Enforcing ensemble averages in molecular dynamics simulations using the Maximum Entropy principle

A thesis submitted for the degree of
Philosophiæ Doctor

(October 2018)

Candidate
Andrea Cesari

Supervisor
Prof. Giovanni Bussi

Molecular and Statistical Biophysics Sector

PhD course in Physics and Chemistry of Biological Systems
Scuola Internazionale Superiore di Studi Avanzati - SISSA
Trieste

Contents

Chapter 1

Introduction

Molecular dynamics (MD) simulations in explicit solvent are nowadays a fundamental tool used to complement experimental investigations in biomolecular modeling [1]. Typical molecular dynamics simulations are usually limited to the microseconds timescale, although milliseconds timescales can be achieved with ad-hoc machines [2]. To overcome timescale limitations, over the years several enhanced sampling techniques have been developed [3–5], allowing to sample events that would require a much longer time in order to spontaneously happen. Simulations length is only one of the two factors contributing to simulations accuracy. The second important factor is the ability of the employed potential energy function, also called force field, to correctly describe the physics of the simulated system. The continuous refinement of enhanced sampling techniques, together with the constant growth of computing power, made the force field the major responsible of simulations inaccuracy. It is then necessary to always validate molecular simulations against experiments when possible. The usual procedure consists in performing a simulation and computing some observable for which an experimental value has been already measured. If the calculated and experimental values are compatible, the simulation can be trusted and other observables can be estimated in order to make genuine predictions. If the discrepancy between calculated and experimental values is significant, one is forced to make a step back and perform a new simulation with a refined force field. For instance, current force fields still exhibit visible limitations in the study of protein-protein interactions [6], in the structural characterization of protein unfolded states [7], in the simulation of the conformational dynamics of unstructured RNAs [8–10],and in the blind prediction of RNA structural motifs [10–12]. Force fields improvement is a very challenging task with many groups involved in this "undertaking". In fact, many correlated parameters should be adjusted, and modifications of one of them could easily lead to unpredictable effects on all the others. Furthermore, it is not guaranteed that

the employed potential energy functional form is sufficient to describe the real energy function of the system. As a consequence, an emerging strategy is to restrain the simulations in order to enforce the agreement with experimental data. It must be noticed that experimental knowledge is usually already encoded in the simulation of complex systems (e.g., a short simulation starting from an experimental structure will then be biased toward it). If properly combined with simulations, experiments can be a valuable alternative to quantum chemistry based force-field refinement. Moreover, it must be noticed that usually quantum chemistry calculations are performed on short fragments while experiments are usually performed on much longer molecules. Particular care should be taken when interpreting bulk experiments that measure averages over a large number of copies of the same molecule. These experiments are valuable in the characterization of dynamical molecules, where heterogeneous structures might be mixed and contribute with different weights to the experimental observation. In such cases, a proper combination of them with molecular simulations can allow to construct a high-resolution picture of molecular structure and dynamics [13–15].

This thesis describes two different methods that can be used in order to refine available molecular dynamics force-field based on experimental data. Although the difference will be more clear in next chapters, where both methods will be introduced and discussed, we propose here a very short introduction of both together with a summary table which will facilitate the reading of the thesis.

Both methods can be used either to just enforce experimental data or to perform force-field refinement based on the enforced experimental data. In both method, the employed experimental observations, are to be considered as ensemble measurements. The first method is based on the Maximum Entropy principle (first and second column of Table 1.1). Maximum Entropy is a natural choice when enforcing ensemble averages in molecular dynamics simulations. As we will see in Chap. 2, the effect of the Maximum Entropy principle is to add a bias which is a linear function of the forward model used to back-calculate the enforced constraints (e.g. Karplus relation in the case of ${}^3J$ scalar couplings). An important effect of this result is that when using Maximum Entropy in a force-field refinement context, two important limitations subsist: the first one is that it is possible to refine only variables (e.g. torsions, distances, etc) for which an experiment is available while the second one is that, if we are not allowed (or we don't want) to modify the functional form of the force-field, we are limited to use only those experimental data having a forward model compatible with the employed force-field (e.g. ${}^3J$ scalar couplings and Karplus relations). To overcame these limitations, we developed another method (third column of Table 1.1), which is not based

on the maximum entropy principle and allows the refinement using arbitrary bias functions, and experimental data (e.g. it is in principle possible to enforce a given average for the torsion X having experimental data on the torsion Y only). While the Maximum Entropy based methods can can be used both on-the-fly during MD simulations (first row in Tab. 1.1) and by reweighting already performed simulations, the second method (third column in Tab. 1.1) can only be used with a reweighting procedure. This method requires in fact to compute the variance of the enforced observables, for which is impossible to build an unbiased estimator to be used on-the-fly.

		MaxEnt	Self-Consistent MaxEnt Refinement	Refinement using arbitrary functions
On-the-fly	*Method*	Sec. 2.5.3	Sec. 3.2	See Introduction
	Application	Sec. 4	Chap. 5	X
Reweighting	*Method*	Subsec. 2.5.1	Sec. 5.2	Sec. 3.3
	Application	[16]	X	Sec. 6

Table 1.1: Summary of the introduced methods and their applications.

The results discussed in Chapter 4, 5 and 6 are largely based on the following publications:

- Cesari A., Gil-Ley A. and Bussi G. *Combining Simulations and Solution Experiments as a Paradigm for RNA Force Field Refinement*. Journal of Chemical Theory and Computation. 2016, 12 (12), 6192-6200.

- Cesari A., Reißer S. and Bussi G. *Using the Maximum Entropy Principle to Combine Simulations and Solution Experiments*. Computation. 2018, 6 (1).

- Cesari A., Bottaro S., Banáš P., Šponer J., Lindorff-Larsen K. and Bussi G. *Automated force-field parametrization guided by multi-system ensemble averages*. (In preparation)

Publications in collaborations with other groups:

- Rangan R., Bonomi M., Heller G. T., Cesari A., Bussi G and Vendruscolo M. *Determination of Structural Ensembles of Proteins: Restraining vs Reweighting*. (Submitted)

Chapter 2

The Maximum Entropy Principle

2.1 The Maximum Entropy Principle

The content of this chapter is mainly adapted from our published works [17, 18].

The maximum entropy principle was introduced for the first time in 1957 by-Jaynes [19, 20] and it was proposed as a link between thermodynamic entropy and information-theory entropy. Before this date, entropy was only used as a validation against laws of thermodynamics [19]. For the first time, Janes proposed to use the entropy as a starting point to be used in building new theories. In particular, distributions that maximize the entropy subject to some physical constraints were postulated to be useful in order to make inference on the system under study. In its original formulation, the maximum entropy principle postulates that, given a set of states describing a physical system, the distribution of such states that maximize the Shannon's entropy is the best probability distribution compatible with a set of observed data. Later, the maximum entropy principle has been extended in order to be invariant with respect to changes of coordinates and coarse-graining [21]. The new formulation, also called maximum relative entropy, has been shown to play an important role in multiscale problems [22]. In this thesis, the entropy is computed relative to a given prior distribution $P_0(\boldsymbol{q})$ which, in our applications, is the one associated to the unrefined potential energy function. For a system described by a set of continuous variables $\boldsymbol{q}$, the relative entropy is then defined as

$$S[P||P_0] = -\int d\boldsymbol{q}\ P(\boldsymbol{q}) \ln \frac{P(\boldsymbol{q})}{P_0(\boldsymbol{q})}\ . \tag{2.1}$$

This quantity should be maximized subject to constraints in order to be compatible

with observations:

$$\begin{cases} P_{ME}(\boldsymbol{q}) = \underset{P(\boldsymbol{q})}{\arg\max}\ S[P||P_0] \\ \int d\boldsymbol{q}\ s_i(\boldsymbol{q})P(\boldsymbol{q}) = \langle s_i(\boldsymbol{q})\rangle = s_i^{exp}; \quad i = 1, \dots, M \\ \int d\boldsymbol{q}\ P(\boldsymbol{q}) = 1 \end{cases} \tag{2.2}$$

$P_0(\boldsymbol{q})$ encodes the knowledge available before the experimental measurement and is thus called *prior* probability distribution. The first equation in the system 2.2 fixes the functional form of the refined probability distribution to be the one maximizing the relative entropy with respect to the prior distribution $P_0(\boldsymbol{q})$. Once the functional form has been derived, M experimental observations constrain the ensemble average of M observables $s_i(\boldsymbol{q})$ computed over the distribution $P(\boldsymbol{q})$ to be equal to s_i^{exp}. The additional constraint, in the third row, ensures that the distribution $P(\boldsymbol{q})$ is normalized. $P_{ME}(\boldsymbol{q})$ represents the best estimate for the probability distribution after the experimental constraints have been enforced and is thus called *posterior* probability distribution. The subscript ME stresses the fact that this is also the distribution that maximizes the entropy.

By noticing that the relative entropy $S[P||P_0]$ is the negative of the Kullback-Leibler divergence $D_{KL}[P||P_0]$ [23], the procedure described above can be also seen as searching the posterior distribution that is as close as possible to the prior distribution and agrees with the given experimental observations. From the information theory point of view, the Kullback-Leibler divergence measures how much information is gained by replacing $P_0(\boldsymbol{q})$ with $P(\boldsymbol{q})$. The maximization problem formulated in Eq. 2.2 can be solved using the method of Lagrangian multipliers, namely searching for the stationary points of the Lagrange function

$$\mathcal{L} = S[P||P_0] - \sum_{i=1}^{M} \lambda_i \left(\int d\boldsymbol{q}\ s_i(\boldsymbol{q})P(\boldsymbol{q}) - s_i^{exp} \right) - \mu \left(\int d\boldsymbol{q}\ (\boldsymbol{q})P(\boldsymbol{q}) - 1 \right) , \tag{2.3}$$

where λ_i and μ are suitable Lagrangian multipliers which will be computed later. The functional derivative of $\mathcal{L}$ with respect to $P(\boldsymbol{q})$ is

$$\frac{\delta \mathcal{L}}{\delta P(\boldsymbol{q})} = -\ln \frac{P(\boldsymbol{q})}{P_0(\boldsymbol{q})} - 1 - \sum_{i=1}^{M} \lambda_i s_i(\boldsymbol{q}) - \mu \ . \tag{2.4}$$

By setting $\frac{\delta \mathcal{L}}{\delta P(\boldsymbol{q})} = 0$ and neglecting the normalization factor, the posterior reads

$$P_{ME}(\boldsymbol{q}) \propto e^{-\sum_{i=1}^{M} \lambda_i s_i(\boldsymbol{q})} P_0(\boldsymbol{q}) \ . \tag{2.5}$$

It is now possible to compute the Lagrangian multipliers λ_i by enforcing the agree-

ment with the experimental data. Several approaches on how to compute the Lagrangian multipliers will be discussed in next sections. In the following, in order to have a more compact notation, we will drop the subscript from the Lagrangian multipliers and write them as a vector whenever possible.

Eq. 2.5 could thus be equivalently written as

$$P_{ME}(\boldsymbol{q}) \propto e^{-\boldsymbol{\lambda}\cdot\boldsymbol{s}(\boldsymbol{q})} P_0(\boldsymbol{q}) \ . \tag{2.6}$$

Notice that the vectors $\boldsymbol{s}$ and $\boldsymbol{\lambda}$ have dimensionality M, whereas the vector $\boldsymbol{q}$ has dimensionality equal to the number of degrees of freedom of the analyzed system.

In short, the maximum relative entropy principle gives a recipe to obtain the posterior distribution that is as close as possible to the prior distribution and agrees with some experimental observation. In the following, we will drop the word "relative" and we will refer to this principle as the maximum entropy principle.

2.2 Combining Maximum Entropy Principle and Molecular Dynamics

In order to enforce experimental data into molecular dynamics simulations, the maximum entropy principle should be properly formalized in the context of molecular dynamics simulations (MD). When combining the maximum entropy principle with MD simulations the prior knowledge $P_0(\boldsymbol{q})$ is represented by the probability distribution associated to the employed potential energy, that is typically an empirical force field in classical MD. In particular, given a potential energy described by the function $V_0(\boldsymbol{q})$, the associated probability distribution $P_0(\boldsymbol{q})$ at thermal equilibrium is the Boltzmann distribution $P_0(\boldsymbol{q}) \propto e^{-\beta V_0(\boldsymbol{q})}$, where $\beta = \frac{1}{k_B T}$, T is the system temperature, and k_B is the Boltzmann constant. According to Eq. 2.6, the posterior will be $P_{ME}(\boldsymbol{q}) \propto e^{-\boldsymbol{\lambda}\cdot\boldsymbol{s}(\boldsymbol{q})} e^{-\beta V_0(\boldsymbol{q})}$. The potential energy generating the posterior distribution can be computed by Boltzmann inversion and is expressed by:

$$V_{ME}(\boldsymbol{q}) = V_0(\boldsymbol{q}) + k_B T \boldsymbol{\lambda}\cdot\boldsymbol{s}(\boldsymbol{q}) \ .2.7 \tag{2.7}$$

The effect of the constraint on the ensemble average is that of adding a bias term to the energy that is linear in the function $\boldsymbol{s}(\boldsymbol{q})$ with prefactors, proportional to the corresponding Lagrangian multipliers, chosen in order to enforce the correct averages. This linear term is guaranteed to make the posterior distribution closer than the prior distribution to an ideal one that has the correct experimental averages [24]. It must be noticed that the effect of such a linear term, is completely

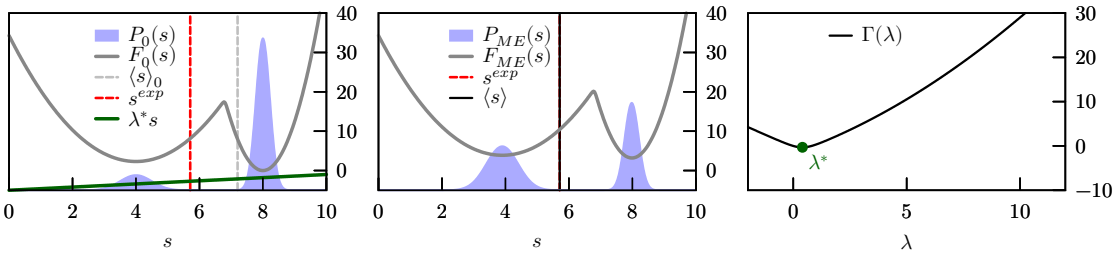


Figure 2.1: The effect of a linear correcting potential on a given reference potential. $P_0(s)$ is the marginal probability distribution of some observable $s(\boldsymbol{q})$ according to the reference potential $V_0(\boldsymbol{q})$ and $F_0(s)$ is the corresponding free-energy profile (left panel). Energy scale is reported in the vertical axis and is given in units of k_BT. Probability scales are not reported. Vertical lines represent the average value of the observable s in the prior ($\langle s \rangle_0$) and in the experiment (s^{exp}). A correcting potential linear in s (green line) shifts the relative depths of the two free-energy minima, leading to a new free energy profile $F_{ME}(s) = F_0(s) + k_BT\lambda^* s$ that corresponds to a probability distribution $P_{ME}(s)$ (central panel). Choosing λ^* equal to the value that minimizes $\Gamma(\lambda)$ (right panel) leads to an average $\langle s \rangle = s^{exp}$.

different from the ones used in constrained MD simulations, where the value of some function of the coordinates is fixed at every step (e.g., using the SHAKE algorithm [25]), or harmonic restraints, where a quadratic function of the observable is added to the potential energy function. Notice that the words constraint and restraint are usually employed when a quantity is exactly or softly enforced, respectively. Strictly speaking, in the maximum entropy context, ensemble averages $\langle \boldsymbol{s}(\boldsymbol{q}) \rangle$ are constrained whereas the corresponding functions $\boldsymbol{s}(\boldsymbol{q})$ are (linearly) restrained.

If one considers the free energy as a function of the experimental observables (also known as potential of mean force), which is defined as

$$F_0(\boldsymbol{s}') = -k_BT \ln \int d\boldsymbol{q}\ \delta(\boldsymbol{s}(\boldsymbol{q}) - \boldsymbol{s}') P_0(\boldsymbol{q}) \ , \tag{2.8}$$

the effect of the corrective potential in Eq. 2.7 is just to tilt the free-energy landscape

$$F_{ME}(\boldsymbol{s}) = F_0(\boldsymbol{s}) + k_BT\boldsymbol{\lambda} \cdot \boldsymbol{s} + C \ , \tag{2.9}$$

where C is an arbitrary constant. A schematic representation of this tilting is reported in Fig. 2.1.

Any experimental data that is the result of an ensemble measurement can be used as a constraint. Typical examples for biomolecular systems are solution nuclear-magnetic-resonance (NMR) experiments such as measures of chemical shifts [26], scalar couplings [27], or residual dipolar couplings [28], and other tech-

niques such as small-angle X-ray scattering (SAXS) [29], double electron-electron resonance (DEER) [30], and Förster resonance energy transfer [31]. In order to correctly enforce the result of an ensemble experiment into MD simulations there should exists a function, called *forward model*, mapping the atomic coordinates of the system to the measured experimental quantity. The output of the forward model can be then compared to the experimental value and restrained if necessary. For instance, in the case of ${}^3J$ scalar couplings, the forward model is given by the so-called Karplus relations [27], that are trigonometric functions of the dihedral angles. A more detailed introduction to Karplus relations can be found in Subsec. 4.1. It must be noted that the formulas used in standard forward models are often parameterized empirically, and one should take into account errors in these parameters on par with experimental errors (see Sec. 2.3).

2.2.1 A Minimization Problem

In this section we show that the optimal values of $\boldsymbol{\lambda}$ can be found either by enforcing the constraints of Eq. 2.2 or by minimization. It is in fact possible to recast the problem of finding the Lagrangian multipliers in a minimization problem, allowing the use of known minimization techniques. In particular, consider the function [32, 33]

$$\Gamma(\boldsymbol{\lambda}) = \ln\left[\int d\boldsymbol{q}\ P_0(\boldsymbol{q}) e^{-\boldsymbol{\lambda}\cdot\boldsymbol{s}(\boldsymbol{q})}\right] + \boldsymbol{\lambda}\cdot\boldsymbol{s}^{exp}. \tag{2.10}$$

Notice that the first term is the logarithm of the ratio between the two partition functions associated to the potential energy functions $V(\boldsymbol{q})$ and $V_0(\boldsymbol{q})$, that is proportional to the free-energy difference between these two potentials. The gradient of $\Gamma(\boldsymbol{\lambda})$ is

$$\frac{\partial\Gamma}{\partial\lambda_i} = s_i^{exp} - \frac{\int d\boldsymbol{q}\ P_0(\boldsymbol{q}) e^{-\boldsymbol{\lambda}\cdot\boldsymbol{s}(\boldsymbol{q})} s_i(\boldsymbol{q})}{\int d\boldsymbol{q}\ P_0(\boldsymbol{q}) e^{-\boldsymbol{\lambda}\cdot\boldsymbol{s}(\boldsymbol{q})}} = s_i^{exp} - \langle s_i(\boldsymbol{q})\rangle. \tag{2.11}$$

It is clear from Eq. 2.11 that minimizing the function Γ, namely searching for the set of λ where $\frac{\partial\Gamma}{\partial\lambda_i} = 0$, is identical to enforcing the constraints in Eq. 2.2. This means that the constraints in Eq. 2.2 can be enforced by searching for a stationary point $\boldsymbol{\lambda}^*$ of $\Gamma(\boldsymbol{\lambda})$ (see Fig. 2.1). The Hessian of $\Gamma(\boldsymbol{\lambda})$ is

$$\frac{\partial\Gamma}{\partial\lambda_i\partial\lambda_j} = \langle s_i(\boldsymbol{q}) s_j(\boldsymbol{q})\rangle - \langle s_i(\boldsymbol{q})\rangle\langle s_j(\boldsymbol{q})\rangle \tag{2.12}$$

and is thus equal to the covariance matrix of the forward models in the posterior distribution. It can be shown that, the Hessian is always positive definite except when some of the enforced observables are correlated. In such case the Hessian

is positive semi-definite [33]. The solution of Eq. 2.2 will thus correspond to a minimum of $\Gamma(\boldsymbol{\lambda})$ that can be searched for instance by a steepest descent procedure or stochastic gradient. However particular care should be taken in cases where such minimum not exist. In particular, one should pay attention to the following cases:

- When data are incompatible with the prior distribution.
- When data are mutually incompatible. As an extreme case, one can imagine two different experiments that measure the same observable and report different values.

In both cases $\Gamma(\boldsymbol{\lambda})$ will have no stationary point. Clearly, there is a continuum of possible intermediate situations where data are almost incompatible. In Sec. 2.4 we will see what happens when the maximum entropy principle is applied to model systems designed in order to highlight these difficult situations.

2.2.2 Connection with Maximum Likelihood Principle

Having defined the function $\Gamma(\boldsymbol{\lambda})$ in Eq. 2.10, it is possible to easily highlight a connection between maximum entropy and maximum likelihood principles. Let's suppose that exist a set of N_s molecular structures $\boldsymbol{q}_t$ generating an experimental value $\boldsymbol{s}^{exp} = \frac{1}{N_s}\sum_{t=1}^{N_s} \boldsymbol{s}(\boldsymbol{q}_t)$. It is possible to rewrite $e^{-N_s\Gamma(\boldsymbol{\lambda})}$ as

$$\begin{aligned} e^{-N_s\Gamma(\boldsymbol{\lambda})} &= \frac{e^{-N_s\boldsymbol{\lambda}\cdot\boldsymbol{s}^{exp}}}{\left[\int d\boldsymbol{q} P_0(\boldsymbol{q}) e^{-\boldsymbol{\lambda}\cdot\boldsymbol{s}(\boldsymbol{q})}\right]^{N_s}} = \frac{e^{-\boldsymbol{\lambda}\cdot\sum_t \boldsymbol{s}(\boldsymbol{q}_t)}}{\left[\int d\boldsymbol{q} P_0(\boldsymbol{q}) e^{-\boldsymbol{\lambda}\cdot\boldsymbol{s}(\boldsymbol{q})}\right]^{N_s}} = \\ &= \prod_{t=1}^{N_s} \frac{e^{-\boldsymbol{\lambda}\cdot\boldsymbol{s}(\boldsymbol{q}_t)}}{\int d\boldsymbol{q} P_0(\boldsymbol{q}) e^{-\boldsymbol{\lambda}\cdot\boldsymbol{s}(\boldsymbol{q})}} = \prod_{t=1}^{N_s} \frac{P(\boldsymbol{q}_t)}{P_0(\boldsymbol{q}_t)} \end{aligned} \tag{2.13}$$

The last term is the ratio between the probability of drawing the structures $\boldsymbol{q}_t$ from the posterior distribution and that of drawing the same structures from the prior distribution. Since the minimum of $\Gamma(\boldsymbol{\lambda})$ corresponds to the maximum of $e^{-N_s\Gamma(\boldsymbol{\lambda})}$, the distribution that maximizes the entropy under experimental constraints is identical to the one that, among an exponential family of distributions, maximizes the likelihood of a set of structures with average value of the observables $\boldsymbol{s}$ equal to the experimental value [34, 35]. This equivalence can be considered as an added justification for the maximum entropy principle [34]. If the notion of selecting a posterior $P(\boldsymbol{q})$ that maximizes the entropy is not compelling enough, one can consider that this same posterior is, among the distributions with the exponential form of Eq. 2.5, the one that maximizes the likelihood of being compatible with the experimental sample.

2.2.3 Equivalence to the Replica Approach

Maximum Entropy is not the only possibility to enforce ensemble averages in molecular dynamics simulations. Among the different methods developed during the years, restrained ensemble [36–38] should be mentioned for its equivalence to maximum entropy in some particular cases. In restrained ensembles N_{rep} identical copies (replicas) of the system are simulated in parallel, each of which having its own atomic coordinates. The set of replicas is then used to mimic the ensemble of structure. The agreement with the M experimental data is then enforced by adding a harmonic restraint for each observable, centered on the experimental reference and acting on the average over all the simulated replicas. This results in a restraining potential with the following form:

$$V_{RE}\left(\boldsymbol{q}_1, \boldsymbol{q}_2, \ldots, \boldsymbol{q}_{N_{rep}}\right) = \sum_{i=1}^{N_{rep}} V_0\left(\boldsymbol{q}_i\right) + \frac{k}{2}\sum_{j=1}^{M}\left(\frac{1}{N_{rep}}\sum_{i=1}^{N_{rep}} s_j(\boldsymbol{q}_i) - s_j^{exp}\right)^2 , \quad (2.14)$$

where k is a suitably chosen force constant. It has been shown [33, 39, 40] that this method produces the same ensemble as the maximum entropy approach in the limit of large number of replicas ($N_{rep} \to \infty$). Although this can be demonstrated in different ways[40], the simplest explanation proposed by Chodera[33] is the following. The potential in Eq. 2.14 results in the same force $-\frac{k}{N_{rep}}\left(\frac{1}{N_{rep}}\sum_{i=1}^{N_{rep}} s_j(\boldsymbol{q}_i) - s_j^{exp}\right)$ applied to the observable $s_j(\boldsymbol{q})$ in each replica. As the number of replicas grows, the fluctuations of the average decrease and the applied force becomes constant in time, so that the explored distribution will have the same form as Eq. 2.5 with $\boldsymbol{\lambda} = \frac{k}{N_{rep}k_BT}\left(\frac{1}{N_{rep}}\sum_{i=1}^{N_{rep}} \boldsymbol{s}(\boldsymbol{q}_i) - \boldsymbol{s}^{exp}\right)$. If k is chosen large enough, the average between the replicas will be forced to be equal to the experimental one. In order to enforce the desired average, it has been shown that is also necessary that k grows faster than N_{rep} [40]. In practical implementations, k should be finite in order to avoid infinite forces. A direct calculation of the entropy-loss due to the choice of a finite N_{rep} has been proposed to be an useful tool in the search for the correct number of replicas [41].

2.3 Modeling Experimental Errors

The maximum entropy method can be modified in order to account for uncertainties in experimental data. This step is fundamental in order to reduce over-fitting. In this section we will briefly consider how the error can be modeled, following what we introduced in Ref. [17]. In our approach, errors are modeled modifying

the experimental constraints introduced in Eq. 2.2 by introducing an auxiliary variables ϵ_i for each data point. This auxiliary variable represent the discrepancy, or equivalently the residual, between the experimental and the simulated value. The constraints in Eq. 2.2 are then modified as follows:

$$\langle (\boldsymbol{s}(\boldsymbol{q}) + \boldsymbol{\epsilon}) \rangle = \boldsymbol{s}^{exp} \ . \tag{2.15}$$

The auxiliary variable $\boldsymbol{\epsilon}$ is a vector with dimensionality equal to the number of constraints and models all the possible sources of error, including inaccuracies of the forward models (introduced in Sec. 2.2) as well as experimental uncertainties. The desired functional form used to model errors, can be chosen by selecting a proper prior distribution function for the variable $\boldsymbol{\epsilon}$. A common choice is represented by a Gaussian prior with 0 mean and fixed standard deviation σ_i for the i^{th} observable

$$P_0(\boldsymbol{\epsilon}) \propto \prod_{i=1}^{M} \exp\left(-\frac{\epsilon_i^2}{2\sigma_i^2}\right) \ . \tag{2.16}$$

The value of σ_i corresponds to the level of confidence in the i^{th} data point. A value of $\sigma_i = \infty$ implies to completely trust the underlying prior distribution (force-field) and totally discard informations in the data. On the other hand, a value of $\sigma_i = 0$ means having complete confidence in the data, that will be fitted as best as possible introducing all the necessary modifications in the force-field distribution. Notice that the independence of $\boldsymbol{q}$ and $\boldsymbol{\epsilon}$ implies that Eq. 2.15 can be written as:

$$\langle \boldsymbol{s}(\boldsymbol{q}) \rangle = \boldsymbol{s}^{exp} - \langle \boldsymbol{\epsilon} \rangle \tag{2.17}$$

where $\langle \boldsymbol{\epsilon} \rangle$ is computed in the posterior distribution $P(\boldsymbol{\epsilon}) \propto P_0(\boldsymbol{\epsilon}) e^{-\boldsymbol{\lambda}\cdot\boldsymbol{\epsilon}}$. The task of incorporating the experimental error in the maximum entropy approach is then translated in the easy operations of enforcing a different experimental value, corresponding to the one in Eq. 2.17. Notice that, since the value of $\langle \boldsymbol{\epsilon} \rangle$ only depends on its prior distribution $P_0(\boldsymbol{\epsilon})$ and on $\boldsymbol{\lambda}$ it can be computed analytically in some particular cases. For a Gaussian prior with standard deviation σ_i (Eq. 2.16) we have:

$$\langle \epsilon_i \rangle = \frac{\int d\epsilon e^{\left(-\frac{\epsilon_i^2}{2\sigma_i^2}\right)} \epsilon_i e^{\left(-\sum_j \epsilon_j \lambda_j\right)}}{\int d\epsilon e^{\left(-\frac{\epsilon_i^2}{2\sigma_i^2}\right)} e^{\left(-\sum_j \epsilon_j \lambda_j\right)}} = -\lambda_i \sigma_i^2. \tag{2.18}$$

Thus, as λ grows in magnitude, a larger discrepancy between simulation and experiment will be accepted. In addition, it can be seen that applying the same

constraint *twice* is exactly equivalent to applying a constraint with a σ_i^2 reduced by a factor two. This is consistent with the fact that the confidence in the repeated data point is increased. This relation also highlight that a Gaussian prior is not a good choice if outliers are present in the data. In fact, since the value of λ_i is unbound, in case of outliers and hence big values of $\langle \epsilon_i \rangle$, the value of λ_i will growth as much as necessary to fit the outlier.

Other prior functions are also possible in order to better account for outliers and to deal with cases where the standard deviation of the residual is not known a priori. One might consider the variance of the i^{th} residual $\sigma_{0,i}^2$ as a variable sampled from a given prior distribution:

$$P_0\left(\boldsymbol{\epsilon}\right) = \prod_{i=1}^{M} \int_0^\infty d\sigma_{0,i}^2 P_0(\sigma_{0,i}^2) \frac{1}{\sqrt{2\pi\sigma_{0,i}^2}} \exp\left(-\frac{\epsilon_i^2}{2\sigma_{0,i}^2}\right) . \tag{2.19}$$

A flexible functional form for $P_0(\sigma_{0,i}^2)$ can be obtained using the following Gamma distribution

$$P_0(\sigma_{0,i}^2) \propto (\sigma_{0,i}^2)^{\kappa-1} \exp\left(-\frac{\kappa\sigma_{0,i}^2}{\sigma_i^2}\right) . \tag{2.20}$$

In the above equation σ_i^2 is the mean parameter of the Gamma function and must be interpreted as the typical expected variance of the error on the i^{th} data. κ, which must satisfy $\kappa > 0$, is the shape parameter of the Gamma distribution and expresses how much the distribution is peaked around σ_i^2. In practice, it controls how much the optimization is tolerant to large discrepancies between the experimental data and the enforced average. Notice that in Ref. [17] a different convention was used with a parameter $\alpha = 2\kappa - 1$. By setting $\kappa = \infty$ a Gaussian prior on $\boldsymbol{\epsilon}$ will be recovered. Smaller values of κ will lead to a prior distribution on $\boldsymbol{\epsilon}$ with "fatter" tails and thus able to accommodate larger differences between experiment and simulation. For instance, the case $\kappa = 1$ leads to a Laplace prior $P_0(\boldsymbol{\epsilon}) \propto \prod_i \exp\left(-\frac{\sqrt{2}|\epsilon|}{\sigma_i}\right)$. After proper manipulation, the resulting expectation value of $\langle \boldsymbol{\epsilon} \rangle$ can be shown to be

$$\langle \epsilon_i \rangle = -\frac{\lambda_i \sigma_i^2}{1 - \frac{\lambda_i^2 \sigma_i^2}{2\kappa}} . \tag{2.21}$$

In this case, it can be seen that applying the same constraint twice is exactly equivalent to applying a constraint with a σ_i^2 reduced by a factor two and a κ multiplied by a factor two. A detailed comparisons of both the prior $P_0\left(\sigma_0\right)$ and $P_0\left(\epsilon\right)$ as function of k, is provided in Appendix A.

In terms of the minimization problem of Section 2.2.1, modeling experimental

errors as discussed here is equivalent to adding a contribution Γ_{err} to Eq. 2.10:

$$\Gamma(\boldsymbol{\lambda}) = \ln \int d\boldsymbol{q}\ P_0(\boldsymbol{q}) e^{-\boldsymbol{\lambda}\cdot\boldsymbol{s}(\boldsymbol{q})} + \boldsymbol{\lambda}\cdot\boldsymbol{s}^{exp} + \Gamma_{err}(\boldsymbol{\lambda})\ . \tag{2.22}$$

For a Gaussian noise with preassigned variance (Eq. 2.16) the additional term is

$$\Gamma_{err}(\boldsymbol{\lambda}) = \frac{1}{2}\sum_{i=1}^{M} \lambda_i^2 \sigma_i^2\ . \tag{2.23}$$

For a prior on the error in the form of Eqs. 2.19 and 2.20 one obtains

$$\Gamma_{err}(\boldsymbol{\lambda}) = -\kappa \sum_{i=1}^{M} \ln\left(1 - \frac{\lambda_i^2\sigma_i^2}{2\kappa}\right)\ . \tag{2.24}$$

In the limit of large κ, Eq. 2.24 is equivalent to Eq. 2.23. If the data points are expected to all have the same error σ_0, unknown but with a typical value σ, Eq. 2.24 should be modified to $\Gamma_{err}(\boldsymbol{\lambda}) = -\kappa \ln\left(1 - \frac{|\boldsymbol{\lambda}|^2\sigma^2}{2\kappa}\right)$.

Equation 2.24 shows that by construction the Lagrangian multiplier λ_i will be limited in the range $(-\frac{\sqrt{2\kappa}}{\sigma_i}, +\frac{\sqrt{2\kappa}}{\sigma_i})$. The effect of using a prior with $\kappa < \infty$ is thus that of restricting the range of allowed λ in order to avoid too large modifications of the prior distribution. In practice, values of λ chosen outside these boundaries would lead to a posterior distribution $P(\boldsymbol{\epsilon}) \propto P_0(\boldsymbol{\epsilon}) e^{-\boldsymbol{\lambda}\cdot\boldsymbol{\epsilon}}$ that cannot be normalized. A plot, highlighting the dependence of λ_i from the residual $\langle\epsilon_i\rangle$ can be found in Appendix A Fig. A.2.

Except for trivial cases (e.g., for Gaussian noise with $\sigma = 0$), the contribution originating from error modeling has positive definite Hessian and as such it makes $\Gamma(\lambda)$ a strongly convex function. Thus, a suitable error treatment can make the minimization process numerically easier.

