SISSA
Scuola
Internazionale
Superiore di
Studi Avanzati

Physics Area - PhD course in
Molecular and Statistical Biophysics

Development and application of methods to integrate molecular simulations with experimental measurements

Candidate:
Ivan Gilardoni

Advisor:
Prof. Giovanni Bussi

Academic Year 2024-25

26th September 2025

Acknowledgments

This PhD Thesis represents for me a great reward after an unforgettable four-year period of intense growth, both professionally and personally. I am deeply grateful to all the people I met along this journey. In each of them I have always found friendship, helping hands, and supportive words. Although it would be impossible to mention everyone, I would like in particular to thank the following.

First of all, my supervisor, Prof. Giovanni Bussi, for his constant optimism and trust in all of his students. I am truly grateful for his guidance, patience, and unwavering support throughout these years, as well as for his advice and careful proofreading of my papers and this Thesis.

All the members of the BussiLab (Thorben Fröhlking, Valerio Piomponi, Elisa Posani, Salvo Di Marco, Giuseppe Sacco, Maurizio Gilioli, Davide Di Vora, Olivier Languin-Cattöen, Tomàs Silva, and Korbinian Liebl) for their collaboration, insightful discussions, and friendship. I also thank the Molecular and Statistical Biophysics group (professors and colleagues) for fostering such a stimulating and welcoming scientific environment – a reminder that science is made first and foremost through human relationships.

A special thanks to my colleagues and friends at SISSA and beyond (Elisa Posani and Pieter van der Hoek, Vittorio Del Tatto and Marta Monti, Davide Marcato and Giada Brentaro, Riccardo Rende, Francesco Slongo, Matteo Tajana, Enrico Fornasa, and many others) for the countless shared moments, both inside and outside the institute. I would also like to remember Samuel Tamagnone, a dear PhD student and friend who loved the mountains as much as he loved science.

Finally, I would like to thank my family and friends in my hometown. Being far from you has been perhaps the only aspect I disliked about living in Trieste, but it has also made me appreciate even more your constant presence and support. Hopefully, a faster railway connection will make life easier for my Insubre successors at SISSA.

Abstract

Molecular dynamics (MD) simulations are a cornerstone of computational biophysics, providing atomistic insight into the conformational dynamics of complex biomolecules. Despite their predictive power, the reliability of MD simulations is hindered by two key limitations: limited sampling of long-timescale events and inaccuracies in the underlying force fields. To ensure meaningful predictions, simulations must therefore be rigorously validated and refined against experimental data.

This thesis addresses these challenges by advancing the integration of experimental measurements into MD simulations within a unified, Bayesian-inspired framework. Building on recent efforts to simultaneously refine structural ensembles and forward models, we extend the scope to include force-field corrections, enabling a synergistic optimization that improves both sampling consistency and force-field accuracy. To this aim, we design and implemented `MDRefine`, a Python package that provides a flexible platform for the refinement of ensembles, force fields, and forward models within a single workflow. The package incorporates tunable hyperparameters to balance experimental constraints and prior knowledge, and its effectiveness is demonstrated in benchmark applications to RNA oligomers.

Furthermore, we revisit ensemble refinement from a fully Bayesian perspective, moving beyond maximum-a-posteriori estimates to emphasize the importance of sampling the full posterior distribution. In this context, we identify a fundamental normalization issue in existing approaches and resolve it by introducing the Jeffreys uninformative prior, which has been implemented in `MDRefine`.

The methodological advances presented in this thesis contribute to a more robust integration of simulations and experiments, advancing the efforts toward more accurate MD simulations. Applications to nucleic acids demonstrate both the practical utility of the framework and its potential to be generalized to diverse molecular systems.

Contents

Introduction **5**

1 Background **9**
1.1 A pill of structural biology 11
1.1.1 RNA 11
1.1.2 Structure-function relationship 15
1.2 Experimental techniques 16
1.3 Computational techniques 19
1.3.1 Molecular dynamics (MD) simulations 21
1.3.2 Neural-network potentials 25
1.3.3 Predicting structures with deep learning 27
1.4 Integrative methods 30
1.4.1 Ensemble refinement by maximum entropy 31
1.4.2 Force-field refinement 37
1.4.3 Forward-model refinement and combined approaches . 38
1.4.4 Available packages 40

2 Boosting ensemble refinement with transferable force-field corrections **41**
2.1 Methods 42
2.1.1 Interpretation of the hyperparameters 44
2.1.2 Generalization to multiple systems 45
2.1.3 Minimization strategy 46
2.1.4 Calculations using reweighting 48
2.1.5 Cross validation 49
2.2 Toy model 51
2.3 Application to RNA oligomers 54
2.3.1 RNA oligomers under study 54
2.3.2 Application of the method 57
2.4 Discussion 67

3 `MDRefine`: a Python package for refining MD trajectories with experimental data **70**
3.1 `MDRefine` package: materials and methods . . . 71
3.1.1 Structure of `data` objects . . . 72
3.1.2 Basic calculations and reweighting . . . 75
3.1.3 Loss function and search for optimal solutions . . . 77
3.1.4 Searching for optimal hyperparameters via cross validation 78
3.1.5 The main function: `MDRefinement` . . . 80
3.1.6 Limitations and good practices . . . 82
3.2 Results . . . 83
3.2.1 Force-field refinement with denaturation experiments . 85
3.2.2 Simultaneous refinement of forward model and structural ensembles . . . 89
3.2.3 Fully combined refinement . . . 93
3.3 Discussion . . . 98

4 Bayesian sampling of the posterior distribution **100**
4.1 Methods . . . 101
4.1.1 BELT (Bayesian energy landscape tilting) . . . 102
4.1.2 Jeffreys uninformative prior . . . 103
4.1.3 Importance of a restricted functional form . . . 105
4.1.4 Alternative ways for ensemble counting . . . 106
4.2 Results . . . 108
4.2.1 Quadratic potential . . . 108
4.2.2 Realistic case: refinement of RNA oligomers . . . 114
4.3 Discussion . . . 119

Conclusions and perspectives **121**

A Kish sample size vs. entropy **123**
A.1 Effects of coarse-graining . . . 124

B Derivatives of the loss function **127**
B.1 Computation of the derivatives . . . 127
B.2 Sign of the derivatives . . . 130

C Supplementary for `MDRefine` **135**
C.1 Implementation of `minimizer` . . . 135
C.2 Fully combined refinement – comparison with test set . . . 136

Introduction

Molecular dynamics (MD) simulations in explicit solvent are nowadays an essential tool for unraveling the conformational dynamics of molecular systems. They offer a powerful mean to investigate the structure and dynamics of complex biomolecules, functioning as a "virtual microscope" that allows to visualize and predict biophysical processes at atomistic resolution. In this capacity, MD simulations complement experimental studies by providing detailed mechanistic insights. However, limitations in sampling efficiency and inaccuracies in the employed potential-energy functions – namely, the force fields – can compromise the reliability of the predictions.

The limitation of MD simulations in sampling efficiency is related to the accessible timescale, typically restricted to the microsecond regime. Although specialized hardware[1] can extend simulations into the millisecond range, such resources are not always available. To overcome this, a variety of enhanced-sampling methods have been developed, enabling the exploration of rare events that would otherwise take place too infrequently to be observed within the simulation times.

Yet, simulation length is only one of the two factors contributing to the applicability of MD simulations. Equally crucial is the reliability of the underlying force field, which describes the physical interactions within the concerned molecular system. Therefore, rigorous validation of MD results against experimental measurements is necessary. To this aim, observables are computed from simulations (through the so-called forward models) and compared to the measured quantities. The agreement between calculated and experimental values supports the validity of the simulation, allowing for confident prediction of other observables. Conversely, significant discrepancies require a careful analysis of their origin.

[1]Notable cases include: Anton, a supercomputer specifically designed and built by D. E. Shaw Research in New York with the purpose of running MD simulations; Frontier, the world's first exascale supercomputer, hosted at the Oak Ridge Leadership Computing Facility in Tennessee; Jupiter, the first exascale supercomputer in Europe, hosted at the Julich Research Centre in Germany.

These inconsistencies may derive from either the employed forward models – often involving empirical parameters – or the MD simulation itself. In the latter case, errors may arise from insufficient sampling, which can be mitigated through enhanced-sampling techniques, or from deficiencies in the force field. Force-field refinement is a complex process due to the interdependence of its parameters, where altering one of them can unpredictably affect the others. Moreover, even an extensively refined force field may fall short of capturing the true physics of the system, due to the intrinsic limitations of the employed functional form and to the usually neglected quantum nuclear effects.

To address these challenges, integrating experimental data into MD simulations has become a widely accepted solution. This integration enhances the accuracy of simulations by constraining them to match experimental observations. At the same time, it improves the interpretability of experimental data by providing a detailed microscopic framework through which macroscopic phenomena can be understood.

This approach leverages existing experimental knowledge and shifts the paradigm from bottom-up refinement based on quantum-chemistry calculations to a more pragmatic, top-down integration of empirical data. Importantly, while quantum chemistry typically targets small molecular fragments, many experiments probe full-length biomolecules under physiological conditions.

Care must also be taken when interpreting ensemble-averaged experimental data, which may hide structural heterogeneity. Complex molecular systems exhibit several conformations structurally different among each other, so these average values reflect contributions from multiple coexisting states, each with different populations. When combined carefully with MD simulations, such data can help reconstruct a high-resolution, ensemble-based picture of molecular structure and dynamics.

Several integrative methods have been proposed in the literature, each aiming at a specific aspect of refinement, such as structural ensembles, force fields or forward models. Traditionally, researchers are expected to select the most appropriate technique for their particular case of interest, relying on personal expertise and applying these methods independently from each other. However, the sources of error are inherently linked across ensembles, force fields and forward models. Consequently, combined approaches that address these interdependencies should be preferable.

A notable step in this direction was made by T. Fröhlking, a PhD graduate from our group, who proposed [1] a combined refinement of ensembles and forward models. Building upon this base, this Thesis advances the integration further by incorporating force-field fitting beside ensemble refinement, as discussed in Chapter 2.

In addition, we present the development of a unified strategy, implemented as a Python package named `MDRefine`, described in Chapter 3. This tool enables seamless interpolation between the refinement of ensembles, force fields and forward models within a single framework. The method can be tuned via some hyperparameters, regulating the confidence to experimental measurements and previous knowledge of the molecular systems, and is demonstrated on benchmark cases involving RNA oligomers.

Finally, Chapter 4 focuses on a fully Bayesian treatment of the loss function, beyond maximum-a-posteriori estimate. We highlight the critical role of proper sampling to avoid issues related to normalization. To this aim, we propose the use of the Jeffreys "uninformative" prior, which we have implemented in `MDRefine`.

The results discussed in this Thesis are largely based on the following publications:

- I. Gilardoni, T. Fröhlking, and G. Bussi, *Boosting ensemble refinement with transferable force-ield corrections: synergistic optimization for molecular simulations*, J. Phys. Chem. Lett. 15, 1204–1210 (2024).
- I. Gilardoni, V. Piomponi, T. Fröhlking, and G. Bussi, ***`MDRefine`**: a Python package for refining molecular dynamics trajectories with experimental data*, J. Chem. Phys. 162, 192501 1-13 (2025).
- I. Gilardoni and G. Bussi, *Bayesian ensemble refinement revisited: the entropy matters* (in preparation).

While pursuing my PhD, I also participated in several CECAM workshops and schools. In particular, during the CECAM workshop *Machine-learned potentials in molecular simulation: best practices and tutorials*, held in Vienna in July 2023, I contributed to developing Python tutorials for applications of unsupervised learning in molecular dynamics. They are available on GitHub[2] and described in the manuscript (under submission):

- R. Barrett, A. Brahmasandra, B. Cheng, G. Fonseca, I. Gilardoni, R. Mercado, T. Oesterich, M. Pinheiro Jr and H. Yazdani, *Best practices for unsupervised learning in molecular systems.*

[2] https://github.com/rhyan10/unsupervised-learning-molecular-systems

This Thesis does not include my effort in the supervision of a Master's student, Davide Di Vora (Università degli Studi di Trieste), for a duration of four months. In his Master's project, Davide focused on developing on-the-fly methods for the simultaneous refinement of structural ensembles, force fields and forward models. His work extended the applicability of the approaches presented in this Thesis to cases where the overlap between prior and experimental ensembles is limited.

Chapter 1

Background

Biological molecules, such as proteins and nucleic acids, exhibit complex structural behavior that plays a fundamental role in their biological functions. Understanding their conformational dynamics at atomic resolution is essential for unraveling the molecular mechanisms underlying key cellular processes. Experimental techniques offered by structural biology – such as X-ray crystallography, nuclear-magnetic-resonance (NMR) spectroscopy, and cryo-electron microscopy – have made remarkable steps in characterizing biomolecular structures. However, they often fall short in capturing the temporal or spatial resolution necessary for a complete picture.

In this landscape, computational approaches, particularly atomistic molecular dynamics (MD) simulations, have emerged as a powerful complementary tool, generating high-resolution trajectories that describe molecular motions in detail. However, the predictive power of MD simulations critically depends on the accuracy of the underlying force fields that model atomic interactions. While modern force fields have been carefully parametrized and refined over decades, they still struggle to capture some effects such as those arising from quantum mechanics and solvent interactions. Even relatively small systems, like RNA oligomers, can be inadequately described by standard force fields, resulting in structural ensembles that deviate from experimental observations.

In recent years, new machine learning paradigms for molecular modeling have been introduced. Neural-network potentials (NNPs) have emerged as data-driven alternatives to traditional force fields, capable of capturing complex interactions with quantum-level accuracy when trained on high-quality reference data. Meanwhile, deep-learning methods like AlphaFold have demonstrated unprecedented success in predicting native protein structures directly from amino acid sequences. While these innovations mark significant progress, challenges remain in applying them to flexible and complex biological molecules such as RNA.

To improve agreement between simulations and experiments, integrative techniques have been developed. One class of methods, known as ensemble refinement, leverages the maximum entropy principle to minimally reweight simulation frames to best match experimental averages. Beyond system-specific refinement, force-field fitting approaches seek to directly refine potential energy functions to improve agreement with experimental data. Though less flexible than ensemble refinement, force-field adjustments offer the advantage of producing transferable improvements applicable to diverse molecular systems. Moreover, they allow the researchers to encode an additional information, namely the fact that some of the force-field parameters might be less reliable than others.

Another critical aspect involves forward models, which link simulated structures to experimentally measured observables – for example, the Karplus equations relating dihedral angles to the ^3J scalar couplings measured in NMR spectroscopy. These forward models introduce additional sources of uncertainty, as their parameters are often determined empirically and come with inherent errors. Refinement techniques can be extended to simultaneously optimize both structural ensembles and forward models, offering a more comprehensive approach.

Collectively, these integrative methods (encompassing ensemble refinement, force-field correction and forward-model adjustment) enable a synergistic combination of simulations and experiments. The ultimate goal is not only to faithfully reproduce existing data but also to build more accurate and predictive models of molecular behavior that transfer effectively to new molecular systems.

1.1 A pill of structural biology

Biological molecules are the building blocks of life and play a crucial role in the structure, function, and regulation of any living system. These organic molecules include proteins, nucleic acids (DNA and RNA), carbohydrates and lipids. In particular, DNA, RNA and proteins are responsible for the storage, transmission and execution of genetic information. This mechanism is described by the *central "dogma" of molecular biology*: the genetic information of any biological system flows from DNA to proteins through RNA. More specifically, DNA is the long-term repository of genetic information, which is transcribed into RNA, that in turn directs the synthesis of proteins. The proteins then carry out the essential functions required for the cell to survive, grow and reproduce. DNA, RNA and proteins are also involved in various cellular processes such as regulation, signaling and catalysis. In this Thesis we mainly focus on RNA molecules.

1.1.1 RNA

Ribonucleic acid (RNA) is one of the most important and versatile chemical species in molecular biology, present in all living cells. RNA is a single-stranded polymer composed of nucleotides, structurally related to DNA, and plays a central role in the translation of genetic information from DNA into proteins. However, RNA is now recognized for performing a wide range of essential cellular functions. Several types of RNA – such as messenger RNA (mRNA), transfer RNA (tRNA), ribosomal RNA (rRNA), and lastly regulatory RNA – contribute to diverse biological processes besides information storage and transmission, including catalysis and regulatory control. In particular, RNA can act both as a carrier of genetic information and as a catalyst in chemical reactions, the latter carried out by RNA enzymes, or ribozymes. For this twofold ability, it has been hypothesized that, early in the history of life, prior to the evolution of DNA and protein-based enzymes, an RNA-world existed, in which RNA both served as storage method for genetic information (a role fulfilled today by DNA, except in the case of RNA viruses) and potentially performed catalytic functions in cells (a function performed today by protein enzymes, with the notable exception of ribozymes).

From a chemical perspective, RNA is a linear polymer composed of nucleotides (also denoted as residues), each consisting of a nitrogeneous base (nucleobase) covalently bonded to a ribose sugar and a phosphate group. Each ribose can adopt two different conformations (sugar puckering): if the C2′ atom faces towards the nucleobase, then the ribose is in the C2′-endo conformation, whereas it is in the C3′-endo conformation if the C3′ atom

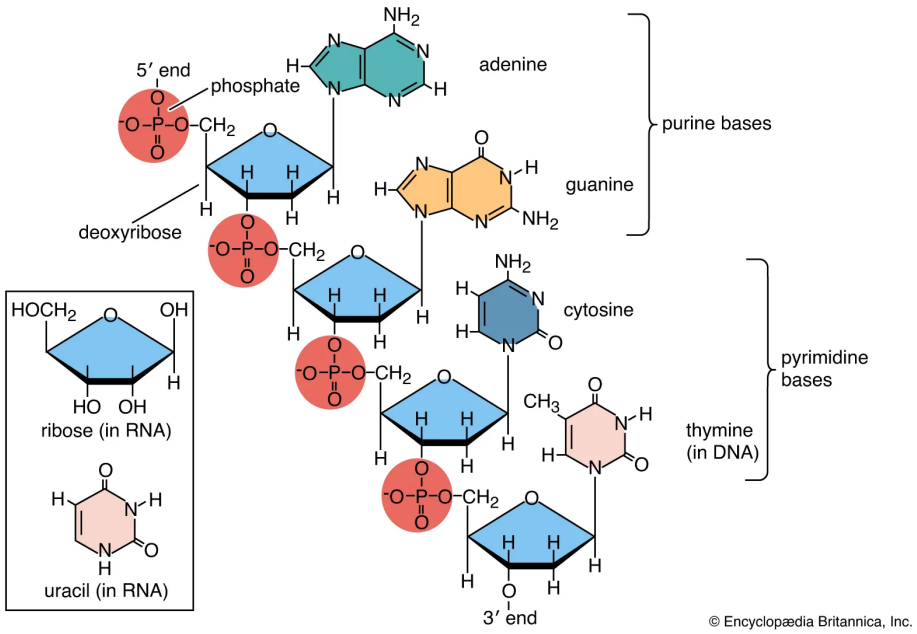


Figure 1.1: RNA is a nucleic acid that typically exists as a single strand, consisting of the nucleotides cytosine, guanine, adenine, and uracil (thymine in DNA).

faces towards the nucleobase. As in DNA, the nucleotides are connected through phosphodiester bonds between the sugar of one nucleotide and the phosphate group of the adjacent nucleotide. The most common nucleobases of RNA are: adenine (A), cytosine (C), guanine (G) and uracil (U), with the latter substituting thymine of DNA, where thymine lacks a methyl group on the 5-carbon atom. Cytosine and uracil are derivatives of the pyrimidine ring, while adenine and guanine are purines, featuring a double-ring which comprises a pyrimidine and an imidazole ring. In physiological solution, each phosphate group carries a net charge of $1e^-$, making RNA a negatively charged polyanion. A crucial structural difference from DNA is the presence of a hydroxyl group OH at the 2′-position of the ribose ring, in contrast to the hydrogen atom in deoxyribose of DNA. This 2′-OH group enhances RNA's chemical reactivity and structural versatility by serving as both a hydrogen-bond donor and acceptor, contributing to RNA's dynamic folding and functional complexity.

To fulfill its diverse biological functions, RNA folds into specific and often complex three-dimensional structures. These conformations are stabilized by two types of non-covalent interactions between nucleobases [2, 3]: stacking and hydrogen bonds. Whereas the contribution of base-stacking interactions

to the structural stability of RNA is the largest, edge-to-edge hydrogen-bond interactions provide directionality and specificity. Like DNA, RNA bases can pair through Watson-Crick hydrogen bonding. Canonical Watson-Crick base pairs include guanine-cytosine (G-C), which forms three hydrogen bonds, and adenine-uracil (A-U), which forms two. These pairs stabilize double-helical regions of RNA, similar to DNA.

However, RNA is able to form a wider variety of interactions, resulting in increased flexibility. Beyond canonical Watson-Crick pairing, more complex interactions are observed, such as the stacking and hydrogen bonding of adenine bases A-A within a bulge, or the guanine-adenine G-A pairing found in the GNRA tetraloop (where N stands for a generic canonical RNA nucleotide and R for either guanine or adenine). These are non-canonical base pairings, namely, hydrogen bonds between nucleobases which differ from the patterns observed in Watson-Crick base pairs, as in classical double-helical DNA. While non-canonical base pairs are relatively rare in biological, double-stranded DNA, in RNA molecules their occurrence is much more frequent. The variety of RNA base pairing is described using the Leontis-Westhof classification system, which defines three distinct interaction edges for each nucleobase: Watson-Crick, Hoogsteen and sugar edges [2]. Unlike double-stranded DNA, the high flexibility of the RNA backbone allows single-stranded RNA to fold into a wide variety of compact and stable conformations, enabling it to perform a broad range of cellular processes.

