *RNA*, 18(4):610–625, 2012.

[99] Wipapat Kladwang, Christopher C VanLang, Pablo Cordero, and Rhiju Das. A two-dimensional mutate-and-map strategy for non-coding RNA structure. *Nature Chemistry*, 3(12):954–962, 2011.

[100] Rhiju Das and David Baker. Automated de novo prediction of native-like RNA tertiary structures. *Proceedings of the National Academy of Sciences*, 104(37):14664–14669, 2007.

[101] Carol A Rohl, Charlie EM Strauss, Kira MS Misura, and David Baker. Protein structure prediction using Rosetta. *Methods in Enzymology*, 383:66–93, 2004.

[102] Rhiju Das, Madhuri Kudaravalli, Magdalena Jonikas, Alain Laederach, Robert Fong, Jason P Schwans, David Baker, Joseph A Piccirilli, Russ B Altman, and Daniel Herschlag. Structural inference of native and partially folded RNA by high-throughput contact mapping. *Proceedings of the National Academy of Sciences*, 105(11):4144–4149, 2008.

[103] Stanley D Dunn, Lindi M Wahl, and Gregory B Gloor. Mutual information without the influence of phylogeny or entropy dramatically improves residue contact prediction. *Bioinformatics*, 24(3):333–340, 2008.

[104] Julien Y Dutheil, Fabrice Jossinet, and Eric Westhof. Base pairing constraints drive structural epistasis in ribosomal RNA sequences. *Molecular Biology and Evolution*, 27(8):1868–1876, 2010.

[105] Simona Cocco, Remi Monasson, and Martin Weigt. From principal component to direct coupling analysis of coevolution in proteins: low-eigenvalue modes are needed for structure prediction. *PLoS Computational Biology*, 9(8):e1003176, 2013.

[106] Debora S Marks, Lucy J Colwell, Robert Sheridan, Thomas A Hopf, Andrea Pagnani, Riccardo Zecchina, and Chris Sander. Protein 3D structure computed from evolutionary sequence variation. *PloS One*, 6(12):e28766, 2011.

[107] Tong Zhang. Adaptive forward-backward greedy algorithm for sparse learning with linear models. In *Advances in Neural Information Processing Systems*, pages 1921–1928, 2009.

[108] Martin Riedmiller and Heinrich Braun. A direct adaptive method for faster backpropagation learning: the RPROP algorithm. In *Neural Networks, 1993., IEEE International Conference on*, pages 586–591. IEEE, 1993.

[109] Benedikt Obermayer and Erel Levine. Inverse Ising inference with correlated samples. *New Journal of Physics*, 16(12):123017, 2014.

Methods for statistical inference on correlated data: application to genomic data

Abstract

The availability of huge amounts of data has changed the role of physics with respect to other disciplines. Within this dissertation I will explore the innovations introduced in molecular biology thanks to statistical physics approaches. In the last 20 years the size of genome databases has exponentially increased, therefore the exploitation of raw data, in the scope of extracting information, has become a major topic in statistical physics. After the success in protein structure prediction, surprising results have been finally achieved also in the related field of RNA structure characterisation. However, recent studies have revealed that, even if databases are growing, inference is often performed in the under sampling regime and new computational schemes are needed in order to overcome this intrinsic limitation of real data. This dissertation will discuss inference methods and their application to RNA structure prediction. We will discuss some heuristic approaches that have been successfully applied in the past years, even if poorly theoretically understood. The last part of the work will focus on the development of a tool for the inference of generative models, hoping it will pave the way towards novel applications.

Keywords: inference, RNA, mean-field, Potts model, generative models, regularisation, structure prediction

Résumé

La disponibilité de quantités énormes de données a changé le rôle de la physique par rapport aux autres disciplines. Dans cette thèse, je vais explorer les innovations introduites dans la biologie moléculaire grâce à des approches de physique statistique. Au cours des 20 dernières années, la taille des bases de données sur le génome a augmenté de façon exponentielle : l'exploitation des données brutes, dans le champ d'application de l'extraction d'informations, est donc devenu un sujet majeur dans la physique statistique. Après le succès dans la prédiction de la structure des protéines, des résultats étonnamment bons ont été finalement obtenus aussi pour l'ARN. Cependant, des études récentes ont révélé que, même si les bases de données sont de plus en plus grandes, l'inférence est souvent effectuée dans le régime de sous-échantillonnage et de nouveaux systèmes informatiques sont nécessaires afin de surmonter cette limitation intrinsèque des données réelles. Cette thèse va discuter des méthodes d'inférence et leur application à des prédictions de la structure de l'ARN. Nous allons comprendre certaines approches heuristiques qui ont été appliquées avec succès dans les dernières années, même si théoriquement mal comprises. La dernière partie du travail se concentrera sur le développement d'un outil pour l'inférence de modèles génératifs, en espérant qu'il ouvrira la voie à de nouvelles applications.

Mots-clés: Inférence, ARN, champ moyen, modèl de Potts, modèles génératifs, régularisation, prédiction structurelle