It is worth mentioning that a very similar formalism can be used to include not only errors but more generally any quantity that influences the experimental measurement but cannot be directly obtained from the simulated structure. For instance, in the case of residual dipolar couplings [28], the orientation of the considered molecule with respect to the external field is often unknown. The orientation of the field can be then used as an additional vectorial variable to be sampled with a Monte Carlo procedure, and suitable Lagrangian multipliers can be obtained in order to enforce the agreement with experiments [42]. Notice that in this case the orientation contributes to the ensemble average in a non additive manner so that Eq. 2.17 cannot be used. Interestingly, thanks to the equivalence between multi-replica simulations and maximum entropy restraints, equivalent results can be obtained using the tensor-free method of Ref. [43].

Finally, we note that several works introduced error treatment using a Bayesian

framework [44–47]. Bayesian ensemble refinement [45] introduces an additional parameter (θ) that takes into account the confidence in the prior distribution. This parameter enters as a global scaling factor in the errors σ_i for each data point. Thus, the errors σ_i discussed above can be used to modulate both our confidence in experimental data and our confidence in the original force field. The equivalence between the error treatment of Ref. [45] and the one reported here is further discussed in Appendix A.2

2.4 Exact results on model systems

In this section we illustrate the effects of adding restraints using the maximum entropy principle on simple model systems. We build a simple system in which the prior function consists in a sum of N_G Gaussians with center $\boldsymbol{s}_\alpha$ and covariance matrix A_α, where $\alpha = 1, \ldots, N_G$:

$$P_0(\boldsymbol{s}) = \sum_{\alpha=1}^{N_G} \frac{w_\alpha}{\sqrt{2\pi \det A_\alpha}} e^{-\frac{(\boldsymbol{s}-\boldsymbol{s}_\alpha)A_\alpha^{-1}(\boldsymbol{s}-\boldsymbol{s}_\alpha)}{2}} \ . \tag{2.25}$$

The coefficients w_α provide the weights of each Gaussian and are normalized ($\sum_\alpha w_\alpha = 1$). We here assume that the restraints are applied on the variable $\boldsymbol{s}$. For a general system, one should first perform a dimensional reduction in order to obtain the marginal prior probability $P_0(\boldsymbol{s})$. By constraining the ensemble averages of the variable $\boldsymbol{s}$ to an experimental value $\boldsymbol{s}^{exp}$ the posterior becomes:

$$P_{ME}(\boldsymbol{s}) = \frac{e^{-\boldsymbol{\lambda}\cdot\boldsymbol{s}}}{Z(\boldsymbol{\lambda})} \sum_\alpha \frac{w_\alpha}{\sqrt{2\pi \det A_\alpha}} e^{-\frac{(\boldsymbol{s}-\boldsymbol{s}_\alpha)A_\alpha^{-1}(\boldsymbol{s}-\boldsymbol{s}_\alpha)}{2}} \ . \tag{2.26}$$

With proper algebra it is possible to compute explicitly the normalization factor $Z(\boldsymbol{\lambda}) = \sum_\alpha w_\alpha e^{\frac{\boldsymbol{\lambda}A_\alpha\boldsymbol{\lambda}}{2} - \boldsymbol{\lambda}\cdot\boldsymbol{s}_\alpha}$. The function $\Gamma(\boldsymbol{\lambda})$ to be minimized is thus equal to:

$$\Gamma(\boldsymbol{\lambda}) = \ln\left(\sum_\alpha w_\alpha e^{\frac{\boldsymbol{\lambda}A_\alpha\boldsymbol{\lambda}}{2} - \boldsymbol{\lambda}\cdot\boldsymbol{s}_\alpha}\right) + \boldsymbol{\lambda}\cdot\boldsymbol{s}^{exp} + \Gamma_{err}(\boldsymbol{\lambda}) \tag{2.27}$$

and the average value of $\boldsymbol{s}$ in the posterior is

$$\langle \boldsymbol{s} \rangle = \frac{\sum_\alpha w_\alpha e^{\frac{\boldsymbol{\lambda}A_\alpha\boldsymbol{\lambda}}{2} - \boldsymbol{\lambda}\cdot\boldsymbol{s}_\alpha} (\boldsymbol{s}_\alpha - A_\alpha\boldsymbol{\lambda})}{\sum_\alpha w_\alpha e^{\frac{\boldsymbol{\lambda}A_\alpha\boldsymbol{\lambda}}{2} - \boldsymbol{\lambda}\cdot\boldsymbol{s}_\alpha}} \ . \tag{2.28}$$

Notice that although the system is quite simple, we could not find a close formula to compute $\boldsymbol{\lambda}^*$ given $\boldsymbol{s}^{exp}$ and Γ_{err}. However, the solution can be found numerically with the gradient descent procedure which is discussed later in Section 2.5 (see Eq. 2.29).

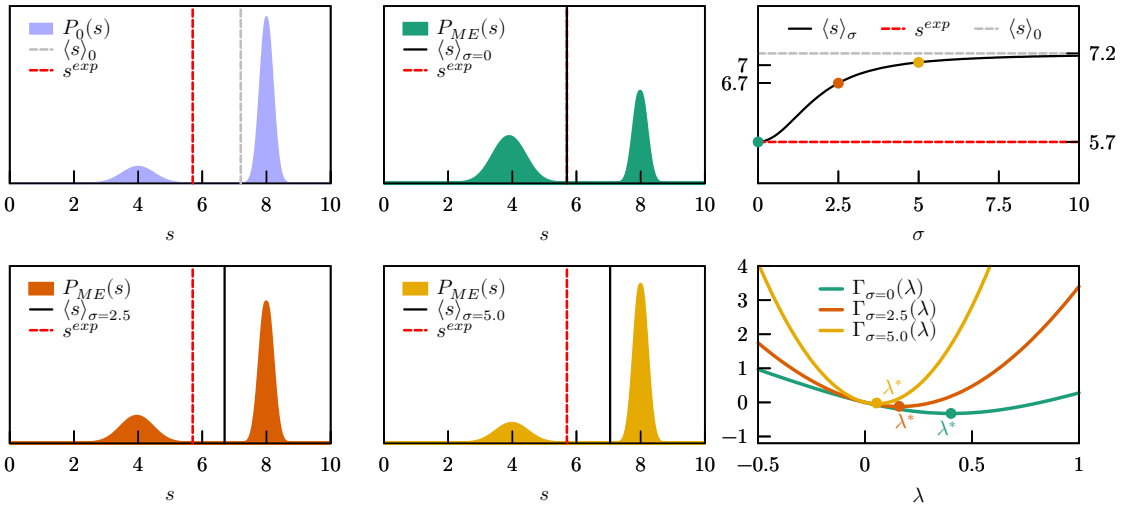


Figure 2.2: Effect of modeling error with a Gaussian probability distribution with different standard deviations σ on the posterior distribution $P_{ME}(s)$. The experimental value is here set to $s^{exp} = 5.7$, which is compatible with the prior distribution. Left and middle column: prior $P_0(s)$ and posterior $P_{ME}(s)$ with $\sigma = 0,\ 2.5,\ 5.0$. Right column: ensemble average $\langle s\rangle$ plotted as a function of σ and $\Gamma(\lambda)$ plotted for different values of σ. λ^* denotes that value of λ that minimizes $\Gamma(\boldsymbol{\lambda})$.

2.4.1 Consistency between Prior Distribution and Experimental Data

As mentioned in Sec. 2.2.1 there are particular cases in which the function $\Gamma(\boldsymbol{\lambda})$ is not strongly convex and hence the minimization could be difficult. Among the great number of cases in which this problem could happen, we selected two general cases of interest. We here show the case in which the enforced experimental data are not compatible with the prior distribution.

To do so, we consider a one dimensional model with a prior expressed as a sum of two Gaussians, one centered in $s_A = 4$ with standard deviation $\sigma_A = 0.5$ and one centered in $s_B = 8$ with standard deviation $\sigma_B = 0.2$. The weights of the two Gaussians are $w_A = 0.2$ and $w_B = 0.8$, respectively. The prior distribution is thus $P_0(s) \propto \frac{w_A}{\sigma_A} e^{-(s-s_A)^2/2\sigma_A^2} + \frac{w_B}{\sigma_B} e^{-(s-s_B)^2/2\sigma_B^2}$, has an average value $\langle s\rangle_0 = 7.2$, and is represented in Fig. 2.2, left column top panel.

We first enforce a value $s^{exp} = 5.7$, which is compatible with the prior probability. If we are absolutely sure about our experimental value and set $\sigma = 0$, the λ^* which minimizes $\Gamma(\lambda)$ is $\lambda^* \approx 0.4$ (Fig. 2.2 right column, bottom panel). In case values of $\sigma \neq 0$ are used, the $\Gamma(\boldsymbol{\lambda})$ function becomes more convex and the optimal value λ^* is decreased. As a result, the average s in the posterior distribution is approaching its value in the prior as the value of σ becomes larger and larger. The

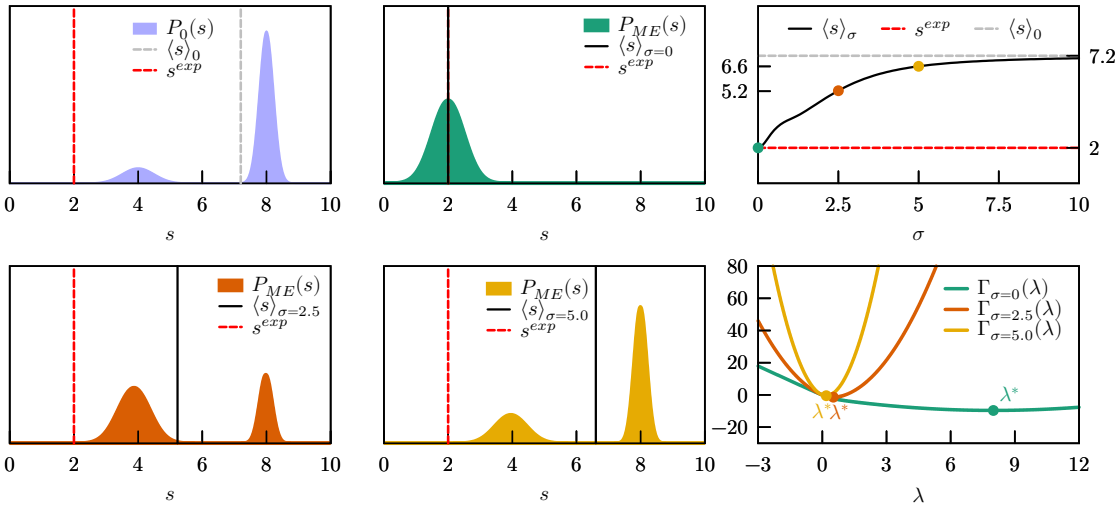


Figure 2.3: Same as Fig. 2.2, but the experimental value is here set to $s^{exp} = 2$, which is not compatible with the prior distribution.

evolution of the ensemble average $\langle s\rangle_\sigma$ varying σ between 0 and 10, with respect to the initial $\langle s\rangle_0$ and the experimental s^{exp}, is shown in Fig. 2.2, right column top panel. In all these cases the posterior distributions remain bimodal and the main effect of the restraint is to change the relative population of the two peaks (Fig. 2.2, left and middle columns). Notice that in case simple harmonic restraints were applied, the posterior distribution would have been a Gaussian distribution centered around the enforced s^{exp}.

We then enforce an average value $s^{exp} = 2$, which is far outside the original probability distribution (see Figure 2.3). It is thus very unlikely that the value of s^{exp} has been generated by a distribution similar to the prior. The enforced value of s^{exp} can be then assumed to be incompatible with the prior distribution. If we are absolutely sure about our experimental value and set $\sigma = 0$, the λ^* which minimizes $\Gamma(\lambda)$ is very large, $\lambda^* \approx 8$ (Fig. 2.3 right column, bottom panel). Assuming zero error on the experimental value is equivalent to having poor confidence in the probability distribution sampled by the force field, and leads in fact to a $P_{ME}(s)$ completely different from $P_0(s)$. The two peaks in $P_0(s)$ are replaced by a single peak centered around the experimental value, which is exactly met by the ensemble average ($\langle s\rangle_{\sigma=0} = s^{exp} = 2$; Fig. 2.3 middle column top panel). If we instead have more confidence in the distribution sampled by the force field and assume that there might be an error in our experimental value, by setting $\sigma = 2.5$ we obtain a value of λ^* which is more than one order of magnitude lower ($\lambda^* \approx 0.52$) than in the case with $\sigma = 0$. The two peaks in $P_0(s)$ are only slightly shifted towards lower s, while their relative populations are shifted in favor of the peak centered around 4 (Fig. 2.3, left column bottom panel). In case we have very high confidence in

the force field and very low confidence in the experimental value and set $\sigma = 5.0$, the correction becomes very small ($\lambda^* \approx 0.18$) and the new ensemble average $\langle s \rangle_{\sigma=5.0} \approx 6.6$, very close to the initial $\langle s \rangle_0 = 7.2$ (Fig. 2.3, middle column bottom panel). The evolution of the ensemble average $\langle s \rangle_\sigma$ with σ values between zero and ten, with respect to the initial $\langle s \rangle_0$ and the experimental s^{exp}, is shown in Fig. 2.3, right column top panel.

In conclusion, when data that are not consistent with the prior distribution are enforced, the posterior distribution could be severely distorted. Clearly, this could happen either because the prior is completely wrong or because the experimental values are affected by errors. By including a suitable error model in the maximum entropy procedure, such as the one introduced in Sec.2.3, it is possible to easily interpolate between the two extremes in which we completely trust the force field or the experimental data.

2.4.2 Consistency between Data Points

A second case in which the function $\Gamma(\boldsymbol{\lambda})$ is not convex is when trying to enforce data that are inconsistent among each other. A simple example is trying to enforce the same quantity to have two different experimental references. To show an example of inconsistent data points we consider a two dimensional model with a prior expressed as a sum of two Gaussians centered in $\boldsymbol{s}_A = (0, 0)$ and $\boldsymbol{s}_B = (3, 3)$ with identical standard deviations $\sigma_A = \sigma_B = 0.2$ and weights $w_A = w_B = 0.5$. The prior distribution is represented in Fig. 2.4.

This model is particularly instructive since, by construction, the two components of $\boldsymbol{s}$ are highly correlated and is hence possible to see what happens when inconsistent data are enforced. To this aim we study the two scenarios (i.e., consistent and inconsistent data) using different error models (no error model, Gaussian prior with $\sigma = 1$, and Laplace prior with $\sigma = 1$), for a total of six combinations. In the *consistent* case we enforce $\boldsymbol{s}^{exp} = (1, 1)$, whereas in the *inconsistent* one we enforce $\boldsymbol{s}^{exp} = (1, 0)$. Figure 2.4 reports the posterior distributions obtained in all these cases.

When consistent data are enforced the posterior distribution is very similar to the prior distribution, the only difference being a modulation in the weights of the two peaks needed to enforce the constraints. The optimal value $\boldsymbol{\lambda}^*$, marked with a $\star$ in Figure 2.4, does not depend significantly on the adopted error model. The main difference between including or not including error models can be seen in the form of the $\Gamma(\boldsymbol{\lambda})$ function. When errors are not included, $\Gamma(\boldsymbol{\lambda})$ is almost flat in a given direction, indicating that one of the eigenvalues of its Hessian is very small. On the contrary, when error modeling is included, the $\Gamma(\boldsymbol{\lambda})$ function

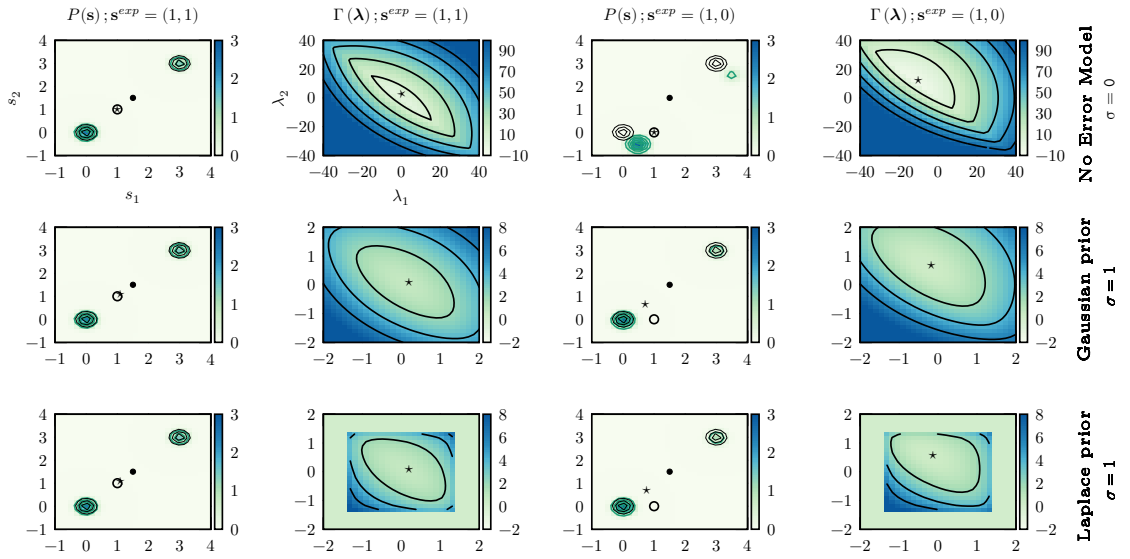


Figure 2.4: Effect of different prior distributions for the error model in a two-dimensional system. In the first (last) two columns, compatible (incompatible) data are enforced. In first and third column, prior distributions are represented as black contour lines and posterior distributions are shown in color scale. A black dot and a $\star$ are used to indicate the average values of $\boldsymbol{s}$ in the prior and posterior distributions respectively while an empty circle is used to indicate the target $\boldsymbol{s}^{exp}$. In second and fourth columns, the function $\Gamma(\boldsymbol{\lambda})$ is shown, and its minimum $\boldsymbol{\lambda}^*$ is indicated with a $\star$. The first row reports results where errors are not modeled, whereas the second and third rows report results obtained using Gaussian and Laplace prior for the error model respectively. Notice that the a different scale is used to represent $\Gamma(\boldsymbol{\lambda})$ in the first row. For the Laplace prior, the region of $\boldsymbol{\lambda}$ where $\Gamma(\boldsymbol{\lambda})$ is undefined is marked as light green.

becomes clearly convex in all directions. In practical applications, the numerical minimization of $\Gamma(\boldsymbol{\lambda})$ would be more efficient.

When enforcing inconsistent data without taking into account experimental error, the behavior is significantly different. Indeed, the only manner to enforce data where the value of the two components of $\boldsymbol{s}$ are different is to significantly displace the two peaks. On the contrary, the distortion is significantly alleviated when taking into account experimental errors. Obviously, in this case the experimental value is not exactly enforced and, with both Gaussian and Laplace prior, we obtain $\langle \boldsymbol{s} \rangle \approx (0.7, 0.7)$.

By observing $\Gamma(\boldsymbol{\lambda})$ it can be seen that the main effect of using a Laplace prior instead of a Gaussian prior for the error is that the range of suitable values for λ is limited. This allows one to decrease the effect of particularly wrong data points on the posterior distribution.

In conclusion, when enforcing data that are not consistent among themselves the posterior distribution could be severely distorted. Inconsistency between data

could either be explicit (as in the case where constraints with different reference values are enforced on the same observable) or more subtle. In the reported example, the only way to know that the two components of $\boldsymbol{s}$ should have similar values is to observe their distribution according to the original force field. In the case of complex molecular systems and of observables that depend non-linearly on the atomic coordinates, it is very difficult to detect inconsistencies between data points a priori. By properly modeling experimental error it is possible to greatly alleviate the effect of these inconsistencies on the resulting posterior. Clearly, if the quality of the prior is very poor, correct data points might artificially appear as inconsistent.

2.5 Strategies for the Optimization of Lagrangian Multipliers

In order to find the optimal values of Lagrangian multipliers, one has to minimize the function $\Gamma(\boldsymbol{\lambda})$. The simplest possible strategy is gradient descent (GD), that is an iterative algorithm in which Lagrangian multipliers are adjusted by following the opposite direction of the gradient of $\Gamma(\boldsymbol{\lambda})$. By using the gradient in Eq. 2.11 and the constraint in Eq. 2.17, the value of λ at the iteration $k+1$ can be obtained from the value of λ at the iteration k as:

$$\lambda_i^{(k+1)} = \lambda_i^{(k)} - \eta_i \frac{\partial \Gamma}{\partial \lambda_i} = \lambda_i^{(k)} - \eta_i \left(s_i^{exp} - \langle s_i(\boldsymbol{q}) \rangle_{\boldsymbol{\lambda}^{(k)}} - \langle \epsilon_i \rangle_{\boldsymbol{\lambda}^{(k)}} \right) \ , \tag{2.29}$$

where η represents the step size or *learning rate* at each iteration and might be different for different observables. Notice that the average $\langle s_i(\boldsymbol{q}) \rangle$ should be computed using the Lagrangian multipliers at the k^{th} iteration $\boldsymbol{\lambda}^{(k)}$. In order to compute this average it is in principle necessary to sum over all the possible values of $\boldsymbol{q}$. This is possible for the simple model systems discussed in Section 2.4, where integrals can be done analytically. However, for a real molecular system, summing over all the conformations would be virtually impossible. Below we discuss some possible alternatives.

Notice that although this thesis focuses mainly on equality restraints, in the form of Eq. 2.2, the methods discussed here can be applied to inequality restraints as well as discussed in Sec. 2.5.3.

2.5.1 Ensemble Reweighting

If a trajectory has been already produced using the prior force field $V_0(\boldsymbol{q})$, samples from this trajectory might be used to compute the function $\Gamma(\boldsymbol{\lambda})$. In particular, the integral in Eq. 2.10 can be replaced by an average over N_s snapshots $\boldsymbol{q}_t$ sampled from $P_0(\boldsymbol{q})$:

$$\tilde{\Gamma}(\boldsymbol{\lambda}) = \ln\left(\frac{1}{N_s}\sum_{t=1}^{N_s} e^{-\boldsymbol{\lambda}\cdot\boldsymbol{s}(\boldsymbol{q}_t)}\right) + \boldsymbol{\lambda}\cdot\boldsymbol{s}^{exp} + \Gamma_{err}(\boldsymbol{\lambda}) \ . \tag{2.30}$$

A gradient descent on $\tilde{\Gamma}$ results in a procedure equivalent to Eq. 2.29 where the ensemble average $\langle \boldsymbol{s}(\boldsymbol{q})\rangle_{\boldsymbol{\lambda}^{(k)}}$ is computed as a weighted average on the available frames:

$$\lambda_i^{(k+1)} = \lambda_i^{(k)} - \eta_i \frac{\partial\tilde{\Gamma}}{\partial\lambda_i} = \lambda_i^{(k)} - \eta_i\left(s_i^{exp} - \frac{\sum_{t=1}^{N_s} \boldsymbol{s}(\boldsymbol{q}_t) e^{-\boldsymbol{\lambda}^{(k)}\cdot\boldsymbol{s}(\boldsymbol{q}_t)}}{\sum_{t=1}^{N_s} e^{-\boldsymbol{\lambda}^{(k)}\cdot\boldsymbol{s}(\boldsymbol{q}_t)}} - \langle\epsilon_i\rangle_{\boldsymbol{\lambda}^{(k)}}\right) \ . \tag{2.31}$$

It is also possible to use conjugated gradient or more advanced minimization methods. Once the multipliers $\boldsymbol{\lambda}^*$ have been found one can compute any other expectation value by just assigning a weight $w_t = e^{-\boldsymbol{\lambda}^*\cdot\boldsymbol{s}(\boldsymbol{q}_t)} / \sum_{t'=1}^{N_s} e^{-\boldsymbol{\lambda}^*\cdot\boldsymbol{s}(\boldsymbol{q}_{t'})}$ to the snapshot $\boldsymbol{q}_t$.

A reweighting procedure related to this one is at the core of the ensemble-reweighting-of-SAXS method [48], that has been used to construct structural ensembles of proteins compatible with SAXS data [48, 49]. Similar reweighting procedures were used to enforce average data on a variety of systems [44, 45, 47, 50–54]. These procedures are very practical since they allow incorporating experimental constraints a posteriori without the need to repeat the MD simulation. For instance, in Ref. [53] it was possible to test different combinations of experimental restraints in order to evaluate their consistency. However, reweighting approaches must be used with care since they are effective only when the posterior and the prior distributions are similar enough [55]. In case this is not true, the reweighted ensembles will be dominated by a few snapshots with very high weight, leading to a large statistical error. The effective number of snapshots with a significant weight can be estimated using the Kish's effective sample size [56]. A deep analysis of reweighting performance, together with a critical comparison between reweighting and restraining methods (both MaxEnt and Replica averaging) on non-trivial systems, has been the subject of a recent collaboration with the group of Prof. Michele Vendruscolo. The results of this collaboration are in publication in a work by Ramya Rangan [16].

2.5.2 Iterative Simulations

In order to decrease the statistical error, it is convenient to use the modified potential $V(\boldsymbol{q}) = V_0(\boldsymbol{q}) + k_B T \boldsymbol{\lambda} \cdot \boldsymbol{s}(\boldsymbol{q})$ to run a new simulation, in an iterative manner. For instance, in the iterative Boltzmann method, pairwise potentials are modified and new simulations are performed until the radial distribution function of the simulated particles does match the desired one [57].

It is also possible to make a full optimization of $\Gamma(\boldsymbol{\lambda})$ using a reweighting procedure like the one illustrated before in Sec. 2.5.1 at each iteration. Particular care should be taken if the posterior and the prior distributions are expected to be quite different from each other. In this case a single iteration reweighting would lead to a very low statistical efficiency. One would first perform a simulation using the original force field and, based on samples taken from that simulation, find the optimal $\boldsymbol{\lambda}$ with a gradient descent procedure. Only at that point a new simulation would be required using a modified potential that includes the extra $k_B T \boldsymbol{\lambda} \cdot \boldsymbol{s}(\boldsymbol{q})$ contribution. This whole procedure should be then repeated until the value of $\boldsymbol{\lambda}$ stops changing. This approach was used in Ref. [58] in order to adjust a force field to reproduce ensembles of disordered proteins. The same scheme was later used in a maximum entropy context to enforce average contact maps in the simulation of chromosomes [59, 60]. A similar iterative approach was used in Refs. [61, 62].

In principle, iterative procedures are supposed to converge to the correct values of $\boldsymbol{\lambda}$. However, this happens only if the simulations used at each iteration are statistically converged. For systems that exhibit multiple metastable states and are thus difficult to sample it might be difficult to tune the length of each iteration so as to obtain good estimators of the $\Gamma(\boldsymbol{\lambda})$ gradients.

2.5.3 On-the-fly Optimization with Stochastic Gradient Descent

Instead of trying to converge the calculation of the gradient at each individual iteration and, only at that point, modify the potential in order to run a new simulation, one might try to change the potential on-the-fly so as to force the system to sample the posterior distribution:

$$V(\boldsymbol{q}, t) = V_0(\boldsymbol{q}) + k_B T \boldsymbol{\lambda}(t) \cdot \boldsymbol{s}(\boldsymbol{q}) \ . \tag{2.32}$$

The simplest choice in order to minimize the $\Gamma(\boldsymbol{\lambda})$ function is to use a stochastic gradient descent (SGD) procedure, where an unbiased estimator of the gradient is used to update $\boldsymbol{\lambda}$. In particular, the instantaneous value of the forward model computed at time t, $\boldsymbol{s}(\boldsymbol{q}(t))$, can be used to this aim. The update rule for $\boldsymbol{\lambda}$ can

thus be rewritten as a differential equation:

$$\dot{\lambda}_i(t) = -\eta_i(t)\left(s_i^{exp} - s_i(\boldsymbol{q}(t)) - \langle\epsilon_i\rangle_{\boldsymbol{\lambda}_i(t)}\right). \tag{2.33}$$

Notice that now the learning rate η depends on the simulation time t. This choice is motivated by the fact that approximating the true gradient with its unbiased estimator introduces a noise into its estimate. In order to decrease the effect of such noise, a common choice when using SGD is to reduce the learning rate as the minimization (learning) process progresses with a typical schedule $\eta(t) \propto 1/t$ for large times. In our work [17] we adopted a learning rate from the class *search then converge* [63], which prescribes to choose $\eta_i(t) = k_i\left(1 + \frac{t}{\tau_i}\right)^{-1}$. Here k_i represents the initial learning rate and τ_i represents its damping time. In this manner, the learning rate is large at the beginning of the simulation and decreases proportionally to $1/t$ for large simulation times. The parameters k_i and τ_i are application specific and must be tuned by a trial and error procedure. In particular, a very small value of τ will cause the learning rate to decrease very fast, increasing the probability to get stuck in a suboptimal minimum. On the other hand, a very large value of τ will prevent step-size shrinking and thus will hinder convergence. Analogous reasoning also applies to k. Also notice that the k_i's are measured in units of the inverse of the observable squared multiplied by an inverse time and could thus in principle be assigned to different values in case of heterogeneous observables. It appears reasonable to choose them inversely proportional to the observable variance in the prior, in order to make the result invariant with respect to a linear transformation of the observables. On the other hand, the τ_i parameter should probably be independent of i in order to avoid different λ_i's to converge on different timescales.

The update procedure for $\boldsymbol{\lambda}$ proceeds can be schematized as follow. In the algorithm scheme below, we also report the case in which inequalities restraints are applied.

All the Lagrangian multipliers λ_i are first initialized to zero. Then, at each MD step:

1. For each value of i

   (a) $s_i(\boldsymbol{q})$ and $\langle\epsilon_i\rangle$ are computed.

   (b) λ_i is updated using:

$$\lambda_i(t + \Delta t) = \lambda(t) + k_i \frac{s_i(\boldsymbol{q}) + \langle\epsilon_i\rangle - s_i^{exp}}{1 + \frac{t}{\tau}} \Delta t \tag{2.34}$$

(c) In case of inequality restraint with $\langle s_i(\boldsymbol{q})\rangle \leq s_i^{exp}$, if $\lambda < 0$ the correcting potential is ignored.

(d) In case of inequality restraint with $\langle s_i(\boldsymbol{q})\rangle \geq s_i^{exp}$, if $\lambda > 0$ the correcting potential is ignored.

2. Positions and velocities are propagated using a bias potential $V(\boldsymbol{x}) = \sum_i^M \lambda_i s_i(\boldsymbol{q})$

The update of λ keeps the system out of equilibrium. The work performed updating λ can be computed by accumulating at each step the value of

$$\sum_i^M \left(\lambda_i(t+\Delta t) - \lambda(t)\right) s_i(\boldsymbol{q}) \tag{2.35}$$

The out of equilibrium effects become less and less important as the simulation proceed and the learning rate decreases.

Once Lagrangian multipliers are converged or, at least, stably fluctuating around a given value, the optimal value $\boldsymbol{\lambda}^*$ can be estimated by taking a time average of $\boldsymbol{\lambda}$ over a suitable time window $[t_{min}, t_{max}]$. We will call "learning phase" the initial part of the simulation ($t < t_{max}$), "averaging phase" the portion of the learning phase where λ is averaged ($t_{min} < t < t_{max}$), and "production phase" the later part of the simulation ($t > t_{max}$), where $\boldsymbol{\lambda}$ is kept equal to the computed average $\boldsymbol{\lambda}^*$.

At this point, a new simulation could be performed using a static potential $V^*(\boldsymbol{q}) = V_0(\boldsymbol{q}) + k_B T \boldsymbol{\lambda}^* \cdot \boldsymbol{s}(\boldsymbol{q})$, either from a different molecular structure or starting from the structure obtained at the end of the averaging phase. Such a simulation done with a static potential can be used to rigorously validate the obtained $\boldsymbol{\lambda}^*$. Notice that, if errors have been included in the model, such validation should be made by checking that $\langle \boldsymbol{s}\rangle \approx \boldsymbol{s}^{exp} - \langle \boldsymbol{\epsilon}\rangle$. Even if the resulting $\boldsymbol{\lambda}^*$ are suboptimal, it is plausible that such a simulation could be further reweighted (Sec. 2.5.1) more easily than the one performed with the original force field. When modeling errors, if an already restrained trajectory is reweighted one should be aware that restraints will be overcounted resulting in an effectively decreased experimental error (see Section 2.3).

As an alternative, one can directly analyze the learning simulation. Whereas strictly speaking this simulation is performed out of equilibrium, this approach has the advantage that it allows the learning phase to be prolonged until the agreement with experiment is satisfactory.

The optimization procedure discussed in this section was used in order to enforce NMR data on RNA nucleosides and dinucleotides (see Sec. 4), where it was

further extended in order to simultaneously constrain multiple systems by keeping their force fields chemically consistent (see Sec. 5).

2.5.4 Other On-the-fly Optimization Strategies

Other optimization strategies have been proposed in the literature. Target metadynamics ([64, 65]) provides a framework to enforce experimental data, and was applied to enforce reference distributions obtained from more accurate simulation methods [64], from DEER experiments [65], or from conformations collected over structural databases [66]. It is however not clear if it can be extended to enforce individual averages.

Also the VES method [67] is designed to enforce full distributions. However, in its practical implementation, the correcting potential is expanded on a basis set and the average values of the basis functions are actually constrained, resulting thus numerically equivalent to the other methods discussed here. In VES, a function equivalent to $\Gamma(\boldsymbol{\lambda})$ is optimized using the algorithm by Bach and Moulines [68] that is optimally suitable for non-strongly-convex functions. This algorithm requires to estimate not only the gradient but also the Hessian of the function $\Gamma(\boldsymbol{\lambda})$. We recall that $\Gamma(\boldsymbol{\lambda})$ can be made strongly convex by suitable treatment of experimental errors (see Section 2.3). However, there might be situations where the Bach-Moulines algorithm outperforms the SGD.

The experiment-directed simulation (EDS) approach [69] instead does not take advantage of the function $\Gamma(\boldsymbol{\lambda})$ but rather minimizes with a gradient-based method [70] the square deviation between the experimental values and the time-average of the simulated ones. A later paper tested a number of related minimization strategies [71]. In order to compute the gradient of the ensemble averages $\langle s_i \rangle_{\boldsymbol{\lambda}}$ with respect to $\boldsymbol{\lambda}$ it is necessary to compute the variance of the observables s_i in addition to their average. Average and variance are computed on short simulation segments. It is worth observing that obtaining an unbiased estimator for the variance is not trivial if the simulation segment is too short. Errors in the estimate of the variance would anyway only affect the effective learning rate of the Lagrangian multipliers. In the applications performed so far, a few tens of MD time steps were shown to be sufficient to this aim, but the estimates might be system dependent. A comparison of the approaches used in Refs. [69, 71] with the SGD proposed here (reported from Ref. [17]) in practical applications would be useful to better understand the pros and the cons of the two algorithms. EDS was used to enforce the gyration radius of a 16-bead polymer to match the one of a reference system [69]. Interestingly, the restrained polymer was reported to have not only the average gyration radius in agreement with the reference one, but also its distribution. This is a clear

case where a maximum entropy (linear) restraint and a harmonic restraint give completely different results. The EDS algorithm was recently applied to a variety of systems (see, e.g., Refs. [24, 71, 72]).