The set of base pairs that define a particular conformation of an RNA molecule is referred to as secondary structure. Typically, an RNA molecule can adopt multiple conformations, each characterized by a distinct secondary structure. In many cases, RNA exists in a dynamic equilibrium of several stable secondary structures. However, some RNAs adopt a single structure, thermodynamically favored, which is known as the native structure. The alternation of base-paired and unpaired nucleotides in RNA gives rise to recurring structural elements. The basic one is the helix, formed by at least two stacked base pairs. Helices are often interrupted by unpaired nucleotides on one or both strands, leading to the formation of bulges and internal loops, structural elements with diverse functional roles. Loops facilitate folding, stabilize local structures and provide binding sites for proteins and amino acids. Due to the rotational flexibility of its sugar-phosphate backbone, RNA can fold beyond these local secondary motifs into more complex tertiary structures, involving interactions between helices and loops. For example, adjacent helices can stack coaxially, or distant helices may dock into one another's grooves. Loops can also form long-range interactions by base pairing with complementary regions giving rise to pseudoknots.

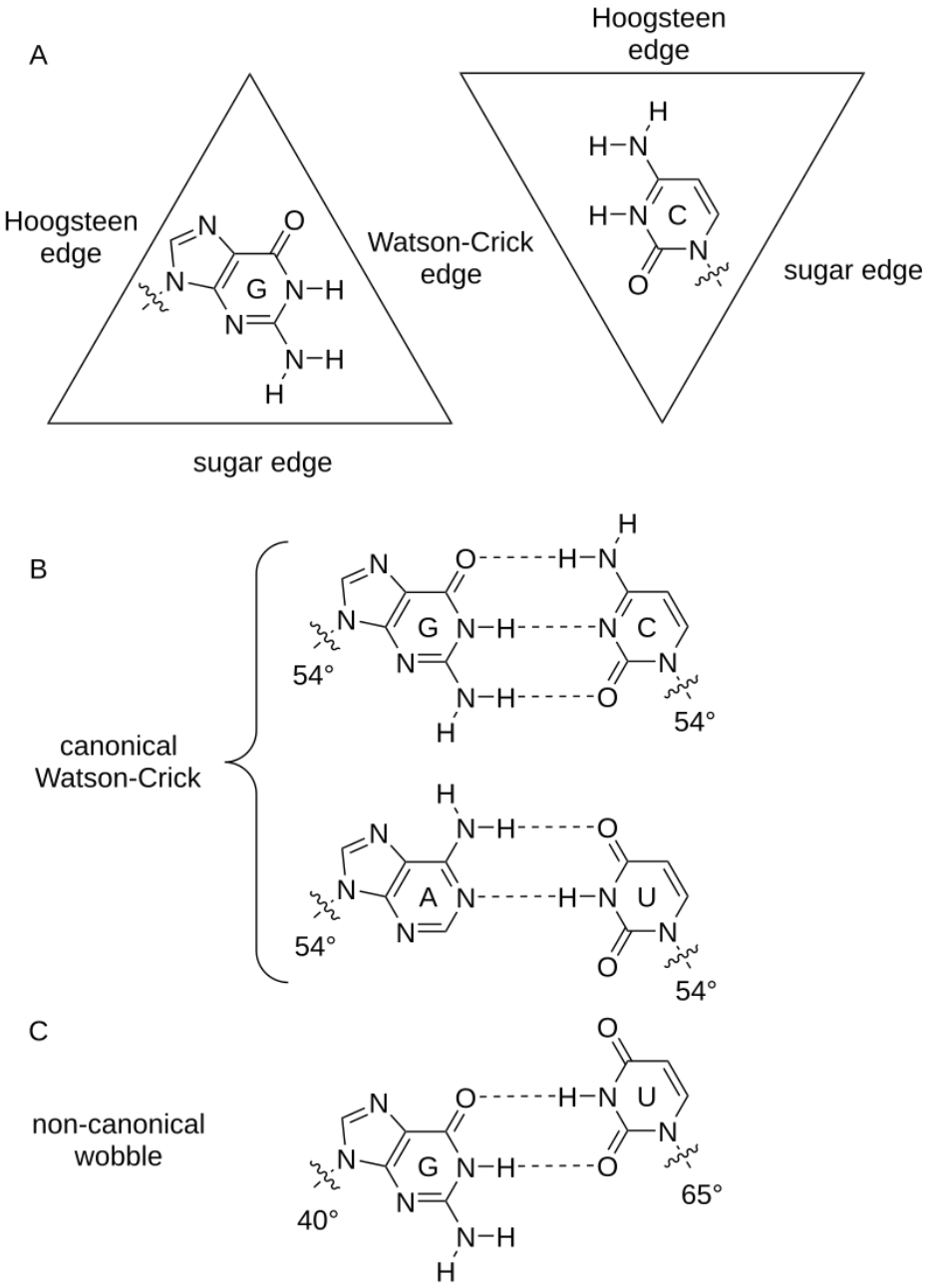


Figure 1.2: (A) The three interacting nucleobase edges described in the Leontis-Westhof classification, (B) canonical base-pair interactions, and (C) the G–U wobble type non-canonical interaction.

Beyond standard RNA nucleobases, a large number of chemically modified nucleotides is also observed. These modifications most commonly occur on the nucleobase moiety and are introduced by the effect of enzymes after the transcription process has been completed (post-transcriptional modifications). For instance, N^6-methyladenosine m^6A is a nucleotide naturally modified through methylation of adenine, namely, the substitution of an hydrogen atom $-H$ on N^6 with a methyl group $-CH_3$ (methylation).

The intracellular ionic environment itself has a direct impact on RNA structure, function and folding. The negative charges on the phosphate groups of RNA backbone create electrostatic repulsion between different parts of the RNA chain, especially when the molecule folds to bring distant regions into close proximity. Consequently, ions play a crucial role in neutralizing this charge and stabilizing both secondary and tertiary structures. They include both monovalent cations (such as Na^+ and K^+) and divalent cations, especially Mg^{2+}. Even the subcellular environment plays a significant role in shaping RNA structures and folding dynamics, depending on whether RNA is in the nucleus or the cytoplasm, due to changes in crowding and ion balance.

1.1.2 Structure-function relationship

The structure of a biomolecule – whether RNA or a protein – is deeply intertwined with its function. This is a core principle of molecular biology, often summarized as "structure determines function". The specific three-dimensional shapes of proteins and nucleic acids enable these molecules to perform their biological roles with high specificity and efficiency. Their function is inherently tied to their structure, interactions and dynamics in a physiological environment.

As described in the previous Section, RNA molecules are highly flexible due to their ability to fold into a variety of secondary and tertiary structures. These structures enable RNA to perform its functions in information transfer, catalysis, regulation, and recognition. Among secondary-structure motifs, hairpins and stems provide structural scaffolding and recognition motifs (as in tRNA or regulatory RNA), internal loops and bulges are often sites of protein interaction or conformational switching, pseudoknots are important for catalytic activities in ribozymes. In terms of tertiary structure, ribozymes fold into specific three-dimensional shapes to position functional groups for catalysis. Coaxial stacking and long-range interactions stabilize the architecture required for molecular recognition, as in the case of tRNA, which must fold into an L-shaped tertiary structure to align amino acids correctly during translation.

Similar to RNA molecules, intrinsically-disordered proteins (IDPs) are highly flexible and lack a stable three-dimensional structure under physiological conditions – yet they are highly functional. Rather than folding into a stable well-defined tertiary structure, they exist as dynamic, structurally heterogeneous ensembles. Despite their disorder, they perform critical biological roles, especially in regulation, signaling and control. While traditional structural biology has linked fixed three-dimensional structures to specific functions, IDPs show that structural flexibility itself can be functional. Their intrinsic disorder enables regulation, adaptability and multi-functionality, thereby challenging the traditional "structure equals function" paradigm.

Investigating how the three-dimensional structures of biological molecules relate to their function is one of the central tasks of structural biology. By resolving structures, researches aim to predict molecular functions, design new drugs, engineer proteins or RNAs for synthetic biology and understand how disease-causing mutations alter normal function, as in the case of neurodegenerative diseases. To achieve these challenges, both experimental and computational methods are jointly employed.

1.2 Experimental techniques

A variety of experimental techniques have been developed to investigate the structural and dynamic properties of biological molecules, each offering its own strengths and limitations. Key methods include X-ray crystallography, nuclear magnetic resonance (NMR) spectroscopy, cryo-electron microscopy (cryo-EM), and small-angle X-ray (or neutron) scattering (SAXS/SANS), all of which provide valuable insights into the atomic and molecular arrangements of biomolecules. Moreover, thermal-denaturation experiments allow for measuring free-energy differences between the folded and unfolded states. By integrating data from multiple techniques, researchers can achieve a more comprehensive picture of biomolecular structures – an essential step in characterizing their functions and interactions. In this Thesis we use data from NMR spectroscopy and thermal-denaturation experiments. In the following, we briefly review all the mentioned techniques highlighting their strengths and weaknesses.

X-ray crystallography [4] leverages on the crystallization of the biomolecule and subsequent irradiation with X-ray beams. The X-rays are diffracted by the electrons in the crystal, producing a diffraction pattern that is analyzed to reconstruct the three-dimensional electron density map, which is then used to infer the positions of the individual atoms. This technique yields high-resolution structural data at atomic resolution. However, it requires

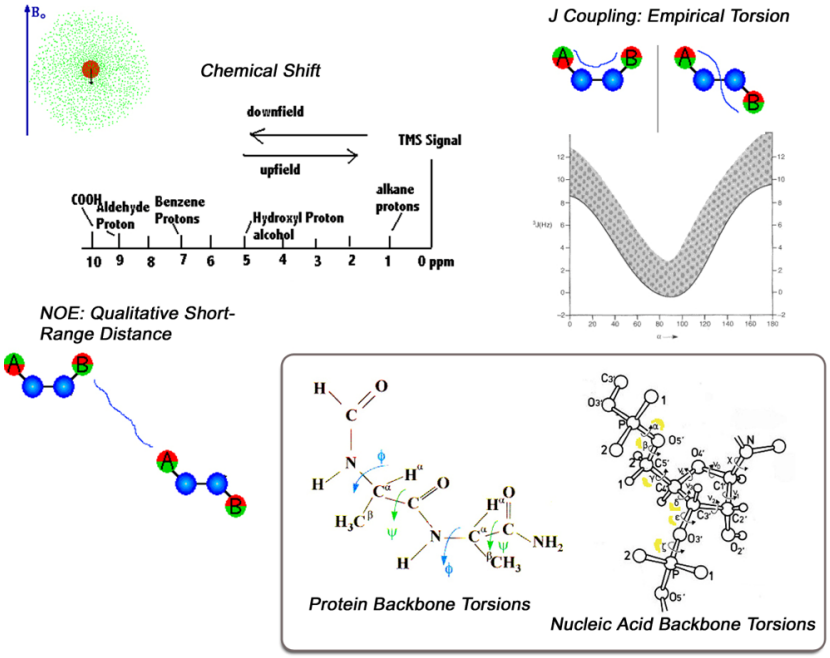


Figure 1.3: Main quantities measured through NMR: chemical shifts, J couplings, and NOE signals.

crystallization, which can alter the molecule's native conformation and may not fully capture its physiological state. Further, X-ray crystallography provides only a static picture of the molecule, offering limited insight into its conformational flexibility and transient structural states.

Nuclear magnetic resonance (NMR) spectroscopy[1] is another experimental technique devoted to structural determination and offers several advantages over X-ray crystallography. In particular, it allows to probe biological molecules in solution, which more closely resembles the physiological conditions than the crystalline state used in X-ray crystallography. This makes NMR particularly valuable for the study of proteins and nucleic acids, including those that are difficult to crystallize. NMR exploits the magnetic properties of certain atomic nuclei – including hydrogen, carbon, nitrogen and phosphorous. Applying a strong magnetic field with radio-frequency pulses, these nuclei resonate at characteristic frequencies that depend on their local chemical environment. Key quantities measured in NMR spectroscopy include chemical shifts, scalar J couplings and nuclear Overhauser effect (NOE).

The **chemical shifts** are the resonant frequencies of specific atomic nucleus

[1] A comprehensive guide on this topic is available at https://organicchemistrydata.org/hansreich/resources/nmr/.

in a magnetic field, indicative of the surrounding electronic environment. Circulation of electrons around the nucleus create local magnetic fields that shield the nucleus from the external field, revealing information about the local chemical environment.

Scalar couplings and NOE signals both arise from the interactions between nuclei with non-zero spin and fall into two distinct categories: direct and indirect interactions. Direct interactions, due to dipole-dipole coupling, are very strong but not observed in mobile solutions because they are averaged to zero by the rapid tumbling of the molecules. However, this tumbling causes the dipole-dipole coupling to vary in time, creating a fluctuating magnetic field at each nucleus. These fluctuations lead to dipole-dipole relaxation, which underlies the **nuclear Overhauser effect (NOE)**. The NOE manifests as a change in population of one nucleus when a nearby magnetic nucleus is saturated. This interaction is distance-dependent, with the NOE signal intensity proportional to $1/r^6$, allowing for the determination of intra-molecular distances, even between atoms that are not directly bonded.

Conversely, indirect interactions (also known as scalar couplings) arise from the polarization of electrons by nuclear spins. In **scalar couplings**, the spin of one nucleus perturbs the spins of the surrounding electrons, which in turn influences the energy levels of neighboring magnetic nuclei. These couplings are transmitted through atomic covalent bonds and their magnitude falls off rapidly with the number of intervening bonds. Couplings over one (^1J), two (^2J) and three (^3J) bonds usually dominates the fine structure of NMR spectra. The values of scalar coupling constants, measured as frequency differences, provide information about the bonding between nuclei and the relative geometry, gaining insights into the three-dimensional molecular structure. In particular, ^3J scalar couplings are related to the dihedral torsion angles ϕ through the **Karplus equations**

$$^3J(\phi) = A\cos^2\phi + B\cos\phi + C \tag{1.1}$$

where A, B, C are empirically-derived parameters that depend on the atoms involved in the specific scalar coupling. The estimation of these parameters is not an easy task and several groups [1, 5–7] have developed different sets of parameters by fitting the Karplus curves with the ones obtained by experiments or quantum-chemistry calculations.

Beyond structural information, NMR also offers **dynamic insights** through relaxation measurements. By analyzing relaxation rates of nuclear spin populations after radio-frequency perturbation, researches can extract information on timescales and amplitudes of atomic motions within the biomolecules, such as side-chain rotations, loop flexibility and folding/unfolding events.

Thermal-denaturation experiments [8] are used to estimate free-energy differences between two different conformations, such as folded/unfolded or single-strand/duplex, under the assumption of a two-state transition. In this technique, a biological molecule is slowly heated in a controlled environment. A spectroscopic tool tracks a property that changes between the folded and unfolded states, like the UV absorbance at a specific wavelength. The measured absorbance is the weighted average of the two values for the baseline absorbance of folded and unfolded states. In this way, for each temperature the equilibrium constant K (namely, the ratio between folded and unfolded concentrations) can be computed. Then, free-energy differences are derived by fitting data from varied concentrations to the van't Hoff equation, relating the equilibrium constant K to the temperature T.

Other experimental techniques include cryo-EM and SAXS/SANS. **Cryo-electron microscopy (cryo-EM)** [9] is particularly useful for large macromolecules. It involves rapid freezing of samples to cryogenic temperatures, avoiding the artifacts often introduced by crystallization. The sample is then analyzed through an electron microscope, which collects scattered electrons to produce thousands of high-resolution two-dimensional images, representing different orientations of the molecule. **Small-angle X-ray (or neutron) scattering (SAXS/SANS)** [10] probe the overall shape, size and conformational flexibility of biomolecules in solution. A beam of monochromatic X-rays (or neutrons) is directed at a sample and the scattered radiation is measured at small angles relative to the incident beam, reconstructing its three-dimensional shape.

1.3 Computational techniques

Computational methods are essential tools for investigating biomolecular structure and dynamics. These techniques complement experimental approaches, providing mechanistic insights and predictive power. The spectrum of computational methods spans from quantum-mechanical models to data-driven approaches based on deep learning, with a pivotal role played by molecular dynamics (MD) simulations, based on classical force fields. Each approach occupies a niche in terms of resolution, system size and computational cost. Increasingly, hybrid methods – combining, for instance, quantum mechanics with molecular mechanics (QM/MM) or deep learning with physics-based simulations – are being developed to exploit the strengths of different techniques.

At the most fundamental level, **quantum-mechanical (ab initio) calculations** model the electronic structure of molecules by solving the Schrödinger

equation for a system of electrons and nuclei. These methods, including Density Functional Theory (DFT), are highly accurate and capable of describing bond formation and breaking, proton transfer and charge distributions (namely, polarization effects). However, they are computationally extremely expensive – so much that their practical use is restricted to very small systems and short timescales, preventing their application to large molecules, like those of biological interest. This limits quantum-mechanical calculations to small fragments of larger biomolecules, like enzymatic active sites, in combination with atomistic MD simulations (QM/MM approaches). Although in principle highly accurate, ab-initio computations are so slow that it is difficult to effectively verify their correctness on molecules with biological interest and on the relevant timescales.

Conversely, **MD simulations** based on classical atomistic force fields (such as AMBER or CHARMM) enable to investigate large biomolecules over biologically relevant timescales (at least in principle), reproducing their conformational flexibility and folding pathways. However, their efficiency comes at the cost of reduced accuracy, as these models use empirical potentials and simplified representations of interactions, mostly derived by fitting with quantum-mechanical computations on small systems (accuracy-versus-efficiency tradeoff).

To study large-scale conformational changes on longer timescales, **coarse-grained models** have been developed to simplify biomolecular systems by grouping atoms into pseudo-particles or "beads". These models aim to preserve essential physics while significantly reducing computational cost. Whereas scalable to large systems (such as ribosomes and viral capsids), coarse-grained models are less detailed than atomistic ones and their accuracy depends on the quality of the coarse-graining strategy and parametrization.

In parallel, recent advances in **machine-learned interatomic potentials**, which leverages neural-network architectures, are helping to bridge quantum accuracy with molecular simulation efficiency. These models are trained on ab-initio calculations to learn the potential-energy or free-energy surfaces. They are orders of magnitude faster than quantum-mechanical methods, while retaining much of their accuracy for small molecules. However, they are significantly slower than classical force fields and currently limited to relatively small systems. Their application to complex biological systems such as full proteins or RNA molecules remains an active area of research. Importantly, besides their bottom-up training to map quantum-mechanical computations into atomistic potentials, these potentials can also be trained using atomistic simulations to generate coarse-grained models, either in a bottom-up or top-down strategy [11–15]. The main limitation lies in the diversity of the training set, which must be sufficiently representative to

ensure good generalization to different molecular systems.

Lastly, **structure-prediction methods based on deep learning** like AlphaFold or RosettaFold represent a largely data-driven, top-down approach. These models infer structure from sequence by training on big datasets of known protein structures. They achieve remarkable accuracy for structured proteins, especially in their native ground states, but generally lack physical interpretability and do not model dynamics or folding pathways. Furthermore, they perform poorly on intrinsically disordered proteins (IDPs) and RNA molecules, especially disordered RNAs, such as the small RNA oligomers investigated in this Thesis. These systems lie outside the domain where AlphaFold-like models were trained and optimized.

When aiming to understand the performance and limitations of computational methods, classical MD simulations remain the gold standard. They offer the unique advantage of being fast enough to achieve converged ensemble sampling in systems like small RNA oligomers – molecules that are small and tractable, yet still complex enough to reveal discrepancies with experimental data [16–23]. In these regimes, classical force fields – less accurate for RNA than for proteins – allow systematic testing and refinement, making them an ideal test-bed for integrative modeling approaches, aimed to combine experimental measurements with computational models. When simulations are fast enough to reach convergence, the resulting ensembles reflect the underlying force field rather than the starting structure. This allows rigorous comparison to experimental data and can help identify specific weaknesses in the force field, enabling targeted improvements. The simulation trajectories and structural ensembles generated by these methods – whether based on ab-initio calculations, classical MD, or neural-network potentials – can serve as valuable input for integrative modeling such as ensemble refinement or transferable force-field fitting.

1.3.1 Molecular dynamics (MD) simulations

Molecular dynamics (MD) simulations are a powerful computational tool used to explore and predict the structural, dynamical and thermodynamic properties of molecular systems, complementing experimental measurements. In MD, molecules are modeled as collection of atoms, approximated as points connected by springs that mimic covalent bonds. Interactions between atoms are described by an empirical classical potential, known as force field, which regulates the time evolution of the system (its dynamic trajectory) through numerical integration of Newton's equations of motion. To simulate conditions of constant temperature (as in the canonical ensemble), a thermostat is applied to couple the system to a thermal bath, ensuring proper temperature

regulation. While force fields are computationally efficient and reasonably accurate for describing biomolecules, they are inherently limited because they do not account for quantum effects such as bond breaking or formation, making them unsuitable for simulating chemical reactions. Furthermore, most classical force fields do not model electronic polarizability explicitly, and the motion of electrons is neglected. Initial atomic coordinates for MD simulations are often taken from experimental structures, such as those derived from X-ray crystallography or NMR, or built through specific tools such as `x3DNA` [24, 25].

To overcome the timescale limitations of plain MD, which may fail to capture rare but biologically relevant events, several enhanced sampling techniques have been developed and applied [26, 27]. These methods aim to accelerate the exploration of conformational space and improve sampling convergence, particularly in systems with complex or rugged free-energy surfaces. Enhanced sampling techniques generally fall into two categories:

1. temperature- or Hamiltonian-based methods, such as replica exchange, which increase sampling efficiency by simulating multiple replicas at different temperatures (or Hamiltonians), allowing the system to more readability cross energetic barriers;

2. biasing techniques, which apply external forces to selected collective variables, namely degrees of freedom known or suspected to determine slow dynamics; examples include umbrella sampling and metadynamics, which guide the system to overcome free-energy barriers accessing otherwise infrequent transitions.

Force fields

A force field refers to the function (both the functional form and its set of parameters) used to model the potential energy of a system of atoms. Typically, a specific set of parameters is derived in order to reproduce quantum-chemistry calculations or experimental data on small systems, in either a bottom-up or top-down strategy. Compared with other computational techniques, they represent a good balance between computational efficiency and accuracy.

The functional form includes a bonded part, characterizing the interactions of atoms linked by covalent bonds, and a non-bonded part that describe the long-range electrostatics and van der Waals forces. The bonded part includes bond and angle terms (modeled as harmonic potentials – bond breaking are neglected) and terms acting on the dihedral angles (modeled as sinusoidal functions). Electrostatics is captured by Coulomb potentials while van der Waals forces are described by Lennard-Jones interactions.