Chapter 3

Force-Field Refinement using experimental data

3.1 RNA Force Fields

In the context of all-atom molecular dynamics simulation, a force field refers to the functional form and parameter sets used to calculate the potential energy of a system of atoms. In general, force fields can have different functional forms and parameters. Different parameters are usually derived in order to reproduce experimental data or quantum chemistry calculations. In all-atom force fields each type of atom, including hydrogens, have a different set of parameters. In atomistic force fields, the basic functional form of the potential energy function comprises a *bonded* part, that characterizes interactions of atoms linked by covalent bonds, and a *nonbonded* part that describe the long-range electrostatic and van der Waals forces. The general form of the total energy can be then expressed, as the sum of both contributions, by $V = V_{bonded} + V_{nonbonded}$ where $V_{bonded} = V_{bond} + V_{angle} + V_{dihedral}$ and $V_{nonbonded} = V_{electrostatics} + V_{van\,der\,Waals}$. A pictorial representation of each single interaction is shown in Fig. 3.1.

The bond and angle terms are modeled as harmonic potentials disallowing bond breaking. Dihedral angles are modeled as a series of cosine functions with different multiplicities. Electrostatic interactions are modeled by a Coulomb potential while other intermolecular interactions are modeled by a van der Waals potential. Notice that all the methods described in this thesis were applied to the refinement of the torsional component, although the application of some of them is not limited to torsions only.

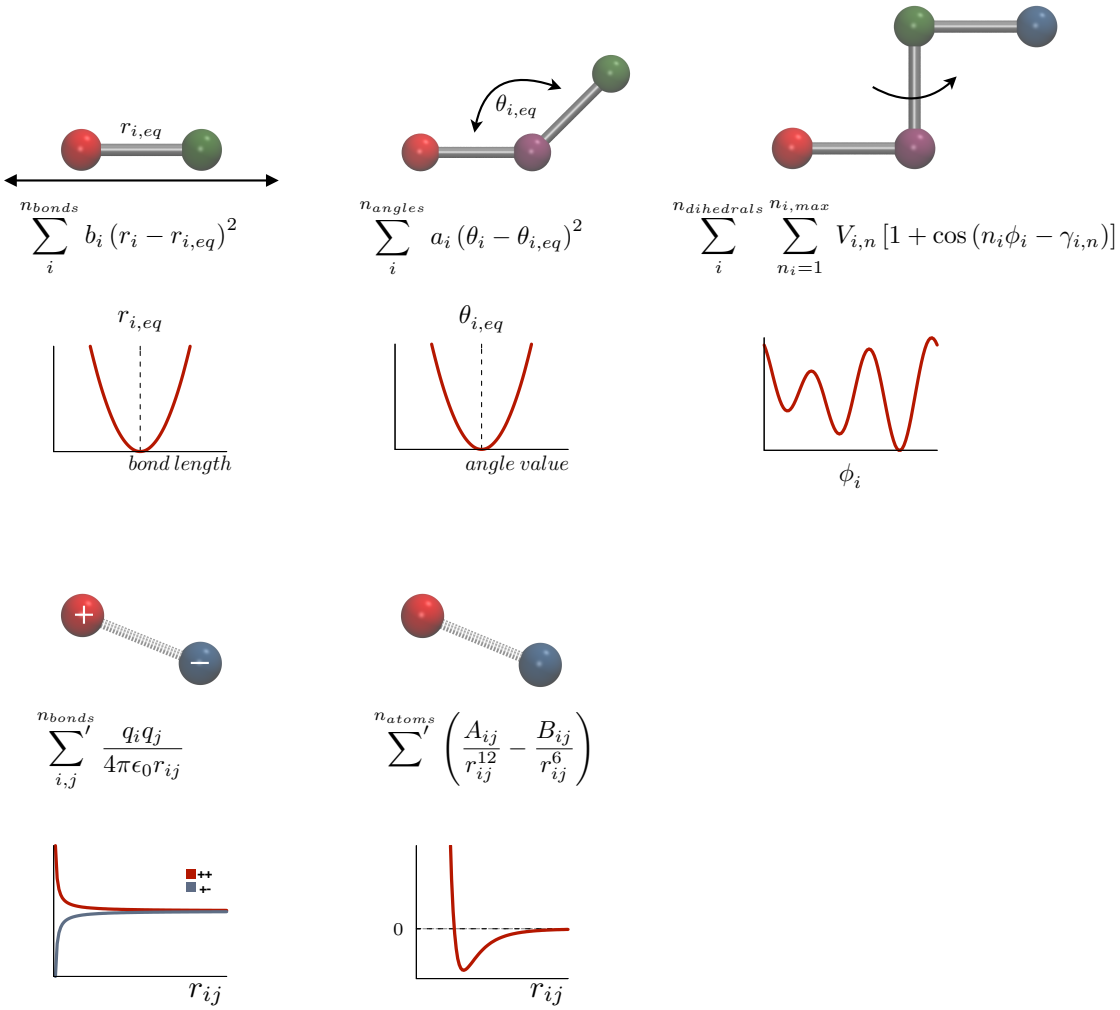


Figure 3.1: Pictorial representation of the different terms composing the potential energy function of a typical atomistic force field. For each term a sketch of the associated functional form is also shown.

3.2 Force-Field Refinement using self-consistent MaxEnt

Once the set of $\{\lambda^*\}$ satisfying the constraints in 2.15 are determined using one of the methods described in Sec. 2.5, the potential energy used in the MaxEnt framework (2.7) is equivalent to the original force field plus a correction linear in the experimental observables. We here propose a strategy in order to derive corrections which are potentially transferable to systems different to the ones used in the fitting procedure. The strategy proposed here is only applicable when the forward model used to fit experimental data can be expressed in any of the functional form composing the force field and reported in Fig. 3.1. This is particularly appealing in the case of ${}^3J$ couplings, where the functional form of the correction

is comparable to the standard torsional terms that are present in biomolecular force fields. We propose to transfer the parameters directly during the learning phase. If experimental data are available for a number of similar systems, one should simulate all the systems in parallel. Each of the simulated systems will be affected both by the corrections arising from the experiments performed on the same system and by some of the correcting potentials determined by the other simulated systems. This procedure allows to fit force-field corrections in a self-consistent way that restrains some of the terms to be equivalent to each other, applying the same corrections to all of them. The group of variables to be considered as equivalent, which will then feel the same corrections, can be in principle arbitrarily decided or based on physical-chemistry considerations. In the case of corrections derived by ^3J couplings, it is possible to enforce the same correction on dihedrals that are chemically equivalent to each other. For instance, the torsional potential around the glycosidic bond (χ angle) in an adenine is expected to be the same irrespectively of its position in the sequence.

We aim at finding a unique set of Lagrangian multipliers which will improve the agreement with experimental values of all the considered systems. The procedure we used to self-consistently fit different systems can be schematized as follows where, for simplicity, we consider to have only two systems A and B. The following list of actions is performed at each time step during the MD simulation:

1. Lagrangian multipliers are updated according to (2.34) so as to satisfy experimental constraints on system A;

2. At the same time, Lagrangian multipliers are updated so as to satisfy experimental constraints on system B;

3. Lagrangian multipliers estimated on system A are communicated to system B which will evolve feeling the sum of the potentials estimated on the two systems.

4. Lagrangian multipliers estimated on system B are communicated to system A which will evolve feeling the sum of the potentials estimated on the two systems.

At convergence this procedure will provide a set of Lagrangian multipliers where each of the consistent observables on system A will feel the same correction of its equivalent counterpart on system B. This guarantees that the corrected force field have chemically consistent corrections. However, the interplay between the two potentials does not guarantee that experimental constraints will be satisfied for both systems. For this reason, the self-consistent force-field fitting requires experimental errors to be explicitly modeled.

3.3 Force-Field refinement using arbitrary functional forms

In the previous section, we proposed a method to derive portable force-field corrections based on the fit of experimental data using the maximum entropy principle. The advantage of using the maximum entropy principle is that we are not using any extra information beside the one contained in the enforced constraints. On the other side the main limitation is, as far as concern the application to force field refinement, the availability of experimental data expressible with a forward model compatible with the functional form of the employed potential energy function. To overcome this limitation, we here propose a method which allow to enforce arbitrary ensemble averages using arbitrary correcting functional form. Since the correcting functional form is arbitrary, it can be also chosen to be compatible with the force field, allowing then the derived corrections to be incorporated in the force field itself and then transferred to other systems. Also in this case, the portability is enhanced by fitting multiple data on multiple different system at the same time. Of course in such case it is not guaranteed that extra information has not been injected in the system. Indeed the chosen functional forms contain extra information which is independent from the enforced constraints. Thus, the use of one approach with respect to the other strongly depends on the available experimental data and corresponding forward models.

Let $P_0(\boldsymbol{x})$ be the Boltzmann probability (*p.d.f.*) associated to the original force field with potential energy $V_0(\boldsymbol{q})$ in Eq. 2.7. Our aim is to construct a refined probability distribution of the form $P(\boldsymbol{x}, \{\boldsymbol{\lambda}\}) \propto P_0(\boldsymbol{x}) \exp\left(-\beta \sum_i^N f_i(\boldsymbol{x})\lambda_i\right)$. The correcting potential is thus expanded on a set of N basis functions $\boldsymbol{f}$ (dihedral angles or non-bonded interactions). To each of the N basis functions is associated a weight λ_i that is proportional to the strength of the correction and must be found in order to simultaneously reproduce M experimental observables. Notice the difference w.r.t Maximum Entropy based methods discussed in Chap. 2 in which the basis functions are by construction identical to the forward-model used to back-calculate the experimental observables, and thus the number N of parameters is equal to the number of the enforced experiments. In our approach the number of experimental observables M is in general different from N, usually being much larger ($M \gg N$). Notice that experimental observables are defined as averages over the refined ensemble. In order to find the optimal weights $\lambda_i (i = 1 \ldots, N)$, we define an error function E encoding the overall discrepancy between observable averages in the refined ensemble and the relative experimental values. The error function is built such that $E = 0$ if all observables are correctly reproduced. Given

a set of M experimental observables denoted by $O_j (j = 1, \ldots, M)$, it is possible to enforce both equalities (i.e. $\langle O_j \rangle = O_j^{exp}$) or inequalities (i.e. $\left(\langle O_j \rangle < O_j^{exp}\right) \vee \left(\langle O_j \rangle > O_j^{exp}\right)$). The averages are meant to be taken in the refined probability distribution $P(\boldsymbol{x}, \{\boldsymbol{\lambda}\})$. We will here compute such averages by reweighting the unrefined ensemble. The accuracy of the procedure will then depend on how close the refined ensemble is to the unrefined one. The error function, which depend on the observables averages, will indirectly depend on $\boldsymbol{\lambda}$. We introduce a general expression of the error function E such as:

$$E\left(\langle O_1 \rangle (\boldsymbol{\lambda}), \ldots, \langle O_M \rangle (\boldsymbol{\lambda})\right) + \alpha |\boldsymbol{\lambda}|^2 \tag{3.1}$$

which must be minimized to in order to enforce the M ensemble averages. We will denote with $\boldsymbol{\lambda}^\star$ the set of parameters which minimize 3.1. Notice that the dimensionality of the minimization is given by the size of the vector $\boldsymbol{\lambda}$ and is equal to N. The second term in 3.1 is a l^2 regularization term needed to avoid over-fitting. The strength of the regularization can be tuned with the parameter α, choosing a value in the range $0 \leq \alpha < \infty$. Practically, this term avoids the values of $\boldsymbol{\lambda}$ to become too large. Setting $\alpha = 0$ will maximally fit the data at the cost of a very large correcting bias potential. This will lead to a new ensemble which will be potentially very different from the unrefined one, generating poor reweighting performance. In the opposite case of $\alpha = \infty$ the data will not be fitted. The optimal value of α must be then chosen carefully. A common strategy to find the optimal value of α, is represented by the cross-validation method which is explained in subsec. 3.3.1. In order to minimize the error function (Eq. 3.1) in the parameters space of $\boldsymbol{\lambda}$, we compute the gradient of E as function of $\boldsymbol{\lambda}$:

$$\begin{aligned} \frac{\partial E}{\partial \lambda_j} &= \sum_{i=1}^{M} \frac{\partial E}{\partial \langle O_i \rangle} \frac{\partial \langle O_i \rangle}{\partial \lambda_j} = \\ &= \sum_{i=1}^{M} \frac{\partial E}{\partial \langle O_i \rangle} \left(\langle f_j \rangle \langle O_i \rangle - \langle f_j O_i \rangle\right) + 2\alpha\lambda_j \end{aligned} \tag{3.2}$$

where $j = 1, \ldots, N$. The still unknown term $\frac{\partial E}{\partial \langle O_i \rangle}$ depends on the functional form of the chosen error function E and is then application specific. Knowing the value of E and its derivatives allow then to minimize the function itself by using a proper gradient based minimizer. In our applications we will use the limited memory version of the Broyden-Fletcher-Goldfarb-Shanno algorithm (L-BFGS). Once the optimal $\boldsymbol{\lambda}^\star$ are found, the final estimation of the observable averages can

be found by reweighting the unrefined ensemble:

$$\langle O_i \rangle = \frac{\sum_{t=1}^{N_{frames}} O_i(t)\, e^{\sum_{d=1}^{N} f_d(t)\lambda_d}}{\sum_{t=1}^{N_{frames}} e^{\sum_{d=1}^{N} f_d(t)\lambda_d}} \tag{3.3}$$

where t denotes the t^{th} frame of the unrefined ensemble.

The described methodology has been implemented in a C++ program.

```
#load observables for system A (sytemA0, systemA1,…,systemA[NumObservablesToFit-1]
     add_system systemA PathToBasisFunctionsFile NumObservablesToFit

#load observables for system B (sytemB0, systemB1,…,systemA[NumObservablesToFit-1]
     add_system systemB PathToBasisFunctionsFile NumObservablesToFit

# stop criterion for minimization
     epsilon 1e-3

#maximum minimization iterations
     maxiter 400

#regularization parameter
     alpha 1500.0

#start from a given set of weights
     lambda  λ1 λ2 … λN

#equality restraint: ⟨systemA0⟩ = ref0 on observable 0 of system A
     function ω1*(systemA0-ref0)^2

#equality restraint: ⟨systemA1⟩ = ref1 on observable 1 of system A
     function ω1*(systemA1-ref1)^2

#inequality restraint: ref2 ≤ systemB0 ≤ ref3 on observable 0 of system B
     function ω2*max(ref2-systemB0,0)^2
     function ω2*max(systemB0-ref3,0)^2

#print any function of the observables (at each minimization step)
     print OutFileName systemA0+systemA1+systemB1

#save optimal parameters on file
     lambdafile OutFileName
```

Figure 3.2: Sample input file for the code written to perform force-field refinement using arbitrary functional forms

The program reads the basis functions stored in external files and can either fit data on a single system or perform a multi-system fit like in the force-field refinement application (see Sec. 6). See a sample input file in Fig. 3.2. By using the Lepton library to perform symbolic differentiation, custom error functions can be defined in the minimization procedure (see Fig. 3.2). This is particularly usefull when using inequality restraints. In such case, as shown in Sec. 6, the `max` and `min` functions can be used.

3.3.1 Cross Validation

As we anticipated the efficiency of the reweighting procedure is inversely related to the distance between the refined and unrefined ensembles. Such distance can be kept relatively small by using a proper regularization parameter α in Eq. 3.1. The optimal value of α, to which we will refer as $\alpha^\star$, can be found via cross validation strategies. We will here use the k-fold cross validation method. In k-fold cross validation, the data set is split in k blocks. For each trial value of α, k minimizations are performed. During the i^{th} minimization, with $i = 1, \ldots, k$, the i^{th} block is left out as validation set while the remaining $k-1$ blocks are used as training set. After the optimal $\lambda^\star$ are found, the error function E is evaluated on the validation set. When evaluating the error on the validation set the unregularized error function must be used (setting $\alpha = 0$). At the end of the k^{th} minimization, a final cross validation error E_{cv}, for the given α, is computed as the average of the validation errors on each of the k blocks. The optimal value of α will then be the one minimizing the cross validation error E_{cv}. The rationale behind is that in this way we will choose the more conservative value of α which will generalize better than other values of α to data that were not seen in the training set. In the present contest, the optimal α is thus expected to result in force field corrections that will be better transferable to systems not considered in the fitting procedure. As a practical example we show later the application of the method to the refinement of RNA force-field torsions. As training data we use 6 different RNA systems, 4 RNA tetranucleotides and 2 RNA tetraloops.

Chapter 4

Enforcing experimental data on nucleosides using MaxEnt

In this chapter we will show the application of the MaxEnt restraining procedure to the fit of ^3J scalar couplings NMR data on RNA nucleosides. Results reported here are for Adenosine only. Results of the fitting procedure on the remaining nucleosides are reported in Appendix B. Most of the content of this chapter is reported and adapted from Sec. 3.1 of our published work [17].

4.1 RNA structure and ^3J scalar coupling

RNA (Ribonucleic acid) is a polymeric molecule formed by a combination of 4 different nucleotides [73–76]. Each nucleotide contains a ribose sugar ring composed by 5 carbons numbered $1'$ through $5'$, an aromatic base attached to the $1'$ carbon, and a phosphate group. The nucleotides are linked to one another in a linear manner, by phosphodiester bonds between the sugar of one nucleotide and the phosphate group of the adjacent nucleotide. The phosphate group of adjacent nucleotides is attached to the $3'$ carbon from one side and to the $5'$ carbon on the other side. The most common nucleobase types are: adenine (A), cytosine (C), guanine (G), and uracil (U). Cytosine and uracil are derivatives of the pyrimidine (Py) ring, while adenine and guanine have a purine (Pu) scaffold, a pyrimidine ring fused to an imidazole ring. Each phosphate group have a negative charge. The structures of a nucleoside (nucleotide without the phosphate group) and a dinucleotide, together with relevant RNA torsions (torsion angles $\alpha\,,\beta\,,\gamma\,,\delta$, ϵ and ζ) are shown in Fig. 4.2.

Experimental informations about RNA torsion angles can be obtained by NMR experiments measuring ^3J scalar couplings. ^3J scalar couplings belong to the general family of ^nJ scalar couplings, where $n = 1, \ldots, 5$ indicates the number of

bonds separating the nuclei A and X between which the magnetic interaction is measured.

In general, the scalar coupling J is a through-bond interaction, in which the spin of one nucleus perturbs (polarizes) the spins of the intervening electrons, and the energy levels of neighboring magnetic nuclei are in turn perturbed by the polarized electrons. This leads to a lowering of the energy of the neighboring nucleus when the perturbing nucleus has one spin, and a raising of the energy when it has the other spin. The J coupling (always reported in Hz) is field-independent (i.e. J is constant at different external magnetic field strength), and is mutual (i.e. $J_{AX} = J_{XA}$). Because the effect is usually transmitted through the bonding electrons, the magnitude of J falls off rapidly as the number of intervening bonds increases. Coupling over one (${}^1J$), two (${}^2J$) and three (${}^3J$) bonds usually dominates the fine structure of NMR spectra, but coupling across four and five (${}^4J$, ${}^5J$) bonds is often seen, especially through π bonds (double and triple bonds, aromatic carbons). ${}^3J$ scalar coupling are always positive. The forward model used to back-calculate ${}^3J$ from molecular dynamics structures is given by the Karplus equations [27]. The equation gives an approximate value for ${}^3J_{HH}$ as a function of dihedral angle between the protons. The Karplus equation is based on the observation, supported by theoretical considerations, that vicinal H-H couplings will be maximal with protons with 180° and 0° dihedral angles (anti or eclipsed relationship results in optimal orbital overlap) and that coupling will be minimal (near 0) for protons that are 90° from each other. The general functional form of the Karplus equation is the following:

$$ {}^3J_{HH}(\theta) = A\cos^2(\theta+\phi_k) + B\cos(\theta+\phi_k) + C\sin(\theta+\phi_k)\cos(\theta+\phi_k) + D \quad (4.1) $$

which is plotted in Fig. 4.1 using an example set of parameters.

The estimation of Karplus equation parameters it's not an easy task and several groups [8, 77–83] have developed different parameters A, B, C and D by fitting the Karplus curves with the ones obtained by experiments or quantum chemistry calculations. To have and idea of the variability among different parametrizations see Tab. B.3b. The impact of different Karplus parameters on a real application is discussed in Subsec. 4.4. Notice that the error treatment procedure introduced in previous chapters should also alleviate the effect of wrong forward model functional forms, although we would not expect this to happen in the investigated cases.

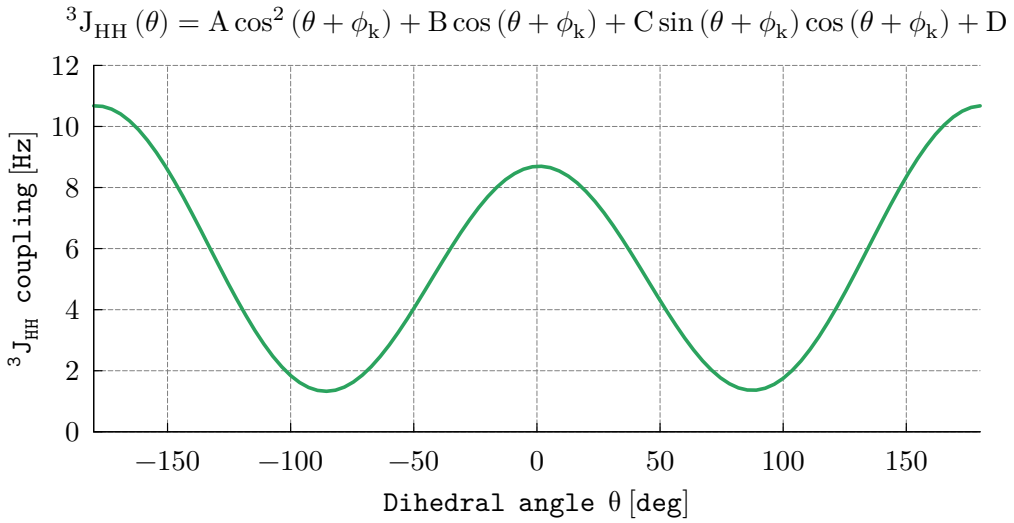


Figure 4.1: General functional form of Karplus equation (shown as plot title) and plot for an example set of Karplus parameters. In this plot $A = 8.313$, $B = -0.99$, $C = 0.27$, $D = 1.373$.

4.2 Molecular dynamics parameters

We performed molecular dynamics on all RNA nucleosides (A, C, G, U). Molecular dynamics simulations were performed using the GROMACS software package [84] in combination with a modified version of the PLUMED plugin [85]. RNA, explicit water, and ions were modeled using the most recent parametrizations within the Amber force field [86–91] (ff99bsc0 + χ_{OL3}). Parameters are available at http://github.com/srnas/ff. Bonds were constrained using the LINCS algorithm[92], allowing for a time-step of 2 fs. The particle-mesh Ewald algorithm [93] was used for long-range electrostatic interactions with a cut-off distance of 1 nm. Simulations were performed at temperature T = 300 K and pressure P = 1 bar [94, 95]. To allow for a fast convergence of the simulated ensembles, sampling was enhanced using replica-exchange with collective-variable tempering (RECT) [96] on selected collective variables. Biased variables are the torsional angles χ, γ, and the puckering variables Z_x and Z_y [97]. Four replicas were used for each system, with bias factors ranging from 1 to 5.

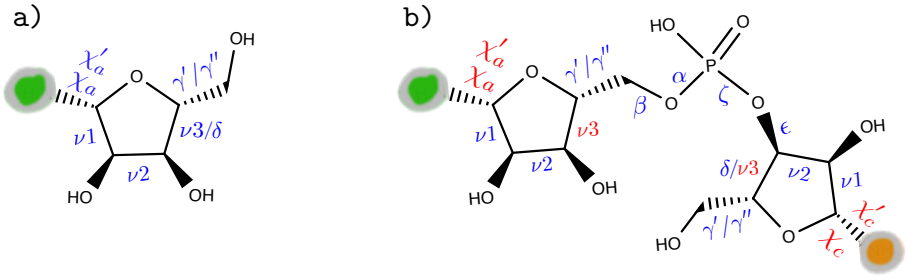
4.3 MaxEnt algorithm parameters

The minimization strategy adopted is the stochastic gradient descent described in Sec. 2.5.3 using the update rule in Eq. 2.33. We performed 200 ns MD per replica using the first 100 ns as learning phase. Lagrangian multipliers were averaged from $t_{min} = 50$ ns to $t_{max} = 100$ ns and these averages were used in the

production phase for the last 100 ns. The parameters for the learning phase were chosen as $k = 0.001\ Hz^{-2}ps^{-1}$, $\tau = 3\ ps$, $\sigma = 2.0\ Hz$. A Laplace prior for the error was used. The biased replicas were simulated using Lagrangian multipliers estimated on the fly from the reference replica, so as to maximize the acceptance rate for the replica-exchange procedure. Each system was simulated with 4 RECT replicas. PLUMED input files are provided in D.1. The modifications to PLUMED required to perform this simulations are implemented in PLUMED since version 2.4 and can be activated with the keyword `MAXENT`. User manual for the `MAXENT` keyword can be found at https://plumed.github.io/doc-v2.4/user-doc/html/_m_a_x_e_n_t.html

4.4 Results

We here report and discuss results for Adenosine only. Results for other nucleosides (uridine, cytidine, and guanosine) are similar and are summarized in Appendix B (Tab. B.1). For this system, $M = 7$ experimental ^3J scalar couplings are available [98], involving dihedral angles both on the backbone and on the nucleobase (see Figure 4.2 on page 36a).



Adenosine — A

Adenylyl-(3'→ 5')-cytidine — ApC

Figure 4.2: Torsional angles associated to the available experimental ^3J scalar couplings for the Adenosine (panel a) and the ApC dinucleoside (panel b). Atoms associated to each torsion are: ν_1 = H1'-C1'-C2'-H2', ν_2 = H2'-C2'-C3'-H3', ν_3 = H3'-C3'-C4'-H4', δ=C5'-C4'-C3'-O3', γ=O5'-C5'-C4'-C3', γ' = H4'-C4'-C5'-H5', γ'' = H4'-C4'-C5'-H5", ϵ_1 = C4'-C3'-03'-P, ζ_1 = C3'-03'-P-05', α_2 = 03'-P-05'-C5', β_2 = P-05'-C5'-C4', χ_A = 04'-C1'-N9-C4, χ'_A = H1'-C1'-N9-C8+60°, χ_C = O4'-C1'N1-C2, χ'_C = H1'-C1'-N1-C6+60°

We assess the deviation between simulation and experiments by computing the

RMSE of the back calculated data from their experimental values, defined as:

$$RMSE = \sqrt{\frac{1}{M}\sum_{i=1}^{M}({}^3J_{i,simulated} - {}^3J_{i,exp})^2}. \tag{4.2}$$

The RMSE has the same units of the analyzed quantities which in this case are Hz. We first computed the scalar couplings using the standard Amber force field (see Tab. 4.1) obtaining a value of 1.3 Hz. This number is significantly larger than the expected experimental error on such data. However, it is important to consider also errors in the parametrization of the Karplus equations.

The robustness against the choice of Karplus parameters can be assessed by computing the standard deviation of each coupling, when the same is back calculated using different sets of Karplus parameters. Assuming we have M torsions and $K_{i;i=1,\dots,M}$ different sets of Karplus parameters for the torsion i we denote with $J_i(n)$ $(n = 1, \dots, K_i)$ the ^{3}J scalar coupling, associated to torsion i, back calculated using the n^{th} set of Karplus parameters. We can hence compute, for each torsion i, the standard deviation among all the sets K_i as $\sigma_i = \sqrt{\frac{1}{K_i-1}\sum_{n=1}^{K_i}(J_i(n) - \langle J_i \rangle)}$. The standard deviation among all the couplings $\Sigma = \sqrt{\frac{1}{M}\sum_{i=1}^{M}\sigma_i^2}$ is then computed as a measure of the overall variance of Karplus parameters. In Tab. B.3a we report the values of the couplings obtained using different sets of Karplus parameters which are reported in Tab. B.3b.

The above mentioned test was carried out on a trajectory corresponding to the ApC dinucleoside monophosphate, since more torsions are available on dinucleosides (see Fig. 4.2 panel b). The obtained standard deviation Σ is 0.6 Hz, which is significantly smaller than the RMSE observed for the Amber force field. This test also sets a lower bound for the RMSE indicating that enforcing an RMSE between simulation and experiment lower than 0.6 could lead to results dependent on the choice of the Karplus equation parameters.

Additionally, we estimated the ability of random conformations to reproduce the experimental ^{3}J scalar couplings. To this aim, we computed the RMSE between simulation and experiments assuming a flat distribution on all the torsions used in the ^{3}J coupling calculation. The torsions considered were, again, the ones available for the ApC dinucleoside with the same set of Karplus parameters which was used to produce all the results in this section. The resulting RMSE is approximately 2.9 Hz, indicating that random conformations do not reproduce experimental data with the accuracy of MD ensembles.

We then use the proposed iterative procedure to determine the correcting potentials. Although we use a Laplace prior for the error, we notice that since the correcting potential has as many degrees of freedom as experimental data, one

cannot expect to detect inconsistencies in the dataset. In the next chapters we will see that when less parameters than experimental datapoints are used such inconsistencies are implicitly taken into account. A crucial parameter in the fitting procedure is σ, which controls the width of the prior distribution for the deviation between experiment and theory, and encodes the confidence that we have in the force field. Results for $\sigma = 2.0$ Hz are shown in Tab. 4.1.

	^3J coupling (Hz)		
torsion	Exp.[98]	Amber	Amber$_{MaxEnt}$
ν_1	6.0	8.5	6.9
ν_2	5.0	5.1	5.1
ν_3	3.4	3.5	4.2
γ'	3.0	3.2	3.1
γ"	3.4	1.5	2.6
χ	3.6	4.7	4.1
χ'	3.9	3.6	3.5
	$RMSE\ (Hz)$		
	0.0	1.3	0.6

Table 4.1: ^3J scalar coupling for the Adenosine nucleoside. Experimental values and back calculated values are shown, both using the Amber force field and the MaxEnt corrections. Angle χ' for the Adenosine nucleoside is defined as the $H1' - C1' - N9 - C8$ torsion along with a shift of 60°. Statistical errors on the values obtained from MD as well as on the calculated RMSE are less than $0.1Hz$.

As it can be seen, the RMSE is greatly reduced compared to the original Amber force field. Final Lagrangian multipliers are shown in Tab. B.2. We recall that the greater the value of σ the higher the confidence in the force field and the lower the correcting MaxEnt potential. To have an idea of the influence of the σ parameter on overall RMSE, a plot of RMSE vs σ is provided in Fig. 4.3.

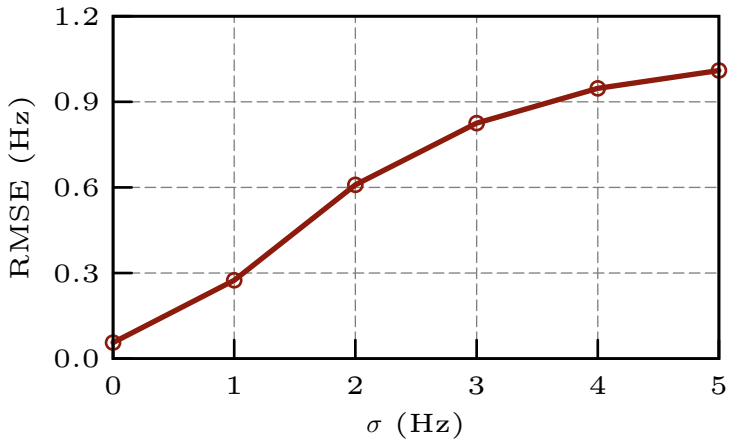


Figure 4.3: RMSE as functions of σ for the Adenosine with a Laplace prior on the error.

We notice that in this case an arbitrary small RMSE can be obtained by choosing a negligible value of σ however this is often not a good practice. We recall that enforcing a RMSE smaller than the typical RMSE between different set of parameters in Karplus relations (≈ 0.6 Hz) is in fact not meaningful as explained before. Moreover, this would introduce much larger corrections to the force field (see Fig. B.1) that could lead to uncontrolled artifacts. For instance, in some of the simulations using $\sigma = 0$ we obtained stereoisomerizations of the C2' atom of the sugar (data not shown). With the adopted value of $\sigma = 2$ the effect of the corrections on the one-dimensional free-energy profiles of the refined dihedral angles is $\leq 2\ K_bT$. Free-energy profiles for a set of representative torsional angles are shown in Fig. 4.4.

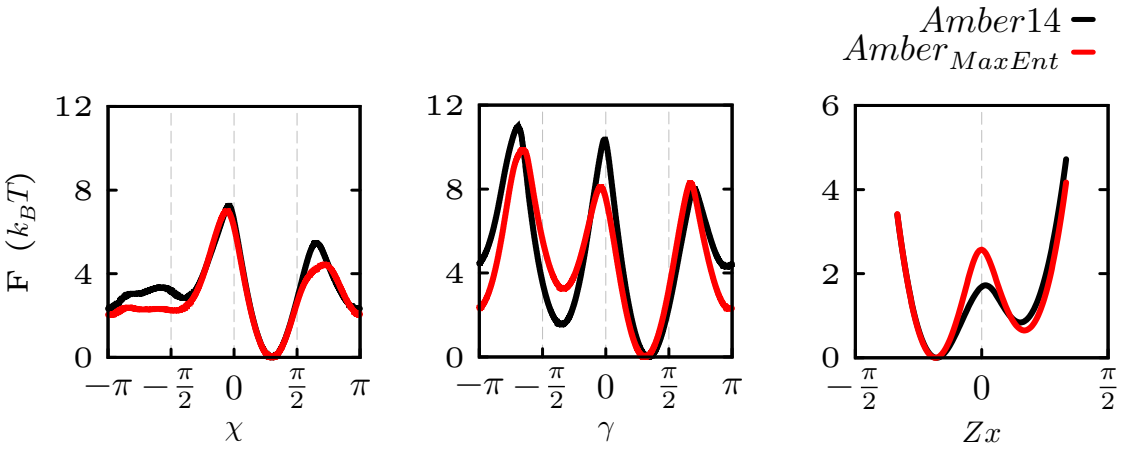


Figure 4.4: One-dimensional free-energy profiles for a representative group of the corrected dihedral angles obtained with Amber and with the refined Amber$_{MaxEnt}$ force fields. Z_x variable [97] is related to sugar conformations C3'-endo ($Z_x > 0$) and C2'-endo ($Z_x < 0$).

Chapter 5

RNA Force-Field Refinement using self-consistent MaxEnt

5.1 On-the-fly refinement

We use the procedure introduced and explained in Sec. 3.2, to perform a self-consistent force-field refinement, on-the-fly, on a set of different RNA nucleosides and dinucleosides mono-phosphate. To this aim we performed molecular dynamics on RNA nucleosides (A and C) and dinucleosides mono-phosphate (ApA, ApC, CpA, and CpC). In Fig. 4.2b the ApC dinucleoside is shown in order to highlight the torsions considered in the refinement procedure. As explained in Sec. 3.2 one has first to identify which torsions must be considered chemically equivalent. Such torsions will then feel the same correcting potential. In this application chemically equivalent torsions are those highlighted in blue in Fig. 4.2b.

5.1.1 Molecular dynamics parameters

Molecular dynamics simulations were performed with the same setup and force-field introduced in 4.2. For the dinucleosides mono-phosphate we additionally included, among the variables biased for the RECT method, torsional angles α, β, ϵ, and ζ as well as the distance between the two nucleobases. Four replicas were used for each system, with bias factors ranging from 1 to 5 both for the nucleosides and dinucleosides mono-phosphate.

5.1.2 MaxEnt algorithm parameters

The MaxEnt algorithm parameters k,τ and σ are the same to the one used in Sec. 4.3 where RNA nucleosides only were simulated. For the dinucleosides mono-phosphate we performed 600 ns using first 300 ns as learning phase and averaging

Lagrangian multipliers between $t_{min} = 150\ ns$ and $t_{max} = 300\ ns$. In both cases a Laplace prior for the error was used. The biased replicas were simulated using Lagrangian multipliers estimated on the fly from the reference replica, so as to maximize the acceptance rate for the replica-exchange procedure. To implement the self-consistent force-field fitting described above, we simultaneously simulated six systems (A, C, ApA, ApC, CpA, and CpC). The replica exchange framework of GROMACS was used, disallowing unphysical exchanges between replicas simulating different systems. Each system was simulated with 4 RECT replicas, resulting in a total of 24 replicas. Lagrangian multipliers were adjusted to fit experimental data available for each of the systems and transmitted on the fly to the other replicas so as to be applied on all the equivalent dihedrals. Input files are provided in Appendix (see D.2, D.3, D.4, D.5). The modifications to PLUMED required to perform this simulations are implemented in PLUMED since version 2.4 and can be activated with the keyword *MAXENT*.

5.1.3 Enforcing ^3J scalar couplings

The obtained Lagrangian multipliers for each torsional angle are summarized in Table 5.1 on page 43. When fitting systems involving different nucleobases (e.g A and C), torsions around the glycosidic bond were considered as base dependent, together with the ν_3 torsion, which controls the balance between C2'-endo and C3'-endo sugar conformations and we empirically observed to be the sugar torsion that is most correlated with the base/sugar relative orientation. Such torsions will feel a different correcting potential depending on whether they belong to and Adenosine or Cytosine. Base dependent torsions are highlighted in red in Figure 4.2 on page 36b. In case of a duplicated term in a single simulation (e.g., the χ angle in an adenine which appears twice in the ApA dinucleoside monophosphate), we do not enforce their individual values but the sum of the two scalar couplings to match the sum of the corresponding experimental values. This implicitly makes sure that both torsional angles feel the same correction.

RMSEs for each system are shown in Figure 5.1 on page 43. Here it can be appreciated that all the resulting RMSEs are below 1 Hz. We notice that in this case the number of non-equivalent dihedrals (16) is significantly lower than the number of experimental data (78). This means that data are redundant and the procedure can detect potential inconsistencies between experimental data.

Back calculated ^3J couplings for each torsion and Karplus parameters are provided in Appendix C (see Tab. C.1, C.2 and Tab. C.3). The effect of the corrections on the one-dimensional free-energy profiles associated with all the dihedral angles is shown in Appendix C (Fig. C.1and Fig. C.2)

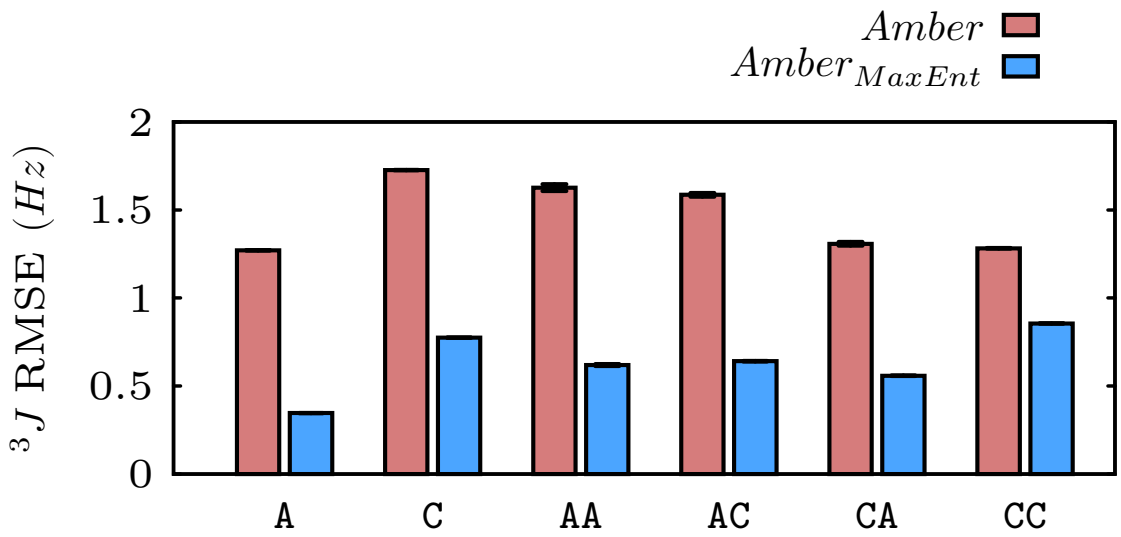


Figure 5.1: ^3J RMSE for each system with the Amber force-field and the Amber$_{MaxEnt}$ force-field obtained with the self consistent refinement.

Coupling	Torsion θ	Base	Lagrangian multiplier (Hz^{-1})
$^3J_{H1'H2'}$	ν_1	A,C	0.4393
$^3J_{H2'H3'}$	ν_2	A,C	0.0570
$^3J_{H3'H4'}$	ν_3	A	0.4009
		C	0.3316
$^3J_{H4'H5'}$	γ'	A,C	0.3643
$^3J_{H4'H5''}$	γ''	A,C	-0.2077
$^3J_{H3'P}$	ϵ_1	A,C	-0.2358
$^3J_{H5'P}$	β_2	A,C	-0.0237
$^3J_{H5''P}$	β_2	A,C	-0.0700
$^3J_{C2'P}$	ϵ_1	A,C	0.2015
$^3J_{C4'P}$	ϵ_1	A,C	0.2010
	β_2	A,C	0.1923
$^3J_{H1'C4}$	χ	A	0.1758
$^3J_{H1'C2}$		C	0.4270
$^3J_{H1'C8}$	χ'	A	-0.4068
$^3J_{H1'C6}$		C	-0.7401

Table 5.1: Lagrangian multipliers associated to each torsional angle used in the self consistent procedure together with the associated Karplus parameters used to back calculate ^3J scalar couplings. The third column specifies to which system the corrections have to be applied. Karplus relations used are in the form $^3J(\theta) = A\cos^2(\theta+\varphi) + B\cos(\theta+\varphi) + C\sin(\theta+\varphi)\cos(\theta+\varphi) + D$. χ' is defined as $H1' - C1' - N1/N9 - C6/C8$ along with a phase shift of 60°.

5.1.4 Validation on RNA Tetranucleotides

The derived corrections are then validated on two RNA tetranucleotides, AAAA and CCCC. In a previous work [66] it has been shown that on such systems a significant improvement of the agreement with NMR solution experiments can be obtained penalizing structures with $\alpha(g+)/\zeta(g+)$ conformations. These conformations are associated to intercalated structures [8, 66] that are incompatible with solution experiments. We call here $\text{Amber}_{\alpha\zeta}$ a potential obtained adding to Amber a two dimensional Gaussian potential centered on the $\alpha(g+)/\zeta(g+)$ conformation with a standard deviation of 0.7 rad and height 8 $\frac{kJ}{mol}$. The Lagrangian multipliers discussed above were obtained as corrections to be applied on the Amber force field. We here perform a new self-consistent fit with identical simulation parameters using as prior distribution the $\text{Amber}_{\alpha\zeta}$ potential and call $\text{Amber}_{\alpha\zeta MaxEnt}$ the resulting force field. We also define the $\text{Amber}_{\alpha\zeta+MaxEnt}$ force field as the one obtained by adding the corrections obtained in the previous section on top of the $\text{Amber}_{\alpha\zeta}$ force-field, without repeating the self-consistent refinement. In order to asses the performance of Amber, $\text{Amber}_{\alpha\zeta}$, $\text{Amber}_{\alpha\zeta+MaxEnt}$ and $\text{Amber}_{\alpha\zeta MaxEnt}$ we performed the same analysis as in refs [8, 66] on AAAA and CCCC. This analysis is made by reweighting the trajectories described in Ref. [66]. For each force field, we evaluate the RMSE associated to scalar coupling as well as the number of violations and false positives in contacts predicted by nuclear Overhauser experiments (NOEs). NOEs are particularly important in tetranucleotides since they are sensitive to intercalated structures erroneously obtained using the Amber force field that have been previously reported [8, 9, 66, 99]. We notice that NOEs might not be visible for many reasons other than the distance is too large. This often happens with large RNAs and proteins and can be due to (1) one or both of the involved resonances are broader than others due to local conformational flexibility at an intermediate rate (microsecond to millisecond), or (2) chemical exchange with solvent protons. All the observed signals in these small systems have similar line widths (i.e. no intermediate conformational exchange) and only non-exchangeable protons are analyzed. Additionally, for a similar tetranucleotide (GACC) it was shown that intercalated structures would lead to peaks that would be easy to detect because they would appear in unique and uncrowded regions of the spectra [100]. Comparison of MD with NMR for the tetranucleotides is reported in Fig. 5.2. As it can be seen, the MaxEnt corrections improve the agreement with experimental scalar couplings for AAAA and CCCC with respect to both Amber and $\text{Amber}_{\alpha\zeta}$ force fields. When considering the NOEs, it can be appreciated that the largest improvement with respect to Amber originates from the $\alpha\zeta$ correction, as previously suggested. Interestingly, the MaxEnt corrections further decrease the

number of false positives in CCCC and the number of violations in AAAA. We summarize the agreement with experimental NOEs using the NMR score defined in Ref. [8]. When comparing Amber$_{\alpha\zeta MaxEnt}$ with Amber$_{\alpha\zeta+MaxEnt}$ it can be noticed that performing a new self-consistent fit starting from Amber$_{\alpha\zeta}$ represent a better choice since it improves both the RMSE and the total NMR agreement. We remark that this is a completely independent validation since experimental data for AAAA and CCCC were not considered in the self-consistent force-field refinement procedure. Moreover, we stress that the validation is made on systems that are different from those used in the fitting procedure. This suggests the corrections to be portable to larger RNA molecules. We finally notice that if the magnitude of the correcting potential is larger than a few k_BT the reweighting procedure can lead to very poor sampling [16, 55, 101]. To assess the confidence in the reweighting we computed both the Kish's effective sample size [56] and the statistical error on the RMSE. The Kish's effective sample sizes are respectively 10 (CCCC) and 29 (AAAA) for the Amber$_{\alpha\zeta MaxEnt}$ potential, to be compared to 4000 frames in the unbiased trajectories. Despite these numbers might seem low, the impact of the reweighting procedure on the estimated RMSE is better described by its statistical error. Although the statistical error is significantly increased in the reweighted ensemble (see Figure 5.2 on page 46), its value is still small enough to allow for a proper comparison between the RMSEs. The structural ensembles obtained with Amber, Amber$_{\alpha}\zeta$ and Amber$_{\alpha\zeta MaxEnt}$ are also shown in Figure 5.3 on page 46. It can be appreciated that in both AAAA and CCCC the effect of the MaxEnt corrections is to penalize structures with high value of root-mean-square deviation (RMSD) after optimal superposition from the ideal A-form conformation, which are related to wrongly predicted intercalated conformations.

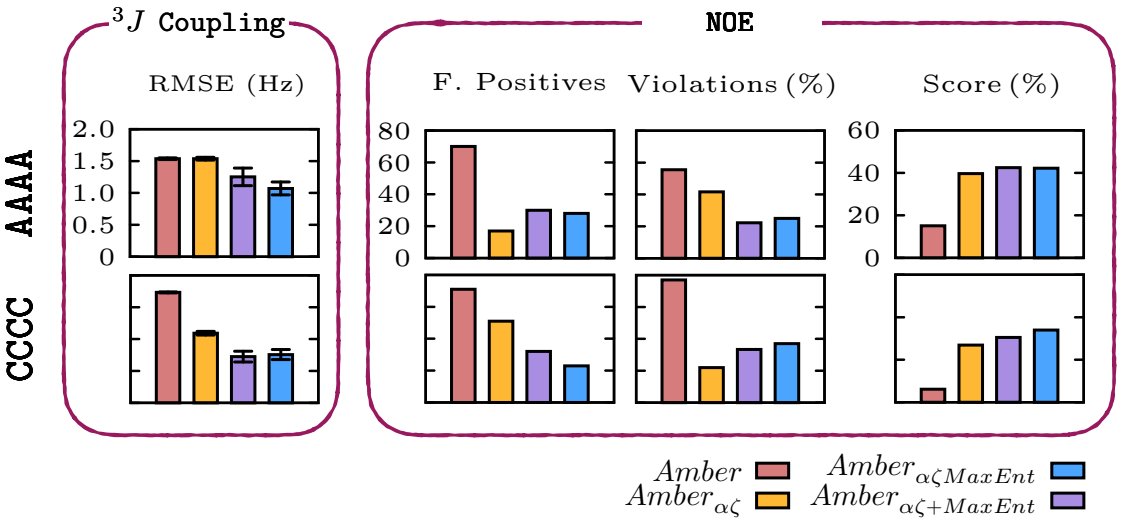


Figure 5.2: Agreement with the NMR solution experiments for Amber, Amber$_{\alpha\zeta}$ and Amber$_{\alpha\zeta MaxEnt}$. The number of distance false positives represent the MD predicted NOEs not observed in the experiments.

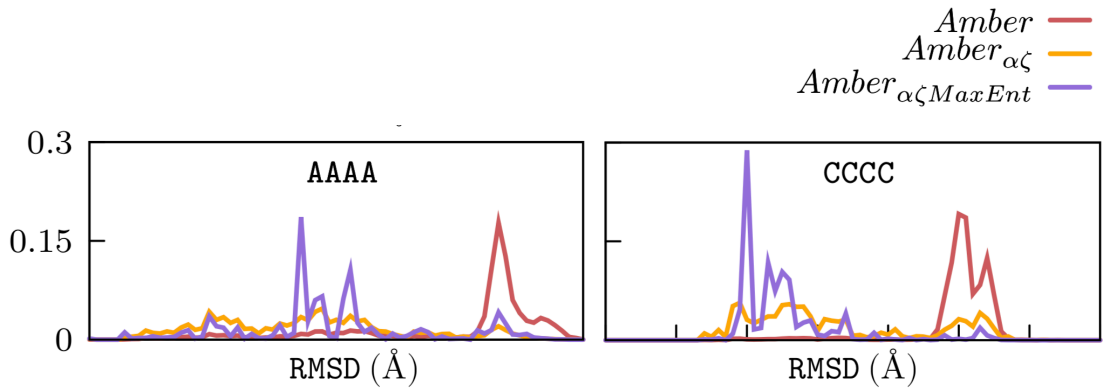


Figure 5.3: Structural ensembles obtained with Amber, Amber$_{\alpha}\zeta$ and Amber$_{\alpha\zeta MaxEnt}$. Ensembles are represented by showing the histogram of the RMSD from the ideal A-form conformation. Both Amber$_{\alpha}\zeta$ and Amber$_{\alpha\zeta MaxEnt}$ show a significant decrease in the population of the high RMSD structures which are associated to intercalated conformations.

5.2 Self-Consistent MaxEnt refinement by reweighting

The procedure explained in previous section can be slightly modified in order to perform a self-consistent MaxEnt refinement by reweighting MD simulations previously produced. In such case, it is not needed anymore to use a stochastic gradient descent since the gradient can be now exactly estimated when all the trajectory has been scanned. Notice that in this framework, a single step of the minimization

procedure consists in scanning all the trajectories for all the considered systems in order to compute the reweighted averages and then the gradient. The update rule is reported in Subsec. 2.5.1. Interestingly, this procedure can be used exploiting the same PLUMED input files used when performing the refinement on-the-fly (with the addition of the `REWEIGHT BIAS` option), but using the PLUMED `DRIVER` combined with the `--multi` option, which instruct the software to analyze previous performed trajectories in a multi replica approach. However, when trying to use this procedure to repeat the same refinement done in previous section, we were not able to obtain satisfactory results. We believe that the main limitation is again to be searched in the exhaustiveness of the sampling of the available trajectories. The effect is clearly enhanced when the posterior distribution is very different from the prior one. In the worse case it could be even impossible to find the correct Lagrangian multipliers by reweighting. In general, when multiple systems are involved, it might be necessary to perform several iterations. Notice that the time to perform an iteration depends on the number of systems involved and on the number of frames present in each trajectory. It is then possible that in some cases it would be more time-convenient to perform a self consistent refinement on-the-fly rather then by reweighting.

5.3 Mapping ^3J scalar couplings MaxEnt corrections to Gromacs force-field

As already discussed in the thesis, ^3J scalar couplings are particularly appealing to be used in a force-field refinement context. This is due to the fact that the forward model used to back-calculate ^3J scalar couplings from molecular simulations is very similar to the functional form used to model dihedral angles in standard Amber force-field (See Fig. 3.1). We here show how to implement the maximum entropy corrections obtained before in Sec. 5.1.3 into a Gromacs compatible force-field.

In Chapter 2 we showed that the effect of a maximum entropy correction is to add a linear bias to the underlying potential energy function. The total potential energy is then the one derived in Eq. 2.7 which we report here:

$$V_{ME}(\boldsymbol{q}) = V_0(\boldsymbol{q}) + k_B T \boldsymbol{\lambda} \cdot \boldsymbol{s}(\boldsymbol{q}) \ . \tag{5.1}$$

In the particular case of ^3J scalar couplings, the forward model $s(\boldsymbol{q})$ is given by the Karplus formula in Eq. 4.1 which we report here with some notation changes needed to simplify the notation. In particular we will: call the forward model $J(\theta)$ to remind that we are referring to ^3J scalar couplings, rename the

parameters A, B, C and D to k_1, k_2, K_3 and k_4 and drop the dependence on $\boldsymbol{q}$ from the dihedral angle θ. The simplified forward model is then:

$$J(\theta) = k_1 \cos^2(\theta + \phi_\theta) + k_2 \cos(\theta + \phi_\theta) + k_3 \cos(\theta + \phi_\theta) \sin(\theta + \phi_\theta) + k_4 \,. \quad (5.2)$$

Notice that Eq. 5.2 is not anymore written in vectorial formalism, and hence is referred to just one of the modified dihedral angles. The potential in Eq. 5.2, multiplied by the corresponding Lagrangian multiplier, must be then summed to the potential energy of Gromacs, which is parametrized as:

$$V_0(\theta) = \sum_n k_\theta^{FF} \left(1 + \cos(nx - \phi_\theta^{FF})\right), \; n = 1, 2, 3 \quad (5.3)$$

where n represent the multiplicity of the cosine function. We used the apex FF to emphasize that those parameters are the ones used in the un-refined force-field. In order the two function to be summed, we need to first write Eq. 5.2 as a series of cosine function only with proper multiplicity and then sum all the terms with same multiplicity. To this aim, the following trigonometric equalities have been used:

$$\cos^2(x) = \frac{\cos(2x) + 1}{2} \quad (5.4)$$

$$2 \sin(x) \cos(x) = \sin(2x) = \cos\left(2x - \frac{\pi}{2}\right) \quad (5.5)$$

$$\begin{cases} A \cos(x + \phi_A) + B \cos(x + \phi_B) = K_{norm} \cos(x + \phi_B + t) \\ K_{norm} = \sqrt{A^2 + B^2 + 2AB \cos(\phi_A - \phi_B)} \\ t = atan2\left(\frac{A \sin(\phi_A - \phi_B)}{K_{norm}}, \frac{A \cos(\phi_A - \phi_B) + B}{K_{norm}}\right) \end{cases} \quad (5.6)$$

where $atan2$ is the "2-argument arctangent" and is defined as the angle in the Euclidean plane, given in radians, between the positive x-axis and the ray to the point $(x, y) \neq (0, 0)$. The $atan2$ function is already implemented in most of the common programming languages. Using these equalities we can rewrite Eq. 5.2 as:

$$J(\theta) = k_2 \cos(\theta + \phi_\theta) + \frac{1}{2}\sqrt{k_1^2 + k_3^2} \cos\left[2\theta + 2\phi_\theta - atan2\left(\frac{k_3}{\sqrt{k_1^2 + k_3^2}}, \frac{k_1}{\sqrt{k_1^2 + k_3^2}}\right)\right] \quad (5.7)$$

where terms with multiplicity $n = 1$ and $n = 2$ are now clearly separated.

From this equation is also clear, if needed, that Karplus equations in the form of Eq. 5.2 give additional contribution up to multiplicity $n = 2$. Eq. 5.3 and Eq. 5.7 can now be summed "multiplicity-wise". We report here, as an example, the summation for $n = 1$.

The force-field term with $n = 1$ in Eq. 5.3 is $k_\theta^{FF} \cos\left(\theta + \phi_\theta^{FF}\right)$ and must be summed with the $n = 1$ term of Eq. 5.7 which is $k_2 \cos(\theta + \phi_\theta)$. Using Eq. 5.6 we obtain:

$$\begin{gathered} k_\theta^{FF} \cos\left(\theta + \phi_\theta^{FF}\right) + k_2 \cos(\theta + \phi_\theta) = K_{norm}^1 \cos\left(\theta + \phi_\theta^{FF} + h\right) \quad , \; where \\ K_{norm}^1 = \sqrt{k_2^2 + \left(k_\theta^{FF}\right)^2 + 2\cos(\phi_\theta - \phi_\theta^{FF}) k_2 k_\theta^{FF}} \\ h = atan2\left(\frac{k_2 \sin(\phi_\theta - \phi_\theta^{FF})}{K_{norm}^1}, \frac{k_2 \cos(\phi_\theta^{FF} - \phi_{FF}) + k_\theta^{FF}}{K_{norm}^1}\right) \quad . \end{gathered} \tag{5.8}$$

At this point, it is possible to modify the Gromacs force-field by using K_{norm}^1 in place of k_θ^{FF} and $\phi_\theta^{FF} + h$ in place of ϕ_θ^{FF}. Terms with $n = 2$ can be treated in a similar way. A practical example, can be found on SRNAS group Github page ([https://github.com/srnas/ff/blob/nmr-corrections/amber_na.ff/ffbonded.itp||Github]), where we used this procedure to implement the Maximum Entropy corrections into the Gromacs RNA force field.

5.4 Discussions

In chapter 2 we introduced a framework to enforce on the fly noisy data from bulk experiments on molecular dynamics simulations. In the first part (see Sec. 2.2) we discussed the case of experiments without noisy tolerance. This procedure is completely equivalent to the MaxEnt procedure discussed by Chodera and Pitera [33] and share many similarities with the experimentally directed simulation (EDS) introduced by White and Voth [69]. In particular, the only difference between the implementation of the MaxEnt procedure used in this thesis and EDS is that we here used a different optimization procedure to find the Lagrangian multipliers (see 2.5.3). In section 2.3 we extend the previous approach so as to take into account experimental uncertainties. Several Bayesian approaches have been discussed to model experimental errors in similar contexts (see e.g. [45–47, 102, 103]). Methods have been described to reweight a pre-computed ensemble of structures so as to match experimental averages [44, 45, 47]. In the proposed formulation, we applied the MaxEnt procedure on an extended system where fictitious variables are introduced that take into account the discrepancy between theory and experiment. A suitably chosen prior distribution for these variables allows one to control the ac-

curacy of the fitting and to embed in the calculation the confidence in the original force field.

The procedure is iterative and is completely encoded in the update rule stated in Eq. 2.34. It is important to notice that a similar equation could be obtained using theoretical approaches different from the one introduced in this thesis in Sec. 2. For instance, one could decide to maximize the posterior as a function of the residuals ϵ as it is done in Ref. [45], instead of computing their average value. More comments on this analogy can be found in Appendix A.2.

We notice that other methods have been proposed in the past to model noisy data within the MaxEnt framework. For instance, Chen and Rosenfeld [35] have proposed to introduce a Gaussian prior on the Lagrangian multipliers which will essentially constraints Lagrangian multipliers to be bounded. The Laplace prior on the additional variables used here has a similar effect, and allows the range of values for the Lagrangian multipliers to be explicitly controlled.

An alternative formulation of the MaxEnt procedure discussed here can be obtained by replacing the time averages with averages performed on an ensemble of molecular dynamics simulations [37, 38]. Replica averaging only converges to MaxEnt when an infinite number of replicas is simulated [39, 40] and implies an intrinsic statistical error in the averages when used with a finite number of replicas. Replica formalism has been extended so as to take into account experimental errors [45, 46]. In this context, we preferred to use an iterative procedure since it allows Lagrangian multipliers to be estimated on the fly. The statistical error that in our procedure arises from the finite length of the simulation can be assessed by standard blocking analysis.

The tests that we performed on model systems (see section 2.4) allow to easily understand the effects of the chosen parameters on the resulting ensembles. In particular, the variance of the prior distribution used for the additional variables can be used to tune the relative weight of the original model and of the enforced experimental data. A Laplace prior for these variables allows for outliers to be tolerated.

We then applied the method to an important open problem, that is the refinement of a force field in order to reproduce available NMR data for RNA oligomers. At first we use our method to enforce all the ^3J scalar couplings available for the four RNA nucleosides. Since the free-energy landscape of nucleosides have significant barriers, we combined the approach with an enhanced-sampling method based on multiple replicas. This can be straightforwardly done in our formulation since Lagrangian multipliers can be estimated on-the-fly in the unbiased replica and instantaneously transferred to the biased ones. The results display a significantly

reduced RMSE with respect to experimental data when compared to the original Amber force field. This is expected, since the validation is made against the same dataset used for the training. However, this confirms that the methodology converges to the correct result also in a non trivial model system. We also observe that the employed couplings are unevenly distributed along the RNA backbone. If desired, one could associate a lower value of σ to the individual couplings that are considered more relevant so as to increase their weight in the fitting procedure.

The method is then applied to the self-consistent force-field fitting for two RNA nucleotides (A and C), employing a variety of data measured for several systems (A and C nucleosides, as well as ApA, ApC, CpA, and CpC dinucleosides monophosphate). Also here, the procedure takes implicitly advantage of the on-the-fly transferability of the Lagrangian multipliers. Our approach reminds the spirit behind the restrained ESP charge model [104], where equivalent atoms are restrained to have equivalent charges. This is translated here in having same correcting potentials on chemically equivalents dihedrals independently of their position in the sequence. Notice that using a self-consistent procedure where several terms are restrained to be identical, effectively reduces the flexibility of the resulting force field and implicitly decreases its capability to match the experimental data. For instance, in the case of a duplicated term in a single simulation (e.g., the χ angle in an adenine which appears twice in the ApA dinucleoside monophosphate), our approach is only controlling the sum of the two scalar couplings and not their individual values. In our specific application, the number of independent parameters in the force field is 16, which should be compared with 78 independent experimental data. In this respect, it is important to notice that in this application the calculation of the RMSE, which depends also on the non-explicitly controlled observables, allows for a rigorous cross validation of the method.

The functional form of the corrections derived here, which is proportional to the Karplus equations, is compatible with the one of dihedral potentials. This suggests the use of scalar coupling data as an alternative to quantum chemistry calculations for force-field parametrization or as a refinement tool on top of quantum-chemistry derived torsions. One might be concerned about the fact that corrections developed to match experimental data on small systems are not necessarily portable to larger systems. However, it must be observed that the standard procedure used in the Amber force field is to refine dihedral potentials based on quantum chemistry calculations performed on small fragments, whose typical size is often below the size of the systems considered in this work [91, 105]. As a validation, we performed a reweighting of previously published trajectories for two RNA tetranucleotides (AAAA and CCCC). In spite of their apparent simplicity these

unstructured oligomers are not described properly by any of the current versions of the Amber force field [9]. Our results show that the corrections are portable and significantly improve the description of these tetranucleotides. The resulting RMSEs are below 1 Hz, which is the typical difference between alternate Karplus equations. The development of a force field that consistently describes all nucleotides and dinucleosides, as well as its validation on tetranucleotides and larger systems, is left as a subject for a future investigation.

In conclusion, we introduced a novel procedure that allows experimental errors to be explicitly modeled in a MaxEnt framework. The method is applied to the self-consistent force-field fitting on RNA systems. Results indicate that the obtained force-field corrections are portable and suggest a new paradigm for empirical force-field refinement.

Chapter 6

RNA Force-Field refinement using arbitrary functional forms

We show here an application of the reweighting procedure introduced in Section 3.3 to the refinement of the standard Amber force-field. In order to enhance the transferability of the corrections, we included multiple systems and used different types of experimental data. We devised our procedure to only re-parametrize torsional parameters, although it could be used to optimize other terms. To this aim we choose the basis functions to be of the same functional type of those used in Amber force field. In particular, our correcting potential looks like:

$$V_{corr} = \sum_{t \in \{torsions\}} \sum_{i=1}^{N_t} \sum_{n=1}^{3} \lambda_{1tn} \cos\left(n\phi_{ti}\right) + \lambda_{2tn} \sin\left(n\phi t_{ti}\right), \tag{6.1}$$

where $torsions = \{\alpha,\ \beta,\ \gamma,\ \delta,\ \epsilon,\ \zeta,\ \chi_{Pur},\ \chi_{Pyr}\}$ is the set of torsion types feeling the correcting potential, N_t is the number of nucleotides involved in the refinement, $\lambda_{1tn}(\lambda_{2tn})$ is the weight associated to the cosine (sine) with multiplicity n relative to the torsion type t and $\phi_{ti}$is the torsion of type t in the nucleotide i. The systems considered in our refinement procedure were 4 RNA tetranucleotides (AAAA, CCCC, UUUU and GACC) and 2 RNA tetraloops (GAGA and UUCG). Tetranucleotides simulation data were taken from Ref. [106] while tetraloops simulation from Ref. [12]. All systems were simulated using the ff99bsc0 + χ_{OL3} Amber force field with corrections to van der Waals oxygen radii ([107]) and using the OPC water model ([108]). We will simply refer to this force field as Amber. Data for tetranucleotides involve both NOE and scalar couplings NMR measurements. For tetraloops we require the native state to be the most populated one. Native configurations were arbitrarily chosen as those with `eRMSD` ([109])<0.8 from the X-ray reference structure (figure 6.5 first line). Forward models and the applied constraints for all systems are summarized in Tab. 6.1. Parameters $\omega_1, \omega_2, \omega_3$

represent the weights of each system in the error function used in the fitting procedure and should have units corresponding to inverse variances. Following the introduced interpretation we heuristically choose these values in order to reproduce the expected experimental error of a given experiment type.

System	Forward Model	Constraint	Error Function
AAAA CCCC UUUU GACC	NOE		
	$\langle O_i \rangle = \langle \frac{1}{d_i^6} \rangle^{-\frac{1}{6}}$	$d_{i,min}^{exp} \leq \langle O_i \rangle \leq d_{i,max}^{exp}$	$\omega_1 \max\left(\langle \frac{1}{d_i^6} \rangle^{-1/6} - d_{i,max}^{exp}, 0\right)^2 + +\omega_1 \max\left(d_{i,min}^{exp} - \langle \frac{1}{d_i^6} \rangle^{-1/6}, 0\right)^2$
	^{3}J Couplings		
	$\langle O_i \rangle = A\cos(2x) + B\cos(x) + C = \langle {}^3J_i \rangle$	$\langle O_i \rangle = O_i^{exp} = \langle {}^3J^{exp} \rangle$	$\omega_2 \left(\langle {}^3J_i \rangle - {}^3J_i^{exp}\right)^2$
ccGAGAgg ccUUCGgg	Native Fraction		
	$\langle O_i \rangle = \langle p_f^i(t) \rangle$; $p_f^i(t) = \begin{cases} 1 & \text{eRMSD} \leq 0.8 \\ 0 & otherwise \end{cases}$	$\langle p_f^i(t) \rangle \geq 0.5$	$\omega_3 \max\left(0, \log(0.5) - \log\left(\langle p_f^i(t) \rangle\right)\right)^2$

Table 6.1: Systems composing the training set in the the force-field refinement procedure. Employed data type for each system are reported (e.g. NOE) with the relative forward model used to back-calculate them from simulations data. In the case of NOE data, index i runs over the proton pairs for which NOE data are available. NOE and ^3J couplings experimental data were taken from Refs. [8, 53, 110]. In the case of ^3J couplings, the index i runs over all torsions for which scalar couplings are available. In the case of tetraloops, index i refers to the system for which the folded fraction is computed (i.e. ccGAGAgg or ccUUCGgg)

The error functions reported in Tab. 6.1 are relative to a single experimental data point. In order to fit all the experimental data, the single error functions must be summed over all the available data. The resulting total error function can be then written as:

$$
\begin{aligned}
E = \omega_1 \sum_{i=1}^{N_{NOE}} & \left[\max\left(\langle \frac{1}{d_i^6} \rangle^{-1/6} - d_i^{max}, 0 \right)^2 + \max\left(d_i^{min} - \langle \frac{1}{d_i^6} \rangle^{-1/6}, 0 \right)^2 \right] + \\
+ \omega_2 & \sum_{i=1}^{N_{couplings}} \left(\langle {}^3J_i \rangle - {}^3J_i^{exp} \right)^2 + \\
+ \omega_3 & \left[\max\left(0, \log(0.5) - \log\left(\langle p_f \rangle\right)_{\texttt{GAGA}}\right)^2 + \max\left(0, \log(0.5) - \log\left(\langle p_f \rangle\right)_{\texttt{UUCG}}\right)^2 \right] + \\
+ \alpha & \boldsymbol{\lambda}^2
\end{aligned}
\tag{6.2}
$$

The last term, as explained in 3.3.1, is a regularization term needed to prevent overfitting. In order to find the best estimate of the parameter α in Eq. 6.2, we used the k-fold cross validation procedure, explained in Subsec. 3.3.1, with $k = 3$. We divided the training set in 3 blocks, each of which containing data from different experiment type. In Tab. 6.2 we summarize how the dataset was divided.