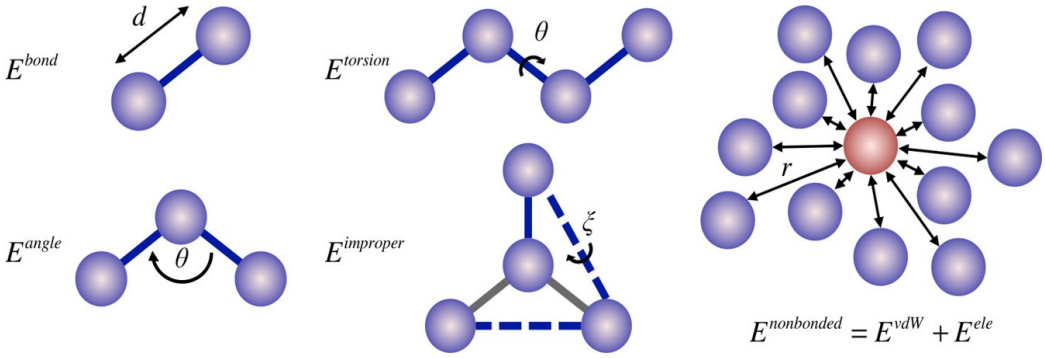


Figure 1.4: The force field comprises multiple contributions: bond stretching, angle bending, torsional rotations, improper torsions, and non-bonded interactions. Image from [28].

At present, the most recommended force field for RNA MD simulations is *AMBER ff99 bsc0* $\chi OL3$ [17, 18, 29–31]. This force field builds upon the original AMBER ff99 force field [29] with two modifications: the bsc0 correction [17], which improves the description of the backbone α, γ torsional angles, and the $\chi OL3$ correction [18], which refines the glycosidic torsion angle χ between ribose and nucleobase. These improvements specifically target common inaccuracies in RNA simulations, such as overstabilized backbone conformations and inaccurate base stacking, and have been well validated against NMR and crystal structures. A further refinement has been proposed based on the modification of phosphate oxygen radii [32].

The parameters of AMBER ff99 (including elastic strengths, equilibrium lengths, torsional potentials and atomic charges) were derived from quantum-mechanical calculations on small molecular fragments, typically up to a few dozen atoms. While this approach provides chemically accurate parameters in localized contexts, it has some limitations when applied to large biomolecules. For example, the original AMBER ff99 was found to overstabilize certain backbone conformations mismodeling the dynamics of RNA or DNA.

These inconsistencies with empirical observations led to the development of *parmbsc0* ("Barcelona correction 0") reparameterization of DNA backbone [17], where the parameters corresponding to the α (O3′ − P − O5′ − C5′) and γ (C5′ − C4′ − C3′ − O3′) torsional angles were fitted against quantum-chemistry calculations. This correction significantly improved the backbone geometries of RNA by reducing the sampling of conformers with non-canonical torsion angles.

Later, similar issues were identified in RNA simulations, such as the formation of ladder-like structures due to poor modeling of base-sugar interactions.

To address this, the $\chi OL3$ correction was introduced, reparametrizing the χ torsional angle ($O4' - C1' - N1/9 - C2/4$). This allowed to accurately reproduce the *syn/anti* conformational preferences of nucleotides (related to the rotation between sugar and nucleobase, crucial for RNA folding and stacking), based on quantum-mechanical calculations and NMR data. This refinement prevented problems like spurious base flipping and incorrect base stacking that previous force fields suffered from.

A general trend in force-field development is that parameters are typically derived from small molecular systems, whereas validation is performed on larger macromolecules. However, discrepancies observed in simulations of complex macromolecular systems play a crucial role in identifying and correcting limitations in the underlying force-field parameters. In recent years, some efforts have moved towards fitting parameters directly from simulations of short oligonucleotides. A remarkable example is the *gHBfix* refinement [33], which targets specific intramolecular hydrogen bonds to improve the accuracy of RNA simulations.

To achieve realistic RNA dynamics, it is important to pair the RNA force field with accurate models for solvent and ions. A suitable model for water is OPC (Optimal Point Charge) [34], a four-site representation that enhances the three-site architecture by introducing a dummy atom with negative partial charge. This atom is positioned near the oxygen along the bisector of the H-O-H angle, improving the model's ability to reproduce key thermodynamic and structural properties of water.

Interactions between ions and either nucleic acids or water molecules are described through Lennard-Jones (LJ) potentials, which model the van der Waals (non-bonded) interactions between atoms. Each atom type is assigned two LJ parameters: the distance σ_i related to the effective atomic radius and the depth ϵ_i of the potential well. Commonly used values for the ion parameters are those derived by Joung & Cheatham [35] for monovalent ions Na+, K+, Cl- and those by Li & Merz [36] for Mg2+ and other metal ions. When two different atom types interact, their cross-interaction parameters are typically computed using the Lorentz-Berthelot mixing rules as:

$$\sigma_{ij} = \frac{\sigma_i + \sigma_j}{2}, \quad \epsilon_{ij} = \sqrt{\epsilon_i \epsilon_j}. \tag{1.2}$$

Force fields for chemically-modified nucleotides

Over the last decades, the parametrization of RNA force fields was mostly focused only on the four standard nucleotides: adenine A, uracil U, cytosine C, and guanine G. To simulate RNA molecules containing modified nucleotides, it is necessary to derive specific force-field parameters for each modification. Parameter sets for approximately 100 naturally-occurring modified nucleotides have been derived both within the AMBER and CHARMM frameworks. A major contribution came from *modrna08* force field by Aduri et al. [37], a full parametrization for 107 naturally occurring modifications. These parameters were derived following the standard AMBER protocol, fitting torsional angles and atomic charges to reproduce quantum-mechanical calculations. Validation was performed through MD simulations of a tRNA molecule with several modifications covered by the force field. Since its release, *modrna08* has been widely adopted in RNA simulations performed with the AMBER suite.

However, later studies revealed some limitations. In particular, *modrna08* failed to accurately capture the conformational preferences of certain modified uridines, as demonstrated by discrepancies with NMR experimental data [38]. This highlighted the need for residue-specific refinement, producing a revised set of tosional parameters named $\chi IDRP$ [39]. Another widely studied modification is N6-methyladenosine (m^6A). Hurst et al. [40] evaluated the destabilizing effect of the methyl group on RNA duplexes and shown that their force-field parameters correctly reproduced data from thermal denaturation experiments. Still, the original *modrna08* force field was shown to fall short in reproducing experimental denaturation profiles across a broader range of RNA sequences. In particular, it misrepresented both the extent of duplex destabilization and the relative stability of the two possible conformations of the methyl group. A more recent reparameterization of both the partial charges close to the methyl group and the η_6 dihedral angle [41] has brought the MD simulations into quantitative agreement with experimental measurements (both conformational behaviour of the methyl group and thermodynamic stability of RNA duplexes).

1.3.2 Neural-network potentials

In recent years, machine-learned interatomic potentials have emerged as a powerful class of methods that aim to fill the gap between quantum-mechanical accuracy and the computational efficiency required for large-scale MD simulations. A major milestone in this field was the introduction of high-dimensional neural-network potentials (NNPs) by J. Behler and M. Parrinello in 2007 [42]. Their approach decomposes the total energy of

a system into atomic contributions, each depending on the local chemical environment. These atomic environments are described through atom-centered symmetry functions used as descriptors fulfilling the requirements of rotational, translational and permutation invariance. In this way, neural networks efficiently learn the potential-energy surface through direct fit with quantum-mechanical reference data. This framework captures many-body interactions and is particularly effective for systems where local interactions dominate. Further extensions to include long-range interactions, such as electrostatics and charge transfer, have been developed. For a comprehensive account of NNPs and recent developments, we refer the reader to recent reviews by J. Behler and coworkers [43–45]. Here, we provide a brief overview of their capabilities and current applications.

An alternative family of machine-learned potentials is represented by the Gaussian Approximation Potential (GAP), introduced in 2010 by G. Csányi and collaborators [46]. GAP employs Gaussian process regression to fit the potential-energy surface of a system based on quantum-mechanical data, typically obtained through DFT calculations. At the heart of GAP, a descriptor, such as the Smooth Overlap of Atomic Positions (SOAP) [47, 48], encodes the local atomic environments capturing symmetry and chemical identity in a rotational and translationally invariant way. Like NNPs, GAP expresses the total energy of a configuration as a sum of atomic contributions, each predicted through Gaussian process regression over local environments [49]. This framework has been applied extensively in materials science, from bulk solids to amorphous phases, including atomistic simulations of carbon [50, 51] and silicon [52], with several practical applications, for example for carbon-based electrodes [53], and even protein-ligand interactions [54].

A key breakthrough in the development of NNPs was their successful application to the simulation of liquid water, a notoriously challenging system. In pioneering works by C. Dellago and colleagues [55, 56], NNPs trained on ab-initio data were able to efficiently capture the relatively-weak isotropic van der Waals forces that, in addition to the strongly-directional hydrogen bonds, are crucial for reproducing water's anomalous properties, such as its density maximum and negative volume change upon melting.

Since 2017, various neural-network architectures have been developed for potential-energy modeling. Remarkable examples include: ANI [57], tailored for organic molecules; SchNet [58, 59] and DimeNet [60, 61]), based on directional message-passing algorithms [62] that capture chemical and spatial orientational information directly from molecular graphs; equivariant neural networks [63], which enforce the physical symmetries of three-dimensional space improving data efficiency; DeepMD [64] and TorchMD [65, 66], designed for scalable simulations of molecular systems. Equivariant models, in

particular, ensure that predictions transform coherently under rotations and translations, an essential property for physically-meaningful representations of atomic systems.

Beyond all-atom simulations, NNPs have also been extended to coarse-grained (CG) modeling. The main idea is to derive effective CG interactions by training NNPs on atomistic trajectories using the force-matching principle [67], following a bottom-up strategy. For instance, the group led by J. Zavadlav employed DimeNet++ to develop CG potentials for alanine dipeptide and water, using a loss function based on relative entropy. This approach demonstrated improved accuracy over traditional force-matching techniques [15]. A complementary top-down strategy was used in [14], where DimeNet++ was trained to augment existing reference potentials by fitting to experimental data. This method was applied to an atomistic model of diamond (starting from the Stillinger-Weber potential) and a coarse-grained description of liquid water (starting from Lennard-Jones interactions).

Coarse-graining based on the force-matching principle has also been extensively employed by C. Clementi and collaborators. Her group, in collaboration with researchers such as F. Noé and G. De Fabritiis, has developed CG models based on neural networks (particularly CGNets) that preserve key features of protein folding while enabling access to millisecond-scale dynamics [11–13, 68, 69]. These models are particularly effective for investigating biomolecular folding and conformational transitions at long timescales, as exemplified by studies on chignolin, a small synthetic polypeptide. On-going efforts to develop machine-learned force fields for more complex biological systems, especially proteins, are being actively pursued by the group of G. De Fabritiis [70].

Despite these advancements, NNPs still face significant challenges. Key limitations include: system size and complexity (biomolecules are large and flexible, requiring careful design of descriptors and training data to capture relevant conformational spaces); data scarcity (high-quality quantum-chemical reference datasets for large biomolecules are expensive to generate, limiting training data availability); transferability (achieving generalizable NNPs that work across diverse biological systems remains an open challenge).

1.3.3 Predicting structures with deep learning

Deep learning has strikingly advanced protein structure prediction, with landmark contributions from AlphaFold by DeepMind [71–73] and RoseTTAFold by the Baker lab [74–77]. These tools use attention-based neural networks [78] and large databases of protein sequence/structure couplings to predict three-dimensional folding structures directly from amino acid sequences, often with

near-experimental accuracy. AlphaFold2, in particular, set a new benchmark during the CASP14 [71, 79] competition[2] (2020), demonstrating accuracy comparable to experimental methods for many structured proteins.

AlphaFold2, ad its successor AlphaFold3, have proven remarkably effective at predicting high-resolution structures of globular proteins, including single chains and some complexes, especially near native conformations (namely, global minima of the molecular energy function). Their power lies in pattern recognition across evolutionary, spatial and physico-chemical constraints, enabled by training deep-learning models on huge databases of known structures, with multiple sequence alignments and structural templates. These models have accelerated structure prediction for large protein complexes and are increasingly being adapted to protein-RNA complexes, with widespread applications in drug discovery and molecular design.

Despite their impact, current models exhibit important limitations. Most notably, they perform poorly on intrinsic disordered proteins (IDPs) and disordered regions in structured proteins [80]. Because they predict a single static structure, they cannot capture the dynamic ensembles peculiar of IDPs. These shortcomings are often flagged by low confidence levels for the predicted structures. Moreover, these deep-learning models fail to represent metastable states, transient intermediates, context-depending folding, or binding-induced conformational changes, all of which are essential for understanding flexible proteins and phase separation.

Recent efforts attempt to overcome these issues by combining AlphaFold with MD simulations to explore conformational landscapes, or by integrating ensemble-based approaches such as Bayesian inference with NMR data [81, 82]. However, AlphaFold alone remains insufficient for capturing the complex, rugged free-energy landscapes of highly flexible biomolecules with multiple metastable states.

While deep learning has transformed protein structure prediction, no comparable model yet exists for RNA [83–85]. RNA presents unique challenges: greater structural flexibility, fewer high-resolution structures for training, and the existence of multiple near-degenerate functional conformations rather than a single native folding state. These factors make RNA structures

[2]CASP (Critical Assessment of Structure Prediction) is a biennal, community-driven experiment initiated in 1994 to evaluate models for protein structure prediction, and more recently extended to RNA structure prediction. It provides research groups with a platform for objectively benchmarking and delivers independent assessments of state-of-the-art modeling techniques. Notably, CASP13 (2018) and CASP14 (2020) were won by DeepMind's AlphaFold and AlphaFold2, respectively. AlphaFold was released as an open source in 2021, and during CASP15 (2022), despite DeepMind's absence, nearly all top-ranking teams employed AlphaFold or its variants.

prediction substantially harder and current computational tools do not yet match AlphaFold's level of performance for proteins.

Multiple state-of-the-art methods for RNA prediction exist, each tailored to specific tasks – such as secondary or tertiary structure prediction, or even structure ranking. Secondary-structure predictors, either thermodynamics-based as ViennaRNA [86, 87] or machine-learned models like CONTRAfold [88], perform well for short RNAs but struggle with long-range interactions. Tertiary-structure methods, like Rosetta FARFAR2 [89], use fragment assembly and coarse-grained physics, but require significant experimental input and are often limited to RNAs under 100 nucleotides due to computational cost. More recent machine-learning approaches, such as EteRNAFold [90], aim to improve secondary-structure prediction by leveraging large-scale datasets obtained through chemical-probing experiments as SHAPE[3]. Conversely, other deep-learning models like ARES [93] rank candidate tertiary structures given as input. However, these are still in early stages and have not yet reached AlphaFold-level accuracy, especially for large or complex RNA molecules.

Since RNA molecules are highly flexible and often exist as a set of different conformations rather than a single native structure, achieving AlphaFold-like performance for RNA will likely require ensemble-based models, capable of predicting distributions over multiple conformations rather than single structures. It will also demand a substantial increase in RNA structural data – from cryo-EM or SHAPE experiments, for instance – to expand beyond the current limited datasets (hundreds of RNA structures vs. over 200.000 proteins structures in the PDB). In addition, better treatment of ions and solvent effects, as well as robust modeling of folding intermediates and metastable states, will be essential. This is an active area of research, and future progress will hinge on integrating data-driven approaches with the physics of RNA folding, ideally leading to predictive models that reflect the inherent flexibility and complexity of RNA molecules. For a comprehensive overview on the current state of RNA structure prediction, we refer the reader to consult recent surveys [85, 94].

[3]SHAPE (Selective 2'-Hydroxyl Acylation analyzed by Primer Extension) [91, 92] is a chemical-probing technique used to investigate RNA secondary structure. It quantifies the flexibility and local dynamics of each nucleotide by measuring the accessibility of the 2'-OH group of the ribose sugar to chemical modification. The degree of reactivity correlates with the structural context of each nucleotide: flexible, unpaired nucleotides (often in loops or bulges) are highly reactive, whereas base-paired or structured regions (stems, helices) are less reactive or protected.

1.4 Integrative methods

While computational methods offer powerful tools for investigating biological molecules, both in simulating dynamics and predicting structures, they are inherently limited by approximations, sampling challenges and model-specific assumptions. MD simulations, for instance, depend on empirical force fields that approximate interatomic interactions using classical physics. Despite considerable refinement efforts, these models are still subject to inaccuracies and often fall short in capturing the full complexity of real molecular behaviour. Conversely, experimental techniques provide empirical observations, yet they are limited by resolution constraints, difficulties in sample preparation, and a tendency to produce static snapshots rather than dynamic insights.

This is where integrative modeling comes into play [95]: by combining computational and experimental approaches, it leverages the strengths of each to compensate for their respective weaknesses. Incorporating experimental data into simulations allows computational models to be corrected and refined, improving both the reliability and accuracy of their predictions. At the same time, simulations can guide experimental design, help interpret ambiguous results, and explore conditions that are difficult to access experimentally.

Different experimental methods are also complementary. For example, X-ray crystallography provides high-resolution data but captures only static conformations, whereas NMR spectroscopy offers dynamic information but with lower spatial resolution. Integrative approaches can bridge these gaps by validating and improving computational models with experimental data.

This iterative, synergistic framework is especially valuable for investigating complex, highly flexible systems – such as IDPs and RNA molecules – that are difficult to characterize through computational or experimental methods alone. By integrating data and simulation, we can achieve a more complete and accurate understanding of biomolecular behaviour.

The refinement of empirical force fields or structural ensembles through experimental data follows a *top-down* strategy, in contrast to the *bottom-up* approach, which leverages data from ab-initio or quantum-mechanical simulations. Within the top-down paradigm, an important distinction exists between approaches based on reweighting or restraining [96].

Reweighting methods adjust the overall outcomes of MD simulations to improve agreement with experimental data by integrating both the prior information encoded in the simulation and the experimental observations. A common strategy involves minimizing a loss function which includes a relative-entropy term, balancing fidelity to the original simulation with agreement to the data. Reweighting can be applied either to structural ensembles or to the force-field parameters. It is often performed as a post-processing step after the

MD simulation has been completed. However, if the experimental observables lie far from the original simulation ensemble, a-posteriori reweighting becomes statistically inefficient. This limitation can be quantified using metrics like the Kish effective sample size, discussed in Appendix A. If this value drops below a certain threshold, it is advisable to initiate a new simulation informed by the insights gained so far [14]. Alternatively, on-the-fly corrections can be introduced by dynamically modifying the potential during the simulation run in response to experimental data. This adaptive, data-driven approach provides more efficient sampling while relying less heavily on the original ensembles.

In contrast, restraint-based integration incorporates experimental data directly into the MD simulation by imposing biases on specific degrees of freedom – typically, experimental observables such as interatomic distances or angles. These restraints, implemented as energy penalties, guide the simulation toward conformations consistent with the experimental data. Harmonic restraints penalize deviations quadratically, tightly enforcing agreement, while linear restraints apply softer corrections, allowing greater conformational flexibility. Notably, a formal link connects restraint-based biasing and reweighting: as reported in the following Section, reweighting based on the maximum-entropy principle is equivalent to introducing linear potentials during the simulation, providing a way to satisfy experimental constraints with minimal bias.

Another important distinction concerns which elements of the simulation framework are modified: the structural ensemble, the force field, or the forward model used to calculate observables from structure. We now proceed to examine each of these techniques in greater detail.

1.4.1 Ensemble refinement by maximum entropy

Ensemble refinement [97–104] is a method originally developed to constrain simulations in such a way that they are consistent with experimental observations. While one might argue that refining ensembles to match known data limits their predictive value, it is important to note that validation can still be performed against independent experimental observables. By leveraging the maximum entropy principle, this method constructs conformational ensembles that agree with experimental measurements while remaining as close as possible to the distribution generated by the original force field.

In its original formulation, ensemble refinement applies the maximum-entropy principle by enforcing rigid constraints corresponding to the experimental values. However, because experimental data are inherently noisy and subject to uncertainties, it is often preferable to adopt soft restraints. This

is typically achieved by introducing a cost term – such as a chi-squared χ^2 – into the loss function, which penalizes deviations from the experimental data without enforcing strict equality. Minimizing this loss function yields the ensemble that best reconciles both sources of information.

A key feature of ensemble refinement is its system-specificity: the procedure acts independently on each molecular system, without jointly considering information from multiple systems. Consequently, even under similar experimental constraints, applying ensemble refinement can yield different correction coefficients for different systems, simply because the data are not integrated simultaneously. This does not mean that the derived corrections are inherently non-transferable, but rather that transferability is not built into the framework. For example, consider the case of ${}^3J$ scalar couplings. The functional form of the correction, given by the Karplus equations, closely resembles the standard torsional terms present in force fields. When applying the maximum entropy principle one might obtain different correction coefficients not only across different molecular system, but even among chemically equivalent residues within the same oligomer. However, in many cases, it would be more appropriate to enforce the same correction on dihedrals that are chemically equivalent, promoting consistency and potential transferability [19].

This reflects a fundamental trade-off: ensemble refinement is highly flexible and capable of accurately fitting complex experimental observables, but this flexibility comes at the cost of transferability across systems. In contrast, force-field fitting methods are explicitly designed to integrate experimental data from multiple systems simultaneously, optimizing shared parameters across the dataset. This approach promotes transferability, but comes at the cost of reduced variational flexibility compared to ensemble refinement, which can tailor corrections more finely to system-specific data.

Maximum-entropy principle

The maximum entropy principle was introduced by E. T. Jaynes in 1957 [105, 106], highlighting a deep connection between statistical mechanics and information theory. The concept of entropy itself is deeply rooted in both domains, thanks to the foundational work of Boltzmann and Shannon. In particular, the relative entropy of a distribution P with respect to a reference one P_0 – which encodes prior knowledge about the system – is defined in the discrete case as

$$S[P||P_0] = -\sum_x P(x) \log \frac{P(x)}{P_0(x)} = -\left\langle \log \frac{P(x)}{P_0(x)} \right\rangle_P. \tag{1.3}$$

Unlike Shannon's original definition of differential entropy, relative entropy is invariant under coordinate transformations and coarse-graining, making it

a robust tool in physical modeling. A central result of statistical mechanics – the derivation of the canonical ensemble – can be viewed as a special case of the maximum entropy principle. By maximizing the relative entropy under a constraint on the average energy $\langle E \rangle$, and assuming a uniform prior P_0 over all microstates in the system's phase space, one recovers the canonical Boltzmann distribution. Jaynes generalized this idea into a framework for statistical inference, postulating that the probability distribution best representing the current state of knowledge (given a prior information P_0) is the one that maximizes relative entropy subject to experimental constraints.