Block n.	Data Type
1	^{3}J Couplings
2	NOE
3	Native Fraction

Table 6.2: Training set splitting scheme. The training set is divided in 3 blocks. Each block contains data resulting from different experiments. Block 1 contains experiments about ^{3}J Couplings only. Block 2 contains data coming from NOE experiments. Block 3 contains data relative to the stability of the tetraloops folded structure.

By performing a grid search on α we performed several minimizations, following the k-fold method prescription. For each trial value of alpha the error function is computed on the cross-validation set. Results for all the tested α values are reported in Fig. 6.1

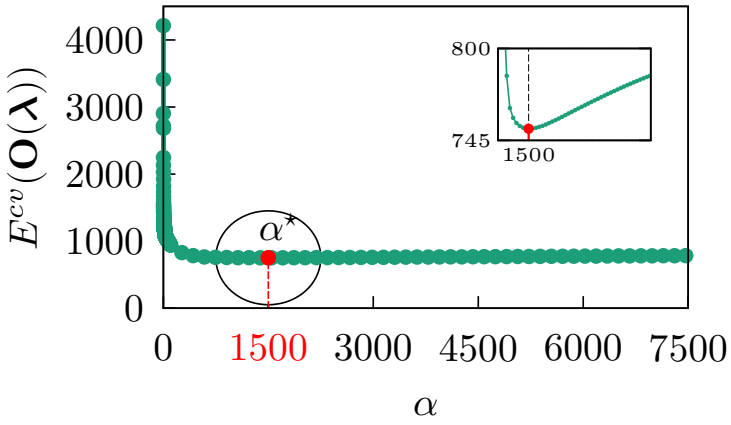


Figure 6.1: Cross-validation procedure. The error functions is evaluated on the validation set for different trial values of α following the procedure introduced in Subsec. 3.3.1.

In Fig. 6.1, we can see that the cross-validation error function has a minimum for a value of $\alpha^\star = 1500$. This means that for the given dataset, 1500 represent the optimal value of α giving the best balance between overfitting and predictiveness on different data and, presumably, systems not seen in the training set. The optimal value $\alpha^\star = 1500$ can be then used to train a potential with which one can perform new simulations on systems different from those used in the training set.

Notice that since we are enforcing different data on multiple systems, mutual compatibility is not *a priori* guaranteed. In other words, minimizing the error function does not guarantee that the enforced constraints are satisfied on all systems. Constraints on individual systems must be then checked after the final reweight procedure. We first checked the enforced constraints on the systems in the training set. We will call $\texttt{Amber}_{\texttt{RW}}$ the ensemble obtained by reweighting, without resampling, the reference `Amber` force-field using the optimal weights obtained by re-fitting all the training set (without leaving out any data) using the optimal value of $\alpha^\star = 1500$. In the case of tetranucleotides we computed the value of the enforced ^3J scalar couplings and `NOE` distances. Similarly to the analysis made in Subsec. 5.1.4, we validate the estimated corrections by computing the $RMSE$ and the percentage of violations for ^3J couplings and `NOE` respectively. Results are summarized in Fig. 6.2.

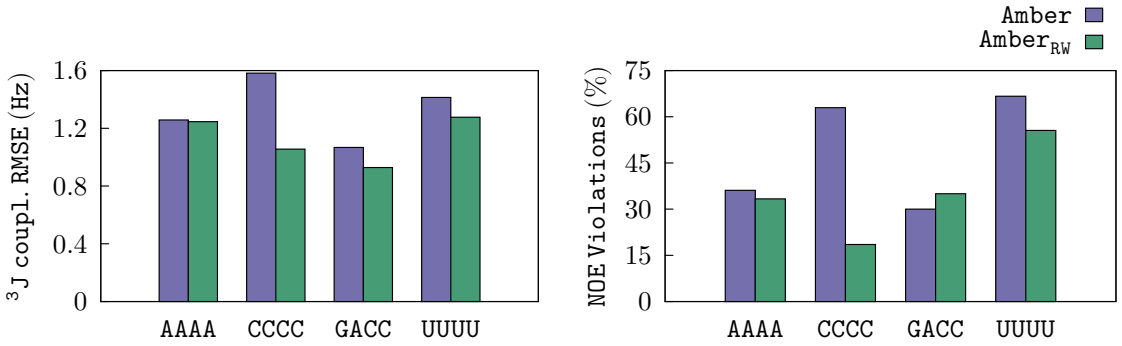


Figure 6.2: ^3J scalar coupling RMSE and NOE violations for RNA tetranucleotides. For both $RMSE$ and percentage of violations, the lower the better.

Fig. 6.2 shows that in the case of ^3J couplings, the reweighted ensemble (green bars in Fig. 6.2) better reproduces experimental data for all the considered tetranucleotides when comparing it to the unrefined `Amber` force field (purple bars in Fig. 6.2), although in some case the improvement is very limited. In any case, all the resulting RMSEs are compatible with the expected error for the forward model used to compute scalar couplings. Looking at the percentage of violated `NOE`, we can see that, except for the `GACC` tetranucleotide, the new ensemble better reproduced `NOE` experimental distances compared to the original `Amber` force field. The improvement is particularly visible for `CCCC`.

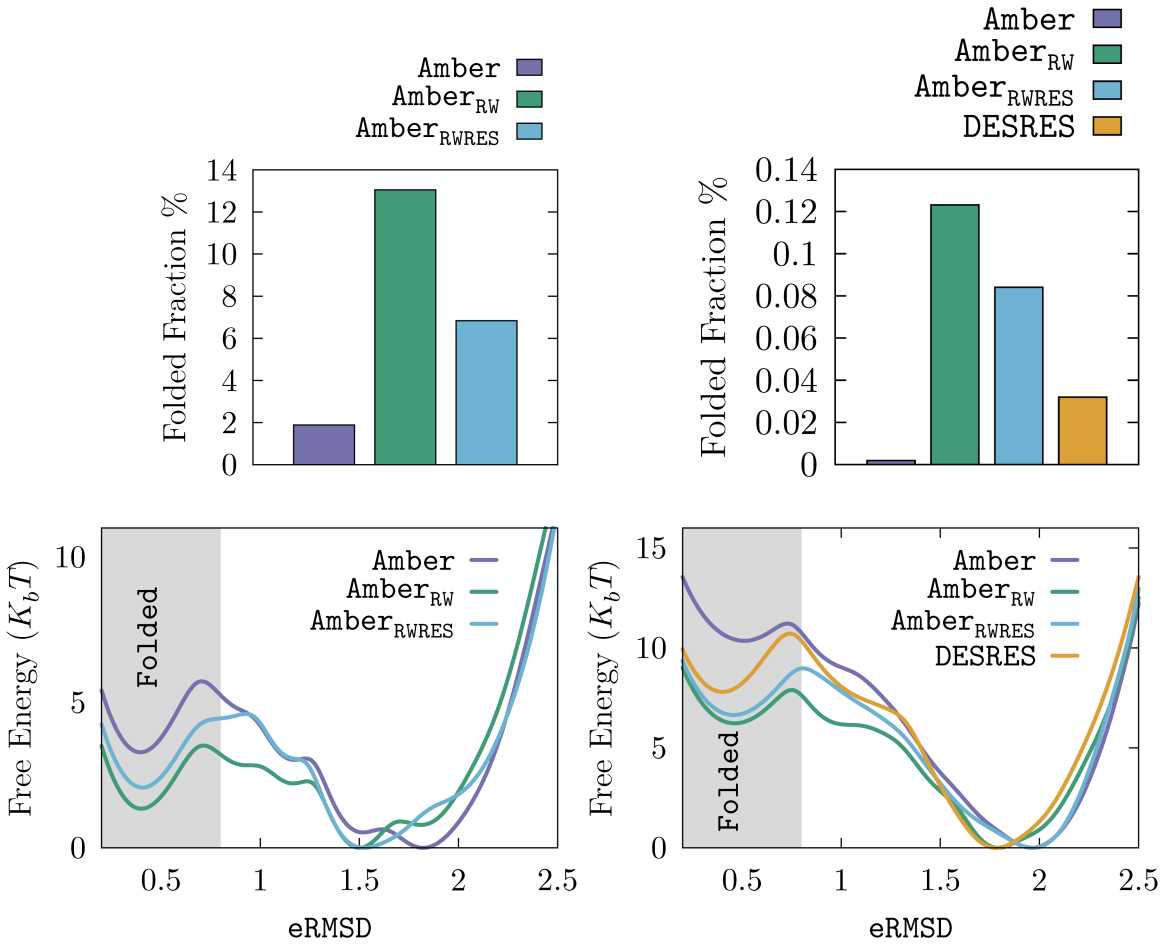


Figure 6.3: RNA ccGAGAgg (left) and ccUUCGgg(right) tetraloops. Fraction of folded structures (top) and free energy surface (bottom) with different force-field parametrizations. Unrefined Amber force-field (Amber)in purple, reweighted Amber force-field (Amber$_{RW}$) before resampling in green, refined Amber force-field (Amber$_{RWRES}$) after resampling in light blue, D. E. Shaw ([111])(DESRES) force field in yellow.

As regards the 2 tetraloops, we computed the folded fraction and the free energy as function of the eRMSD from native structure, both with the unrefined Amber force field and with the reweighted one Amber$_{RW}$. Results are reported in Fig. 6.3. As we can see, the fraction of folded structure is significatively increased for both systems (first row of Fig. 6.3). The introduced correction indeed reduces the relative weight of some of the unfolded structures observed in the Amber ensemble.

All the results shown before were obtained performing a reweighting of a given simulated ensemble. The statistical accuracy of the reweighting procedure depends however on the distance between the unrefined ensemble and the reweighted one. In case the two ensemble are too different, reweighting might be inefficient since

t	α	β	γ	δ	ϵ	ζ	χ_{Pur}	χ_{Pyr}
$\sum\cos(t)$	0.028743	−0.118088	0.369240	−0.068875	−0.061779	−0.083563	−0.007880	0.067727
$\sum\cos(2t)$	−0.135683	0.199287	−0.269494	0.035808	0.086680	0.091654	−0.054599	−0.116106
$\sum\cos(3t)$	0.122662	−0.064958	−0.001949	0.109930	0.042434	0.061681	0.016977	0.037457
$\sum\sin(t)$	0.045960	−0.085707	0.080645	−0.011406	0.038967	0.220101	−0.086418	−0.014430
$\sum\sin(2t)$	0.064641	0.008920	0.190771	−0.109477	0.054064	0.150732	−0.073151	−0.046067
$\sum\sin(3t)$	0.032150	0.016087	0.020993	0.051061	−0.077758	0.015956	0.143994	0.140334

Table 6.3: We report here the values of the $\lambda_{\texttt{RWRES}}$ coefficients obtained with the proposed procedures, and used to perform the $\texttt{Amber}_{\texttt{RESRW}}$ simulation. The sum in the first column is meant to be taken on all the torsions of type t present in the considered system. The quantity $K_B T\sqrt{6\frac{\sum_{i=1}^{48}\lambda_i^2}{48}}$ can be used as an indicator of the average corrections per torsion. In this case we obtain a value of 0.68 $\frac{kJ}{mol}$.

there may be very few frames in the original ensemble with a significant weight. A rough estimate of the reweighting accuracy is given by the Kish's effective sample size $n_{eff} = \frac{\left(\sum_{i=1}^{N_{frames}} w_i\right)^2}{\sum_{i=1}^{N_{frames}} w_i^2}$. A deep analysis of reweighting performance, together with a critical comparison between reweighting and restraining methods can be found in Ref. [16]. The value of n_{eff} is bounded to the range $1 \leq n_{eff} \leq N_{frames}$ and indicates how many frames, among all the available ones, are effectively used in the reweighting procedure. The value of the Kish effective size can be controlled by adjusting the regularization parameter α. Although there is no rule of thumb on a suitable value of n_{eff} which ensures good reweighting performance, in the reported results the chosen value of $\alpha^\star$ guarantees at least 30% of effective samples, for each system.

In order to properly validate the predictiveness of the force field corrections obtained using the reweighing procedure, one should perform a new simulation (resampling) using these corrections. We performed a resampling for the tetraloops only. The conformational ensemble for the tetranucleotides obtained in the Amber simulation was sufficiently overlapping with the experimental one ([106]). We will call $\texttt{Amber}_{\texttt{RWRES}}$ the ensemble obtained after resampling conformational space using the optimal values of λ (see Tab. 6.3) obtained with $\alpha^\star = 1500$.

Results for tetraloops are reported in Fig. 6.3 and Fig. 6.5. Looking at first row of Fig. 6.3 we can compare results obtained by reweight only ($\texttt{Amber}_{\texttt{RW}}$) with the ones obtained after resampling ($\texttt{Amber}_{\texttt{RWRES}}$). We can notice that although $\texttt{Amber}_{\texttt{RWRES}}$ increases the stability of both tetraloops by a factor of 4 and 44 for `ccGAGAgg` and `ccUUCGgg` respectively, the improvement is not as big as the one obtained by reweighting only. This is a consequence of both the inaccuracy of the reweighting due to poor sampling of the reference ensemble and a small unavoidable overfitting on the specific conformations produced during the reference Amber simulation. In any case, thanks to the regularization term, the correction

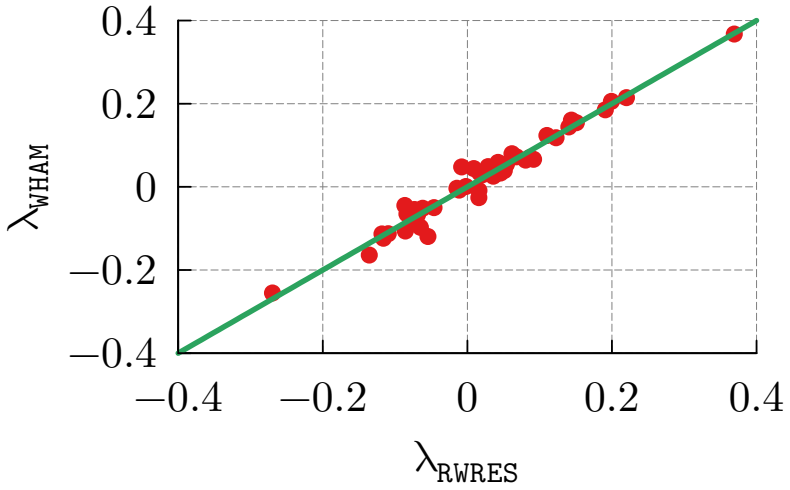


Figure 6.4: Comparison between the coefficients $\lambda_{\texttt{RWRES}}$ (reported in Tab. 6.3) and the $\lambda_{\texttt{WHAM}}$ coefficients obtained by performing the reweighting procedure on the WHAM-combined trajectory. Notice that with the formalism used in this thesis the coefficients of λ are unitless and they must be multiplied by the value of k_bT when used for reweighting.

is limited, the Kish sample size is relatively large, and the trends observed in the reweight calculations are the same of those observed in the resampling calculation. At an early stage we tried the same procedure without including any regularization term and, whereas the reweight procedure was reporting a high stability for the tetraloops, the resampling results were *worst* than those obtained with the original force field, indicating that our parameters were highly overfitted on the conformations sampled in the specific run.

As a further check, we computed the native populations by combining the `Amber` and the $\texttt{Amber}_{\texttt{RESRW}}$ simulations using a WHAM procedure ([112]) in order to take into account both the bias introduced by the meta dynamics potential and the difference between the two employed force fields. The resulting populations corresponding to both `Amber` and $\texttt{Amber}_{\texttt{RESRW}}$ force fields were very close to those reported above. We notice that, as suggested in Ref. [113] one might reiterate the fitting procedure.

In our case, the λ coefficients obtained by fitting on the WHAM simulation using the same regularization parameter $\alpha = 1500$ were very close to those used in the $\texttt{Amber}_{\texttt{RESRW}}$ simulation ($RMSD = 0.02$). This suggests that further iterations are not required. Comparison between λ coefficients and the values used to perform the $\texttt{Amber}_{\texttt{RESRW}}$ are reported in Fig. 6.4.

In Fig. 6.5 we report the dynamic secondary structure ([114]) for selected ensembles for both tetraloops as obtained using all the employed force fields, together with the native structures. When the ensemble is selected to only contain con-

formations where both the stem and the loop are formed (first row), the dynamic secondary structure is highly homogenous and, by construction, consistent with native. This ensemble was selected with the same criterion used to identify the native conformations in the force-field fitting procedure.

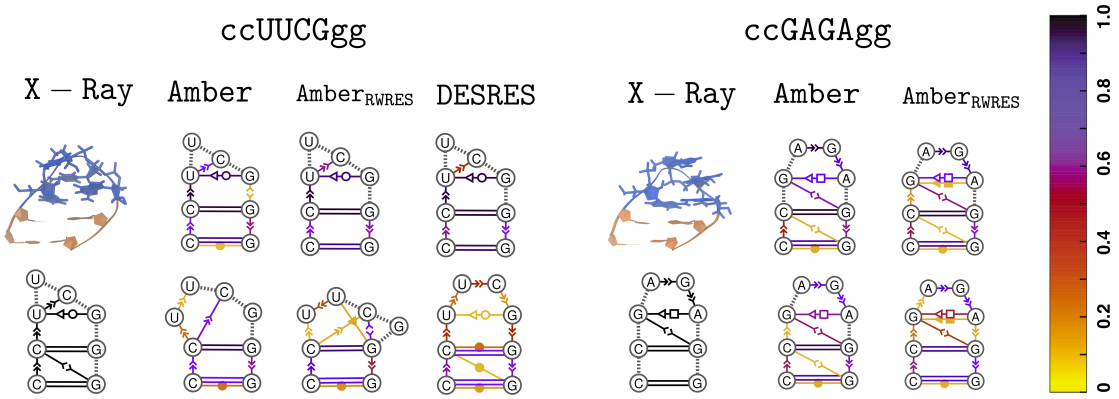


Figure 6.5: First and fifth column show, respectively, ccUUCGgg and ccGAGAgg tetraloops X-ray structures. Both tertiary (first row) and secondary (second row) structures are shown. Secondary structure were obtained with the BARNABA software ([114]). For all the other columns, we show dynamic secondary structure representations obtained with different force fields as indicated in the column name. The color scheme shows the relative number of frames for which the interaction is formed. Structures are sampled from the unbiased distribution in order to remove the effect of the metadynamics bias. In the first row, we report ensembles where the eRMSD of the whole system is within 0.8 from native. In the second row instead we report ensembles where only the stem has eRMSD from native less than 0.8. In this second case the loop portion is then free to assume any conformation allowed by the employed force-field.

Conversely, if we extract ensembles where only the stem is assumed to be formed (second row) we can quantify the capability of the employed force fields to reproduce the native loop structure assuming the stem to be formed. For the GAGA tetraloop, the secondary structure is consistent with the native structure both using the Amber and the Amber$_{RESRW}$ force fields. This indicates that already in the original force field the loop would have the correct structure if the stem is folded. Our correction does not perturb significantly the result. For the UUCG tetraloop, both the original Amber and the Amber$_{RESRW}$ force fields are not capable to reproduce the correct loop structure, indicating that further corrections would be required to this aim. For this system we also report results obtained using the DESRES force field [111]. We notice that the overall stability of the tetra loop

using this force field is lower than the one obtained with the $\texttt{Amber}_{\texttt{RESRW}}$ force field (Fig. 6.3). However, when we select conformations where the stem is formed, the loop displays a partly correct native structure where the trans-sugar/Watson-Crick pair (U3-G6) is observed although with a low population. On the other hand the parallel stacking U3-C5 is not observed. A comparison between the results of the DESRES force field with the force field corrections derived here is however difficult since the DESRES force field modified all the nonbonded interactions, whereas our correction only impact the torsional angles.

6.1 Discussions

The introduced method is based on an existing procedure ([113]) that is here applied to the fitting of an atomistic force field for RNA systems. An important extension presented here is the introduction of a regularization procedure that allows overfitting to be decreased. The method is used to combine data of different type on multiple systems, which is crucial in order to achieve transferable parameters. Since the optimization of the parameters is done with a reweighting procedure, its application in a single iteration as shown here is limited to small corrections, such as dihedral terms or other solute-solute non-bonded terms. Fitting parameters such as partial charges would lead to larger changes in the ensemble, also involving solvent molecules, and might require to apply the procedure in an iterative manner by resampling new conformations at every change of the lambda parameters, in order to progressively shift the simulated ensemble towards the correct one. Especially in this case, it would be convenient to proceed in the refinement using all the performed simulations and combining them with the WHAM method.

The method is very flexible in that arbitrary error functions can be optimized. In this specific case, we used NMR data and population of native structures. Other possible choices for nucleic acid systems could be helical parameters or other structural quantities for which ranges of acceptable values can be identified a priori.

For the investigated systems, we have shown that very small corrections to dihedral angles can perturb significantly the population of the native structure in RNA tetraloops. By only correcting dihedrals we were not able to obtain a force field capable to fold the investigated loops to the native structure with a significant population. However, the resulting populations were improved with respect to the original ones and, for the UUCG tetra loop, higher than those obtained with a recently proposed reparametrization ([111]). By including several systems and

a large number of datapoints, we were able to mitigate the side effects of the corrections in an automatic fashion.

We decided to use as starting point the Amber force field including non-standard modifications on the phosphate oxygens and in combination with the OPC water model. This choice was motivated by the good performance shown by this force field in moderating the population of intercalated structures in RNA tetranucleotides ([106]). However, these modifications might lead to side effects on systems not tested here. The procedure introduced here might lead to better results if based on a more accurate starting point. Since torsional potentials are usually fitted as the last step in force field derivation, we suggest that a final refinement could be performed with the procedure introduced here, including all the desired structural features in the minimized error function.

As a word caution, we would like to point out that before suggesting the derived corrections to be used on new systems they should be validated on a larger set of RNA motifs.

Chapter 7

Conclusions and Perspectives

The subject of this thesis was the development of techniques to enforce NMR solution experiments in molecular dynamics simulations. In Chap. 2 we developed a framework based on the Maximum Entropy principle were experiments resulting from ensemble measurements can be used to complement molecular dynamic simulations (prior). Among the many possible ensembles compatible with the enforced experimental averages, the Maximum Entropy solution finds the least biased one with respect to the prior ensemble. The algorithm builds, on-the-fly, an additional bias potential which is linear in the forward model used to compute the enforced quantity as function of the atomic coordinates. Usage in combination with enhanced sampling methods is also possible. In our applications shown in Chap. 4 and Chap. 5 we combined the proposed method with the RECT enhanced sampling [96]. Overfitting has been addressed with particular care as well as tolerance to outliers in experimental data. The method can be applied on-the-fly during MD simulations or by reweighting previously available trajectories. In Sec. 3.2 the framework is then extended and proposed as a tool to be used in a force field refinement procedure. The proposed strategy consists in simulating several systems in parallel (in the same way as in multi replica simulations) allowing them to share the same corrections on a set of variables considered to be chemically equivalent among all the considered systems. The method is applied to the refinement of state of the art RNA force field by fitting ^3J scalar coupling NMR data on a set of RNA nucleosides and dinucleotides (A, C, ApA, ApC, CpA, CpC). The resulting corrections are validated by reweighting previously available trajectories of RNA tetranucleotides which were not included in the training set. Results show an overall good improvement with respect to the starting Amber force field.

In Sec. 3.3 we proposed a new strategy aimed at overcoming the limitations of the Maximum Entropy approach, in which one is limited to corrections of the same functional form of the forward model. This approach is not anymore based on the

Maximum Entropy formalism and can only be applied by reweighting available trajectories. In this new formulations, any ensemble average can be enforced using arbitrary functional forms for the correcting bias potential. Also in this framework the method is applied on multiple systems in parallel, with proper overfitting treatment, allowing to derive transferable force-field corrections. We applied this method to the refinement of RNA force-field, using both RNA tetranucleotides and tetraloops as training set. The derived corrections were validated by performing new simulations of ccUUCGgg and ccGAGAgg tetraloops systems with the refined force-field. Results show significant improvements on the stability of the considered tetraloops both when comparing to the original Amber force field and when comparing with a recent parametrization of RNA force field[111].

It must be anyway noticed that none of the used parametrization is able to reproduce the experimental stability of tetra loops. This is a challenging task, that will require additional work and improvements.

In perspective, the force field refinement methods introduced here could be applied starting from different non bonded parameters. For instance, one could start from the reparametrization proposed in Ref. [111], from the corrected hydrogen-bond parameters proposed in Ref. [115], or from any other force field. It might also be beneficial to include more experimental information in the form of structural quantities for which ranges of acceptable values can be identified a priori, such as, for instance, helical parameters. The accuracy and transferability of the derived corrections can be further improved by including more and heterogeneous systems in the training set.

Whereas the applications here were limited to RNA oligonucleotides, the introduced methodologies could certainly be applied to other molecular systems for which the accuracy of empirical force field is suboptimal, such as for instance disordered proteins.

In general, we expect that methods based on the combination of experimental data and molecular simulations will find significant application in the future, both to study specific systems for which experimental data are available and to refine empirical force fields.

Appendix A

More on prior error

A.1 Generic error prior

Here we discuss in detail the relationships between the prior on the discrepancy between simulation and experiment $P_0(\epsilon)$, the prior on its variance $P_0(\sigma_0)$, and the expected value of the discrepancy $\xi(\lambda) = \frac{\int d\epsilon P_0(\epsilon) e^{-\lambda\epsilon}\epsilon}{\int d\epsilon P_0(\epsilon) e^{-\lambda\epsilon}}$.

We here assume that ϵ is Gaussian distributed with unknown variance, and introduce a prior $P_0(\sigma_0)$ on its variance:

$$P_0(\epsilon) = \int_0^\infty P_0(\sigma_0) \frac{e^{-\frac{\epsilon^2}{2\sigma_0^2}}}{\sqrt{2\pi}\sigma_0} d\sigma_0 \tag{A.1}$$

From this equation we can compute $\xi(\lambda)$ as

$$\xi(\lambda) = \frac{\int d\epsilon e^{-\lambda\epsilon} P_0(\epsilon)\epsilon}{\int d\epsilon e^{-\lambda\epsilon} P_0(\epsilon)} = -\frac{\partial}{\partial\lambda} \log \int_{-\infty}^\infty d\epsilon e^{-\lambda\epsilon} P_0(\epsilon)$$

This can be written equivalently using $P_0(\sigma_0)$ resulting in:

$$\xi(\lambda) = -\frac{\partial}{\partial\lambda} \log \left[\int_0^\infty d\sigma_0 \frac{P_0(\sigma_0)}{\sqrt{2\pi}\sigma_0} \int_{-\infty}^\infty d\epsilon e^{-\lambda\epsilon - \frac{\epsilon^2}{2\sigma_0^2}} \right] = -\frac{\partial}{\partial\lambda} \log \left[\int_0^\infty d\sigma_0 P_0(\sigma_0) e^{\frac{\lambda^2\sigma_0^2}{2}} \right] \tag{A.2}$$

Clearly, there is a large degree of arbitrariness in the choice of the prior $P_0(\sigma_0)$. We here introduce a class of functions $P_0(\sigma_0; \sigma, \alpha)$ defined as

$$P_0(\sigma_0; \sigma, \alpha) = C\sigma_0^\alpha e^{-\frac{\sigma_0^2(\alpha+1)}{2\sigma^2}} \tag{A.3}$$

Here $C = 2^{\frac{1}{2}(\alpha-1)} \left(\frac{\sigma^2}{\alpha+1}\right)^{\frac{\alpha+1}{2}} \Gamma\left(\frac{\alpha+1}{2}\right)$ is a normalization factor, σ is a parameter that can be interpreted as the typical expected error, and $\alpha > -1$ is a parameter that determines how peaked is the prior. In the limit of $\alpha \to \infty$ the prior turns into a Dirac delta function,

resulting in a Gaussian priors on ϵ. The parameter αhere is related to the parameter k, introudced in the main text, by the relation $\alpha = 2k - 1$. In the general case one can find by straightforward manipulation that

$$\xi(\lambda; \sigma, \alpha) = -\frac{\lambda\sigma^2}{1 - \frac{\lambda^2\sigma^2}{1+\alpha}}$$

Here it is possible to see that $\xi(\lambda; \sigma, \alpha = \infty) = -\lambda\sigma^2$, which corresponds to a Gaussian prior on ϵ, and $\xi(\lambda; \sigma, \alpha = 1) = -\frac{\lambda\sigma^2}{1-\frac{\lambda^2\sigma^2}{2}}$, which corresponds to a Laplace prior on ϵ. A plot for different choices of α is provided in figure A.1 showing how α affects the priors.

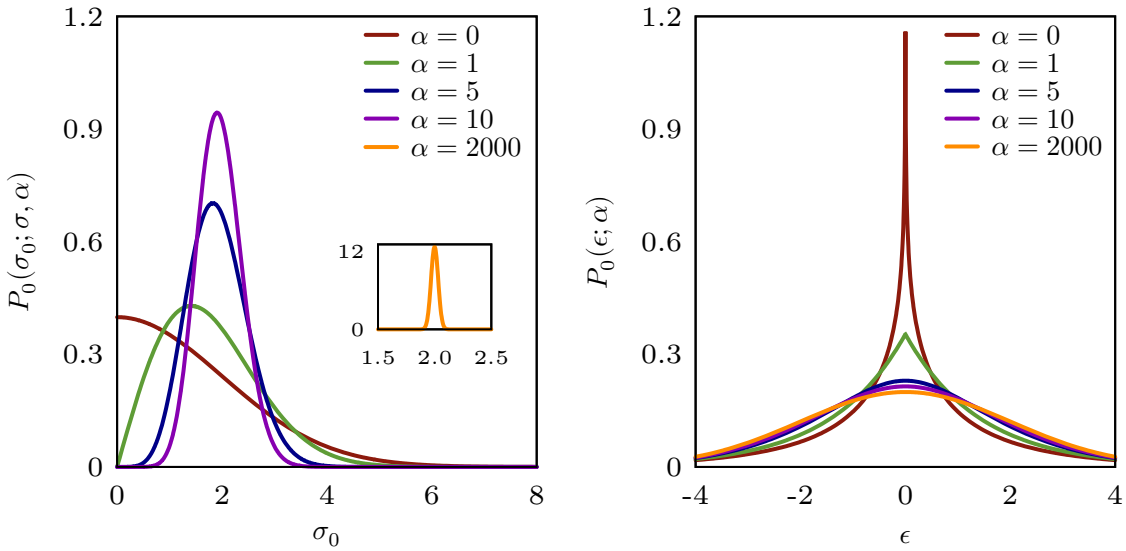


Figure A.1: Prior functions both on the variance σ_0 and on the additional variable ϵ for a set of chosen values for α. All the priors have the same value for the typical error $\sigma = 2$. Left panel shows the family of priors $P_0(\sigma_0; \sigma = 2, \alpha)$. The prior $P_0(\sigma_0; \sigma = 2, \alpha = 2000)$ is represented in an inset with a different scale and shows that for large values of α the prior converges to a delta function centered in $\sigma = 2$. Right panel shows the corresponding family of priors on ϵ obtained through equation (A.1).

A.2 Maximum a posteriori vs Maximum Entropy

In the main text we argued that different theoretical frameworks would lead to identical algorithms whenever they result in the same function $\xi(\lambda)$, which represents the tolerated discrepancy between experiments and simulations for a given value of the Lagrangian multiplier λ. Here we cover the relationship between our Maximum Entropy (MaxEnt) procedure and the Maximum a posteriori (MAP) approach of ref. [45]. A maximum a posteriori framework would lead to the definition $\xi(\lambda) = \arg\max_{\epsilon} \left[P_0(\epsilon)e^{-\lambda\epsilon}\right]$. For a Gaussian prior $P_0(\epsilon)$ the posterior $P(\epsilon) \propto P_0(\epsilon)e^{-\lambda\epsilon}$ is also Gaussian and its mode and average coincide. Thus, the iterative procedure introduced in this thesis would lead to a result equivalent to that of the method discussed in Ref[45]. However, for a non-Gaussian prior this is not true. Since $\xi(\lambda)$ is the only expression entering directly in the minimization procedure, we argue that different theoretical frameworks would lead to identical results whenever they result in the same function $\xi(\lambda)$. In MAP one aims at finding the optimal probability density maximizing a suitable posterior functional. In Hummer[45] such functional is defined as:

$$\mathcal{P}[p(\boldsymbol{x})|data] \propto \exp\left(-\theta \int d\boldsymbol{x} p(\boldsymbol{x}) \ln \frac{p(\boldsymbol{x})}{p_0(\boldsymbol{x})}\right) \mathcal{P}[data|p(\boldsymbol{x})]$$

where $\mathcal{P}[data|p(\boldsymbol{x})]$ represent the likelihood function of the observed data. To keep the notation consistent with the one we used in our approach we denote as $\xi = f_{exp} - \int d\boldsymbol{x} p(\boldsymbol{x}) f(\boldsymbol{x}) = f_{exp} - \langle f \rangle$ the discrepancy between the experiments and the simulations. We then notice that the likelihood $\mathcal{P}[data|p(\boldsymbol{x})]$ only depends on ξ and we thus write it as $P(\xi)$. For simplicity we are considering a single experiment here. One can then find the optimal probability density $p^{(opt)}(\boldsymbol{x})$, which maximizes the posterior, for a generic likelihood functional $\mathcal{P}(\xi)$:

$$p^{(opt)}(\boldsymbol{x}) \propto p_0(\boldsymbol{x}) \exp\left(-\frac{f}{\theta} \frac{\partial \ln \mathcal{P}[data|\xi]}{\partial \xi}\right)$$

This expression is identical to the probability density that would be sampled with a MaxEnt procedure $P(\boldsymbol{x}) \propto P_0(\boldsymbol{x})e^{-\lambda f}$ with the following definition of λ:

$$\lambda = \frac{1}{\theta} \frac{\partial \ln \mathcal{P}(\epsilon)}{\partial \epsilon} \tag{A.4}$$

In the case of a Gaussian likelihood $\mathcal{P}(\xi) \propto e^{-\frac{\xi^2}{2\sigma^2}}$ the MAP approach leads to $\lambda = -\frac{\xi}{\sigma^2}$ which will give a $p^{(opt)}(\boldsymbol{x})$ equivalent to the one of equation 21 of Ref[45]. In our MaxEnt approach, this is equivalent to using a Gaussian prior on the additional variable ϵ which leads to $\xi(\lambda) = \langle\epsilon\rangle = -\lambda\sigma^2$, namely $\lambda = -\frac{\langle\epsilon\rangle}{\sigma^2}$. Since λ is the same in both approach we conclude that both MAP and MaxEnt lead to the same optimal probability density.