In the context of molecular simulations, the reference distribution P_0 corresponds to the ensemble generated by an unrefined MD simulation, which (a part from the case of enhanced sampling based on biasing) is uniform over the sampled frames x. Experimental data $\vec{g}_{exp}$ are associated to the average values of the observables $\vec{g}_x$ over the ensemble $P(x)$ – denoted $\langle \vec{g}(x) \rangle_P$. The maximum entropy principle then determines the optimal ensemble P^* by solving:

$$\begin{cases} P^*(x) = \arg\max\limits_{P(x)} S[P||P_0] \\ \sum\limits_x P(x) = \langle 1 \rangle_P = 1 \\ \langle \vec{g}(x) \rangle_P = \vec{g}_{exp} \end{cases} . \tag{1.4}$$

Here, the vectors $\vec{g}(x)$ and $\vec{g}_{exp}$ have dimension equal to the number of experimental constraints M, whereas the configuration $\vec{x}$ lies in a space of dimension equal to the number of degrees of freedom (e.g., $3n$ for a system of n atoms in 3d space). The sum over x is understood as summing over the N frames sampled by the MD simulation.

Since the relative entropy $S[P||P_0]$ is the opposite of the Kullback-Leibler divergence $S[P||P_0] = -D_{KL}[P||P_0]$, the maximum (relative) entropy principle can also be interpreted as finding the posterior distribution P^* that is as close as possible to the prior P_0, while being consistent with experimental data – where "closeness" is quantified by the Kullback-Leibler divergence.

This maximization problem Eq. 1.4 can be carried out via the method of Lagrange multipliers, by searching for the stationary points of the Lagrangian

$$\mathcal{L}(P(q), \vec{\lambda}, \mu) = S[P||P_0] - \vec{\lambda} \cdot (\langle \vec{g} \rangle_P - \langle g_{exp} \rangle) - \mu \Big(\sum_x P(x) - 1 \Big) \tag{1.5}$$

where $\vec{\lambda}$ and μ are the Lagrange multipliers. Taking the functional derivative of $\mathcal{L}$ with respect to $P(x)$, we get:

$$\frac{\delta \mathcal{L}}{\delta P(x)} = -\log \frac{P(x)}{P_0(x)} - 1 - \vec{\lambda} \cdot \vec{g}_i(x) - \mu. \tag{1.6}$$

Setting this to zero $\frac{\delta \mathcal{L}}{\delta P(x)} = 0$ yields a proportionality relationship

$$P^*(x) \propto P_0(x)\, e^{-\vec{\lambda}\cdot\vec{g}(x)}, \tag{1.7}$$

which mirrors the form of the canonical ensemble $P^*(x) = \frac{1}{Z} e^{-\beta E(x)}$, with $\beta^{-1} = k_B T$ (being k_B the Boltzmann constant and T the temperature of the system). Taking advantage of this formal result, we can restrict the numerical minimization to the subset of ensembles parametrized by $\vec{\lambda}$ coefficients P_λ:

$$P_{\vec{\lambda}}(x) = \frac{1}{Z_{\vec{\lambda}}} P_0(x) \exp\Big(-\vec{\lambda}\cdot\vec{g}(x)\Big) \tag{1.8}$$

where $Z_{\vec{\lambda}}$ is the normalization factor $Z_{\vec{\lambda}} = \langle e^{-\vec{\lambda}\cdot\vec{g}(x)}\rangle_{P_0}$. Substituting $P_{\vec{\lambda}}$ to P in the relative entropy $S[P||P_0]$ and exploiting the constraint $\langle\vec{g}(x)\rangle = \vec{g}_{exp}$, the original problem Eq. 1.4 becomes equivalent to minimizing a different function[4]

$$\Gamma(\vec{\lambda}) = \log Z_\lambda + \vec{\lambda}\cdot\vec{g}_{exp}. \tag{1.9}$$

This formulation yields the same optimal coefficients λ^* and the stationary value of the Lagrangian satisfies $\mathcal{L}(P_{\lambda^*}) = \Gamma(\vec{\lambda}^*)$. Note the switch from a maximization to a minimization problem[5]: maximizing the relative entropy under constraints is equivalent to minimizing $\Gamma(\vec{\lambda})$. The key practical advantage is a significant reduction in dimensionality, as the optimization now occurs in $\mathbb{R}^M$, where M is the number of experimental observables, usually much lower than the number of frames $M \ll N$. This optimization problem can be efficiently solved using numerical methods such as the BFGS algorithm [107], taking advantage of the convexity of $\Gamma(\vec{\lambda})$.

It is important to note that a solution to the maximization problem in Eq 1.4 is not guaranteed to exist. For example, if every conformation x in the sampled ensemble satisfies $g(x) < g_{exp}$, no ensemble average can reach the constraint. Similarly, mutually inconsistent experimental constraints may lie outside the convex hull of the original ensemble. In such cases, relaxing the constraints (via flexible restraints or regularization) may be needed.

When the maximum (relative) entropy principle is applied to MD simulations, $P_0(x)$ samples the canonical distribution corresponding to the reference

[4] In all the rest of this Thesis, we will omit vector arrows over symbols (e.g., $\vec{\lambda}$ in $P_{\vec{\lambda}}$, $Z_{\vec{\lambda}}$) when the meaning is unambiguous from the context.

[5] From a mathematical perspective, the function $\Gamma(\vec{\lambda})$ is the dual function obtained via the Legendre transform of the entropy maximization problem, in the same way as the thermodynamic entropy is expressed in terms of the free energy $F = -\beta^{-1}\log Z_\beta$ by the relationship $TS = E - F$. Indeed, Eq 1.9 with the constraint on the average energy E reads: $\Gamma(\beta) = \log Z_\beta + \beta E = \beta(\beta^{-1}\log Z_\beta + E) = \beta(E - F) = S(\beta)$.

force field $V_0(x)$ and the posterior distribution defines an effective potential

$$V^*(x) = V_0(x) + k_B T \, \vec{\lambda} \cdot \vec{g}(x). \tag{1.10}$$

Hence, the effect of maximum entropy principle is to add a bias term to the force field that is linear in the observables $\vec{g}(x)$. This stands in contrast to methods used in constrained MD simulations, such as SHAKE [108], which enforce observables at each time step, and to harmonic restraints, which introduce quadratic penalties.

Any experimental observable that is the result of an ensemble-averaged measurement (bulk experiment) can be used as a constraint. In the context of biomolecular systems, typical examples include NMR observables such as chemical shifts, scalar couplings or NOEs (see subsection 2.1.5). Mapping atomic configurations to measured quantities relies on forward models, such as the Karplus equations 1.1 for ^3J scalar couplings, which are often empirically parametrized and may carry associated uncertainties.

Extension to experimental uncertainties (flexible restraints)

The method described so far, based on the maximum entropy principle, has been extended in order to account for uncertainties in the experimental data. Notable contributions in this direction include the work of A. Cesari, a former PhD student in Giovanni Bussi's lab [19, 96, 109], and the development of the `BioEn` package in 2018 by G. Hummer, J. Köfinger, and collaborators [99, 102, 110, 111], available at `https://github.com/bio-phys/BioEn/tree/master`.

To relax rigid constraints into flexible restraints, these approaches introduce an error function that quantifies the discrepancy between the ensemble averages and the experimental measurements. A common choice is the χ^2 function:

$$\chi^2[P] = \sum_{i=1}^{M} \left(\frac{\langle g_i \rangle_P - g_{i,exp}}{\sigma_{i,exp}} \right)^2 \tag{1.11}$$

where $\sigma_{i,exp}$ denotes the estimated uncertainty of the experimental measurement $g_{i,exp}$. By replacing the rigid constraints in 1.4 with this soft penalization term, the loss function takes the form

$$\mathcal{L}[P] = \frac{1}{2}\chi^2[P] + \alpha D_{KL}[P||P_0] \tag{1.12}$$

where the factor $1/2$ is included for analogy with harmonic potentials, and $\alpha > 0$ is a hyperparameter controlling the trade-off between agreement

with experimental data and closeness to the original ensemble P_0.[6] This regularization introduces a tunable balance between trusting experimental measures and original ensemble. Small values of α prioritize agreement with experiment (allowing large deviations from P_0), while large values favor preserving the original simulation ensemble at the cost of poorer agreement. The optimal range of values for α is typically chosen via cross-validation, selecting the regime (intermediate between under- and over-fitting) that minimizes the χ^2 on a set of left-out observables.

From an optimization perspective, this loss function represents a Pareto compromise [104] between two objectives: minimizing both the discrepancy with experiment and the deviation from the prior distribution. This is implemented as a linear scalarization of a multi-objective optimization problem, with $\alpha \geq 0$ setting the relative weight of the two objective functions $1/2\chi^2[P]$ and $S[P||P_0]$. Furthermore, this loss function admits a Bayesian interpretation as (minus) a log-posterior: the term $-\frac{1}{2}\chi^2[P]$ corresponds to the log-likelihood of the data $\mathcal{D}$ given the ensemble P, while $-\alpha D_{KL}[P||P_0]$ acts as a log-prior favoring distributions close to the unrefined ensemble P_0:

$$\mathcal{P}(P|\mathcal{D}, P_0) \propto \mathcal{P}(\mathcal{D}|P)\,\mathcal{P}(P|P_0) = e^{-\frac{1}{2}\chi^2[P]}\, e^{-\alpha D_{KL}[P|P_0]}. \tag{1.13}$$

Consistently performing Bayesian sampling over a full set of ensembles is, to the best of our knowledge, an open issue. Several techniques have been proposed in the literature [98, 99, 112], each with its own limitations (such as prior normalization and invariance upon coarse-graining). One of the goals of this Thesis (discussed in Chapter 4) is to derive a suitable strategy to correctly sample from the posterior distribution over a set of ensembles.

Through analytical calculations, one can show that the optimal ensemble P^* minimizing $\mathcal{L}$,

$$P^* = \arg\min_P \mathcal{L}[P], \tag{1.14}$$

retains the same form as in the original maximum-entropy formalism 1.8, but the interpretation of the coefficients $\vec{\lambda}$ changes. Instead of Lagrange multipliers, the λ_i coefficients are now determined by the system of implicit equations

$$\lambda_i = \frac{\langle g_i \rangle_{P_\lambda} - g_{i,exp}}{\alpha \sigma^2_{i,exp}}. \tag{1.15}$$

[6] Note that the hyperparameter α is unrelated to the α dihedral angle in the RNA backbone, despite the reuse of the symbol; the meaning should be clear from the context. The same clarification applies to the symbols β, γ, which will also denote hyperparameters introduced below, as well as to χ, which refers both to a dihedral angle and to the statistical χ^2 error.

Taking advantage of this formal structure, we again reduce the optimization to the low-dimensional parameter space of $\vec{\lambda}$. Substituting P_λ into the loss function 1.12 and using the relationship above 1.15, the minimization of $\mathcal{L}$ becomes equivalent to minimizing:

$$\tilde{\Gamma}(\lambda) = \Gamma(\lambda) + \frac{1}{2}\alpha \sum_{j=1}^{M} \sigma_j^2 \lambda_j^2 = \log Z_\lambda + \vec{\lambda} \cdot \vec{g}_{exp} + \frac{1}{2}\alpha \sum_{j=1}^{M} \sigma_j^2 \lambda_j^2, \tag{1.16}$$

which provides substantial computational savings. This reformulated objective function has the same minimum λ^* as the original loss function 1.12, and the relationship $\mathcal{L}(P_{\lambda^*}) = -\alpha\tilde{\Gamma}(\lambda^*)$ holds. Additionally, derivative computations are simplified in this parameterized space. For instance, the covariance matrix of the observables $C_{ij}(\lambda) = \langle g_i g_j \rangle_\lambda - \langle g_i \rangle_\lambda \langle g_j \rangle_\lambda$, appears in the gradient $\partial\mathcal{L}/\partial\lambda_i$, but simplifies in the gradient of $\tilde{\Gamma}(\vec{\lambda})$, further facilitating optimization.

1.4.2 Force-field refinement

While ensemble refinement operates independently on each molecular system, there may exist corrections to the force field that are common across different systems. The idea behind force-field refinement [13, 14, 19, 113–121] is to exploit these transferable corrections, that, once learned, can be used to perform new MD simulations of a broader range of molecules in better agreement with experiments. Instead of optimizing system-specific correction terms as in ensemble refinement, force-field refinement imposes a fixed functional form, typically motivated by chemical intuition (e.g., dihedral potentials) or, alternatively, parameterized through machine-learning models such as neural-network potentials (NNPs).

We denote this functional form for the correction to the force field as $\Delta V(q;\vec{\phi})$, where q represents molecular coordinates and $\vec{\phi}$ is a set of tunable parameters. In the simplest case, the correction is linear in the coefficients $\Delta V(q;\vec{\phi}) = \vec{\phi} \cdot \vec{f}(q)$ where $\vec{f}(q)$ is a vector of basis functions. More complex and non-linear functional forms for $\Delta V(q;\vec{\phi})$ are also possible.

To determine the optimal parameters $\vec{\phi}^*$, one minimizes a loss function that combines a χ^2 term, quantifying the discrepancy between predicted and experimental observables across multiple systems, with a regularization term $R[\vec{\phi}|\vec{\phi}_0]$ to prevent overfitting:

$$\mathcal{L}(\vec{\phi}) = \frac{1}{2}\chi^2[P_{\vec{\phi}}] + \beta R[\vec{\phi}|\vec{\phi}_0]. \tag{1.17}$$

Here, $P_{\vec{\phi}}$ is the ensemble corresponding to the modified force field

$$P_{\vec{\phi}}(x) = \frac{1}{Z_\phi} P_0(x) \exp(-\Delta V_\phi(x)/k_B T) \tag{1.18}$$

and $\vec{\phi}_0$ is a reference parameter set from the original force field. The regularization term $R[\vec{\phi}|\vec{\phi}_0]$ can take various forms, such as: the $L2$ distance $||\vec{\phi}-\vec{\phi}_0||^2$ (corresponding to a Gaussian prior centered on ϕ_0) or the Kullback-Leibler divergence $D_{KL}[P_{\vec{\phi}}||P_{\vec{\phi}_0}]$, which penalizes distributional shifts [118]. The hyperparameter $\beta > 0$ controls the trade-off between data fidelity and adherence to the prior force field, playing a role analogous to α in ensemble refinement.

While this loss function closely resembles that of ensemble refinement, a key difference lies in the restricted variational freedom: the ensemble $P_{\vec{\phi}}$ is now constrained by the selected functional form of the force-field correction ΔV_ϕ. This constraint enables transferability of the learned corrections to new molecular systems, though at the cost of reduced flexibility compared to ensemble refinement.

A major advantage of force-field refinement is that it allows simultaneous optimization across multiple molecular systems, rather than performing independent minimizations as in ensemble refinement. The overall loss function becomes a sum over different systems

$$\mathcal{L}(\vec{\phi}) = \sum_{i=1}^{N_{sys}} \Big[\chi^2[P^i_{\vec{\phi}}] + \beta R_i[\vec{\phi}|\vec{\phi}_0]\Big]. \tag{1.19}$$

Unlike ensemble refinement, where each system yields its own set of $\vec{\lambda}$ coefficients, here a single set of parameters $\vec{\phi}$ is learned and applied to all systems simultaneously.

In summary, force-field refinement supports generalization by identifying transferable corrections, enabling application to systems beyond the training set. However, this benefit comes at the cost of reduced flexibility, as the corrections are constrained to a predefined functional form – unlike ensemble refinement, which allows system- and even residue-specific adjustments.

1.4.3 Forward-model refinement and combined approaches

When comparing MD simulations with experimental measurements, an additional source of error arises from the forward models – the functions mapping structural features of MD configurations to experimentally measurable observables. These forward models involve empirical parameters that have been determined using experimental or quantum-mechanical data, and are thus associated with their own uncertainties. Since the forward models might significantly contribute to the overall discrepancy between simulation and experiment, refining them in isolation from the structural ensemble can lead to suboptimal results. A more consistent approach refines them simultaneously

with the structural ensembles. This strategy was adopted by T. Fröhlking and collaborators [1], who focused on the case of Karplus equations for RNA oligomers.

Let $\vec{\theta}$ denote the coefficients of the forward model that we wish to refine. The loss function used for ensemble refinement is then modified as:

$$\mathcal{L}[P,\vec{\theta}] = \frac{1}{2}\chi^2[P,\vec{\theta}] + \alpha D_{KL}[P||P_0] + \gamma R(\vec{\theta}|\vec{\theta}_0) \tag{1.20}$$

where $R(\vec{\theta}|\vec{\theta}_0)$ is a regularization term that penalizes large deviations from the original forward model coefficients $\vec{\theta}_0$, and $\gamma > 0$ is the associated hyper-parameter (analogous to α and β introduced earlier). A typical choice is the $L2$ regularization

$$R(\vec{\theta}|\vec{\theta}_0) = \frac{1}{2}\sum_i (\theta_i - \theta_{0i})^2. \tag{1.21}$$

Since the parameters $\vec{\theta}$ are shared across different molecular systems, this joint refinement is generalizable to multiple systems. Similar to standard ensemble refinement, a dimensionality reduction from the number of frames N to the number of observables M (with $M \ll N$) can be exploited. To do so, the optimization is split into an inner minimization over P at fixed $\vec{\theta}$, and an outer minimization over $\vec{\theta}$. Using results from ensemble refinement, the optimization becomes:

$$\arg\min_{P,\vec{\theta}} \mathcal{L}[P,\vec{\theta}] = \arg\min_{\vec{\theta}} \Big[-\alpha \min_{\lambda} \tilde{\Gamma}(\vec{\lambda};\vec{\theta}) + \gamma R(\vec{\theta}|\vec{\theta}_0) \Big]. \tag{1.22}$$

Here, observables are computed using the forward model parametrized by $\vec{\theta}$, leading to an optimal ensemble of the form

$$P_{\lambda,\theta} = \frac{1}{Z_{\lambda\theta}} P_0(x) e^{-\vec{\lambda}\cdot\vec{g}(x;\vec{\theta})} \tag{1.23}$$

with normalization

$$Z_{\lambda\theta} = \langle e^{-\vec{\lambda}\cdot\vec{g}(x;\vec{\theta})} \rangle_{P_0}. \tag{1.24}$$

The modified function $\tilde{\Gamma}(\vec{\lambda};\vec{\theta})$ depends on the forward-model parameters $\vec{\theta}$, since these enter ensemble definition through the partition function $Z_{\lambda\theta}$:

$$\tilde{\Gamma}(\vec{\lambda};\vec{\theta}) = \log Z_{\lambda\theta} + \vec{\lambda}\cdot\vec{g}_{exp} + \frac{1}{2}\alpha \sum_{j=1}^{M} \sigma_j^2 \lambda_j^2. \tag{1.25}$$

One of the goals of this Thesis (see Chapter 2 and Chapter 3) is to extend this combined refinement strategy to include flexible force-field refinement, thereby integrating ensemble optimization, forward-model calibration and force-field correction into a single, consistent framework.

1.4.4 Available packages

General-purpose scripts have been developed to implement some of the integrative techniques described so far. For ensemble refinement, two notable Python tools based on the Bayesian maximum entropy principle with soft restraints are available. The first, developed by G. Hummer, J. Köfinger, and collaborators, is *Bayesian Inference Of ENsembles* (abbreviated as `BioEn`), available at `https://github.com/bio-phys/BioEn` [99, 102, 110, 111]. The second, developed by S. Bottaro, T. Bengtsen and K. Lindorff-Larsen, is known as *Bayesian/MaxEnt* `BME`, and is available at the GitHub repository `https://github.com/KULL-Centre/BME` [103]. Both tools have been applied to a wide range of experimental techniques including NMR, SAXS and SANS [22, 110, 122].

In parallel, several software packages have been developed for top-down force-field refinement via minimization of the χ^2, complementing bottom-up and coarse-grained approaches. These Python tools leverage the automatic differentiation via the `Jax` framework, enabling differentiable MD simulations as implemented in `JAX, M.D.` [123], developed by Google Research and available at `https://github.com/jax-md/jax-md`. Among these tools are `DMFF` (*Differentiable Molecular Force Field*) [124], created by the DeepModeling community (`https://github.com/deepmodeling/DMFF`), and `chemtrain` [125], developed by P. Fuchs and collaborators in the group led by J. Zavadlav (`https://github.com/tummfm/chemtrain`). Moreover, the usage of automatic differentiation facilitates the implementation and optimization of complex neural networks for machine-learned potentials, as performed by `chemtrain`, with several applications in materials science and biophysics [14, 15]. However, in their standard implementation, these packages do not directly support the refinement of structural ensembles or forward models.

The generality of the process of adjusting force-field parameters including multiple systems, and the possibility to combine it with refinements of ensembles and forward models result in a significant increase in complexity of these methods, which often require substantial amount of programming. To address this, and as part of the work presented in this Thesis, we developed `MDRefine`, a Python toolkit designed to facilitate the analysis and comparison of MD-generated trajectories against experimental data. The structure and functionality of this package are detailed in Chapter 3, along with its application to representative benchmark systems.

Chapter 2

Boosting ensemble refinement with transferable force-field corrections

In this Chapter, we present a novel method that combines ensemble refinement based on the maximum entropy principle (ER) with force-field refinement (FFR). The content is largely adapted from our published work [126]. Several notebooks reproducing the results are available in the GitHub repository https://github.com/bussilab/force-field-ensemble-refinement.