This is not true in general for non-Gaussian priors. We here discuss the case of a Laplace prior $P(\xi) \propto e^{-\sqrt{2}\frac{|\xi|}{\sigma_0}}$. Using this prior in MAP leads to $\lambda_{MAP}(\xi) = -\frac{\sqrt{2}}{\sigma}\text{sgn}(\xi)$.

The equivalent relationship in our MaxEnt approach is $\xi = -\frac{\lambda_{MaxEnt}\sigma^2}{1-\frac{\lambda^2_{MaxEnt}\sigma^2}{2}}$, which can be inverted leading to $\lambda_{MaxEnt}(\xi) = \frac{1-\sqrt{\sigma^2+2\xi^2}}{\xi}$. A plot comparing $\lambda_{MAP}(\epsilon)$ and $\lambda_{MaxEnt}(\xi)$ for $\sigma = 1.0$ is provided in figure A.2a. In both cases the Lagrangian multiplier is limited to the range $[-\frac{\sqrt{2}}{\sigma}, +\frac{\sqrt{2}}{\sigma}]$. Interestingly, by using in the MaxEnt procedure a prior from the class discussed in equation (A.3) and setting $\alpha = 1$ results in $\xi = -\frac{\lambda\sigma^2}{1-\frac{\lambda^2\sigma^2}{2}}$ which correspond to the Laplace prior introduced before. This indicates that different priors in the two different frameworks could lead to exactly the same relationship between Lagrangian multipliers and discrepancy between theory and experiments.

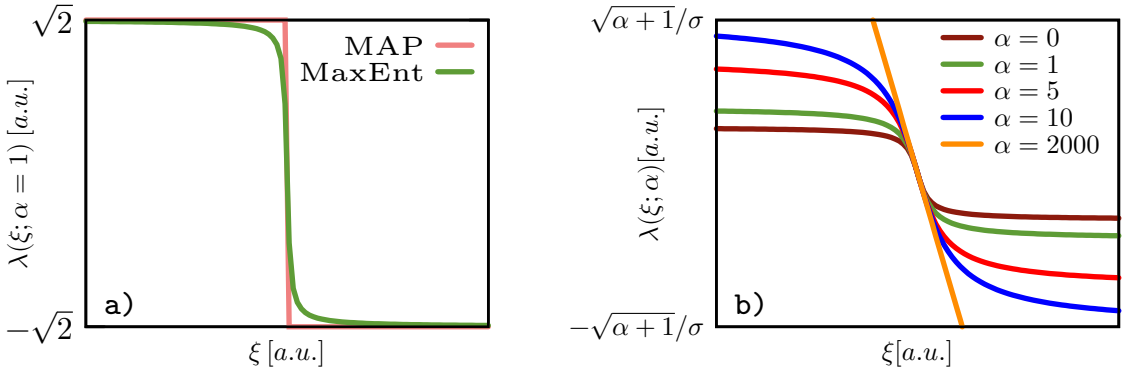


Figure A.2: In panel a) is shown the dependence of the Lagrangian multipliers from the variable ξ both in MaxEnt with Laplace prior (green) and in maximum a posteriori approach (pink). In panel b) is shown the α dependence of λ when using a prior $P_0(\sigma_0, \alpha)$ from the family introduced in A.3. For this example $\sigma = 1$ was used.

Appendix B

^3J scalar coupling on RNA nucleosides

		$J_{HH}(Hz)$					$J_{HC}(Hz)$		$RMSE(Hz)$
		$H1'H2'$	$H2'H3'$	$H3'H4'$	$H4'H5'$	$H4'H5''$	$H1'C2/C4$	$H1'C6/C8$	
seq.	source	$H1^{'}C1^{'}C2^{'}H2^{'}$	$H2^{'}C2^{'}C3^{'}H3^{'}$	$H3^{'}C3^{'}C4^{'}H4^{'}$	$H4^{'}C4^{'}C5^{'}H5^{'}$	$H4^{'}C4^{'}C5^{'}H5^{''}$	χ	χ'	
A	NMR[98]	6.0	5.0	3.4	3.0	3.4	3.6	3.9	
	Amber14	8.5	5.1	3.5	3.2	1.5	4.7	3.6	1.3
	Amber+MaxEnt	6.9	5.1	4.2	3.1	2.6	4.1	3.5	0.6
G	NMR[98]	5.5	5.1	3.9	3.2	3.3	2.5	4.5	
	Amber14	6.8	5.3	5.4	3.4	1.6	4.2	2.2	1.5
	Amber+MaxEnt	6.3	5.1	4.8	3.4	2.7	3.0	3.1	0.7
C	NMR[98]	3.6	5.0	5.8	2.8	4.2	1.9	3.3	
	Amber	7.4	5.2	5.7	3.5	1.8	2.4	3.8	1.7
	Amber+MaxEnt	5.1	5.0	6.7	3.2	3.4	2.0	3.3	0.7
U	NMR[98]	4.3	5.3	5.6	3.0	4.3	2.3	3.6	
	Amber14	7.5	5.2	5.6	3.5	1.8	2.3	3.9	1.6
	Amber+MaxEnt	5.6	5.1	6.4	3.3	3.3	2.2	3.7	0.7

Table B.1: ^3J scalar coupling for the RNA nucleosides along with the computed RMSE both using the original force field and using the MaxEnt-corrected force field. Statistical errors for each scalar coupling and the for the RMSE, obtained by error propagation, are less than $0.1Hz$.

	Lagrangian multipliers (Hz^{-1})						
seq.	$H1'H2'$	$H2'H3'$	$H3'H4'$	$H4'H5'$	$H4'H5''$	$H1'C2/C4$	$H1'C6/C8$
A	0.2187	0.0242	0.1732	0.0363	−0.1917	0.1113	−0.1084
G	0.1745	0.0002	0.2048	0.0487	-0.1423	0.0910	-0.2729
C	0.2972	0.0044	0.2093	0.1003	-0.1782	0.0340	0.0104
U	0.2742	-0.0397	0.1828	0.0722	-0.2063	-0.0298	0.0232

Table B.2: Average Lagrangian multipliers for each of the seven ^3J scalar coupling on each RNA nucleosides. Lagrangian multipliers are averaged between $50ns$ and $100ns$.

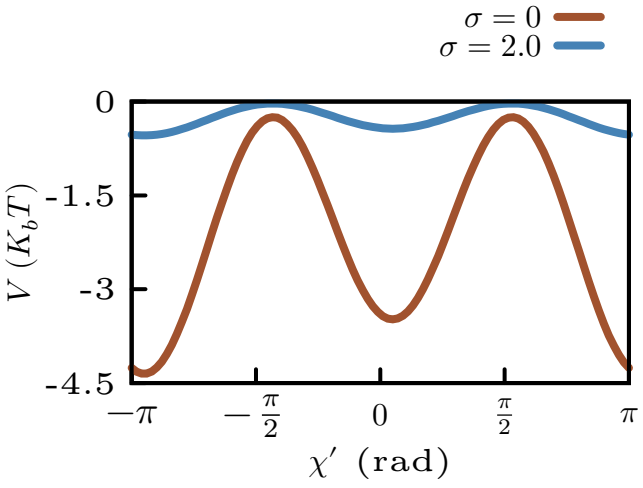


Figure B.1: Correcting potential on the adenosine χ' angle both with $\sigma = 0$ (pure MaxEnt) and with error toleration $\sigma = 2.0$

Coupling	Torsion	Unit	$^3J_i(n)$ (Hz)			σ_i (Hz)	$\Sigma(Hz)$
			n				
			1	2	3		
$^3J_{H1'H2'}$	ν_1	Ap	3.6	3.2	3.0	0.3	0.6
		pC	6.1	5.6	4.8	0.7	
$^3J_{H2'H3'}$	ν_2	Ap	5.0	6.4	5.7	0.7	
		pC	5.0	6.4	5.7	0.7	
$^3J_{H3'H4'}$	ν_3	Ap	8.5	8.0	7.1	0.7	
		pC	6.0	5.5	5.1	0.4	
$^3J_{H4'H5'}$	$\gamma^{'}$	Ap	2.4	3.6	-	0.8	
		pC	1.9	3.2	-	0.9	
$^3J_{H4'H5''}$	$\gamma^{''}$	Ap	2.0	1.6	-	0.3	
		pC	2.0	1.4	-	0.4	
$^3J_{H5'P}$	β_2	pC	3.2	3.1	3.2	0.1	
$^3J_{H5''P}$	β_2	pC	2.7	2.5	2.3	0.2	
$^3J_{H3'P}$	ϵ_1	Ap	6.0	5.9	7.0	0.6	
$^3J_{C4'P}$	β_2	pC	11.3	10.4	-	0.6	

(a)

Coupling	Torsion	A			B			C			D		
		n											
		1	2	3	1	2	3	1	2	3	1	2	3
$^3J_{H1'H2'}$	ν_1	6.965[77]	10.2[78]	9.67[8]	-0.91	-0.8	-2.03	1.026	0	0	1.270	0	0
$^3J_{H2'H3'}$	ν_2	8.289[77]	10.2[78]	9.67[8]	-0.91	-0.8	-2.03	0.668	0	0	0.002	0	0
$^3J_{H3'H4'}$	ν_3	7.964[77]	10.2[78]	9.67[8]	-0.91	-0.8	-2.03	0.772	0	0	-0.262	0	0
$^3J_{H4'H5'}$	$\gamma^{'}$	9.7[8,	8.313[77]	-	-1.8	-0.99	-	0	0.27	-	0	1.373	-
$^3J_{H4'H5''}$	$\gamma^{''}$	9.7[8,	8.313[77]	-	-1.8	-0.99	-	0	-4.752	-	0	1.373	-
$^3J_{H5'P}$	$\beta_2 - 120°$	15.3[79]	15.3[80]	18.1[81]	-6.1	-6.2	-4.8	0	0	0	1.6	1.5	0
$^3J_{H5''P}$	$\beta_2 + 120°$	15.3[79]	15.3[80]	18.1[81]	-6.1	-6.2	-4.8	0	0	0	1.6	1.5	0
$^3J_{H3'P}$	$\epsilon_1 + 120°$	15.3[79]	15.3[80]	18.1[81]	-6.1	-6.2	-4.8	0	0	0	1.6	1.5	0
$^3J_{C4'P}$	β_2	8.0[82]	6.9[83]	-	-3.4	-3.4	-	-	-	-	0.5	0.7	-

(b)

Table B.3: In table a) we show the standard deviation of all the available ^{3}J scalar couplings computed using different sets of Karplus parameters. The various sets used for each scalar coupling are exposed in table b). The Karplus parameters used in the main text are highlighted in red and are also included in the full list reported in table C.3

Appendix C

Self-consistent Maximum Entropy force-field refinement

		$J_{HH}(Hz)$					$J_{HC}(Hz)$		$RMSE(Hz)$
		$H1'H2'$	$H2'H3'$	$H3'H4'$	$H4'H5'$	$H4'H5''$	$H1'C2/C4$	$H1'C6/C8$	
seq.	source	$H1^{'}C1^{'}C2^{'}H2^{'}$	$H2^{'}C2^{'}C3^{'}H3^{'}$	$H3^{'}C3^{'}C4^{'}H4^{'}$	$H4^{'}C4^{'}C5^{'}H5^{'}$	$H4^{'}C4^{'}C5^{'}H5^{''}$	χ	χ'	
A	NMR[98]	6.0	5.0	3.4	3.0	3.4	3.6	3.9	
	Amber14	8.5	5.1	3.5	3.2	1.5	4.7	3.6	1.3
	Amber+MaxEnt	6.5	5.2	3.7	2.5	3.0	3.8	3.6	0.3
C	NMR[98]	3.6	5.0	5.8	2.8	4.2	1.9	3.3	
	Amber14	7.3	5.2	5.7	3.5	1.8	2.4	3.8	1.7
	Amber+MaxEnt	5.4	5.0	6.1	2.6	3.5	2.3	3.8	0.8

Table C.1: ^3J scalar coupling for the adenine and cytidine RNA nucleosides obtained in the self-consistent procedure. Total RMSE is reported both using the original force field and using the MaxEnt-corrected force field. Statistical errors for each scalar coupling and the for the RMSE, obtained by error propagation, are less than $0.1Hz$. ^3J scalar coupling for the Adenosine and Cytidine RNA nucleosides obtained in the self consistent procedure.

			$J_{HH}(Hz)$			$J_{HP}(Hz)$			$J_{CP}(Hz)$		$J_{HC}(Hz)$		
			$H3'H4'$	$H4'H5'$	$H4'H5''$	$H3'P$	$H5'P$	$H5''P$	$C2'P$	$C4'P$	$H1'C2/C4$	$H1'C6/C8$	
seq.	source	unit	$H3'C3'C4'H4'$	$H4'C4'C5'H5'$	$H4'C4'C5'H5''$	ϵ_1	β_2	β_2	ϵ_1	ϵ_1/β_2	χ	χ'	$RMSE(Hz)$
ApA	NMR	1	5.0	2.5	3.6	9.0	-	-	3.7	5.3	1.9	4.2	
		2	5.5	2.8*	3.8	-	3.0*	3.8	-	9.4	2.5	3.1	
	Amber14	1	7.6	3.4	1.5	5.6	-	-	3.1	7.3	4.3	3.5	1.6
		2	6.2	3.6	1.6	-	3.1	2.5	-	10.4	1.7	2.7	
	$\text{Amber}_{\text{MaxEnt}}$	1	5.3	2.6	3.5	8.5	-	-	4.2	5.0	3.3	3.2	0.6
		2	5.6	2.5	2.9	-	3.6	3.7	-	9.7	1.7	3.8	
ApC	NMR	1	6.1	2.4	3.5	8.7	-	-	3.3	4.6	2.1	2.8	
		2	7.1*	1.7	2.0	-	4.0	3.4	-	9.5	1.4	4.5	
	Amber14	1	8.4	2.9	1.9	6.1	-	-	2.5	7.7	3.4	2.8	1.6
		2	5.9	3.4	1.4	-	3.1	2.4	-	10.5	1.6	3.1	
	$\text{Amber}_{\text{MaxEnt}}$	1	7.0	2.6	3.5	8.8	-	-	3.6	5.5	2.1	2.3	0.6
		2	6.9	2.6	3.1	-	3.9	4.5	-	9.1	1.6	3.4	
CpA	NMR	1	6.8	2.6	4.0	8.7	-	-	3.4	5.4	1.6	4.6	
		2	5.5	2.6	3.0	-	4.3	3.8	-	9.3	1.8	4.3	
	Amber	1	7.9	3.4	2.5	5.9	-	-	3.4	6.9	2.5	3.6	1.3
		2	6.1	3.6	1.7	-	3.2	2.2	-	10.5	2.1	3.5	
	$\text{Amber}_{\text{MaxEnt}}$	1	6.1	2.6	3.4	8.3	-	-	4.3	4.9	2.3	3.8	0.6
		2	5.6	2.6	3.1	-	3.3	3.4	-	9.9	1.9	3.9	
CpC	NMR	1	7.3	2.5	3.8	8.9	-	-	3.1	6.0	1.4	4.5	
		2	7.2	2.4	2.4	-	4.3	3.2	-	9.5	1.4	4.6	
	Amber14	1	9.4	3.8	2.9	6.8	-	-	2.3	7.6	1.7	2.9	1.3
		2	7.1	3.3	1.3	-	3.4	1.9	-	10.5	1.8	3.3	
	$\text{Amber}_{\text{MaxEnt}}$	1	9.1	3.1	5.5	8.5	-	-	2.7	6.4	1.7	3.1	0.9
		2	7.5	3.4	2.0	-	4.6	3.0	-	9.5	1.7	3.4	

Table C.2: Scalar coupling for the dinucleosides monophosphate with the self consistent procedure. Experimental values were taken from refs([116–118]). Experimental temperatures 300K and (293K)*.

Coupling	Torsion θ	Base	Lagrangian multiplier (Hz^{-1})	A	B	C	D	φ
${}^3J_{H1'H2'}$	$H1'C1'C2'H2'$	all	0.4393	9.67[8]	-2.03[8]	0	0	0°
${}^3J_{H2'H3'}$	$H2'C2'C3'H3'$	all	0.0570	9.67[8]	-2.03[8]	0	0	0°
${}^3J_{H3'H4'}$	$H3'C3'C4'H4'$	A	0.4009	9.67[8]	-2.03[8]	0	0	0°
		C	0.3316	9.67[8]	-2.03[8]	0	0	0°
${}^3J_{H4'H5'}$	$H4'C4'C5'H5'$	all	0.3643	8.31[77]	-0.99[77]	0.27[77]	1.37[77]	0°
${}^3J_{H4'H5''}$	$H4'C4'C5'H5''$	all	-0.2077	8.31[77]	-0.99[77]	-4.72[77]	1.37[77]	0°
${}^3J_{H3'P}$	ϵ_1	all	-0.2358	15.3[79]	-6.1[79]	0	1.6[79]	120°
${}^3J_{H5'P}$	β_2	all	-0.0237	18.1[81]	-4.8[81]	0	0	−120°
${}^3J_{H5''P}$	β_2	all	-0.0700	18.1[81]	-4.8[81]	0	0	120°
${}^3J_{C2'P}$	ϵ_1	all	0.2015	6.90[83]	-3.4[83]	0	0.7[83]	−120°
${}^3J_{C4'P}$	ϵ_1	all	0.2010	6.90[83]	-3.4[83]	0	0.7[83]	0°
	β_2	all	0.1923	6.90[83]	-3.4[83]	0	0.7[83]	0°
${}^3J_{H1'C4}$	χ	A	0.1758	3.60[119]	1.8[119]	0	0.4[119]	−68.6°
${}^3J_{H1'C2}$		C	0.4270	3.90[119]	1.7[119]	0	0.3[119]	−70.4°
${}^3J_{H1'C8}$	χ'	A	-0.4068	4.20[119]	-0.5[119]	0	0.3[119]	−68.9°
${}^3J_{H1'C6}$		C	-0.7401	4.80[119]	0.7[119]	0	0.3[119]	−66.9°

Table C.3: Lagrangian multipliers associated to each torsional angle used in the self consistent procedure together with the associated Karplus parameters used to back calculate ${}^3J$scalar couplings. Karplus relations used are in the form ${}^3J(\theta) = A\cos^2(\theta+\varphi) + B\cos(\theta+\varphi) + C\sin(\theta+\varphi)\cos(\theta+\varphi) + D$. χ' is defined as $H1' - C1' - N1/N9 - C6/C8$ along with a phase shift of 60°.

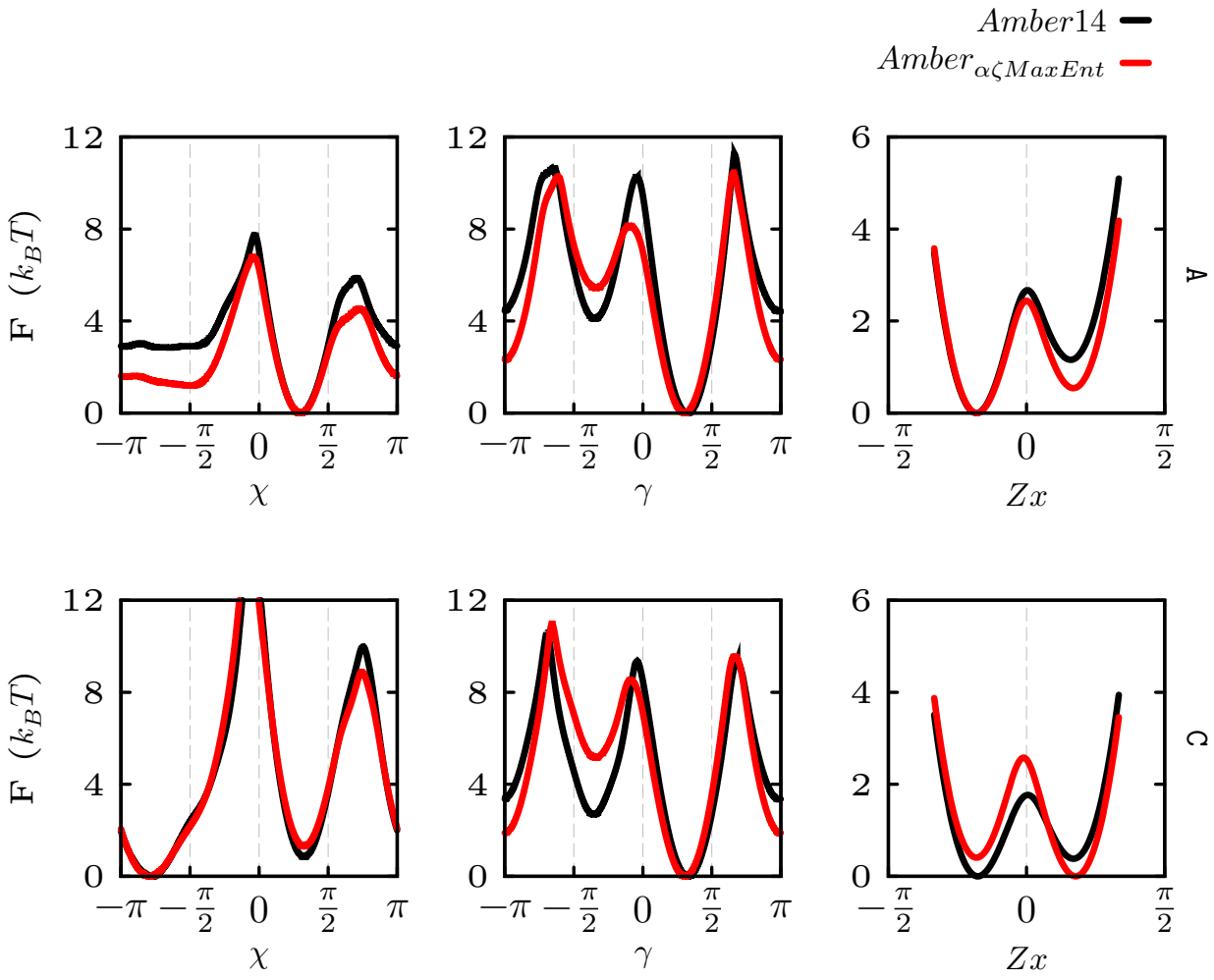


Figure C.1: Free-energy profiles for all the corrected torsional angles of A and C, using the standard AMBER force-field, and the AMBER$_{\alpha\zeta SC}$ obtained with the self-consistent procedure introduced in this thesis starting from the AMBER$_{\alpha\zeta}$ force-field.

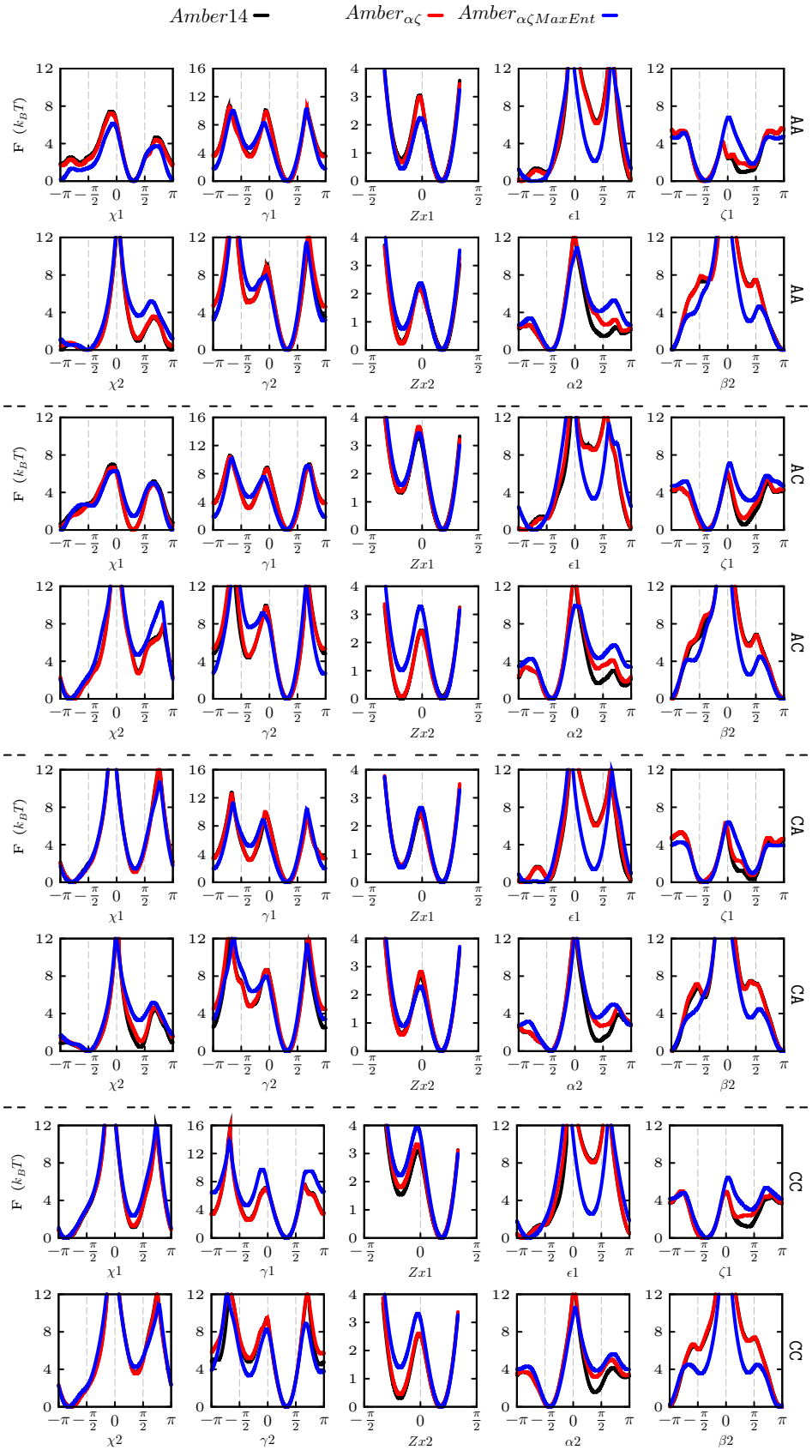


Figure C.2: Free-energy profiles of all corrected torsionals of ApA, ApC, CpA, CpC systems, using the standard AMBER force-field, AMBER$_{\alpha\zeta}$), and the refined force-field AMBER$_{\alpha\zeta SC}$ obtained with the self-consistent procedure introduced in this thesis starting from the AMBER$_{\alpha\zeta}$ force-field.

Appendix D

Plumed Input Files

D.1 Maximum Entropy restraints on RNA Nucleosides

```
##########################
## A nucleoside structure
##########################
MOLINFO STRUCTURE=adenosine.pdb MOLTYPE=rna
########################
#### CVs
########################
n1: TORSION ATOMS=10,9,27,28
n2: TORSION ATOMS=28,27,25,26
n3: TORSION ATOMS=26,25,6,7
n4: TORSION ATOMS=7,6,3,4
n5: TORSION ATOMS=7,6,3,5
c1: TORSION ATOMS=@chi¿1
chi1: TORSION ATOMS=10,9,11,12
g1: TORSION ATOMS=@gamma¿1
puck1: PUCKERING ATOMS=@sugar¿1
########################
## Karplus relations used to back calculate 3J scalar couplings
########################
#JH1'¿H2'
j1: MATHEVAL ARG=n1 FUNC=9.67*cos(x)*cos(x)-2.03*cos(x) PERIODIC=NO
#JH2'¿H3'
j2: MATHEVAL ARG=n2 FUNC=9.67*cos(x)*cos(x)-2.03*cos(x) PERIODIC=NO
#JH3'¿H4'
j3: MATHEVAL ARG=n3 FUNC=9.67*cos(x)*cos(x)-2.03*cos(x) PERIODIC=NO
#JH4'¿H5'
j4: MATHEVAL ARG=n4 FUNC=8.313*cos(x)*cos(x)-0.99*cos(x)+1.373+0.27*cos(x)*sin(x) PERIODIC=NO
#JH4'¿H5''
j5: MATHEVAL ARG=n5 FUNC=8.313*cos(x)*cos(x)-0.99*cos(x)+1.373¿4.752*cos(x)*sin(x) PERIODIC=NO
#JC4¿H1'
MATHEVAL ...
LABEL=j6
ARG=c1
FUNC=3.6*cos(x-(pi/180.0)*68.6)*cos(x-(pi/180.0)*68.6)+1.8*cos(x-(pi/180.0)*68.6)+0.4 PERIODIC=NO
... MATHEVAL
#JC8¿H1'
MATHEVAL ...
LABEL=j7
ARG=chi1
FUNC=4.2*cos(x+(pi/180.0)*(60.0-68.9))*cos(x+(pi/180.0)*(60.0-68.9))-0.5*cos(x+(pi/180.0)*(60.0-68.9))+0.3
PERIODIC=NO
... MATHEVAL
##########################
## Concurrent Metadynamics
# (Is present only in replica 1,2,3 with increasing BIASFACTORS)
##########################
METAD ARG=c1 SIGMA=0.25 PACE=500 TAU=12 TEMP=300 BIASFACTOR=1.5 GRID_MIN=-pi GRID_MAX=pi FILE=HILLS_c1
METAD ARG=g1 SIGMA=0.25 PACE=500 TAU=12 TEMP=300 BIASFACTOR=1.5 GRID_MIN=-pi GRID_MAX=pi FILE=HILLS_c2
METAD ...
ARG=puck1.Zx,puck1.Zy SIGMA=0.15,0.15 PACE=500 TAU=12 TEMP=300 BIASFACTOR=1.5 GRID_MIN=-pi,-pi
GRID_MAX=pi,pi FILE=HILLS_c3
... METAD
########################
###### MaxEnt procedure
########################
MAXENT ...
LABEL=res
ARG=j1,j2,j3,j4,j5,j6,j7
KAPPA=0.01,0.01,0.01,0.01,0.01,0.01,0.01
TAU=3.0,3.0,3.0,3.0,3.0,3.0,3.0
ERROR_TYPE=LAPLACE #Using Laplace priori for error treatment
SIGMA=0.5
TSTART=50000
TEND=100000
TYPE=EQUAL AT=6.0,5.0,3.4,3.0,3.4,3.6,3.9
... MAXENT
PRINT ...
ARG=j1,j2,j3,j4,j5,j6,j7
STRIDE=10 FILE=Adenosine
... PRINT
```

Figure D.1: Sample PLUMED[85] input file for the simulation of the A nucleosides

D.2 Maximum Entropy Force-Field Refinement

```
############################
#### A nucleoside structure
############################
MOLINFO STRUCTURE=adenosine.pdb MOLTYPE=rna

########################
#### CVs
########################
n1: TORSION ATOMS=10,9,27,28
n2: TORSION ATOMS=28,27,25,26
n3: TORSION ATOMS=26,25,6,7
n4: TORSION ATOMS=7,6,3,4
n5: TORSION ATOMS=7,6,3,5
c1: TORSION ATOMS=@chi-1
chi1: TORSION ATOMS=10,9,11,12
g1: TORSION ATOMS=@gamma-1
puck1: PUCKERING ATOMS=@sugar-1

########################
## Karplus relations used to back calculate 3J scalar couplings
########################
#JH1'-H2'
j1: MATHEVAL ARG=n1 FUNC=9.67*cos(x)*cos(x)-2.03*cos(x) PERIODIC=NO
#JH2'-H3'
j2: MATHEVAL ARG=n2 FUNC=9.67*cos(x)*cos(x)-2.03*cos(x) PERIODIC=NO
#JH3'-H4'
j3: MATHEVAL ARG=n3 FUNC=9.67*cos(x)*cos(x)-2.03*cos(x) PERIODIC=NO
#JH4'-H5'
j4: MATHEVAL ARG=n4 FUNC=8.313*cos(x)*cos(x)-0.99*cos(x)+1.373+0.27*cos(x)*sin(x) PERIODIC=NO
#JH4'-H5''
MATHEVAL ...
LABEL=j5
ARG=n5
FUNC=8.313*cos(x)*cos(x)-0.99*cos(x)+1.373-4.752*cos(x)*sin(x) PERIODIC=NO
... MATHEVAL
#JC4-H1'
MATHEVAL ...
LABEL=j6
ARG=c1
FUNC=3.6*cos(x-(pi/180.0)*68.6)*cos(x-(pi/180.0)*68.6)+1.8*cos(x-(pi/180.0)*68.6)+0.4 PERIODIC=NO
... MATHEVAL
#JC8-H1'
MATHEVAL ...
LABEL=j7
ARG=chi1
FUNC=4.2*cos(x+(pi/180.0)*(60.0-68.9))*cos(x+(pi/180.0)*(60.0-68.9))-0.5*cos(x+(pi/180.0)*(60.0-68.9))+0.3 PERIODIC=NO
... MATHEVAL
 ########################
 #Extra CVs to have the same CVs on which to apply refinement procedure
 #Notice if some CVs are only present in a dinucleoside (i.e. dhiedral anglese involving a phospate)
 #in this nucleoside input file they are setted to a fictitious constant.
 ########################
 jA3: COMBINE ARG=j3  PERIODIC=NO
 jA6: COMBINE ARG=j6  PERIODIC=NO
 jA7: COMBINE ARG=j7  PERIODIC=NO
 jC3: CONSTANT VALUE=0.0
 jC6: CONSTANT VALUE=0.0
 jC7: CONSTANT VALUE=0.0
 j8: CONSTANT VALUE=0.0
 j9: CONSTANT VALUE=0.0
 j10: CONSTANT VALUE=0.0
 j11: CONSTANT VALUE=0.0
 j12: CONSTANT VALUE=0.0

########################
#Concurrent Metadynamics
#They present in all 4 replicas, with increasing BIASFACTOR from 1(reference replica) to 5
########################

METAD ARG=c1 SIGMA=0.25 PACE=500 TAU=12 TEMP=300 BIASFACTOR=1.70998 GRID_MIN=-pi GRID_MAX=pi FILE=HILLS_c1
METAD ARG=g1 SIGMA=0.25 PACE=500 TAU=12 TEMP=300 BIASFACTOR=1.70998 GRID_MIN=-pi GRID_MAX=pi FILE=HILLS_c2
METAD ...
ARG=puck1.Zx,puck1.Zy SIGMA=0.15,0.15 PACE=500 TAU=12 TEMP=300 BIASFACTOR=1.70998
GRID_MIN=-pi,-pi GRID_MAX=pi,pi FILE=HILLS_c3
... METAD

INCLUDE FILE=common.dat
```

Figure D.2: Sample input file for A nucleoside used in the self consistent procedure. We recall that this setup consist of 24 replicas, the first 4 of which regards the A nucleoside. BIASFACTOR's for the four replicas: 1.0, 1.70998, 2.92402, 5.0. File "common.dat" containing the paramenter for the MaxEnt procedure is the same for all replicas and is reported after.

```
# vim:ft=plumed

INDEX_TO_AVOID 12
RESTART

### ApA structure
MOLINFO STRUCTURE=AA.pdb  MOLTYPE=rna

#### Torsions
############################

#### alpha
a2: TORSION ATOMS=@alpha-2

#### beta
b2: TORSION ATOMS=@beta-2

#### gamma
g1: TORSION ATOMS=@gamma-1
g2: TORSION ATOMS=@gamma-2

#### epsilon
e1: TORSION ATOMS=@epsilon-1

#### zeta
z1: TORSION ATOMS=@zeta-1

#### chi
c1: TORSION ATOMS=@chi-1
c2: TORSION ATOMS=@chi-2

#### Puckering
############################
puck1: PUCKERING ATOMS=@sugar-1
puck2: PUCKERING ATOMS=@sugar-2

#### Base COM distances
############################

#### Center of Mass
WHOLEMOLECULES ENTITY0=@base-1
com1: COM ATOMS=@base-1
WHOLEMOLECULES ENTITY0=@base-2
com2: COM ATOMS=@base-2

## Distances between COMs
d1_2: DISTANCE ATOMS=com1,com2

###Torsional for 3J COUPLINGS###
n1_1: TORSION ATOMS=10,9,27,28
n1_2: TORSION ATOMS=43,42,60,61
n2_1: TORSION ATOMS=28,27,25,26
n2_2: TORSION ATOMS=61,60,58,59
n3_2: TORSION ATOMS=59,58,39,40
n3_1: TORSION ATOMS=26,25,6,7
n4_1: TORSION ATOMS=7,6,3,4
n4_2: TORSION ATOMS=40,39,36,37
n5_1: TORSION ATOMS=7,6,3,5
n5_2: TORSION ATOMS=40,39,36,38
chi1_1: TORSION ATOMS=10,9,11,12
chi1_2: TORSION ATOMS=43,42,44,45
```

(a) Page1

```
##Karplus relations for 3J couplings back calculation
#J(H1'-H2')_1/2
j1_1: MATHEVAL ARG=n1_1 FUNC=9.67*cos(x)*cos(x)-2.03*cos(x) PERIODIC=NO
j1_2: MATHEVAL ARG=n1_2 FUNC=9.67*cos(x)*cos(x)-2.03*cos(x) PERIODIC=NO
#J(H2'-H3')_1/2
j2_1: MATHEVAL ARG=n2_1 FUNC=9.67*cos(x)*cos(x)-2.03*cos(x) PERIODIC=NO
j2_2: MATHEVAL ARG=n2_2 FUNC=9.67*cos(x)*cos(x)-2.03*cos(x) PERIODIC=NO
#J(H3'-H4')_1/2
j3_1: MATHEVAL ARG=n3_1 FUNC=9.67*cos(x)*cos(x)-2.03*cos(x) PERIODIC=NO
j3_2: MATHEVAL ARG=n3_2 FUNC=9.67*cos(x)*cos(x)-2.03*cos(x) PERIODIC=NO
#J(H4'-H5')_1/2
j4_1: MATHEVAL ARG=n4_1 FUNC=8.313*cos(x)*cos(x)-0.99*cos(x)+0.27*cos(x)*sin(x)+1.373 PERIODIC=NO
j4_2: MATHEVAL ARG=n4_2 FUNC=8.313*cos(x)*cos(x)-0.99*cos(x)+0.27*cos(x)*sin(x)+1.373 PERIODIC=NO
##J(H4'-H5'')_1/2
j5_1: MATHEVAL ARG=n5_1 FUNC=8.313*cos(x)*cos(x)-0.99*cos(x)-4.752*cos(x)*sin(x)+1.373 PERIODIC=NO
j5_2: MATHEVAL ARG=n5_2 FUNC=8.313*cos(x)*cos(x)-0.99*cos(x)-4.752*cos(x)*sin(x)+1.373 PERIODIC=NO
##J(C4-H1')_1/2
j6_1: MATHEVAL ARG=c1 FUNC=3.6*cos(x-(pi/180.0)*68.6)*cos(x-(pi/180.0)*68.6)+1.8*cos(x-(pi/180.0)*68.6)+0.4 PERIODIC=NO
j6_2: MATHEVAL ARG=c2 FUNC=3.6*cos(x-(pi/180.0)*68.6)*cos(x-(pi/180.0)*68.6)+1.8*cos(x-(pi/180.0)*68.6)+0.4 PERIODIC=NO
##J(C8-H1')_1/2
j7_1: MATHEVAL ARG=chi1_1 FUNC=4.2*cos(x+(pi/180.0)*(60.0-68.9))*cos(x+(pi/180.0)*(60.0-68.9))-0.5*cos(x+(pi/180.0)*(60.
0-68.9))+0.3 PERIODIC=NO
j7_2: MATHEVAL ARG=chi1_2 FUNC=4.2*cos(x+(pi/180.0)*(60.0-68.9))*cos(x+(pi/180.0)*(60.0-68.9))-0.5*cos(x+(pi/180.0)*(60.
0-68.9))+0.3 PERIODIC=NO
##J(C4'-P)_1/2
j11_1: MATHEVAL ARG=e1 FUNC=6.9*cos(x)*cos(x)-3.4*cos(x)+0.7 PERIODIC=NO
j11_2: MATHEVAL ARG=b2 FUNC=6.9*cos(x)*cos(x)-3.4*cos(x)+0.7 PERIODIC=NO
#J(H5'-P)
j8: MATHEVAL ARG=b2 FUNC=18.1*cos(x-2.0944)*cos(x-2.0944)-4.8*cos(x-2.0944) PERIODIC=NO
##J(H5''-P)
j9: MATHEVAL ARG=b2 FUNC=18.1*cos(x+2.0944)*cos(x+2.0944)-4.8*cos(x+2.0944) PERIODIC=NO
##J(H3'-P)
j10: MATHEVAL ARG=e1 FUNC=15.3*cos(x+2.0944)*cos(x+2.0944)-6.1*cos(x+2.0944)+1.6 PERIODIC=NO
#J(C2'-P)
j12: MATHEVAL ARG=e1 FUNC=6.9*cos(x-2.0944)*cos(x-2.0944)-3.4*cos(x-2.0944)+0.7 PERIODIC=NO

####Additional ApA variable required for the self consistent fit

## Notice that here (i.e. ApA), j1 is the sum of the two equivalent torsions j1_1 and j1_2 on the nucleobase in 3' and 5
'
j1: COMBINE ARG=j1_1,j1_2 PERIODIC=NO
j2: COMBINE ARG=j2_1,j2_2 PERIODIC=NO
j4: COMBINE ARG=j4_1,j4_2 PERIODIC=NO
j5: COMBINE ARG=j5_1,j5_2 PERIODIC=NO
jA3: COMBINE ARG=j3_1,j3_2 PERIODIC=NO
jA6: COMBINE ARG=j6_1,j6_2 PERIODIC=NO
jA7: COMBINE ARG=j7_1,j7_2 PERIODIC=NO
jC3: CONSTANT VALUE=0.0
jC6: CONSTANT VALUE=0.0
jC7: CONSTANT VALUE=0.0

##Concurrent Metadynamics
METAD ARG=d1_2 SIGMA=0.05 TAU=12 PACE=500 BIASFACTOR=5.0 TEMP=300 GRID_MIN=0 GRID_MAX=4 FILE=HILLSd1_2
METAD ARG=a2 SIGMA=0.25 TAU=12 PACE=500 BIASFACTOR=5.0 TEMP=300 GRID_MIN=-pi GRID_MAX=pi FILE=HILLSa2
METAD ARG=b2 SIGMA=0.25 TAU=12 PACE=500 BIASFACTOR=5.0 TEMP=300 GRID_MIN=-pi GRID_MAX=pi FILE=HILLSb2
METAD ARG=g1 SIGMA=0.25 TAU=12 PACE=500 BIASFACTOR=5.0 TEMP=300 GRID_MIN=-pi GRID_MAX=pi FILE=HILLSg1
METAD ARG=g2 SIGMA=0.25 TAU=12 PACE=500 BIASFACTOR=5.0 TEMP=300 GRID_MIN=-pi GRID_MAX=pi FILE=HILLSg2
METAD ARG=e1 SIGMA=0.25 TAU=12 PACE=500 BIASFACTOR=5.0 TEMP=300 GRID_MIN=-pi GRID_MAX=pi FILE=HILLSe1
METAD ARG=z1 SIGMA=0.25 TAU=12 PACE=500 BIASFACTOR=5.0 TEMP=300 GRID_MIN=-pi GRID_MAX=pi FILE=HILLSz1
METAD ARG=c1 SIGMA=0.25 TAU=12 PACE=500 BIASFACTOR=5.0 TEMP=300 GRID_MIN=-pi GRID_MAX=pi FILE=HILLSc1
METAD ARG=c2 SIGMA=0.25 TAU=12 PACE=500 BIASFACTOR=5.0 TEMP=300 GRID_MIN=-pi GRID_MAX=pi FILE=HILLSc2
METAD ARG=puck1.Zx SIGMA=0.25 TAU=12 PACE=500 BIASFACTOR=5.0 TEMP=300 GRID_MIN=-pi GRID_MAX=pi FILE=HILLSpuck1
METAD ARG=puck2.Zx SIGMA=0.25 TAU=12 PACE=500 BIASFACTOR=5.0 TEMP=300 GRID_MIN=-pi GRID_MAX=pi FILE=HILLSpuck2

INCLUDE FILE=common.dat
```

(b) Page 2

Figure D.3: Sample input file for ApA used in the self consistent procedure. Notice that all the equivalent torsions which are present both in A and ApA, are here expressed as sum of the corresponding torsion on each nucleobase in the 3' and 5' directions respectively. BIASFACTOR's for the four replicas: 1.0, 1.70998, 2.92402, 5.0. Notice also the INDEX_TO_AVOID keyword which is instructing PLUMED to not exchange with different systems. In this case replica 12 is the first replica of ApC. This keyword must be present in all the input belonging to contiguous replicas. (e.g. replica 12 input must have INDEX_TO_AVOID 11)

```
# vim:ft=plumed

INDEX_TO_AVOID 16
RESTART

####  ApC structure
MOLINFO STRUCTURE=AC.pdb  MOLTYPE=rna

#### Torsions
############################

#### alpha
a2: TORSION ATOMS=@alpha-2

#### beta
b2: TORSION ATOMS=@beta-2

#### gamma
g1: TORSION ATOMS=@gamma-1
g2: TORSION ATOMS=@gamma-2

#### epsilon
e1: TORSION ATOMS=@epsilon-1

#### zeta
z1: TORSION ATOMS=@zeta-1

#### chi
c1: TORSION ATOMS=@chi-1
c2: TORSION ATOMS=@chi-2

#### Puckering
############################
puck1: PUCKERING ATOMS=@sugar-1
puck2: PUCKERING ATOMS=@sugar-2

#### Base COM distances
############################

#### Center of Mass
WHOLEMOLECULES ENTITY0=@base-1
com1: COM ATOMS=@base-1
WHOLEMOLECULES ENTITY0=@base-2
com2: COM ATOMS=@base-2

## Distances between COMs
d1_2: DISTANCE ATOMS=com1,com2

####Torsional for 3J COUPLINGS###
n1_1: TORSION ATOMS=10,9,27,28
n1_2: TORSION ATOMS=43,42,58,59
n2_1: TORSION ATOMS=28,27,25,26
n2_2: TORSION ATOMS=59,58,56,57
n3_2: TORSION ATOMS=57,56,39,40
n3_1: TORSION ATOMS=26,25,6,7
n4_1: TORSION ATOMS=7,6,3,4
n4_2: TORSION ATOMS=40,39,36,37
n5_1: TORSION ATOMS=7,6,3,5
n5_2: TORSION ATOMS=40,39,36,38
chi1_1: TORSION ATOMS=10,9,11,12
chi1_2: TORSION ATOMS=43,42,44,45
```

(a) Page1

```
##Karplus relations for 3J couplings back calculation

#J(H1'-H2')_1/2
j1_1: MATHEVAL ARG=n1_1 FUNC=9.67*cos(x)*cos(x)-2.03*cos(x) PERIODIC=NO
j1_2: MATHEVAL ARG=n1_2 FUNC=9.67*cos(x)*cos(x)-2.03*cos(x) PERIODIC=NO
#J(H2'-H3')_1/2
j2_1: MATHEVAL ARG=n2_1 FUNC=9.67*cos(x)*cos(x)-2.03*cos(x) PERIODIC=NO
j2_2: MATHEVAL ARG=n2_2 FUNC=9.67*cos(x)*cos(x)-2.03*cos(x) PERIODIC=NO
#J(H3'-H4')_1/2
j3_1: MATHEVAL ARG=n3_1 FUNC=9.67*cos(x)*cos(x)-2.03*cos(x) PERIODIC=NO
j3_2: MATHEVAL ARG=n3_2 FUNC=9.67*cos(x)*cos(x)-2.03*cos(x) PERIODIC=NO
#J(H4'-H5')_1/2
j4_1: MATHEVAL ARG=n4_1 FUNC=8.313*cos(x)*cos(x)-0.99*cos(x)+0.27*cos(x)*sin(x)+1.373 PERIODIC=NO
j4_2: MATHEVAL ARG=n4_2 FUNC=8.313*cos(x)*cos(x)-0.99*cos(x)+0.27*cos(x)*sin(x)+1.373 PERIODIC=NO
##J(H4'-H5'')_1/2
j5_1: MATHEVAL ARG=n5_1 FUNC=8.313*cos(x)*cos(x)-0.99*cos(x)-4.752*cos(x)*sin(x)+1.373 PERIODIC=NO
j5_2: MATHEVAL ARG=n5_2 FUNC=8.313*cos(x)*cos(x)-0.99*cos(x)-4.752*cos(x)*sin(x)+1.373 PERIODIC=NO
##J(C4-H1') for A in 5' direction
j6_1: MATHEVAL ARG=c1 FUNC=3.6*cos(x-(pi/180.0)*68.6)*cos(x-(pi/180.0)*68.6)+1.8*cos(x-(pi/180.0)*68.6)+0.4 PERIODIC=NO
##J(C2-H1') for C in 3' direction
j6_2: MATHEVAL ARG=c2 FUNC=3.9*cos(x-(pi/180.0)*70.4)*cos(x-(pi/180.0)*70.4)+1.7*cos(x-(pi/180.0)*70.4)+0.3 PERIODIC=NO
##J(C8-H1')
j7_1: MATHEVAL ARG=chi1_1 FUNC=4.2*cos(x+(pi/180.0)*(60.0-68.9))*cos(x+(pi/180.0)*(60.0-68.9))-0.5*cos(x+(pi/180.0)*(60.
0-68.9))+0.3 PERIODIC=NO
##J(C6-H1')
j7_2: MATHEVAL ARG=chi1_2 FUNC=4.8*cos(x+(pi/180.0)*(60.0-66.9))*cos(x+(pi/180.0)*(60.0-66.9))+0.7*cos(x+(pi/180.0)*(60.
0-66.9))+0.3 PERIODIC=NO
##J(C4'-P)_1/2
j11_1: MATHEVAL ARG=e1 FUNC=6.9*cos(x)*cos(x)-3.4*cos(x)+0.7 PERIODIC=NO
j11_2: MATHEVAL ARG=b2 FUNC=6.9*cos(x)*cos(x)-3.4*cos(x)+0.7 PERIODIC=NO
#J(H5'-P)
j8: MATHEVAL ARG=b2 FUNC=18.1*cos(x-2.0944)*cos(x-2.0944)-4.8*cos(x-2.0944) PERIODIC=NO
##J(H5''-P)
j9: MATHEVAL ARG=b2 FUNC=18.1*cos(x+2.0944)*cos(x+2.0944)-4.8*cos(x+2.0944) PERIODIC=NO
##J(H3'-P)
j10: MATHEVAL ARG=e1 FUNC=15.3*cos(x+2.0944)*cos(x+2.0944)-6.1*cos(x+2.0944)+1.6 PERIODIC=NO
#J(C2'-p)
j12: MATHEVAL ARG=e1 FUNC=6.9*cos(x-2.0944)*cos(x-2.0944)-3.4*cos(x-2.0944)+0.7 PERIODIC=NO

### Additional ApC variables required for the self consistent fit

j1: COMBINE ARG=j1_1,j1_2 PERIODIC=NO
j2: COMBINE ARG=j2_1,j2_2 PERIODIC=NO
j4: COMBINE ARG=j4_1,j4_2 PERIODIC=NO
j5: COMBINE ARG=j5_1,j5_2 PERIODIC=NO
jA3: COMBINE ARG=j3_1 PERIODIC=NO
jC3: COMBINE ARG=j3_2 PERIODIC=NO
jA6: COMBINE ARG=j6_1 PERIODIC=NO
jC6: COMBINE ARG=j6_2 PERIODIC=NO
jA7: COMBINE ARG=j7_1 PERIODIC=NO
jC7: COMBINE ARG=j7_2 PERIODIC=NO

## Concurrent Metadynamics
METAD ARG=d1_2 SIGMA=0.05 TAU=12 PACE=500 BIASFACTOR=5.0 TEMP=300 GRID_MIN=0 GRID_MAX=4 FILE=HILLSd1_2
METAD ARG=a2 SIGMA=0.25 TAU=12 PACE=500 BIASFACTOR=5.0 TEMP=300 GRID_MIN=-pi GRID_MAX=pi FILE=HILLSa2
METAD ARG=b2 SIGMA=0.25 TAU=12 PACE=500 BIASFACTOR=5.0 TEMP=300 GRID_MIN=-pi GRID_MAX=pi FILE=HILLSb2
METAD ARG=g1 SIGMA=0.25 TAU=12 PACE=500 BIASFACTOR=5.0 TEMP=300 GRID_MIN=-pi GRID_MAX=pi FILE=HILLSg1
METAD ARG=g2 SIGMA=0.25 TAU=12 PACE=500 BIASFACTOR=5.0 TEMP=300 GRID_MIN=-pi GRID_MAX=pi FILE=HILLSg2
METAD ARG=e1 SIGMA=0.25 TAU=12 PACE=500 BIASFACTOR=5.0 TEMP=300 GRID_MIN=-pi GRID_MAX=pi FILE=HILLSe1
METAD ARG=z1 SIGMA=0.25 TAU=12 PACE=500 BIASFACTOR=5.0 TEMP=300 GRID_MIN=-pi GRID_MAX=pi FILE=HILLSz1
METAD ARG=c1 SIGMA=0.25 TAU=12 PACE=500 BIASFACTOR=5.0 TEMP=300 GRID_MIN=-pi GRID_MAX=pi FILE=HILLSc1
METAD ARG=c2 SIGMA=0.25 TAU=12 PACE=500 BIASFACTOR=5.0 TEMP=300 GRID_MIN=-pi GRID_MAX=pi FILE=HILLSc2
METAD ARG=puck1.Zx SIGMA=0.25 TAU=12 PACE=500 BIASFACTOR=5.0 TEMP=300 GRID_MIN=-pi GRID_MAX=pi FILE=HILLSpuck1
METAD ARG=puck2.Zx SIGMA=0.25 TAU=12 PACE=500 BIASFACTOR=5.0 TEMP=300 GRID_MIN=-pi GRID_MAX=pi FILE=HILLSpuck2

INCLUDE FILE=common.dat
```

(b) Page 2

Figure D.4: Sample input file for ApC used in the self consistent procedure. Here we don't combine j3_1 with j3_2 since they involve highly base dependent torsions which are not suitable for a self consistent fit. The same apply to j6_1, j6_2, j7_1, j7_2. Notice also the INDEX_TO_AVOID keyword which is instructing PLUMED to not exchange with different systems. In this case replica 12 is the first replica of ApC. This keyword must be present in all the input belonging to contiguous replicas. (e.g. in this case since this file correspond to the 4^{th} replica of the ApC system, namely the 15^{th} replica, exchanges with next system must be forbidden). BIASFACTOR's for the four replicas: 1.0, 1.70998, 2.92402, 5.0.

```
##MaxEnt restraining procedure for the A nucleoside
##It will fit data for A and send corrections to all the other systems
#########################
MAXENT ...
LABEL=resA
ARG=j1,j2,jA3,j4,j5,jA6,jA7
KAPPA=0.001,0.001,0.001,0.001,0.001,0.001,0.001
TAU=3.0,3.0,3.0,3.0,3.0,3.0,3.0
ERROR_TYPE=LAPLACE
SIGMA=2.0
TSTART=150000
TEND=300000
TYPE=EQUAL AT=6.0,5.0,3.4,3.0,3.4,3.6,3.9
LEARN_REPLICA=0 #Since reference replica for A is the 0 one
PACE=200
... MAXENT

##MaxEnt restraining procedure for the C nucleoside
##It will fit data for C and send corrections to all the other systems
#########################

MAXENT ...
LABEL=resC
ARG=j1,j2,jC3,j4,j5,jC6,jC7
KAPPA=0.001,0.001,0.001,0.001,0.001,0.001,0.001
TAU=3.0,3.0,3.0,3.0,3.0,3.0,3.0
ERROR_TYPE=LAPLACE
SIGMA=2.0
TSTART=150000
TEND=300000
TYPE=EQUAL AT=3.6,5.0,5.8,2.8,4.2,1.9,3.3
LEARN_REPLICA=4 #Reference replica for C
PACE=200
... MAXENT

##MaxEnt restraining procedure for the ApA dinucleoside monophosphate
##It will fit data for ApA and send corrections to all the other systems
#########################

MAXENT ...
LABEL=resAA
ARG=jA3,j4,j5,jA6,jA7,j8,j9,j10,j11,j12
KAPPA=0.001,0.001,0.001,0.001,0.001,0.001,0.001,0.001,0.001,0.001
TAU=3.0,3.0,3.0,3.0,3.0,3.0,3.0,3.0,3.0,3.0
ERROR_TYPE=LAPLACE
SIGMA=2.0
TSTART=150000
TEND=300000
TYPE=EQUAL AT=10.5,5.3,7.4,4.4,7.3,3.0,3.8,9.0,14.7,3.7
LEARN_REPLICA=8 #Reference Replica for ApA
PACE=200
... MAXENT

##MaxEnt restraining procedure for the ApC dinucleoside monophosphate
##It will fit data for ApC and send corrections to all the other systems
#########################

MAXENT ...
LABEL=resAC
ARG=jA3,jC3,j4,j5,jA6,jC6,jA7,jC7,j8,j9,j10,j11,j12
KAPPA=0.001,0.001,0.001,0.001,0.001,0.001,0.001,0.001,0.001,0.001,0.001,0.001,0.001
TAU=3.0,3.0,3.0,3.0,3.0,3.0,3.0,3.0,3.0,3.0,3.0,3.0,3.0
ERROR_TYPE=LAPLACE
SIGMA=2.0
TSTART=150000
TEND=300000
TYPE=EQUAL AT=6.1,7.1,4.1,5.5,2.1,1.4,2.8,4.5,4.0,3.4,8.7,14.1,3.3
LEARN_REPLICA=12 #Reference Replica for ApC
PACE=200
... MAXENT

##MaxEnt restraining procedure for the CpA dinucleoside monophosphate
##It will fit data for CpA and send corrections to all the other systems
#########################

MAXENT ...
LABEL=resCA
ARG=jA3,jC3,j4,j5,jA6,jC6,jA7,jC7,j8,j9,j10,j11,j12
KAPPA=0.001,0.001,0.001,0.001,0.001,0.001,0.001,0.001,0.001,0.001,0.001,0.001,0.001
TAU=3.0,3.0,3.0,3.0,3.0,3.0,3.0,3.0,3.0,3.0,3.0,3.0,3.0
ERROR_TYPE=LAPLACE
SIGMA=2.0
TSTART=150000
TEND=300000
TYPE=EQUAL AT=5.5,6.8,5.2,7.0,1.8,1.6,4.3,4.6,4.3,3.8,8.7,14.7,3.4
LEARN_REPLICA=16 #Reference Replica for CpA
PACE=200
... MAXENT

##MaxEnt restraining procedure for the CpC dinucleoside monophosphate
##It will fit data for CpC and send corrections to all the other systems
#########################

MAXENT ...
LABEL=resCC
ARG=jC3,j4,j5,jC6,jC7,j8,j9,j10,j11,j12
KAPPA=0.001,0.001,0.001,0.001,0.001,0.001,0.001,0.001,0.001,0.001
TAU=3.0,3.0,3.0,3.0,3.0,3.0,3.0,3.0,3.0,3.0
ERROR_TYPE=LAPLACE
SIGMA=2.0
TSTART=150000
TEND=300000
TYPE=EQUAL AT=14.5,4.9,6.2,2.8,9.1,4.3,3.2,8.9,15.5,3.1
LEARN_REPLICA=20 #Reference Replica for CpC
PACE=200
... MAXENT
```

Figure D.5: File “common.dat” containing all the MaxEnt algorithm parameters which is the same for all the systems.

```
# vim: ft=plumed
alpha2: TORSION ATOMS=@alpha-2
beta2: TORSION ATOMS=@beta-2
alpha3: TORSION ATOMS=@alpha-3
beta3: TORSION ATOMS=@beta-3
alpha4: TORSION ATOMS=@alpha-4
beta4: TORSION ATOMS=@beta-4
alpha5: TORSION ATOMS=@alpha-5
beta5: TORSION ATOMS=@beta-5
alpha6: TORSION ATOMS=@alpha-6
beta6: TORSION ATOMS=@beta-6
alpha7: TORSION ATOMS=@alpha-7
beta7: TORSION ATOMS=@beta-7
alpha8: TORSION ATOMS=@alpha-8
beta8: TORSION ATOMS=@beta-8
chi1: TORSION ATOMS=@chi-1
chi2: TORSION ATOMS=@chi-2
chi3: TORSION ATOMS=@chi-3
chi4: TORSION ATOMS=@chi-4
chi5: TORSION ATOMS=@chi-5
chi6: TORSION ATOMS=@chi-6
chi7: TORSION ATOMS=@chi-7
chi8: TORSION ATOMS=@chi-8
zeta1: TORSION ATOMS=@zeta-1
epsilon1: TORSION ATOMS=@epsilon-1
zeta2: TORSION ATOMS=@zeta-2
epsilon2: TORSION ATOMS=@epsilon-2
zeta3: TORSION ATOMS=@zeta-3
epsilon3: TORSION ATOMS=@epsilon-3
zeta4: TORSION ATOMS=@zeta-4
epsilon4: TORSION ATOMS=@epsilon-4
zeta5: TORSION ATOMS=@zeta-5
epsilon5: TORSION ATOMS=@epsilon-5
zeta6: TORSION ATOMS=@zeta-6
epsilon6: TORSION ATOMS=@epsilon-6
zeta7: TORSION ATOMS=@zeta-7
epsilon7: TORSION ATOMS=@epsilon-7
gamma1: TORSION ATOMS=@gamma-1
delta1: TORSION ATOMS=@delta-1
gamma2: TORSION ATOMS=@gamma-2
delta2: TORSION ATOMS=@delta-2
gamma3: TORSION ATOMS=@gamma-3
delta3: TORSION ATOMS=@delta-3
gamma4: TORSION ATOMS=@gamma-4
delta4: TORSION ATOMS=@delta-4
gamma5: TORSION ATOMS=@gamma-5
delta5: TORSION ATOMS=@delta-5
gamma6: TORSION ATOMS=@gamma-6
delta6: TORSION ATOMS=@delta-6
gamma7: TORSION ATOMS=@gamma-7
delta7: TORSION ATOMS=@delta-7
gamma8: TORSION ATOMS=@gamma-8
delta8: TORSION ATOMS=@delta-8
sumcosalpha: MATHEVAL ARG=alpha2,alpha3,alpha4,alpha5,alpha6,alpha7,alpha8 V
AR=v2,v3,v4,v5,v6,v7,v8 FUNC=cos(1*v2)+cos(1*v3)+cos(1*v4)+cos(1*v5)+cos(1*v
6)+cos(1*v7)+cos(1*v8) PERIODIC=NO
sumsinalpha: MATHEVAL ARG=alpha2,alpha3,alpha4,alpha5,alpha6,alpha7,alpha8 V
AR=v2,v3,v4,v5,v6,v7,v8 FUNC=sin(1*v2)+sin(1*v3)+sin(1*v4)+sin(1*v5)+sin(1*v
6)+sin(1*v7)+sin(1*v8) PERIODIC=NO
2sumcosalpha: MATHEVAL ARG=alpha2,alpha3,alpha4,alpha5,alpha6,alpha7,alpha8
VAR=v2,v3,v4,v5,v6,v7,v8 FUNC=cos(2*v2)+cos(2*v3)+cos(2*v4)+cos(2*v5)+cos(2*
v6)+cos(2*v7)+cos(2*v8) PERIODIC=NO
2sumsinalpha: MATHEVAL ARG=alpha2,alpha3,alpha4,alpha5,alpha6,alpha7,alpha8
VAR=v2,v3,v4,v5,v6,v7,v8 FUNC=sin(2*v2)+sin(2*v3)+sin(2*v4)+sin(2*v5)+sin(2*
v6)+sin(2*v7)+sin(2*v8) PERIODIC=NO
3sumcosalpha: MATHEVAL ARG=alpha2,alpha3,alpha4,alpha5,alpha6,alpha7,alpha8
VAR=v2,v3,v4,v5,v6,v7,v8 FUNC=cos(3*v2)+cos(3*v3)+cos(3*v4)+cos(3*v5)+cos(3*
v6)+cos(3*v7)+cos(3*v8) PERIODIC=NO
```

(a) Page 1

```
3sumsinalpha: MATHEVAL ARG=alpha2,alpha3,alpha4,alpha5,alpha6,alpha7,alpha8
VAR=v2,v3,v4,v5,v6,v7,v8 FUNC=sin(3*v2)+sin(3*v3)+sin(3*v4)+sin(3*v5)+sin(3*
v6)+sin(3*v7)+sin(3*v8) PERIODIC=NO
sumcosbeta: MATHEVAL ARG=beta2,beta3,beta4,beta5,beta6,beta7,beta8 VAR=v2,v3
,v4,v5,v6,v7,v8 FUNC=cos(1*v2)+cos(1*v3)+cos(1*v4)+cos(1*v5)+cos(1*v6)+cos(1
*v7)+cos(1*v8) PERIODIC=NO
sumsinbeta: MATHEVAL ARG=beta2,beta3,beta4,beta5,beta6,beta7,beta8 VAR=v2,v3
,v4,v5,v6,v7,v8 FUNC=sin(1*v2)+sin(1*v3)+sin(1*v4)+sin(1*v5)+sin(1*v6)+sin(1
*v7)+sin(1*v8) PERIODIC=NO
2sumcosbeta: MATHEVAL ARG=beta2,beta3,beta4,beta5,beta6,beta7,beta8 VAR=v2,v
3,v4,v5,v6,v7,v8 FUNC=cos(2*v2)+cos(2*v3)+cos(2*v4)+cos(2*v5)+cos(2*v6)+cos(
2*v7)+cos(2*v8) PERIODIC=NO
2sumsinbeta: MATHEVAL ARG=beta2,beta3,beta4,beta5,beta6,beta7,beta8 VAR=v2,v
3,v4,v5,v6,v7,v8 FUNC=sin(2*v2)+sin(2*v3)+sin(2*v4)+sin(2*v5)+sin(2*v6)+sin(
2*v7)+sin(2*v8) PERIODIC=NO
3sumcosbeta: MATHEVAL ARG=beta2,beta3,beta4,beta5,beta6,beta7,beta8 VAR=v2,v
3,v4,v5,v6,v7,v8 FUNC=cos(3*v2)+cos(3*v3)+cos(3*v4)+cos(3*v5)+cos(3*v6)+cos(
3*v7)+cos(3*v8) PERIODIC=NO
3sumsinbeta: MATHEVAL ARG=beta2,beta3,beta4,beta5,beta6,beta7,beta8 VAR=v2,v
3,v4,v5,v6,v7,v8 FUNC=sin(3*v2)+sin(3*v3)+sin(3*v4)+sin(3*v5)+sin(3*v6)+sin(
3*v7)+sin(3*v8) PERIODIC=NO
sumcosgamma: MATHEVAL ARG=gamma1,gamma2,gamma3,gamma4,gamma5,gamma6,gamma7,g
amma8 VAR=v1,v2,v3,v4,v5,v6,v7,v8 FUNC=cos(1*v1)+cos(1*v2)+cos(1*v3)+cos(1*v
4)+cos(1*v5)+cos(1*v6)+cos(1*v7)+cos(1*v8) PERIODIC=NO
sumsingamma: MATHEVAL ARG=gamma1,gamma2,gamma3,gamma4,gamma5,gamma6,gamma7,g
amma8 VAR=v1,v2,v3,v4,v5,v6,v7,v8 FUNC=sin(1*v1)+sin(1*v2)+sin(1*v3)+sin(1*v
4)+sin(1*v5)+sin(1*v6)+sin(1*v7)+sin(1*v8) PERIODIC=NO
2sumcosgamma: MATHEVAL ARG=gamma1,gamma2,gamma3,gamma4,gamma5,gamma6,gamma7,
gamma8 VAR=v1,v2,v3,v4,v5,v6,v7,v8 FUNC=cos(2*v1)+cos(2*v2)+cos(2*v3)+cos(2*
v4)+cos(2*v5)+cos(2*v6)+cos(2*v7)+cos(2*v8) PERIODIC=NO
2sumsingamma: MATHEVAL ARG=gamma1,gamma2,gamma3,gamma4,gamma5,gamma6,gamma7,
gamma8 VAR=v1,v2,v3,v4,v5,v6,v7,v8 FUNC=sin(2*v1)+sin(2*v2)+sin(2*v3)+sin(2*
v4)+sin(2*v5)+sin(2*v6)+sin(2*v7)+sin(2*v8) PERIODIC=NO
3sumcosgamma: MATHEVAL ARG=gamma1,gamma2,gamma3,gamma4,gamma5,gamma6,gamma7,
gamma8 VAR=v1,v2,v3,v4,v5,v6,v7,v8 FUNC=cos(3*v1)+cos(3*v2)+cos(3*v3)+cos(3*
v4)+cos(3*v5)+cos(3*v6)+cos(3*v7)+cos(3*v8) PERIODIC=NO
3sumsingamma: MATHEVAL ARG=gamma1,gamma2,gamma3,gamma4,gamma5,gamma6,gamma7,
gamma8 VAR=v1,v2,v3,v4,v5,v6,v7,v8 FUNC=sin(3*v1)+sin(3*v2)+sin(3*v3)+sin(3*
v4)+sin(3*v5)+sin(3*v6)+sin(3*v7)+sin(3*v8) PERIODIC=NO
sumcosdelta: MATHEVAL ARG=delta1,delta2,delta3,delta4,delta5,delta6,delta7,d
elta8 VAR=v1,v2,v3,v4,v5,v6,v7,v8 FUNC=cos(1*v1)+cos(1*v2)+cos(1*v3)+cos(1*v
4)+cos(1*v5)+cos(1*v6)+cos(1*v7)+cos(1*v8) PERIODIC=NO
sumsindelta: MATHEVAL ARG=delta1,delta2,delta3,delta4,delta5,delta6,delta7,d
elta8 VAR=v1,v2,v3,v4,v5,v6,v7,v8 FUNC=sin(1*v1)+sin(1*v2)+sin(1*v3)+sin(1*v
4)+sin(1*v5)+sin(1*v6)+sin(1*v7)+sin(1*v8) PERIODIC=NO
2sumcosdelta: MATHEVAL ARG=delta1,delta2,delta3,delta4,delta5,delta6,delta7,
delta8 VAR=v1,v2,v3,v4,v5,v6,v7,v8 FUNC=cos(2*v1)+cos(2*v2)+cos(2*v3)+cos(2*
v4)+cos(2*v5)+cos(2*v6)+cos(2*v7)+cos(2*v8) PERIODIC=NO
2sumsindelta: MATHEVAL ARG=delta1,delta2,delta3,delta4,delta5,delta6,delta7,
delta8 VAR=v1,v2,v3,v4,v5,v6,v7,v8 FUNC=sin(2*v1)+sin(2*v2)+sin(2*v3)+sin(2*
v4)+sin(2*v5)+sin(2*v6)+sin(2*v7)+sin(2*v8) PERIODIC=NO
3sumcosdelta: MATHEVAL ARG=delta1,delta2,delta3,delta4,delta5,delta6,delta7,
delta8 VAR=v1,v2,v3,v4,v5,v6,v7,v8 FUNC=cos(3*v1)+cos(3*v2)+cos(3*v3)+cos(3*
v4)+cos(3*v5)+cos(3*v6)+cos(3*v7)+cos(3*v8) PERIODIC=NO
3sumsindelta: MATHEVAL ARG=delta1,delta2,delta3,delta4,delta5,delta6,delta7,
delta8 VAR=v1,v2,v3,v4,v5,v6,v7,v8 FUNC=sin(3*v1)+sin(3*v2)+sin(3*v3)+sin(3*
v4)+sin(3*v5)+sin(3*v6)+sin(3*v7)+sin(3*v8) PERIODIC=NO
sumcosepsilon: MATHEVAL ARG=epsilon1,epsilon2,epsilon3,epsilon4,epsilon5,eps
ilon6,epsilon7 VAR=v1,v2,v3,v4,v5,v6,v7 FUNC=cos(1*v1)+cos(1*v2)+cos(1*v3)+c
os(1*v4)+cos(1*v5)+cos(1*v6)+cos(1*v7) PERIODIC=NO
sumsinepsilon: MATHEVAL ARG=epsilon1,epsilon2,epsilon3,epsilon4,epsilon5,eps
ilon6,epsilon7 VAR=v1,v2,v3,v4,v5,v6,v7 FUNC=sin(1*v1)+sin(1*v2)+sin(1*v3)+s
in(1*v4)+sin(1*v5)+sin(1*v6)+sin(1*v7) PERIODIC=NO
2sumcosepsilon: MATHEVAL ARG=epsilon1,epsilon2,epsilon3,epsilon4,epsilon5,ep
silon6,epsilon7 VAR=v1,v2,v3,v4,v5,v6,v7 FUNC=cos(2*v1)+cos(2*v2)+cos(2*v3)+
cos(2*v4)+cos(2*v5)+cos(2*v6)+cos(2*v7) PERIODIC=NO
2sumsinepsilon: MATHEVAL ARG=epsilon1,epsilon2,epsilon3,epsilon4,epsilon5,ep
silon6,epsilon7 VAR=v1,v2,v3,v4,v5,v6,v7 FUNC=sin(2*v1)+sin(2*v2)+sin(2*v3)+
```