As described in the previous Chapter, ER and FFR play a major role among the different integrative methods. However, to the best of our knowledge, they have traditionally been applied separately, depending on the specific case of interest. By combining them into a unified framework, we enable a continuous interpolation in between these two methods, which can thus be interpreted as two limiting cases. This integration allows to capture in the same framework both the high flexibility of ER and the transferability across different systems proper of FFR. A cross-validation procedure enables to correctly assess the relative weight of both of them, distinguishing scenarios where the combined approach is optimal from those in which either ER or FFR separately are a best option.

After a description of the employed method, we show the behavior of this approach on a toy model as a proof of concept. Then, we use the method to derive ensembles and force-field corrections for RNA oligomers, for which a carefully performed cross-validation procedure is necessary to tune the hyperparameters. Within the new scheme, MD simulations are integrated with experimental data provided by NMR measures. We show that force-field corrections are in general superior when applied to the appropriate force-field terms but are automatically discarded by the method when applied to inappropriate force-field terms.

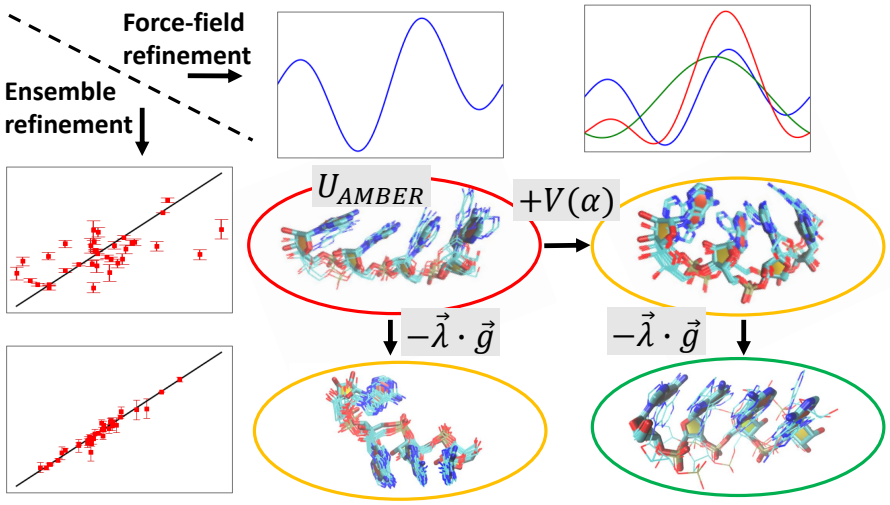


Figure 2.1: Schematic illustration of the method: starting from the reference ensemble generated by the MD simulation (red ellipse), force-field refinement is first applied (horizontal arrow), followed by ensemble refinement (vertical arrow). This sequence of consecutive steps leads to the final ensemble (green ellipse) refined by the combined approach, different from performing ensemble refinement directly on the initial ensemble.

2.1 Methods

Based on the maximum entropy principle, ER selects the ensemble which best describes the experimental measures and is, at the same time, as close as possible to the initially hypothesized one. In doing so, ER is agnostic with respect to the knowledge of the force field parametrization: the functional form of the corrections carried to the initial ensemble only depends on the selected observables [97, 109] and thus the corrections are not directly transferable to different systems. Conversely, FFR is based on a reasonable guess of the force-field correction terms. Their optimal coefficients are determined by minimizing a loss function that includes the discrepancy from experimental data and, in modern implementations, a regularization term [19, 33, 115, 117, 118, 127] that penalizes moving away from the initial force field, in a Bayesian line of thought. This approach enables one to encode prior information about the reliability of a given force-field term, by choosing which specific term should be refined, and makes the resulting corrections transferable to other systems [128]. However, adding the same force-field correction terms to all the

copies of a given residue could be over-limiting and not able to capture further relevant differences among them. Indeed, the functional form of the force-field is limited and might be intrinsically unable to reproduce experimental data.

These two categories of methods have been traditionally derived in a different manner. Only recently, a formulation of the FFR approach that formally relates to the maximum entropy principle has been proposed [118]. Importantly, methods of the two classes have been so far used in a disjoint fashion. The user is thus expected to decide, based on experience, if transferable or non-transferable corrections are performing best for a given system.

We here introduce a procedure to seamlessly combine the ER method with FFR, as schematically illustrated by Fig. 2.1. This allows to preserve the flexibility of ER while at the same time ensuring the transferability of the resulting force-field corrections to different molecules as in FFR.

The main concept at the basis of this combined approach is the following. Starting from force-field refinement, we add system-specific flexibility to the force field corrections by further refining the ensemble P_ϕ before its comparison with experimental measures. This is achieved by minimizing the loss function

$$\mathcal{L}[P, P_\phi] = \frac{1}{2}\chi^2[P] + \alpha D_{KL}[P||P_\phi] + \beta D_{KL}[P_\phi||P_0] \tag{2.1}$$

where the two hyperparameters α, β regulate the reliability of the initial hypotheses (the force-field functional form P_ϕ and the reference ensemble P_0). While the first term $\frac{1}{2}\chi^2$ quantifies the agreement of P with experiments, the following two regulate the closeness of P from P_ϕ and of P_ϕ from the original ensemble P_0, respectively. This amounts to introducing ER as an intermediate step in between the FFR step and the comparison with experiments as illustrated in Fig. 2.1 with the aim of fine-tuning P_ϕ in a new ensemble P – compare the loss functions in Eqs. 2.1 and 1.17. The former step (FFR) is expected to take into account any transferable contribution, and to leverage on the knowledge of which force-field terms might benefit a refinement. The latter step (ER) enables to include the flexibility missing by FFR applied alone, ensuring the final ensemble averages better agree with experiment. This method reduces to FFR only for $\alpha = \infty$, i.e., maximally penalizing flexible deviations from P_ϕ, and to ER only for $\beta = \infty$ (no adjustment of force-field coefficients).

The optimal solution is found by minimizing the loss function:

$$(\phi^*, P^*) = \arg\min_{\phi,P} \mathcal{L}[P, P_\phi]. \tag{2.2}$$

Repeating the same calculations as before (Eqs. 1.15, 1.16), the optimal ensemble P^* (at given ϕ and so also for the optimal set of values ϕ^*) has

functional form

$$P_{\lambda\phi}(x) = \frac{1}{Z_{\lambda\phi}} P_\phi(x) \exp\Big(-\sum_{i=1}^{N_{exp}} \lambda_i g_i(x)\Big), \tag{2.3}$$

where $\vec{\lambda}$ satisfies the condition in Eq. 1.15 and

$$Z_{\lambda\phi} = \langle e^{-\vec{\lambda}\cdot\vec{g}(x)} \rangle_{P_\phi} \tag{2.4}$$

the normalization factor. This allows to employ the $\vec{\lambda}, \vec{\phi}$ parametrization in order to restrict the minimization of the loss function to the ensembles $P_{\lambda\phi}$ parametrized as in Eq. 2.3. By doing so, one avoids the need to explore the full space of all the ensembles, whose dimensionality equals the number of frames (minus 1, due to normalization).

2.1.1 Interpretation of the hyperparameters

The two hyperparameters α, β tune the confidence in the initial assumptions. In particular, α regulates how much we trust the force-field correction ΔV_ϕ to account for missing refinements. Low values of α make the loss function less sensitive to changes in the ensemble, so even significant deviations from P_ϕ result in only a small increase in the loss. This allows the refinement to freely explore alternative ensembles, as long as there is even a modest improvement in agreement with experimental data. In contrast, high values of α heavily penalize deviations from P_ϕ: even small changes lead to a large increase in the loss function. Hence, in this regime, such deviations are only justified if they result in a substantial reduction in the χ^2, i.e., a much better fit to experimental observables. The hyperparameter β, on the other hand, tunes the degree of reliability of the prior potential U_0, regarding only the ΔV_ϕ correction.

The following limiting cases appear:

- both $\alpha, \beta \to \infty$: we completely trust our initial hypothesis (both the force-field model and its coefficients) $P = P_\phi = P_0$ (no refinement); the loss function acquires finite value only at $\vec{\phi} = 0$, $\vec{\lambda} = 0$, corresponding to $\frac{1}{2}\chi^2$ for the reference ensemble;
- only $\alpha \to \infty$: we believe the force field correction ΔV_ϕ to be able to catch all the missing information required to appropriately describe the molecules, and we only need to refine its coefficients; the solution is restricted to the form $P = P_\phi$, i.e., the combined approach reduces to FFR only;

- only $\beta \to \infty$: we believe refining the force field correction ΔV_ϕ does not improve the description of the systems (so we keep $P_\phi = P_0$), and further (possibly system-specific) flexibility is required; the combined approach reduces to ER only;

- $\alpha, \beta \to 0$: the initial assumptions are considered quite unreliable with respect to the accuracy of experimental values. In limiting cases where the hyperparameters are very small, the model may overfit the experimental data, neglecting their intrinsic experimental uncertainty. Setting $\alpha = 0$ (while keeping β finite) completely decouples force-field refinement from comparison with experimental data, nullifying the combined approach. In this case, minimizing $D_{KL}[P_\phi|P_0]$ forces $\vec{\phi} = 0$, thereby eliminating any force-field correction. The optimization then reduces to finding the ensemble P that best fits the experimental data by minimizing $\chi^2[P]$, independently of the force field. In absence of a relative-entropy penalization for P, the ensemble will not acquire the functional form P_λ and we disregarded this possibility.

By suitably choosing finite values of α, β we can optimize the performance of the combined approach. This fine-tuning of the hyperparameters is implemented through cross validation, described in subsection 2.1.5.

2.1.2 Generalization to multiple systems

In order to fully exploit the potentiality of the presented method, it is beneficial to consider more than one molecular system. In this way, one can achieve a better refinement of the force field parameters $\vec{\phi}$, which are common to different systems. Let's use the index k to enumerate the different molecular systems $k = 1...N_{sys}$, with P^k the corresponding ensembles and $\vec{\lambda}_k$ the system-specific ER coefficients (see Eq. 1.8). The loss function can be written as the sum of Eq. 2.1 over different systems

$$\mathcal{L}\left(\{\vec{\lambda}_k\}, \vec{\phi}\right) = \sum_{k=1}^{N_{sys}} \mathcal{L}[P^k_{\lambda_k \phi}, P^k_\phi] \tag{2.5}$$

In this way, each system contributes to the χ^2 with a weight proportional to the number of the experimental data available for that system. Notice that, in principle, each system could be included with separated hyperparameters α, β, or with a separate weight for its χ^2. In this work, we consider a homogeneous set of systems, so we do not exploit this possibility.

2.1.3 Minimization strategy

To introduce the minimization strategy, we first consider a single molecular system. Minimizing the loss function $\mathcal{L}(\vec{\lambda}, \vec{\phi})$ given by Eq. 2.1 through the $\vec{\lambda}$ parametrization Eq. 2.3 requires evaluation of the covariance matrix

$$C_{ij}(\lambda, \phi) = \langle g_i g_j \rangle_{\lambda\phi} - \langle g_i \rangle_{\lambda\phi} \langle g_j \rangle_{\lambda\phi}, \tag{2.6}$$

which appears in the derivatives with respect to $\vec{\lambda}$

$$\frac{\partial \mathcal{L}}{\partial \lambda_i} = -\sum_j \Big(\frac{\langle g_j \rangle_{\lambda\phi} - g_{j,exp}}{\sigma^2_{j,exp}} - \alpha \lambda_j \Big) C_{ij}(\lambda, \phi). \tag{2.7}$$

This step becomes computationally expensive as the number M of available experimental data increases, since the number of independent components in the covariance matrix scales as M^2. To reduce this cost, we adopt a two-step minimization approach, firstly minimizing $\mathcal{L}(\vec{\lambda}, \vec{\phi})$ over $\vec{\lambda}$ at fixed $\vec{\phi}$. This is analogous to Eq. 1.22 for the combination with forward-model refinement – where the parametrization P_λ in Eq. 1.8 has been substituted in the loss function of Eq. 1.20. In this way, we get

$$(\vec{\lambda}^*, \vec{\phi}^*) = \arg\min_{\vec{\lambda}, \vec{\phi}} \mathcal{L}(\vec{\lambda}, \vec{\phi}) = \arg\min_{\vec{\phi}} \Big[-\alpha \min_{\vec{\lambda}} \tilde{\Gamma}(\vec{\lambda}; \vec{\phi}) + \beta D_{KL}[P_\phi || P_0] \Big] \tag{2.8}$$

where we defined

$$\tilde{\Gamma}(\vec{\lambda}; \vec{\phi}) = \frac{1}{2} \alpha \sum_j \sigma_j^2 \lambda_j^2 + \log Z_{\lambda\phi} + \vec{\lambda} \cdot \vec{g}_{exp}. \tag{2.9}$$

Notice that $\tilde{\Gamma}(\vec{\lambda}; \vec{\phi})$ depends on $\vec{\lambda}$ exactly as the function $\tilde{\Gamma}(\vec{\lambda})$ derived in the maximum relative entropy approach [19], except that now it is evaluated with respect to the reference ensemble P_ϕ rather than P_0.

This formulation flipped the minimum in the $\vec{\lambda}$ direction to a maximum, and so the global minimum of $\mathcal{L}$ becomes a saddle (or mini-max) point of a different function $\tilde{\mathcal{L}}$

$$(\lambda^*, \phi^*) = \arg\min_{\lambda, \phi} \mathcal{L}(\lambda, \phi) = \arg\min_{\phi} \max_{\lambda} \tilde{\mathcal{L}}(\lambda, \phi) \tag{2.10}$$

with

$$\tilde{\mathcal{L}}(\vec{\lambda}, \vec{\phi}) = -\alpha \tilde{\Gamma}(\vec{\lambda}; \vec{\phi}) + \beta D_{KL}[P_\phi || P_0]. \tag{2.11}$$

The search for the saddle point is tackled by performing a nested minimization, with the inner one of $\Gamma(\vec{\lambda}; \vec{\phi})$ over $\vec{\lambda}$ (at given $\vec{\phi}$) and the outer

over $\vec{\phi}$ coefficients (see Eq. 2.8). Although the search for a saddle point is in general more challenging than a minimization, the new function $\tilde{\mathcal{L}}(\vec{\lambda}, \vec{\phi})$ is considerably simpler than $\mathcal{L}(\vec{\lambda}, \vec{\phi})$, especially in terms of computational cost. Indeed, the derivatives with respect to $\vec{\lambda}$ simplify to

$$\frac{\partial \tilde{\mathcal{L}}(\vec{\lambda}, \vec{\phi})}{\partial \lambda_j} = \alpha(\langle g_j \rangle_{\lambda\phi} - g_{j,exp} - \alpha\sigma_j^2 \lambda_j). \tag{2.12}$$

In addition, the function $\tilde{\mathcal{L}}$ has the advantage of being concave over $\vec{\lambda}$, since $\tilde{\Gamma}(\vec{\lambda}; \vec{\phi})$ is convex over $\vec{\lambda}$.

Notice that min/max can be exchanged provided that the hypotheses of the minimax theorem are satisfied: $\tilde{\mathcal{L}}(\vec{\lambda}; \vec{\phi})$ is required to be concave as a function of λ $\forall\phi$ and convex as a function of ϕ $\forall\lambda$. While the first condition is always satisfied, the second condition is not guaranteed, so we proceed with the inner minimization over $\vec{\lambda}$, as described.

Assuming corrections to the force field that are linear in the coefficients $\vec{\phi}$, we can explicate D_{KL} in Eq. 2.11 and the loss function $\tilde{\mathcal{L}}$ can be written as

$$\tilde{\mathcal{L}}(\vec{\lambda}, \vec{\phi}) = -\alpha\tilde{\Gamma}(\vec{\lambda}; \vec{\phi}) - \beta(\log Z_\phi + \vec{\phi} \cdot \langle \vec{f} \rangle_\phi), \tag{2.13}$$

with partial derivatives with respect to $\vec{\phi}$

$$\frac{\partial \tilde{\mathcal{L}}(\vec{\lambda}, \vec{\phi})}{\partial \phi_k} = \alpha(\langle f_k \rangle_{\lambda\phi} - \langle f_k \rangle_\phi) + \beta\vec{\phi} \cdot (\langle \vec{f} f_k \rangle_\phi - \langle \vec{f} \rangle_\phi \langle f_k \rangle_\phi). \tag{2.14}$$

Both the inner and outer minimizations can be performed with second-order methods, i.e. preconditioning the gradient with curvature information, by employing BFGS or L-BFGS-B methods (provided by the Python library SciPy). For multiple systems, we proceed analogously, noting that the $\vec{\phi}$ coefficients are shared across different systems and so each system contributes to this gradient component $\frac{\partial \tilde{\mathcal{L}}(\vec{\lambda}, \vec{\phi})}{\partial \phi_k}$. Conversely, the $\vec{\lambda}$ coefficients are system-specific $\vec{\lambda}^{isys}$, so each system only contributes to its own corresponding gradient component $\frac{\partial \tilde{\mathcal{L}}(\vec{\lambda}, \vec{\phi})}{\partial \lambda_j^{isys}}$.

2.1.4 Calculations using reweighting

In the above derivation we assumed to be able to compute averages over the ensembles P_0, $P_{\lambda\phi}$ and P_ϕ. By performing MD simulations, and assuming ergodicity of the sampled trajectories, we can replace averages of any observable $\mathcal{O}$ over the configuration space (discretized into cells x) with averages over frames (with normalized weight w_t) of the MD trajectory sampling the ensemble P:

$$\langle \mathcal{O} \rangle_P = \sum_x \mathcal{O}(x) P(x) \simeq \sum_t \mathcal{O}_t w_t. \tag{2.15}$$

However, this would require to perform MD simulations for each considered ensemble, which is in practice unfeasible. To circumvent this issue, one can exploit the relationships of P_ϕ and $P_{\lambda\phi}$ to the initial ensemble P_0 reported in Eqs. 1.18 and 2.3. In this way, once the initial ensemble P_0 has been accurately sampled through a MD simulation, the expectation values over $P_{\lambda\phi}$, P_ϕ can be traced back to computing averages over P_0, in a reweighting procedure described by the following equations:

$$\langle \mathcal{O} \rangle_\phi = \frac{1}{Z_\phi} \sum_x \mathcal{O}(x) P_0(x)\, e^{-\beta \Delta V_\phi(x)} = \frac{\langle \mathcal{O}(x)\, e^{-\beta \Delta V_\phi(x)} \rangle_0}{\langle e^{-\beta \Delta V_\phi(x)} \rangle_0} \tag{2.16}$$

$$\begin{aligned} \langle \mathcal{O} \rangle_{\lambda\phi} &= \frac{1}{Z_{\lambda\phi}} \frac{1}{Z_\phi} \sum_x \mathcal{O}(x) P_0(x)\, e^{-\beta \Delta V_\phi(x)}\, e^{-\vec{\lambda}\cdot\vec{g}(x)} \\ &= \frac{\langle \mathcal{O}(x)\, e^{-\vec{\lambda}\cdot\vec{g}(x)}\, e^{-\beta \Delta V_\phi(x)} \rangle_0}{\langle e^{-\vec{\lambda}\cdot\vec{g}(x)}\, e^{-\beta \Delta V_\phi(x)} \rangle_0} \end{aligned} \tag{2.17}$$

being $\langle ... \rangle_0$ the averages over the initial ensemble P_0. In particular, the partition function $Z_{\lambda\phi}$ of Eq. 2.4 is

$$Z_{\lambda\phi} = \langle e^{-\vec{\lambda}\cdot\vec{g}(x)} \rangle_\phi = \frac{\langle e^{-\vec{\lambda}\cdot\vec{g}(x) - \beta \Delta V_\phi(x)} \rangle_0}{\langle e^{-\beta \Delta V_\phi(x)} \rangle_0}. \tag{2.18}$$

Clearly, the sampling of P_0 is limited, so moving away from P_0 the fidelity of the reweighted average decreases. The quality of the reweighting can be quantified through the effective number of frames N_{eff} of the reweighted ensemble, estimated by the so-called relative Kish sample size K_{rel} as

$$N_{eff}/N = K_{rel}[\{w_t\}|\{w_{0t}\}] = \Big(\sum_t \frac{w_t^2}{w_{0t}} \Big)^{-1} \tag{2.19}$$

or analogously by the exponential of the Kullback-Leibler divergence $e^{D_{KL}[P||P_0]}$. Both these two indicators measure the average relative ratio w/w_0, as arithmetic or geometric mean respectively. For a description of them, see Appendix A.

2.1.5 Cross validation

The hyperparameters α, β introduced in the loss functions (Eqs. 1.12, 1.17, 1.20, and 2.1) require careful tuning to identify the optimal intermediate regime between underfitting and overfitting. Underfitting typically arises when hyperparameters are too large, favoring prior assumptions over experimental data. Conversely, small hyperparameter values can lead to overfitting, where the model adapts too closely to noise or specific fluctuations in the experimental measurements.

A practical strategy to select suitable hyperparameter values is cross validation, where part of the data is iteratively left out during training and used solely for validation. In the simplest implementation, one performs a grid search over the hyperparameter space. For each combination of hyperparameter values, the loss function is minimized on the training subset, and the agreement with the left-out validation data is evaluated – typically using the reduced χ^2 (χ^2 divided by the number of experimental observables). Normalizing χ^2 by the number of observables enables direct comparison between data subsets containing different number of observables. This process is repeated across multiple random splits of the dataset to ensure statistical robustness. A typical partition includes 70% of both the observables and the trajectory frames for training and the remaining 30% for validation. In cases where molecular dynamics simulations are performed using replica exchange methods (such as replica exchange with solute tempering, as described in this chapter), frame selection can be carried out by choosing approximately 70% of the demultiplexed ("demuxed") trajectories and extracting from these all frames corresponding to the reference (room) temperature. With "demuxed" trajectories we mean continuous paths reconstructed by tracing a single replicaâĂŹs progression through temperature exchanges, ensuring each represents an uninterrupted physical evolution despite the swapping of temperatures among replicas.

As a simple illustrative case, consider a 1-dimensional scan over the hyperparameter α for ensemble refinement, reported in Fig. 2.2. One would:

- split the data into training and validation sets;
- optimize the ensemble for a range of α values using the training set;
- compute χ^2 on the validation set;
- repeat the process across multiple random splits;
- identify the value of α that minimizes the average validation χ^2.

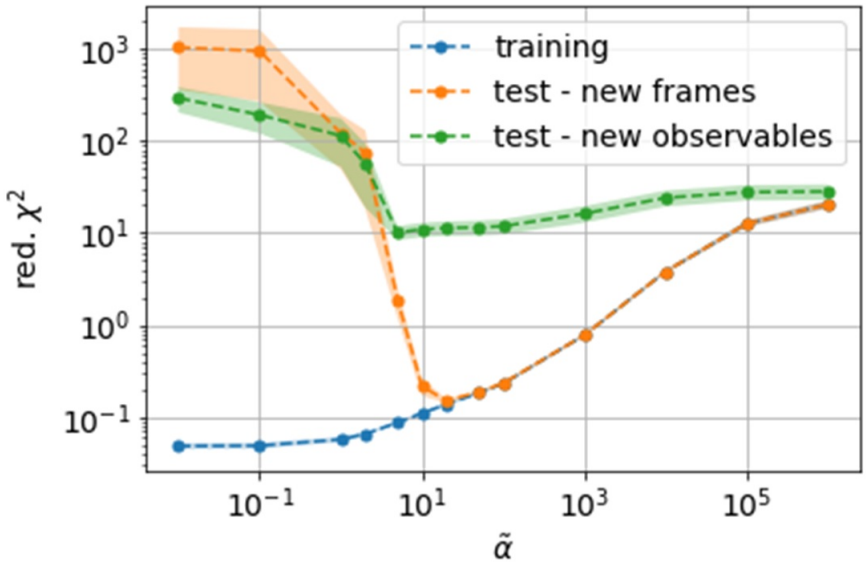


Figure 2.2: Cross-validation for ensemble refinement (scan over the hyperparameter α). The optimal regime is highlighted by the minimum of the χ^2 on left-out observables, which closely aligns with the one on left-out frames.

Starting from a high α value and decreasing it, we expect a transition from underfitting (where prior assumptions dominate) to overfitting (where experimental data are overemphasized). This transition is typically marked by a minimum in the validation χ^2. This procedure provides a practical means to assess the reliability of the fitted model. Once the optimal values of the hyperparameter are identified, a final minimization of the loss function can be performed using the entire dataset to obtain the best-fit ensemble.

To estimate the model's ability to generalize beyond the training data, one must evaluate its performance on data not seen during the training-and-validation process (the test set). One approach, followed in this Chapter, is to evaluate the χ^2 on molecular systems that were not present in training. Alternatively, as will be described in the following Chapter, one can assess generalization by leaving out some observables and/or frames from the training-and-validation process. This second strategy leads to a nested cross-validation setup, where a portion of the data is held out for final evaluation, while the remaining data is used in an inner cross-validation loop to select optimal hyperparameters. After the best hyperparameters are identified, the model is retrained on the full outer training set and evaluated on the held-out test set. This process may be repeated across multiple outer splits. Nested cross-validation provides an estimate of the model's generalization error, which

is particularly important when comparing different models or regularization schemes.

In higher dimensions – when tuning multiple hyperparameters simultaneously – this cross-validation procedure becomes significantly more challenging. This difficulty is known as the "curse of dimensionality": the number of grid points needed to adequately sample the hyperparameter space grows exponentially with the number of hyperparameters. Moreover, for each point in the grid, several minimizations of the loss function must be performed (one per random split of the dataset), each of which is a high-dimensional and computationally expensive task in itself. As a result, naive grid search becomes quickly impractical beyond a few dimensions. While in earlier cases we only considered two two hyperparameters α, β, the next Chapter introduces an additional coupling with forward-model refinement via the hyperparameter γ. Furthermore, one may wish to assign independent hyperparameters to each molecular system, further increasing the dimensionality of the problem. To overcome these limitations, we reformulate the problem as the direct minimization of the validation error $\chi^2_{valid}(\alpha, \beta, \gamma)$ averaged over multiple random splits, as a function of the hyperparameters. This enables the use of gradient-based optimization strategies, which scale more favorably with dimensionality. The implementation of this approach is presented in detail in Chapter 3.

2.2 Toy model

To check the validity of the combined refinement of force field and structural ensembles, we set up a simple toy model which consists of a two-dimensional probability distribution with four peaks (see Fig. 2.3).[1] The initial hypothesis sets most probability in the top-left peak ($x < 0$ and $y > 0$), while the ground-truth probability is distributed also on the top-right peak ($x > 0$ and $y > 0$). So, the average value of the x observable is underestimated by the initial hypothesis, whereas the average value of the y observable is approximately correct. We then correct the initial distribution purely based on the value of the observed averages of x and y.

Ensemble refinement shifts the probability from the two peaks at $x < 0$ to those at $x > 0$, perfectly matching the observed averages. According to the maximum entropy principle, this is the minimal correction to the prior ensemble that allows matching the observed averages. The resulting ensemble

[1] A Jupyter Notebook reproducing this toy model is available on GitHub at https://github.com/bussilab/force-field-ensemble-refinement/blob/main/Analysis_scripts/Toy_model.ipynb.

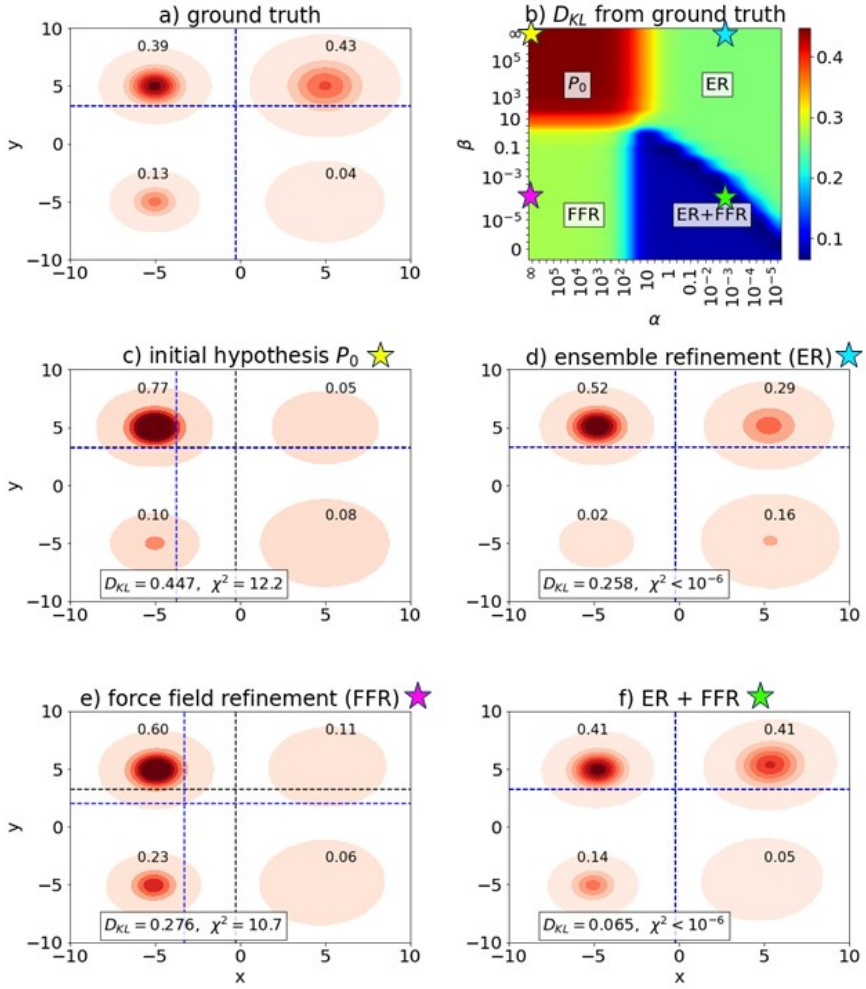


Figure 2.3: Results one a toy model, with populations reported for each quadrant. (a) Ground truth distribution. (c) Prior distribution. (d) Ensemble refinement leads to an ensemble closer, but still not identical, to the ground truth. (e) A careful choice of the force field correction term results in a refinement alternative to the one proposed by ER, nonetheless as good as that (comparable Kullback-Leibler divergences D_{KL} from the ground truth), with the benefit to be transferable. If this correction is not sufficient, (f) further flexibility can be included through the combined approach. In panels (a, c-f), black dashed lines report the averages computed using the ground truth, whereas blue dashed lines report the averages computed using the refined ensemble. (b) Hyperparameters scan: the hyperparameters values prove to be crucial for a proper balancing of the ER and FFR contributions. The values of the hyperparameters used to generate the ensembles in panels (c-f) are indicated with a star of a matching color.

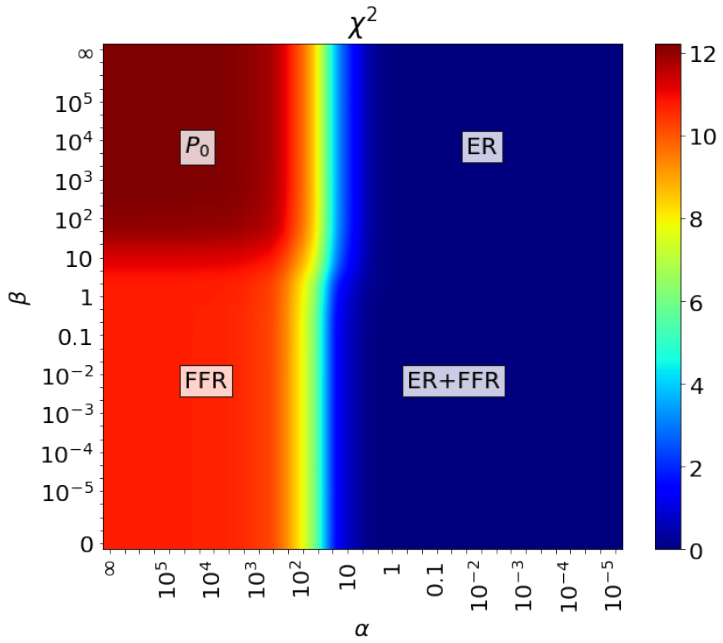


Figure 2.4: Results on a toy model: χ^2 for the optimal ensemble as a function of α, β hyperparameters.

is closer to the ground-truth one [129], but still not necessarily identical. We then assume that a physical knowledge of the system suggests the top-right and bottom-left peaks to be coupled, leading to a specific functional form for the force-field correction $\Delta V(x, y; k) = -kxy$. By performing a force-field refinement with this additional information, the observed averages are not exactly matched, but the obtained ensemble is also getting closer to the ground truth. Including further flexibility through the combined approach here introduced allows to optimally combine the information used in the force-field refinement approach with the maximum entropy principle, resulting in better agreement with the ground-truth ensemble than the one obtained applying any of the two methods separately.

In Fig. 2.3b we report the distance from ground truth for the ensembles obtained using different possible values of α and β. The method interpolates between no correction to ensemble refinement, force-field refinement, and any combination of the two, as indicated in the figure.

A similar figure showing the discrepancy between the predicted and experimental observables (quantified by the χ^2) is reported in Fig. 2.4. Starting from the original ensemble P_0 (ground truth) at $\alpha = \beta = \infty$, which corresponds to $\chi^2 \simeq 12.2$, and decreasing the value of one or both the two hyperparameters, the χ^2 decreases. However, this occurs in a remarkably different way for the α

(ensemble refinement) or β (force field refinement) directions: in the first case, decreasing the hyperparameter α at given β, the χ^2 decreases to values close to $\chi^2 = 0$, while in the other case the "stiffness" of the force-field functional form stops the decrement at the limit value $\chi^2 \simeq 10.7$. Starting from low values of β and allowing for more flexibility of the ensemble (α direction), the χ^2 decreases down to $\chi^2 = 0$, as for ER only. However, although the χ^2 is almost the same, the Kullback-Leibler divergence from the ground truth is significantly different in the two regions ER and ER+FFR: including some knowledge about the force-field results in better refinement.

A suitable choice of the two hyperparameters α, β is required to avoid overfitting. This is particularly relevant considering that experimental data are only known with a given uncertainty. As this effect is not present in this toy model, hyperparameters tuning can be better illustrated using a more realistic example.

2.3 Application to RNA oligomers

In this section we describe an application of our method to a realistic case study involving short RNA molecules (oligomers). The corresponding MD simulations were performed in a previous work by our group [1]. The scripts used to perform the refinements discussed in this work can be found at https://github.com/bussilab/force-field-ensemble-refinement. The analyzed time series are available at https://doi.org/10.5281/zenodo.10185005.

2.3.1 RNA oligomers under study

The RNA systems investigated in this work include five tetramers AAAA, CAAU, CCCC, GACC, UUUU, and two hexamers UCAAUC, UCUCGU. Corresponding MD simulations follow protocols described in previous studies [1, 21] and summarized as follows:

- starting structures: built with x3DNA or Nucleic Acid Builder (AmberTools 14) as described in Ref. [1], corresponding to a single strand of canonical A-form RNA;
- force field: AMBER bsc0 χOL3 [18, 29, 30, 32] with van der Waals modifications for phosphate oxygens [17], but no adjustments to dihedral parameters;
- solvent model: explicit OPC water molecules [34];

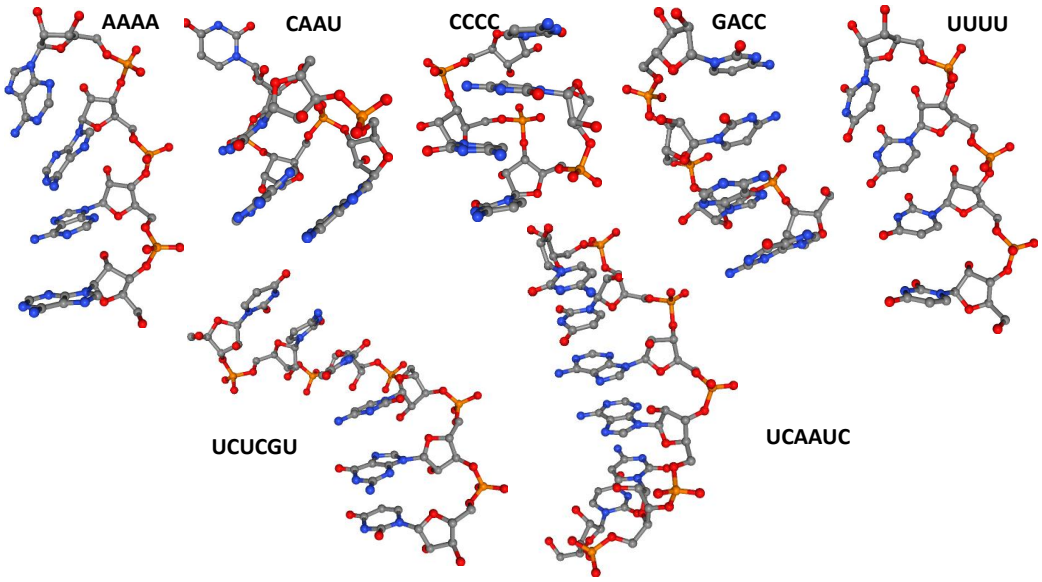


Figure 2.5: RNA oligomers under study: five tetramers and two hexamers.

- ions and salt concentration: KCl with Joung-Cheatham ion parameters [105], matched to experimental conditions;
- enhanced-sampling strategy: Replica Exchange Solute Tempering (REST) using 24 replicas in the temperature range 275-400 K; each replica was simulated for 1 μs; for analysis, only the frames closest to room temperature (300 K) were used.

The CAAU and UCUCGU oligomers were held out as test systems to assess the transferability of force-field corrections trained on the remaining oligomers. For additional simulation details, including temperature ladder and box setup, we refer to Ref. [1].

The MD trajectories of each oligomer show remarkable heterogeneity and flexibility[2], characterized by the presence of multiple structural clusters. Despite the short length of these RNA molecules, hydrogen bonds between nucleobases are occasionally observed, though only in a few number of oligomers and simulation frames. This structural diversity underscores both the complexity and relevance of studying even small biomolecular systems.

[2]These analyses have been performed in a Jupyter Notebook `MD_analysis.ipynb`, which is available on the GitHub repository https://github.com/IvanGilardoni/MD_analysis_oligomers.

Comparison with experimental data

Simulation results were compared against experimental NMR data, including ${}^3J$ scalar couplings, observed NOEs, and unobserved NOEs (uNOEs). For an overview of the physical interpretation of these observables, see section 1.2. In particular, the ${}^3J$ scalar couplings report on three-bond dihedral angles in the backbone and sugar ring (see Fig. 2.6):

- backbone dihedral angles: β (P − O5′ − C5′ − H5′, P − O5′ − C5′ − H5″), ε (H3′ − C3′ − O3′ − P), γ (H5′ − C5′ − C4′ − H4′, H5″ − C5′ − C4′ − H4′);
- sugar puckering angles: ν_1, ν_2, ν_3 (H1′ − C1′ − C2′ − H2′, H2′ − C2′ − C3′ − H3′, H3′ − C3′ − C4′ − H4′).

The experimental values, together with the corresponding estimated uncertainties, are provided by Refs. [7, 130–133]. The relation between ${}^3J$ scalar couplings and dihedral angles is described by the Karplus equations with empirical coefficients

$$ {}^3J(\phi) = A\cos^2\phi - B\cos\phi + C $$

where ϕ is the involved dihedral angle. These equations are used as a forward model to back-calculate the scalar couplings from dihedral angles provided by MD simulations. Their coefficients were estimated in Refs. [5–7].

We also consider NOEs, whose intensity scales as $1/r^6$ where r is the distance between proton pairs. The absence of detected NOE signals – referred to as unobserved NOEs (uNOEs) – indicates that corresponding protons are not in close proximity. Experimental values and uncertainties are provided by Refs. [7, 130–133] and are reported as distances r (in Å). Since the observed quantity is the $1/r^6$ signals and not the distance, we prefer to compare these signals with corresponding averages over MD simulations in the χ^2. To evaluate the associated experimental error, given the discrepancy δr, we compute:

$$ \delta\Big(\frac{1}{r^6}\Big) = \frac{1}{2}\Big(\Big|\frac{1}{(r-\delta r)^6} - \frac{1}{r^6}\Big| + \Big|\frac{1}{(r+\delta r)^6} - \frac{1}{r^6}\Big|\Big). \tag{2.20} $$

For unobserved NOEs, only an upper signal threshold is defined, and the upper uncertainty results:

$$ \delta\Big(\frac{1}{r^6}\Big) = \frac{1}{(r-\delta r)^6} - \frac{1}{r^6}. \tag{2.21} $$

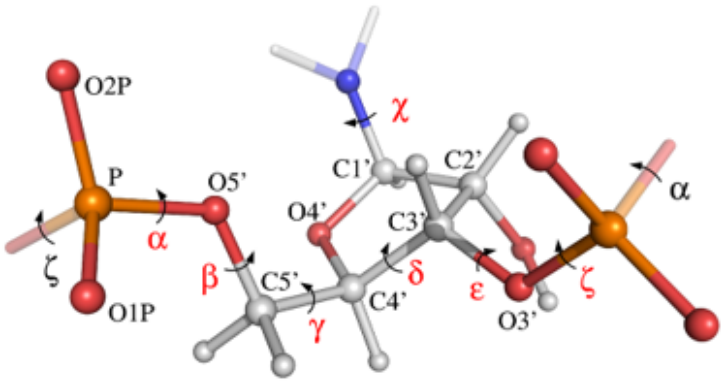


Figure 2.6: RNA backbone structure, with standard atom names and dihedral angles; the nucleobase is truncated so that only two carbon atoms from the cycle are shown. The force-field corrections selected in this work act on the α ($O3'_{i-1} - P - O5' - C5'$), ζ ($C3' - O3' - P_{i+1} - O5'_{i+1}$) and χ ($O4' - C1' - N1' - C2'$ for pyrimidines and $O4' - C1' - N9' - C4'$ for purines) dihedral angles. The analyzed ^3J scalar couplings involve β, ϵ, γ, ν dihedral angles. Image from Frellsen J. et al., PLoS Comput. Biol. (2009) [134].

Hence, in computing χ^2 contributions from unobserved NOEs $g(x)$, the expression is modified as:

$$\chi^2[P] = \left[\max\left(0, \frac{\langle g \rangle_P - g_{exp}}{\sigma}\right)\right]^2, \tag{2.22}$$

so that no error is reported if the average value lies below the experimental threshold g_{exp}.

2.3.2 Application of the method

In this work, we consider correction terms which are linear in the coefficients $\vec{\phi}$, being $\Delta V_\phi(x) = \sum_j \phi_j f_j(x)$. In particular, we specialized to three different cases: sinusoidal corrections on glycosidic χ, on α, and on α and ζ (with equal coefficient for sine and equal one for cosine) dihedral angles (see Fig. 2.6):

$$\Delta V_\phi(x) = \sum_{i=1}^{N_{dihedrals}} (\phi_1 \sin \chi_i(x) + \phi_2 \cos \chi_i(x)); \tag{2.23}$$

$$\Delta V_\phi(x) = \sum_{i=1}^{N_{dihedrals}} (\phi_1 \sin \alpha_i(x) + \phi_2 \cos \alpha_i(x)); \tag{2.24}$$

$$\Delta V_\phi(x) = \sum_{i=1}^{N_{dihedrals}} \{\phi_1[\sin\alpha_i(x) + \sin\zeta_i(x)] + \phi_2[\cos\alpha_i(x) + \phi_2\cos\zeta_i(x)]\}. \tag{2.25}$$

We then apply the method on the set of RNA oligomers described above, using the same experimental dataset previously employed in Ref. [1].

Search for optimal hyperparameters

We perform the minimization of the loss function obtained combining all the oligomers in the training set (AAAA, CCCC, GACC, UUUU, UCAAUC – Fig. 2.5) with a scan in the space of the hyperparameters α and β. The limiting cases of ensemble refinement (scan on the hyperparameter α at $\beta = \infty$) and force field fitting (scan on the hyperparameter β at $\alpha = \infty$) are included, as well as the reference ensemble, corresponding to $\alpha = \beta = \infty$ (i.e., no refinement). For each considered value of the hyperparameters α, β, we use 20 different random seeds to define training sets, each composed of 70% of the frames and 70% of the observables (with the same selections applied across all the sampled hyperparameters), implementing a bootstrap strategy [135]. Frame selection is based on the "demuxed" trajectories, as described in subsection 2.1.5. An exception is made for UCAAUC system, where a corrupted file prevented the reconstruction of continuous trajectories and so we resorted to randomly selecting 70% of frames directly from the full trajectory. The remaining 30% of the observables are used to evaluate the reduced χ^2 on the full trajectory, including frames that were part of the training set (validation step).