(b) Page 2

Figure D.6: Plumed Input files for the `GAGA` resampling simulation

```
sin(2*v4)+sin(2*v5)+sin(2*v6)+sin(2*v7) PERIODIC=NO
3sumcosepsilon: MATHEVAL ARG=epsilon1,epsilon2,epsilon3,epsilon4,epsilon5,ep
silon6,epsilon7 VAR=v1,v2,v3,v4,v5,v6,v7 FUNC=cos(3*v1)+cos(3*v2)+cos(3*v3)+
cos(3*v4)+cos(3*v5)+cos(3*v6)+cos(3*v7) PERIODIC=NO
3sumsinepsilon: MATHEVAL ARG=epsilon1,epsilon2,epsilon3,epsilon4,epsilon5,ep
silon6,epsilon7 VAR=v1,v2,v3,v4,v5,v6,v7 FUNC=sin(3*v1)+sin(3*v2)+sin(3*v3)+
sin(3*v4)+sin(3*v5)+sin(3*v6)+sin(3*v7) PERIODIC=NO
sumcoszeta: MATHEVAL ARG=zeta1,zeta2,zeta3,zeta4,zeta5,zeta6,zeta7 VAR=v1,v2
,v3,v4,v5,v6,v7 FUNC=cos(1*v1)+cos(1*v2)+cos(1*v3)+cos(1*v4)+cos(1*v5)+cos(1
*v6)+cos(1*v7) PERIODIC=NO
sumsinzeta: MATHEVAL ARG=zeta1,zeta2,zeta3,zeta4,zeta5,zeta6,zeta7 VAR=v1,v2
,v3,v4,v5,v6,v7 FUNC=sin(1*v1)+sin(1*v2)+sin(1*v3)+sin(1*v4)+sin(1*v5)+sin(1
*v6)+sin(1*v7) PERIODIC=NO
2sumcoszeta: MATHEVAL ARG=zeta1,zeta2,zeta3,zeta4,zeta5,zeta6,zeta7 VAR=v1,v
2,v3,v4,v5,v6,v7 FUNC=cos(2*v1)+cos(2*v2)+cos(2*v3)+cos(2*v4)+cos(2*v5)+cos(
2*v6)+cos(2*v7) PERIODIC=NO
2sumsinzeta: MATHEVAL ARG=zeta1,zeta2,zeta3,zeta4,zeta5,zeta6,zeta7 VAR=v1,v
2,v3,v4,v5,v6,v7 FUNC=sin(2*v1)+sin(2*v2)+sin(2*v3)+sin(2*v4)+sin(2*v5)+sin(
2*v6)+sin(2*v7) PERIODIC=NO
3sumcoszeta: MATHEVAL ARG=zeta1,zeta2,zeta3,zeta4,zeta5,zeta6,zeta7 VAR=v1,v
2,v3,v4,v5,v6,v7 FUNC=cos(3*v1)+cos(3*v2)+cos(3*v3)+cos(3*v4)+cos(3*v5)+cos(
3*v6)+cos(3*v7) PERIODIC=NO
3sumsinzeta: MATHEVAL ARG=zeta1,zeta2,zeta3,zeta4,zeta5,zeta6,zeta7 VAR=v1,v
2,v3,v4,v5,v6,v7 FUNC=sin(3*v1)+sin(3*v2)+sin(3*v3)+sin(3*v4)+sin(3*v5)+sin(
3*v6)+sin(3*v7) PERIODIC=NO
pursumcoschi: MATHEVAL ARG=chi1,chi2,chi3,chi4,chi5,chi6,chi7,chi8 VAR=v1,v2
,v3,v4,v5,v6,v7,v8 FUNC=0*(v1)+0*(v2)+cos(1*v3)+cos(1*v4)+cos(1*v5)+cos(1*v6
)+cos(1*v7)+cos(1*v8) PERIODIC=NO
2pursumcoschi: MATHEVAL ARG=chi1,chi2,chi3,chi4,chi5,chi6,chi7,chi8 VAR=v1,v
2,v3,v4,v5,v6,v7,v8 FUNC=0*(v1)+0*(v2)+cos(2*v3)+cos(2*v4)+cos(2*v5)+cos(2*v
6)+cos(2*v7)+cos(2*v8) PERIODIC=NO
3pursumcoschi: MATHEVAL ARG=chi1,chi2,chi3,chi4,chi5,chi6,chi7,chi8 VAR=v1,v
2,v3,v4,v5,v6,v7,v8 FUNC=0*(v1)+0*(v2)+cos(3*v3)+cos(3*v4)+cos(3*v5)+cos(3*v
6)+cos(3*v7)+cos(3*v8) PERIODIC=NO
pursumsinchi: MATHEVAL ARG=chi1,chi2,chi3,chi4,chi5,chi6,chi7,chi8 VAR=v1,v2
,v3,v4,v5,v6,v7,v8 FUNC=0*(v1)+0*(v2)+sin(1*v3)+sin(1*v4)+sin(1*v5)+sin(1*v6
)+sin(1*v7)+sin(1*v8) PERIODIC=NO
2pursumsinchi: MATHEVAL ARG=chi1,chi2,chi3,chi4,chi5,chi6,chi7,chi8 VAR=v1,v
2,v3,v4,v5,v6,v7,v8 FUNC=0*(v1)+0*(v2)+sin(2*v3)+sin(2*v4)+sin(2*v5)+sin(2*v
6)+sin(2*v7)+sin(2*v8) PERIODIC=NO
3pursumsinchi: MATHEVAL ARG=chi1,chi2,chi3,chi4,chi5,chi6,chi7,chi8 VAR=v1,v
2,v3,v4,v5,v6,v7,v8 FUNC=0*(v1)+0*(v2)+sin(3*v3)+sin(3*v4)+sin(3*v5)+sin(3*v
6)+sin(3*v7)+sin(3*v8) PERIODIC=NO
pyrsumcoschi: MATHEVAL ARG=chi1,chi2,chi3,chi4,chi5,chi6,chi7,chi8 VAR=v1,v2
,v3,v4,v5,v6,v7,v8 FUNC=cos(1*v1)+cos(1*v2)+0*(v3)+0*(v4)+0*(v5)+0*(v6)+0*(v
7)+0*(v8) PERIODIC=NO
2pyrsumcoschi: MATHEVAL ARG=chi1,chi2,chi3,chi4,chi5,chi6,chi7,chi8 VAR=v1,v
2,v3,v4,v5,v6,v7,v8 FUNC=cos(2*v1)+cos(2*v2)+0*(v3)+0*(v4)+0*(v5)+0*(v6)+0*(
v7)+0*(v8) PERIODIC=NO
3pyrsumcoschi: MATHEVAL ARG=chi1,chi2,chi3,chi4,chi5,chi6,chi7,chi8 VAR=v1,v
2,v3,v4,v5,v6,v7,v8 FUNC=cos(3*v1)+cos(3*v2)+0*(v3)+0*(v4)+0*(v5)+0*(v6)+0*(
v7)+0*(v8) PERIODIC=NO
pyrsumsinchi: MATHEVAL ARG=chi1,chi2,chi3,chi4,chi5,chi6,chi7,chi8 VAR=v1,v2
,v3,v4,v5,v6,v7,v8 FUNC=sin(1*v1)+sin(1*v2)+0*(v3)+0*(v4)+0*(v5)+0*(v6)+0*(v
7)+0*(v8) PERIODIC=NO
2pyrsumsinchi: MATHEVAL ARG=chi1,chi2,chi3,chi4,chi5,chi6,chi7,chi8 VAR=v1,v
2,v3,v4,v5,v6,v7,v8 FUNC=sin(2*v1)+sin(2*v2)+0*(v3)+0*(v4)+0*(v5)+0*(v6)+0*(
v7)+0*(v8) PERIODIC=NO
3pyrsumsinchi: MATHEVAL ARG=chi1,chi2,chi3,chi4,chi5,chi6,chi7,chi8 VAR=v1,v
2,v3,v4,v5,v6,v7,v8 FUNC=sin(3*v1)+sin(3*v2)+0*(v3)+0*(v4)+0*(v5)+0*(v6)+0*(
v7)+0*(v8) PERIODIC=NO
kbt: CONSTANT VALUE=2.5
mybias: MATHEVAL ARG=kbt,sumcosalpha,sumcosbeta,sumcosgamma,sumcosdelta,sumc
osepsilon,sumcoszeta,sumsinalpha,sumsinbeta,sumsingamma,sumsindelta,sumsinep
silon,sumsinzeta,pursumcoschi,pyrsumcoschi,pursumsinchi,pyrsumsinchi,2sumcos
alpha,2sumcosbeta,2sumcosgamma,2sumcosdelta,2sumcosepsilon,2sumcoszeta,2sums
inalpha,2sumsinbeta,2sumsingamma,2sumsindelta,2sumsinepsilon,2sumsinzeta,2pu
rsumcoschi,2pyrsumcoschi,2pursumsinchi,2pyrsumsinchi,3sumcosalpha,3sumcosbet
```

(a) Page 3

```
a,3sumcosgamma,3sumcosdelta,3sumcosepsilon,3sumcoszeta,3sumsinalpha,3sumsinb
eta,3sumsingamma,3sumsindelta,3sumsinepsilon,3sumsinzeta,3pursumcoschi,3pyrs
umcoschi,3pursumsinchi,3pyrsumsinchi VAR=kbt,v1,v2,v3,v4,v5,v6,v7,v8,v9,v10,
v11,v12,v13,v14,v15,v16,v17,v18,v19,v20,v21,v22,v23,v24,v25,v26,v27,v28,v29,
v30,v31,v32,v33,v34,v35,v36,v37,v38,v39,v40,v41,v42,v43,v44,v45,v46,v47,v48
FUNC=kbt*((0.028743*v1)+(-0.118088*v2)+(0.369240*v3)+(-0.068875*v4)+(-0.0617
79*v5)+(-0.083563*v6)+(0.045960*v7)+(-0.085707*v8)+(0.080645*v9)+(-0.011406*
v10)+(0.038967*v11)+(0.220101*v12)+(-0.007880*v13)+(0.067727*v14)+(-0.086418
*v15)+(-0.014430*v16)+(-0.135683*v17)+(0.199287*v18)+(-0.269494*v19)+(0.0358
08*v20)+(0.086680*v21)+(0.091654*v22)+(0.064641*v23)+(0.008920*v24)+(0.19077
1*v25)+(-0.109477*v26)+(0.054064*v27)+(0.150732*v28)+(-0.054599*v29)+(-0.116
106*v30)+(-0.073151*v31)+(-0.046067*v32)+(0.122662*v33)+(-0.064958*v34)+(-0.
001949*v35)+(0.109930*v36)+(0.042434*v37)+(0.061681*v38)+(0.032150*v39)+(0.0
16087*v40)+(0.020993*v41)+(0.051061*v42)+(-0.077758*v43)+(0.015956*v44)+(0.0
16977*v45)+(0.037457*v46)+(0.143994*v47)+(0.140334*v48)) PERIODIC=NO
BIASVALUE ARG=mybias
```

(b) Page 4

Figure D.7: Plumed Input files for the `GAGA` resampling simulation

References

[1] Dror, R. O.; Dirks, R. M.; Grossman, J.; Xu, H.; Shaw, D. E. *Annu Rev Biophys* **2012**, *41*, 429–452.

[2] Shaw, D. E.; Dror, R. O.; Salmon, J. K.; Grossman, J. P.; Mackenzie, K. M.; Bank, J. A.; Young, C.; Deneroff, M. M.; Batson, B.; Bowers, K. J.; Chow, E.; Eastwood, M. P.; Ierardi, D. J.; Klepeis, J. L.; Kuskin, J. S.; Larson, R. H.; Lindorff-Larsen, K.; Maragakis, P.; Moraes, M. A.; Piana, S.; Shan, Y.; Towles, B. In *Proceedings of the Conference on High Performance Computing Networking, Storage and Analysis*; pp 1–11.

[3] Bernardi, R. C.; Melo, M. C.; Schulten, K. *Biochim. Biophys. Acta, Gen. Subj.* **2015**, *1850*, 872–877.

[4] Valsson, O.; Tiwary, P.; Parrinello, M. *Annu. Rev. Phys. Chem.* **2016**, *67*, 159–184.

[5] Mlýnský, V.; Bussi, G. *arXiv preprint arXiv:1709.02342* **2017**.

[6] Petrov, D.; Zagrovic, B. *PLoS Comput. Biol.* **2014**, *10*, e1003638.

[7] Piana, S.; Lindorff-Larsen, K.; Shaw, D. E. *Biophys. J.* **2011**, *100*, L47–L49.

[8] Condon, D. E.; Kennedy, S. D.; Mort, B. C.; Kierzek, R.; Yildirim, I.; Turner, D. H. *J Chem Theory Comput* **2015**, *11*, 2729–2742.

[9] Bergonzo, C.; Henriksen, N. M.; Roe, D. R.; Cheatham, T. E. *RNA* **2015**, *21*, 1578–1590.

[10] Šponer, J.; Bussi, G.; Krepl, M.; Banáš, P.; Bottaro, S.; Cunha, R. A.; Gil-Ley, A.; Pinamonti, G.; Poblete, S.; Jurečka, P.; Walter, N. G.; Otyepka, M.; *RNA Structural Dynamics As Captured by Molecular Simulations: A Comprehensive Overview*; 2018.

[11] Kuhrová, P.; Best, R. B.; Bottaro, S.; Bussi, G.; Šponer, J.; Otyepka, M.; Banáš, P. *J Chem Theory Comput* **2016**, *12*, 4534–4548.

[12] Bottaro, S.; Banáš, P.; Šponer, J.; Bussi, G. *J. Phys. Chem. Lett.* **2016**, *7*, 4032–4038.

[13] Schröder, G. F. *Curr. Opin. Struct. Biol.* **2015**, *31*, 20–27.

[14] Allison, J. R. *Curr. Opin. Struct. Biol.* **2017**, *43*, 79–87.

[15] Bonomi, M.; Heller, G. T.; Camilloni, C.; Vendruscolo, M. *Curr. Opin. Struct. Biol.* **2017**, *42*, 106–116.

[16] Rangan, R.; Bonomi, M.; Heller, G. T.; Cesari, A.; Bussi, G.; Vendruscolo, M. *In review.*

[17] Cesari, A.; Gil-Ley, A.; Bussi, G. *J. Chem. Theory Comput.* **2016**, *12*, 6192–6200.

[18] Cesari, A.; Reißer, S.; Bussi, G. *Computation* **2018**, *6*.

[19] Jaynes, E. T. *Phys. Rev.* **1957**, *106*, 620.

[20] Jaynes, E. T. *Phys. Rev.* **1957**, *108*, 171.

[21] Banavar, J.; Maritan, A. *arXiv preprint cond-mat/0703622* **2007**.

[22] Shell, M. S. *J Chem Phys* **2008**, *129*, 144108.

[23] Kullback, S.; Leibler, R. A. *Ann. Math. Statist.* **1951**, *22*, 79–86.

[24] Dannenhoffer-Lafage, T.; White, A. D.; Voth, G. A. *J Chem Theory Comput* **2016**, *12*, 2144–2153.

[25] Ryckaert, J.-P.; Ciccotti, G.; Berendsen, H. J. *J. Comput. Phys.* **1977**, *23*, 327–341.

[26] Case, D. A. *Curr. Opin. Struct. Biol.* **2013**, *23*, 172–176.

[27] Karplus, M. *J Am Chem Soc* **1963**, *85*, 2870–2871.

[28] Tolman, J. R.; Ruan, K. *Chem. Rev.* **2006**, *106*, 1720–1736.

[29] Bernadó, P.; Mylonas, E.; Petoukhov, M. V.; Blackledge, M.; Svergun, D. I. *J. Am. Chem. Soc.* **2007**, *129*, 5656–5664.

[30] Jeschke, G. *Annu. Rev. Phys. Chem.* **2012**, *63*, 419–446.

[31] Piston, D. W.; Kremers, G.-J. *Trends Biochem. Sci.* **2007**, *32*, 407–414.

[32] Mead, L. R.; Papanicolaou, N. *J Math Phys* **1984**, *25*, 2404–2417.

[33] Pitera, J. W.; Chodera, J. D. *J Chem Theory Comput* **2012**, *8*, 3445–3451.

[34] Berger, A. L.; Pietra, V. J. D.; Pietra, S. A. D. *Comput. Linguist.* **1996**, *22*, 39–71.

[35] Chen, S. F.; Rosenfeld, R.; *A Gaussian prior for smoothing maximum entropy models*; Tech. Rep.; 1999.

[36] Fennen, J.; Torda, A. E.; van Gunsteren, W. F. *J. Biomol. NMR* **1995**, *6*, 163–170.

[37] Best, R. B.; Vendruscolo, M. *J Am Chem Soc* **2004**, *126*, 8090–8091.

[38] Lindorff-Larsen, K.; Best, R. B.; DePristo, M. A.; Dobson, C. M.; Vendruscolo, M. *Nature* **2005**, *433*, 128–132.

[39] Cavalli, A.; Camilloni, C.; Vendruscolo, M. *J Chem Phys* **2013**, *138*, 094112.

[40] Roux, B.; Weare, J. *J Chem Phys* **2013**, *138*, 084107.

[41] Olsson, S.; Cavalli, A. *J. Chem. Theory Comput.* **2015**, *11*, 3973–3977.

[42] Olsson, S.; Ekonomiuk, D.; Sgrignani, J.; Cavalli, A. *J. Am. Chem. Soc.* **2015**, *137*, 6270–6278.

[43] Camilloni, C.; Vendruscolo, M. *J. Phys. Chem. B* **2014**, *119*, 653–661.

[44] Beauchamp, K. A.; Pande, V. S.; Das, R. *Biophys J* **2014**, *106*, 1381–1390.

[45] Hummer, G.; Köfinger, J. *J Chem Phys* **2015**, *143*, 243150.

[46] Bonomi, M.; Camilloni, C.; Cavalli, A.; Vendruscolo, M. *Sci Adv* **2016**, *2*, e1501177–e1501177.

[47] Brookes, D. H.; Head-Gordon, T. *J Am Chem Soc* **2016**, *138*, 4530–4538.

[48] Różycki, B.; Kim, Y. C.; Hummer, G. *Structure* **2011**, *19*, 109–116.

[49] Boura, E.; Różycki, B.; Herrick, D. Z.; Chung, H. S.; Vecer, J.; Eaton, W. A.; Cafiso, D. S.; Hummer, G.; Hurley, J. H. *Proc. Natl. Acad. Sci. U.S.A.* **2011**, *108*, 9437–9442.

[50] Sanchez-Martinez, M.; Crehuet, R. *Phys. Chem. Chem. Phys.* **2014**, *16*, 26030–26039.

[51] Leung, H. T. A.; Bignucolo, O.; Aregger, R.; Dames, S. A.; Mazur, A.; Bernéche, S.; Grzesiek, S. *J. Chem. Theory Comput.* **2015**, *12*, 383–394.

[52] Cunha, R. A.; Bussi, G. *RNA* **2017**, *23*, 628–638.

[53] Bottaro, S.; Bussi, G.; Kennedy, S. D.; Turner, D. H.; Lindorff-Larsen, K. *bioRxiv* **2017**, 230268.

[54] Podbevsek, P.; Fasolo, F.; Bon, C.; Cimatti, L.; Reisser, S.; Carninci, P.; Bussi, G.; Zucchelli, S.; Plavec, J.; Gustincich, S.; *Structural determinants of the SINEB2 element embedded in the long non-coding RNA activator of translation AS Uchl1*; To Be Published.

[55] Shen, T.; Hamelberg, D. *J Chem Phys* **2008**, *129*, 034103.

[56] Gray, P. G.; Kish, L. *Journal of the Royal Statistical Society. Series A (General)* **1969**, *132*, 272.

[57] Reith, D.; Pütz, M.; Müller-Plathe, F. *J. Comput. Chem.* **2003**, *24*, 1624–1636.

[58] Norgaard, A. B.; Ferkinghoff-Borg, J.; Lindorff-Larsen, K. *Biophys. J.* **2008**, *94*, 182–192.

[59] Giorgetti, L.; Galupa, R.; Nora, E. P.; Piolot, T.; Lam, F.; Dekker, J.; Tiana, G.; Heard, E. *Cell* **2014**, *157*, 950–963.

[60] Tiana, G.; Amitai, A.; Pollex, T.; Piolot, T.; Holcman, D.; Heard, E.; Giorgetti, L. *Biophys. J.* **2016**, *110*, 1234–1245.

[61] Zhang, B.; Wolynes, P. G. *Proc. Natl. Acad. Sci. U.S.A.* **2015**, *112*, 6062–6067.

[62] Zhang, B.; Wolynes, P. G. *Phys. Rev. Lett.* **2016**, *116*, 248101.

[63] Darken, C.; Moody, J. *NIPs* **1991**, 1009–1016.

[64] White, A. D.; Dama, J. F.; Voth, G. A. *J. Chem. Theory Comput.* **2015**, *11*, 2451–2460.

[65] Marinelli, F.; Faraldo-Gómez, J. D. *Biophys. J.* **2015**, *108*, 2779–2782.

[66] Gil-Ley, A.; Bottaro, S.; Bussi, G. *J Chem Theory Comput* **2016**, *12*, 2790–2798.

[67] Valsson, O.; Parrinello, M. *Phys Rev Lett* **2014**, *113*, 090601.

[68] Bach, F.; Moulines, E. In *Advances in neural information processing systems*; pp 773–781.

[69] White, A. D.; Voth, G. A. *J Chem Theory Comput* **2014**, *10*, 3023–3030.

[70] Duchi, J.; Hazan, E.; Singer, Y. *J. Mach. Learn. Res.* **2011**, *12*, 2121–2159.

[71] Hocky, G. M.; Dannenhoffer-Lafage, T.; Voth, G. A. *J. Chem. Theory Comput.* **2017**, *13*, 4593–4603.

[72] White, A. D.; Knight, C.; Hocky, G. M.; Voth, G. A. *J. Chem. Phys* **2017**, *146*, 041102.

[73] TS'O, P. O. In *Basic Principles in Nucleic Acid Chemistry*; TS'O, P. O., Ed.; Academic Press, 1974; pp 453 – 584.

[74] Saenger, W. *Principles of Nucleic Acid Structure*; Springer advanced texts in chemistry; Springer, 1984.

[75] Neidle, S. In *Principles of Nucleic Acid Structure*; Neidle, S., Ed.; Academic Press: New York, 2008; pp 20 – 37.

[76] Neidle, S. In *Principles of Nucleic Acid Structure*; Neidle, S., Ed.; Academic Press: New York, 2008; pp 204 – 248.

[77] Haasnoot, C.; de Leeuw, F.; Altona, C. *Tetrahedron* **1980**, *36*, 2783–2792.

[78] Davies, D. B. *Prog Nucl Magn Reson Spectrosc* **1978**, *12*, 135–225.

[79] Lankhorst, P. P.; Haasnoot, C. A.; Erkelens, C.; Altona, C. *J Biomol Struct Dyn* **1984**, *1*, 1387–1405.

[80] Mooren, M. M.; Wijmenga, S. S.; van der Marel, G. A.; van Boom, J. H.; Hilbers, C. W. *Nucleic Acids Res* **1994**, *22*, 2658–2666.

[81] Lee, C.-H.; Sarma, R. H. *J Am Chem Soc* **1976**, *98*, 3541–3548.

[82] Wijmenga, S. S.; van Buuren, B. N. *Prog Nucl Magn Reson Spectrosc* **1998**, *32*, 287–387.

[83] Marino, J. P.; Schwalbe, H.; Griesinger, C. *Acc Chem Res* **1999**, *32*, 614–623.

[84] Pronk, S.; Pall, S.; Schulz, R.; Larsson, P.; Bjelkmar, P.; Apostolov, R.; Shirts, M. R.; Smith, J. C.; Kasson, P. M.; van der Spoel, D.; Hess, B.; Lindahl, E. *Bioinformatics* **2013**, *29*, 845–854.

[85] Tribello, G. A.; Bonomi, M.; Branduardi, D.; Camilloni, C.; Bussi, G. *Comput Phys Commun* **2014**, *185*, 604–613.

[86] Jorgensen, W. L.; Chandrasekhar, J.; Madura, J. D.; Impey, R. W.; Klein, M. L. *J Chem Phys* **1983**, *79*, 926.

[87] Cornell, W. D.; Cieplak, P.; Bayly, C. I.; Gould, I. R.; Merz, K. M.; Ferguson, D. M.; Spellmeyer, D. C.; Fox, T.; Caldwell, J. W.; Kollman, P. A. *J Am Chem Soc* **1995**, *117*, 5179–5197.

[88] Pérez, A.; Marchán, I.; Svozil, D.; Sponer, J.; Cheatham, T. E.; Laughton, C. A.; Orozco, M. *Biophys J* **2007**, *92*, 3817–3829.

[89] Joung, I. S.; Cheatham, T. E. *J Phys Chem B* **2008**, *112*, 9020–9041.

[90] Banáš, P.; Hollas, D.; Zgarbová, M.; Jurečka, P.; Orozco, M.; Cheatham, T. E.; Šponer, J.; Otyepka, M. *J Chem Theory Comput* **2010**, *6*, 3836–3849.

[91] Zgarbová, M.; Otyepka, M.; Šponer, J.; Mládek, A.; Banáš, P.; Cheatham, T. E.; Jurečka, P. *J Chem Theory Comput* **2011**, *7*, 2886–2902.

[92] Hess, B.; Bekker, H.; Berendsen, H. J. C.; Fraaije, J. G. E. M. *J Comput Chem* **1997**, *18*, 1463–1472.

[93] Darden, T.; York, D.; Pedersen, L. *J Chem Phys* **1993**, *98*, 10089–10092.

[94] Parrinello, M.; Rahman, A. *Phys Rev Lett* **1980**, *45*, 1196–1199.

[95] Bussi, G.; Donadio, D.; Parrinello, M. *J Chem Phys* **2007**, *126*, 014101.

[96] Gil-Ley, A.; Bussi, G. *J Chem Theory Comput* **2015**, *11*, 1077–1085.

[97] Huang, M.; Giese, T. J.; Lee, T.-S.; York, D. M. *J Chem Theory Comput* **2014**, *10*, 1538–1545.

[98] Ancian, B. In *Annual Reports on NMR Spectroscopy*; Elsevier BV, 2010; pp 39–143.

[99] Bottaro, S.; Gil-Ley, A.; Bussi, G. *Nucleic Acids Res* **2016**, *44*, 5883–5891.

[100] Yildirim, I.; Stern, H. A.; Tubbs, J. D.; Kennedy, S. D.; Turner, D. H. *J Phys Chem B* **2011**, *115*, 9261–9270.

[101] Ceriotti, M.; Brain, G. A.; Riordan, O.; Manolopoulos, D. E. *Proc. R. Soc. A* **2012**, *468*, 2–17.

[102] Rieping, W. *Science* **2005**, *309*, 303–306.

[103] Olsson, S.; Frellsen, J.; Boomsma, W.; Mardia, K. V.; Hamelryck, T. *PLoS ONE* **2013**, *8*, e79439.

[104] Bayly, C. I.; Cieplak, P.; Cornell, W.; Kollman, P. A. *J Phys Chem* **1993**, *97*, 10269–10280.

[105] Ivani, I.; Dans, P. D.; Noy, A.; Pérez, A.; Faustino, I.; Hospital, A.; Walther, J.; Andrio, P.; Goñi, R.; Balaceanu, A.; Portella, G.; Battistini, F.; Gelpí, J. L.; González, C.; Vendruscolo, M.; Laughton, C. A.; Harris, S. A.; Case, D. A.; Orozco, M. *Nat Methods* **2015**, *13*, 55–58.

[106] Bottaro, S.; Bussi, G.; Kennedy, S. D.; Turner, D. H.; Lindorff-Larsen, K. *Science Advances* **2018**, *4*.

[107] Steinbrecher, T.; Latzer, J.; Case, D. A. *Journal of Chemical Theory and Computation* **2012**, *8*, 4405–4412; PMID: 23264757.

[108] Izadi, S.; Anandakrishnan, R.; Onufriev, A. V. *The Journal of Physical Chemistry Letters* **2014**, *5*, 3863–3871; PMID: 25400877.

[109] Bottaro, S.; DiÂ Palma, F.; Bussi, G. *Nucleic Acids Research* **2014**, *42*, 13306–13314.

[110] Tubbs, J. D.; Condon, D. E.; Kennedy, S. D.; Hauser, M.; Bevilacqua, P. C.; Turner, D. H. *Biochemistry* **2013**, *52*, 996–1010.

[111] Tan, D.; Piana, S.; Dirks, R. M.; Shaw, D. E. *Proceedings of the National Academy of Sciences* **2018**.

[112] Kumar, S.; Bouzida, D.; Swendsen, R. H.; Kollman, P. A.; Rosenberg, J. M. *J. Comput. Chem.* **1992**, *13*, 1011–1021.

[113] Norgaard, A. B.; Ferkinghoff-Borg, J.; Lindorff-Larsen, K. *Biophysical Journal* **2008**, *94*, 182 – 192.

[114] Bottaro, S.; Bussi, G.; Pinamonti, G.; Reisser, S.; Boomsma, W.; Lindorff-Larsen, K. *bioRxiv* **2018**.

[115] Kuhrova, P.; Mlynsky, V.; Zgarbova, M.; Krepl, M.; Bussi, G.; Best, R. B.; Otyepka, M.; Sponer, J.; Banas, P. *bioRxiv* **2018**.

[116] Lee, C.-H.; Ezra, F. S.; Kondo, N. S.; Sarma, R. H.; Danyluk, S. S. *Biochemistry (Mosc)* **1976**, *15*, 3627–3639.

[117] Ezra, F. S.; Lee, C.-H.; Kondo, N. S.; Danyluk, S. S.; Sarma, R. H. *Biochemistry (Mosc)* **1977**, *16*, 1977–1987.

[118] Vokáčová, Z.; Buděšinský, M.; Rosenberg, I.; Schneider, B.; Šponer, J.; Sychrovský, V. *J Phys Chem B* **2009**, *113*, 1182–1191.

[119] Ippel, J. H.; Wijmenga, S. S.; de Jong, R.; Heus, H. A.; Hilbers, C. W.; de Vroom, E.; van der Marel, G. A.; van Boom, J. H. *Magn Reson Chem* **1996**, *34*, S156–S176.