The achievement of the numerical minimizations is demonstrated by the increase in the minimum value of the loss function at fixed random seeds as either α or β increases, consistently with the fact that the optimal loss $\mathcal{L}^*(\alpha, \beta)$ is a monotonically increasing function of the hyperparameters (see section B.2). Given the way trajectories are bootstrapped, the reduced χ^2 on training observables and the one on validating observables coincide within their statistical error at $\alpha = \beta = \infty$, since they both correspond to the χ^2 for the original ensemble.

The two hyperparameters (α and β) control the flexibility of the fitting. By decreasing one or both, the reliability given to the original assumptions on the force field is reduced in favour of the confidence on experimental measures. In other words, low values of the hyperparameters correspond to high flexibility in the correction to the ensemble, which means strong ability to fit the (training) experimental values. This is particularly evident for the hyperparameter α, corresponding to the ensemble refinement direction. For β

(force-field refinement direction), the flexibility is instead intrinsically limited by the constrained functional form of the force-field correction. Whereas a limited flexibility can provide a (physically meaningful) improvement in the description of the molecules, an uncontrolled flexibility may lead to overfit the data, disregarding their intrinsic experimental error. Cross validation deals with assessing the appropriate importance of this flexibility, which in the method here proposed plays on two different directions: those of non-transferable ensemble refinement and force-field refinement. This task is performed by evaluating the error (the reduced χ^2) on left-out observables, namely those which are not used in this training step to determine the optimal ensemble.

Correction	**Method**	**Optimal $\boldsymbol{\alpha, \beta}$**	$\boldsymbol{\chi^2_{red}}$
– (no correction)	No reweighting	$\alpha = \infty$, $\beta = \infty$	28.32
	ER ($\beta = \infty$)	$\alpha = 5$	10.10
χ angles	FFR ($\alpha = \infty$)	$\beta = \infty$	28.32
	ER+FFR	$\alpha = 5$, $\beta = \infty$	10.10
α angles	FFR ($\alpha = \infty$)	$\beta = 100$	9.33
	ER+FFR	$\alpha = 20$, $\beta = 0.01$	5.70
α, ζ angles	FFR ($\alpha = \infty$)	$\beta = 5$	9.81
	ER+FFR	$\alpha = 10^5$, $\beta = 2$	8.02

Table 2.1: Results of cross-validation on training molecules (see also Fig. 2.7). We compare ensemble refinement (ER), force-field refinement (FFR), and their combination (ER+FFR) under different correction schemes. For each case, we report the optimal hyperparameters α and β, along with the minimum reduced chi-squared value χ^2_{red} computed on validating observables. The first row shows the baseline performance without any reweighting or corrections.

Whereas the χ^2_{red} computed on training data is decreasing when decreasing the values of the two hyperparameters (see again section B.2),the χ^2_{red} computed on validating observables (Figs. 2.7- 2.8) does so only over a certain range of the hyperparameters, signaling when under/over-fitting occurs.

Firstly, we consider the case of traditional ensemble refinement, which in the combined method corresponds to $\beta = \infty$ (first row in the plots of Fig. 2.7, independent on the selected force field correction, and Fig. 2.8a). Starting from the prior ensemble P_0 at $\alpha = \infty$, the error χ^2_{red} decreases with α, up to a certain point, where it starts to increase. Such point of minimum (approximately at $\alpha \simeq 5$, with $\chi^2_{red} \simeq 10$) marks the transition from the underfitting to overfitting scenarios. We then move forward to transferable corrections.

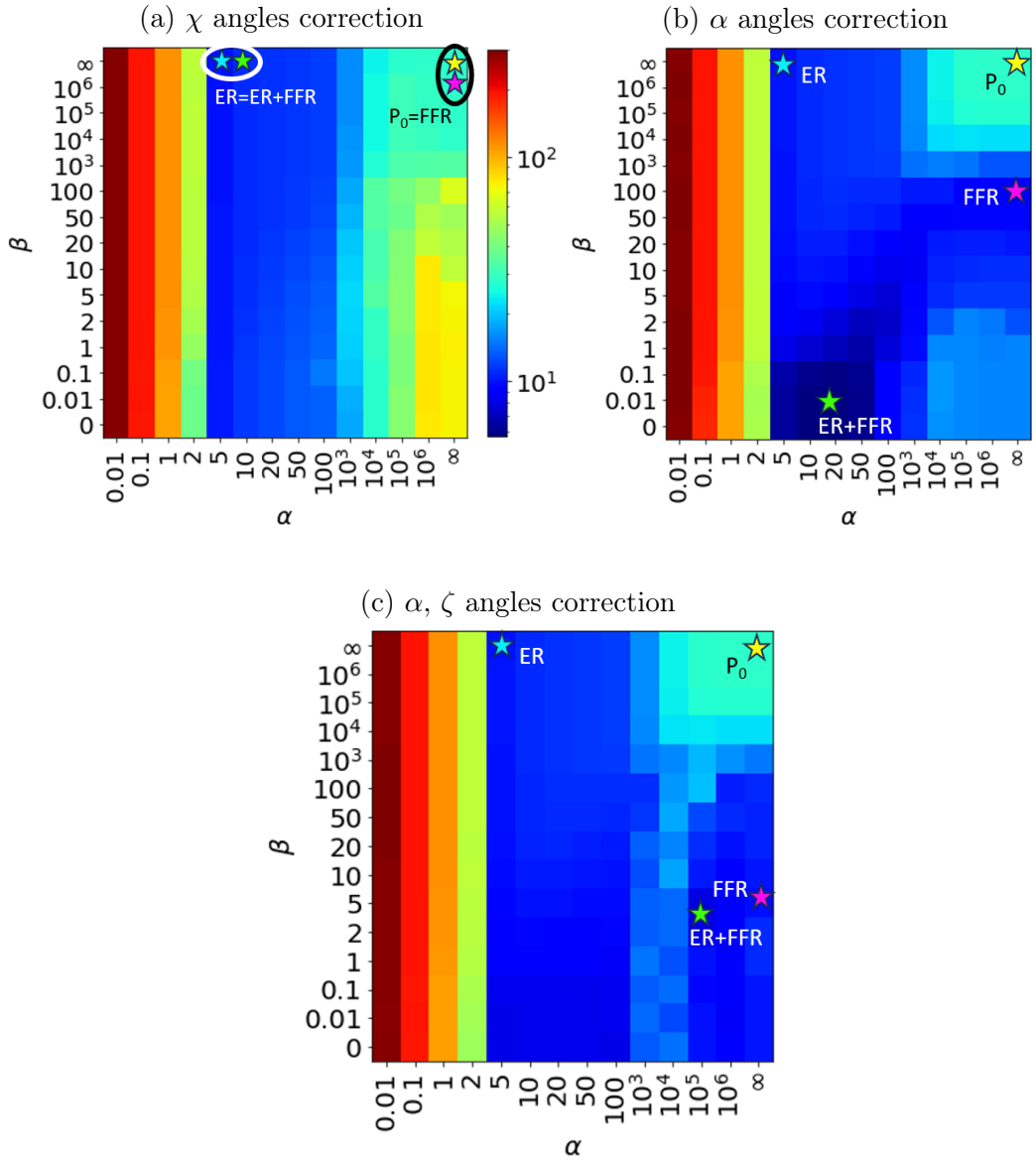


Figure 2.7: Cross-validated reduced χ^2 values on validating observables for three different force-field refinement schemes. (a) Correction of the χ angles alone is ineffective, indicating the importance of non-transferable ensemble refinement. (b) Correction of the α angle improves agreement with experiments, and further flexibility enhances performance. (c) Correction of both α and ζ angles alone is sufficient, making additional flexibility unnecessary. Color bars are the same for all the plots.

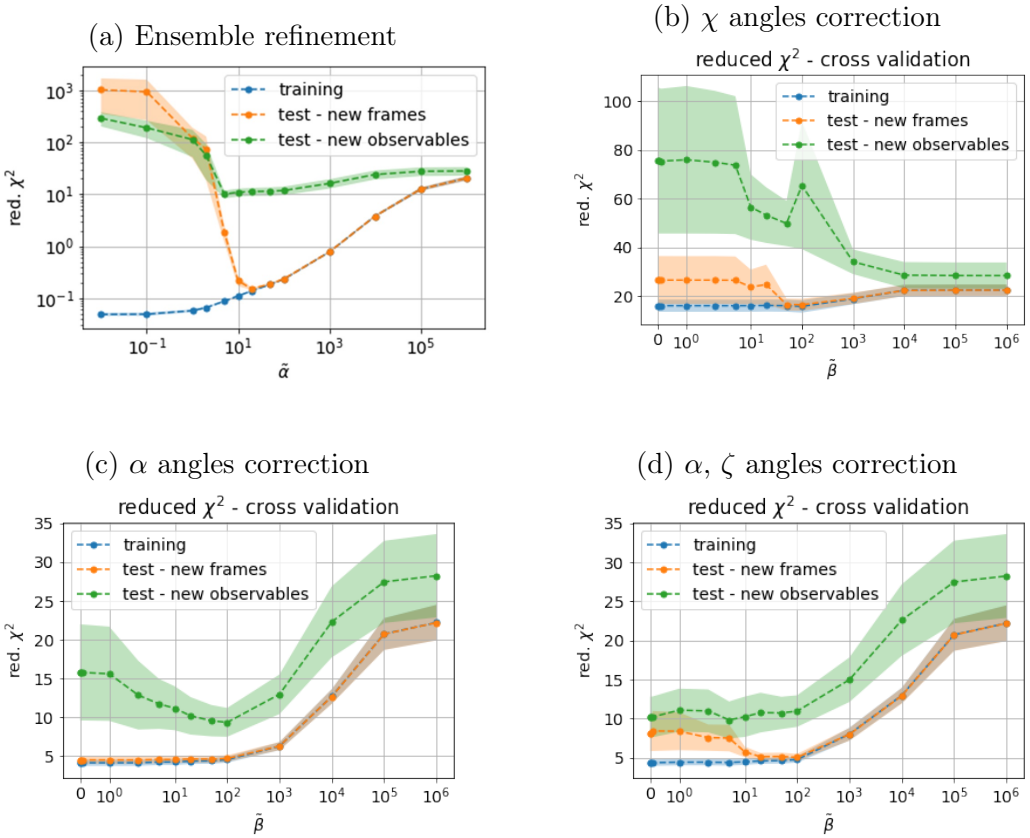


Figure 2.8: Reduced χ^2 in cross validation for ensemble refinement and force field refinement separately applied. Different colors refer to: training data (blue line), training observables and left-out frames (orange line), test observables and full trajectories (green line).

The force-field corrections exhibit the three different behaviors which are expected when applying the combined ER+FFR method. The first attempted correction (acting on χ angles, Fig. 2.7a and Fig. 2.8a) applied alone is fruitless, since the χ^2_{red} at $\alpha = \infty$, i.e. in the FFR regime, is larger than in the original ensemble P_0 for any choice of $\beta < \infty$. The contribution of (non-transferable) ensemble refinement is thus essential: when including it, the minimum χ^2_{red} results at $\alpha = 5, \beta = \infty$. Hence, this case corresponds to the extreme in which adding the contribution of the force field refinement does not improve the description with respect to ensemble refinement alone. Even in this particularly difficult case, the ensemble refinement step is able to report reasonable cross-validation observables (see Table 2.1).

The second attempted correction (on α dihedral angles, not to be confused with the α hyperparameter) is beneficial, as shown at $\alpha = \infty$ (last column in Fig. 2.7b, and Fig. 2.8b). However adding more flexibility with the combination of ER and FFR proves to be beneficial with respect to the separate application of either the two methods since it results in a lower cross-validation error.

Finally, the correction on α, ζ angles alone is sufficient and the inclusion of further flexibility leads to a minor improvement in the agreement with experiment (Fig. 2.7c and Fig. 2.8c). This is close to the other extreme, in which FFR alone shows to be optimal.

To check the statistical efficiency of the reweighting, we report in Fig. 2.9 the Kish ratios K_{rel} for the three methods in the case of force-field correction on α dihedral angles (average over cross validation values). In all the cases the effective number of frames is a significant fraction (around 10%) of the total number of frames, with slightly lower values for UCAAUC hexamer.

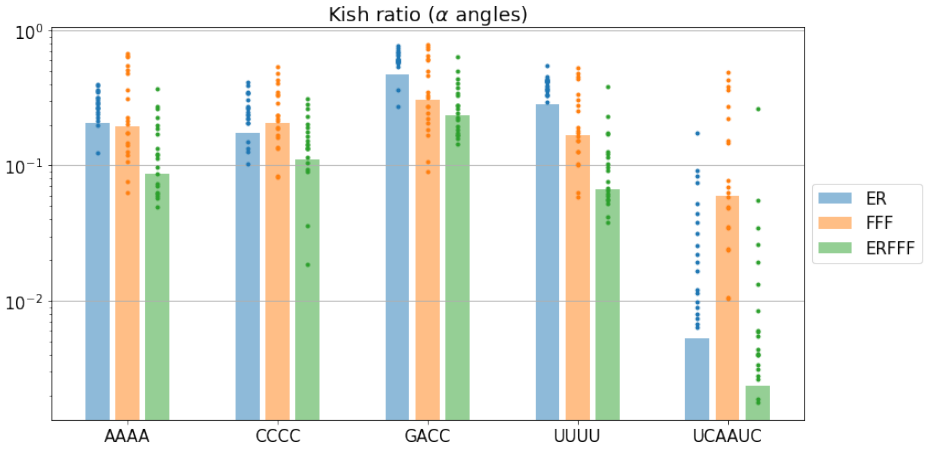


Figure 2.9: Kish ratio (ratio between effective and total number of frames); bars show values from minimization on the whole data set, points display values in cross-validation (force-field correction on α dihedral angles).

Optimal force-field corrections

Once the optimal values of the α, β hyperparameters have been determined through cross validation, we estimate the coefficients of the force-field corrections by minimizing the loss function with the optimized hyperparameters on the whole data set (without distinguishing between training and validation set). The associated uncertainty is quantified by the standard deviation of the coefficients resulting from cross validation, as it is conventionally done in bootstrap analysis [135]. The average values on the bootstrap samples are compatible with the values from whole minimization within the obtained uncertainties. The results are reported in Table 2.2. The correction on the χ dihedral angles is null, since the optimal hyperparameters correspond to ensemble refinement only. This implies that relevant force field corrections are not acting on the χ bond, rather on the backbone structure. Indeed, for the other two cases (correction on α or α, ζ) the force field correction coefficients are significantly different from zero.

We then study how the force-field corrections on the α angles modify the distribution of α, ζ dihedral angles, comparing the three different refinement methods described above. In these constructs, the phosphate group of the first nucleotide is absent (in agreement with experiments) so the $\alpha_i\, \zeta_{i-1}$ dihedral angles encompass the i-th phosphate group, with $i = 2, \ldots, N$. In Fig. 2.10a) we show the overall histogram of the α_i, ζ_{i-1} distributions, averaged over molecules and positions, which highlights two dominant peaks in the original ensembles, corresponding to gauche+ (g+) and gauche− (g−). The population

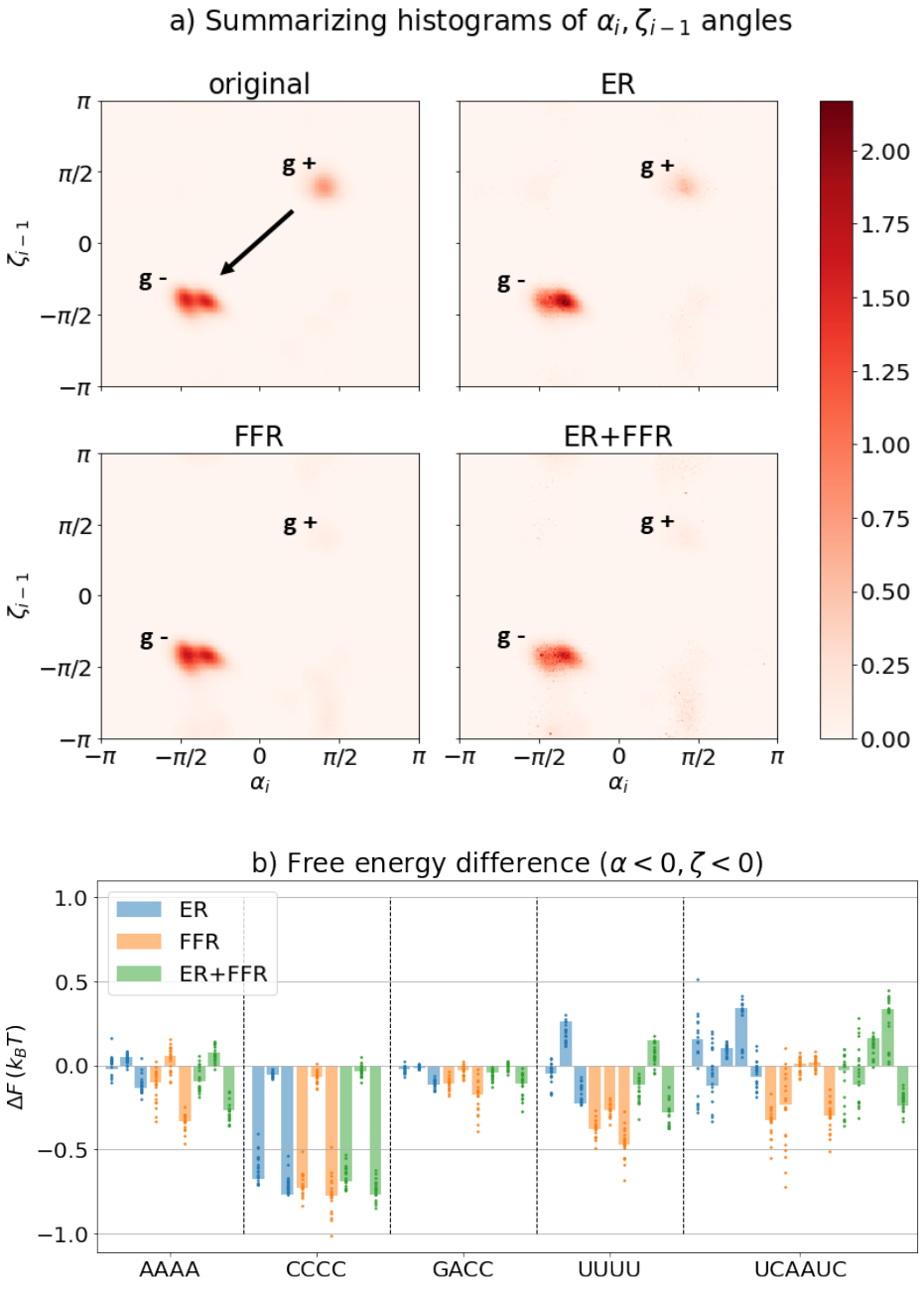


Figure 2.10: (a) Summarizing histograms of α, ζ angles before/after corrections on α dihedral angles, averaged over all the oligomers; (b) free energy difference for the $(\alpha, \zeta) \in (-\pi, 0) \times (-\pi, 0)$ region, for each molecule and phosphate (bars corresponding to minimization on the whole set of observables, dots corresponding to minimizations in cross validation).

Correction	**Method**	$\boldsymbol{\alpha}$	$\boldsymbol{\beta}$	$\boldsymbol{\phi_1}$	$\boldsymbol{\phi_2}$
χ angles ($\sin\chi$, $\cos\chi$)	FFR	∞	∞	0	0
	ER+FFR	5	∞	0	0
α angles ($\sin\alpha$, $\cos\alpha$)	FFR	∞	100	0.91 ± 0.16	1.67 ± 0.85
	ER+FFR	20	0.01	0.51 ± 0.08	1.65 ± 0.36
α, ζ angles (sin, cos)	FFR	∞	5	3.0 ± 1.5	4.0 ± 2.0
	ER+FFR	10^5	2	3.0 ± 1.6	4.0 ± 1.8

Table 2.2: Force-field correction coefficients (unit: $k_BT = 2.49\,\mathrm{kJ/mol}$), reported with uncertainties. For the χ angles, both FFR and ER+FFR yield zero correction, confirming that only non-transferable ensemble refinement contributes (see Fig. 2.7). Corrections follow the forms $V(\chi) = \phi_1 \sin\chi + \phi_2 \cos\chi$, $V(\alpha) = \phi_1 \sin\alpha + \phi_2 \cos\alpha$, and $V(\alpha, \zeta) = \phi_1(\sin\alpha + \sin\zeta) + \phi_2(\cos\alpha + \cos\zeta)$, summed over all instances of the respective dihedral angles in the molecule.

of these two peaks is modified when introducing ensemble and/or force field corrections, with a general increase in the $\alpha_i(g-), \zeta_{i-1}(g-)$ region, to the disadvantage of the $\alpha_i(g+), \zeta_{i-1}(g+)$ area. This is in agreement with previous studies [20, 136, 137]. To better visualize these variations, Fig. 2.10b reports the free energy differences ΔF associated to the $\alpha_i(g-), \zeta_{i-1}(g-)$ region, separately for each oligomer and phosphate group. We notice how, for AAAA, CCCC, and GACC tetramers, the refinements have a significant impact only on the α_i, ζ_{i-1} angles corresponding to the first and last phosphate, while the population for the middle phosphate is almost unchanged. This can be explained on the basis that the employed force-field is well-suited for long RNA molecules, for which the first and last phosphate constitute a small fraction of the whole molecule, so that *ad hoc* correction at the termini might be convenient [138]. In particular, both the FFR and ER+FFR corrections go in the same direction as ER, favouring $\alpha(g-), \zeta(g-)$ angles as expected above. Results for the UUUU tetramer instead show significant free-energy differences also for the intermediate phosphate. Here, ER and FFR suggest opposite corrections, with the former disfavouring $\alpha_3(g-)$, $\zeta_2(g-)$. Also for the UCAAUC hexamer, the α_i, ζ_{i-1} dihedral angles corresponding to $i = 4, 5$ tend to be modified by both the ER and ER+FFR methods, and left unchanged by FFR correction. Given the better performance of the ER+FFR approach in cross validation tests (see χ^2_{red} in Table 2.1), we argue that the ensembles obtained with the ER+FFR approach are more reliable than those obtained using the ER or the FFR approach alone.

Testing the force-field corrections on left-out molecules

We compare the performance of the two force-field corrections that we obtained with the FFR and ER+FFR procedure when transferred to new molecules, not considered in training, namely the CAAU and UCUCGU oligomers. To this aim, we employ the optimal coefficients of the force-field corrections reported in Table 2.2 to reweight the CAAU and UCUCGU ensembles. We do so without performing a new ensemble refinement procedure. The corresponding χ^2_{red} is then evaluated on the whole set of observables measured for the new two systems (see Table 2.3).

These two molecules are quite different in structure and exhibit distinct responses to the correction derived from the training set. Specifically, the original ensemble of CAAU has a very large χ^2 which is dramatically decreased by the force field corrections. The larger the penalty on the gauche+ (g+) rotamers, the better the agreement with experiment. As a consequence, the correction applied on the α, ζ angles results in the best agreement with experiment. Conversely, the UCUCGU hexamer has a moderate χ^2. Interestingly, the force field corrections obtained on the training set are not capable on improving the agreement of the corresponding ensemble with experiment. In this case, all corrections lead to some degree of overfitting. The mixed FFR+ER approach on the α angle, which leads to more conservative force field corrections, results in smaller overfitting and in a χ^2 comparable to the one obtained with the original force field.

Correction	**Method**	**CAAU**	**UCUCGU**	**Both**
No correction	No reweighting	243.8	14.1	211.3
α angles	FFR	24.8	16.4	23.6
	ER+FFR	57.5	14.3	51.4
α, ζ angles	FFR	11.7	23.8	13.4
	ER+FFR	11.6	23.7	13.3

Table 2.3: Reduced χ^2 values on the validating molecules CAAU and UCUCGU (reported both individually and jointly), based on force-field corrections fitted on the training set (reported above). Corrections involving α or α, ζ angles significantly reduce the error compared to the uncorrected (no reweighting).

2.4 Discussion

In summary, this study reports a strategy to boost the efficiency of the ensemble-refinement methods by preceding them with a knowledge-based force-field refinement step. This combined method allows also to obtain force-field corrections which can then be transferred to different systems, and can outperform normal ensemble refinement in cross-validation tests. Differently from force-field refinement alone, these corrections are derived taking explicitly into account the fact that transferable corrections might not be able to match experiments simultaneously in multiple systems. This specificity on the system is guaranteed by the ensemble refinement step. Whereas the force-field refinement and ensemble refinement methods have been separately applied in several works, we are not aware of any attempt made to combine their strengths in a single approach.

We then applied the proposed method to a realistic case study of RNA oligomers. Despite the apparent simplicity of these small RNA molecules, current force-fields are still limited in correctly generating structural ensembles, which therefore can be used to improve molecular potentials. We use a robust cross-validation protocol to select the suitable values of the two hyperparameters α, β and then analyze the predictions about both training and left-out oligomers.

In this work, we designed force-field corrections to be partly independent of the measured observables, by correcting dihedral angles that were not directly measured. However, correlations indirectly appear through other interactions. Correlated observables used in ensemble refinement act cooperatively and are known not to be a problem, because corrections on the different observables lead to equivalent ensembles [19]. Conversely, correlated corrections in force-field refinement or in the combined method introduced here might lead to multiple inequivalent solutions. In the extreme case of identical force-field corrections and observables, we can expect that the cross-validation procedure used here will make sure that a fraction of the correction that can be safely transferred to other copies of the same residue is included in the force-field correction whereas the remaining part is included in the ensemble refinement step.

The introduced method interpolates between ER and FFR, reducing to either of the two whenever the other approach does not provide significant improvement of the χ^2 computed on a validating set of observables. This happens with the force-field correction on glycosidic bond angles, which does not exhibit any improvement in cross validation; therefore, the combined approach selects ER alone, resulting in better agreement with experiments than FFR alone. On the contrary, the correction on α and ζ dihedral angles,

with identical coefficients, has enough flexibility that the inclusion of system-specific refinement turns out to be irrelevant, and FFR alone is the optimal solution. However, as shown by comparison of Fig. 2.7, the best agreement with experimental data is obtained through the flexible correction on α dihedral angles only, using the combined method.

To minimize the loss function we employ a reweighting of the ensembles approach. This might be a weakness of the proposed approach, common to all reweighting methods, due to its potentially low statistical efficiency [96, 139]. This concept can be quantified as the effective number of frames, computed through the Kish sample size or analogously the relative entropy. While for the examined oligomers the effective number of frames still remains a significant fraction of the whole amount (see Fig. 2.9), for more complex systems, it might not be the case. Performing new MD simulations during the minimization, for instance whenever the resulting ensembles move too far from the initial ones, would lead to greater statistical robustness [14].

On-the-fly restraining can also be performed [19, 99, 116, 140, 141]. However, we are not aware of approaches where simultaneous corrections to both force fields and ensembles are applied on-the-fly. This direction is being explored as part of a master's thesis in collaboration with Davide Di Vora. The challenge lies in the computation of the derivatives in Eq. 2.14, which cannot be implemented on-the-fly from the last sampled frame alone, differently from the derivatives in Eq. 2.12, where $\langle g_j \rangle_{\lambda\phi}$ can be substituted by the g_j value in the most recent frame, as in ensemble refinement performed on-the-fly.

A second, related issue arises from regions of the conformational space with limited or no sampling. Due to the way our cross validation procedure is performed, left-out portions of the initial trajectory are used to test for overfitting. However, if samples from a region are never observed in the initial trajectory, it is impossible to use reweighting to predict which will be the effect of the correction on samples from that region. If the region has a very large energy, e.g. because it is sterically forbidden, any change to the potential energy function is irrelevant. But if the region can be sampled in a new simulation for the same or for a different system, overfitting issues will arise [128]. This is a well-known issue in force-field refinement strategies, where it is common to perform new simulations using the refined force-field parameters to test for these artifacts. It might also be an issue in ensemble-refinement strategies based on maximum entropy, if consecutive simulations are performed including linear corrections to the energy function. This issue is instead not expected to be visible if no new simulations are performed and the resulting ensemble is reported *as is*.

Finally, discrepancies between simulated and experimental data may arise not only from inaccuracies in the structural ensembles but also from imper-

fections in the forward models used to compute experimental observables from MD simulations. In extreme cases, the ensembles might be in perfect agreement with the ground truth, still yielding a high χ^2_{red} due to erroneous forward models – for instance, incorrect empirical coefficients in Karplus equations. A recent study introduced a method for the simultaneous refinement of ensembles and forward models [1]. Building on this idea, we extend the approach to enable the joint optimization of force fields, ensembles, and forward models. The method is presented in detail in the following Chapter 3.

Chapter 3

`MDRefine`: a Python package for refining MD trajectories with experimental data

In this Chapter, we introduce `MDRefine`, a Python package designed to perform the simultaneous refinement of structural ensembles, force fields, and forward models by leveraging experimental data in conjunction with MD simulations. The content is largely adapted from our published work [142].

The motivation behind this integrated approach lies in the fact that inaccuracies in each of these components – ensemble, force-field parameters, and forward-model coefficients – can all contribute to discrepancies between simulations and experiments. When these sources of error are addressed independently, their residual inaccuracies may still be present, leading to suboptimal agreement with data. Therefore, a simultaneous refinement strategy could be crucial to disentangle and correct for the contributions of each component, ultimately achieving more accurate and physically consistent models.

The `MDRefine` software is modular, allowing the individual tools to be used independently or combined in flexible ways, supporting a seamless transition between different types of refinement. Using benchmark systems, we show that this joint approach leads to better agreement with experimental data than methods that optimize only one component at a time. `MDRefine` is available as open-source software under the LGPLv2+ license. Source code, documentation and usage examples are available at `https://pypi.org/project/MDRefine` and `https://github.com/bussilab/MDRefine`.

3.1 MDRefine package: materials and methods

The simultaneous refinement of force-field parameters, structural ensembles, and forward models across multiple molecular systems introduces a significant increase in complexity of these methods, which often require substantial programming effort to be performed. To date, general-purpose scripts have been released to perform ensemble refinement alone (see, e.g., BME [103] and BioEn [99, 102, 110, 111]). Other software packages (including DMFF [124] and chemtrain [125]) have been developed for top-down force-field refinement based on χ^2 minimization, often in conjunction with bottom-up or coarse-grained modeling approaches (see also subsection 1.4.4). However, these packages typically do not support ensemble or forward-model refinement in their standard implementations.

Here we present MDRefine, a Python library designed to facilitate the comparison of MD trajectories with experimental data and to implement simultaneous refinement of ensembles, force fields and forward models. The package contains routines to calculate relevant statistical quantities (such as relative entropies and χ^2) and to reweight trajectories using different criteria. Importantly, it supports the full joint refinement pipeline discussed above without requiring users to implement complex algorithms from scratch. MDRefine also includes an optimization algorithm that determines optimal hyperparameters by direct minimization of the cross-validation error, avoiding the need for exhaustive grid searches in the hyperparameter space.

MDRefine adopts a modular architecture, treating input datasets – including both experimental observables and MD-derived quantities – as structured Python objects. Users can define custom force-field corrections or forward models tailored to their systems of interest, and combine different levels of refinement depending on their goals and available data. This modularity makes it possible to either constrain the refinement scope a priori or let the data determine the most effective approach.

The software is implemented as a Python package. It requires Python ≥ 3.8 and builds on powerful, widely-adopted scientific Python libraries such as Numpy [143], Scipy [144], and JAX [145]. The source code is freely available at https://github.com/bussilab/MDRefine.git, and tagged versions can be installed from PyPI. The GitHub repository also provides documentation and illustrative Jupyter notebooks. In the following, we describe the overall architecture of MDRefine and illustrate its capabilities through application to benchmark cases.

3.1.1 Structure of `data` objects

A complete data set is represented as an object `data`, which, once defined, is devised to be immutable. Its exact structure is discussed in the Python documentation. The user can directly provide the `data` object or use the provided `load_data` function to read files saved using the standard Numpy format from a suitably organized hierarchy of directories. The `data` object has two main attributes: `data.properties`, which is a dictionary containing global properties of the data (such as the names of the systems `sys_names` and the original forward-model coefficients), and `data.mol`, which is a dictionary with one item for each molecular system.

For each system, the corresponding item `data.mol[sys_name]` (an instance of `data_class`) contains as attributes both the *experimental values* `gexp` (measured values $g_{i,exp}$ with estimated uncertainties $\sigma_{i,exp}$) and some quantities evaluated for each frame x of the MD trajectory. The latter includes: the `weights` $w(x)$, the *observables* `g` to be compared with g_{exp}, and the quantities `f` required to compute the *force-field corrections* through the method `ff_correction`, whose functional form is specified by the user. We include also the possibility to compute some (or all) of the observables `g` through a user-defined function stored in the method `forward_model`, which acts on the attribute `forward_qs` (namely, quantities required for the forward model) and the corresponding coefficients. For the sake of clarity, both `g` and `gexp` are dictionaries of Numpy arrays split per experiment type *exp_ type*, like for example `backbone1_gamma_3J`, `backbone2_beta_epsilon_3J`, `sugar_3J` (3J scalar couplings computed for several dihedral angles) and `NOEs` (signals detected through nuclear Overhauser effect), with the attribute `names` naming each experimental measure. Also `forward_qs` is a dictionary of different arrays, but its keys do not necessarily correspond to those of `g` and `gexp`. Both `weights`, `g[exp_type]`, `f` and `forward_qs[exp_type]` are Numpy arrays whose rows correspond to (consecutive) frames x of the MD trajectory.

Vast freedom is left to the *units of measure*, provided the experimental values are loaded consistently with the observables (both `g` and output quantities of the forward model). Regarding the force-field corrections, the set of `data.mol[sys_name]` items include a variable named `temperature` corresponding to the thermal energy k_BT at which the MD simulation was performed. It is $k_BT = 1$ by default, namely, energies are expressed in unit of temperature, and in this way force-field corrections must be defined. Alternatively, one can express `temperature` in the preferred unit, as long as the force-field corrections are consistently defined. If the same system has been simulated at multiple temperatures to match different experiments, it is possible to provide the copies as if they were independent systems, each with

its own temperature, as long as they are defined with different `sys_names`.

Experimental values can be passed either as a "central reference" $g_i = g_{i,exp} \pm \sigma_{i,exp}$ or as *upper/lower bound*. The latter is often the case for data such as *unobserved NOEs*, where the experiment only provides an upper threshold for the NOE signal below which the signal cannot be revealed $g_i < g_{i,exp}$ with tolerance $\sigma_{i,exp}$. This is specified by the attribute `ref` ("reference"), which contains the sign of the inequality to be satisfied (`'='` for central references, `'</>'` for upper/lower thresholds), the same sign for each `data.mol[name_sys].g[exp_type]` set of observables. To simplify the implementation, we assume that a tolerance $\sigma_{i,exp}$ is always provided and different from zero, although it can be set to a numerically very small value to reach the regime where the tolerance is negligible. If no value for $\sigma_{i,exp}$ is provided, this is by default assumed to be 10^{-6} times the standard deviation of the corresponding time series.

Two specific methods can be used to inform the algorithm about the preference for adjusting the force-field parameters (`ff_correction`) and the forward models (`forward_model`). The function `ff_correction` takes as input both the full set of force-field correction coefficients ϕ specified by `data.properties.names_ff_pars` and the `f` array previously defined and returns the force-field correction for each frame $\Delta V(x;\phi)$. The function `forward_model` takes as input both the full set of forward-model coefficients θ listed as in `forward_coeffs_0` and the dictionary `forward_qs` and returns calculated observables $g_i(x;\theta)$; the output is structured as a dictionary which is going to update `data.mol[sys_name].g` that can be compared element-wise against the experimental measures `data.mol[sys_name].gexp`. Relying on JAX to compute the derivatives, the functional forms of `ff_correction` and `forward_model` can be defined at discretion of the user by employing `jax.numpy` library. The choice of the regularization terms, external to the `data` object, is discussed in section 3.1.2.

Free-energy differences and thermodynamic cycles

We also included the case of experimental values given by free-energy differences ΔG_{exp} between couples of molecules. We built our implementation based on the prototypical case of computing mutation free-energies in single- (S) and double- (D) stranded nucleic acids, which can be measured through denaturation experiments [119]. This has been used in refining force-fields for RNA molecules with methylated nucleotides [41, 119]. In order to compare these experimental measures with MD simulations, alchemical transformations [119, 146] are performed transforming oligomers with wild-type adenosine (A) to methylated ones (M) both in the single strand S and duplex D cases.

In this framework,

- $\Delta G_{\mathrm{exp},A}$ and $\Delta G_{\mathrm{exp},M}$ are the experimental free-energy differences between duplex and single-stranded structures at fixed chemical composition (i.e., for wild-type A and methylated M, respectively);
- ΔG_S and ΔG_D are the free-energy differences from alchemical calculations between the wild-type and modified oligomers (A $\rightarrow$ M), performed in the single-stranded (S) and duplex (D) forms, respectively.

These quantities are related by the thermodynamic cycle in Fig. 3.4, leading to the key equation

$$\Delta G_{exp,A} - \Delta G_{exp,M} = \Delta G_S - \Delta G_D. \tag{3.1}$$

This identity allows direct comparison between experimental mutation-induced stability changes and the results of alchemical MD simulations.

To this aim, we introduced a new class `data_cycle_class` for the thermodynamic cycles, whose instances `data.cycle[cycle_name]` contain both the experimental value of $\Delta\Delta G$ (with associated uncertainty) and the temperature. In this case, the `data.mol` items of the four molecular systems belonging to the cycle `cycle_name` have to be indicated in `infos['global']['cycle_names']` in the same order as `['AS', 'AD', 'MS', 'MD']`, where one has the experimental ΔG between single- (S) and double- (D) -stranded molecules and the alchemical transformation between A and M molecules. In other words, the four systems should be listed in order such that the alchemical transformations are between the first and the third system and between the second and the fourth system, and the reference experimental value reports the difference between these two alchemical free energies. Each of them has a further attribute `logZ`, which is the natural logarithm of the partition function shifted by a given reference value and related to the alchemical transformation from A to M molecules (either single- or double-stranded). Fixing a reference ensemble P_0 for the energy, then it is $\log Z_A = \log\langle e^{-\beta V_A(x)}\rangle_{P_0}$ and analogous for $\log Z_M$, so the free-energy difference is

$$\Delta G|_A^M = -k_B T(\log Z_M - \log Z_A). \tag{3.2}$$

In most of the cases, one fixes P_A as reference ensemble $P_0 = P_A$, so $\log Z_A = 0$ and the previous eq. results in the Zwanzig equation [147]

$$\Delta G|_A^M = -k_B T \log Z_M = -k_B T \log\langle e^{-\beta \Delta V|_A^M}\rangle_A. \tag{3.3}$$

3.1.2 Basic calculations and reweighting

Once we have loaded all the required data in the `data` object, we are now ready to perform basic calculations and reweighting of the ensembles. First of all, we are interested in quantifying how well the ensemble P_0 predicted by MD simulations agrees with the experimental measures $g_{i,exp} \pm \sigma_{i,exp}$. This is usually done using the χ^2, which depends both on the ensemble P_0 and on the forward model parameters θ:

$$\chi^2[P_0, \theta] = \sum_i \Big(\frac{\langle g_i(x;\theta) \rangle_{P_0} - g_{i,exp}}{\sigma_{i,exp}} \Big)^2, \tag{3.4}$$

where $\langle g_i(x;\theta) \rangle_{P_0}$ is the average value of the observable $g_i(x;\theta)$ on the ensemble P_0. This is computed by the function `compute_chi2`, whose inputs are the `weights` representative of P_0, the observables `g` $g_i(x;\theta)$, the experimental values `gexp` and the `ref` dictionary previously defined. This last variable allows for including also the case of experimental values for upper/lower bounds. For example, the values of the item `gexp['uNOEs']` (unobserved NOEs) are upper bounds $g_{i,exp}$ for the signal observables `g['uNOEs']` with a tolerance $\sigma_{i,exp}$ (`ref['uNOEs'] = '<'`) and the corresponding contribution to the χ^2 is

$$\sum_{i \in uNOEs} \Big[\max\Big(0, \frac{\langle g_i(x) \rangle_{P_0} - g_{i,exp}}{\sigma_{i,exp}}\Big) \Big]^2. \tag{3.5}$$

The output values of the function `compute_chi2` are: the average quantities $\langle g_i(\theta;x) \rangle_{P_0}$, the χ^2 for each observable i, the relative difference $\frac{\langle g_i(x) \rangle_{P_0} - g_{i,exp}}{\sigma_{i,exp}}$ and the total χ^2.

If there is a significant discrepancy, the reference ensemble P_0 might require some adjustments, either by varying the empirical coefficients ϕ of the force field or by refining the ensemble P_0 with full flexibility, or both [126]. Once the dimensionless *correction* $\beta\Delta V(x)$ (namely, the logarithm of the correction factor) for each frame is given, we can reweight the ensemble by employing the function `compute_new_weights`, which takes as input the original weights $w_0(x)$ and the correction $\beta\Delta V(x)$ and returns the new (normalized) weights

$$w(x) = \frac{1}{Z} w_0(x)\, e^{-\beta\Delta V(x)} \tag{3.6}$$

together with $\log Z$. For example, the correction might be the output of `ff_correction` with some coefficients ϕ reasonably chosen.

Once we have got the new (reweighted) ensemble P, represented by the weights $w(x)$, we can both compute the new $\chi^2[P]$ and compare P

with the original ensemble P_0. In the case of alchemical calculations, the function `compute_DeltaDeltaG_terms` allows for computing the new free-energy differences $\Delta G|_{A\phi}^{M\phi}$ and χ^2 for the $\Delta\Delta G$ of the thermodynamic cycle by passing as input variables the `data` object and the dictionary of $\log Z_\phi = \log\langle e^{-\beta\Delta V(x;\phi)}\rangle_\phi$ values for the reweighted ensembles P_ϕ of each molecular system in the cycle.

Regularization terms

Regarding the regularization term comparing P_ϕ with the reference ensemble P_0, we can evaluate the deviation just by computing the $L2$ distance in the parameter space $\sum_i \phi_i^2$ (`l2_regularization`). However, changing by the same quantity some coefficients ϕ_j rather than others may lead to different variations of the ensemble, and the $L2$ norm itself is undefined if the coefficients are expressed in different units. An alternative possibility without these problems is the Kullback-Leibler divergence:

$$D_{KL}[P_\phi|P_0] = \sum_x w(x)\log\frac{w_\phi(x)}{w_0(x)} = -\beta\langle\Delta V_\phi(x)\rangle_P - \log Z \qquad (3.7)$$

with $Z = \langle e^{-\beta\Delta V_\phi(x)}\rangle_{P_0}$ the partition function of P_ϕ, whose logarithm is returned by `compute_new_weights`. This quantity is the negative of the relative entropy between the ensemble P_ϕ and P_0 and is positively definite, the greater it is the further the ensemble P_ϕ is from P_0. It can be computed through the function `compute_D_KL`, whose inputs are the new weights $P_\phi(x)$, the correction $\Delta V_\phi(x)$, the temperature β^{-1} and $\log Z$.

The `regularization` dictionary encodes both the terms acting on the force-field fitting (item `force_field_reg`) and those acting on the forward-model refinement (item `forward_model_reg`). The first item `force_field_reg` can be either a string (`KL divergence`, `plain l2`) or a user-defined function acting on ϕ; the second item `forward_model_reg` is a user-defined function acting on θ and θ_0 (for example, an $L2$ regularization). Now we have all the basic ingredients to compute the loss functions defined in the next section.

3.1.3 Loss function and search for optimal solutions

When comparing MD simulations with experiments, all the errors, both coming from the ensembles, the force field and the forward model, come into play. Hence, we consider the possibility to simultaneously refine all of them. To this aim, and motivated by previous works in these directions [1, 126], we deem the following loss function to be suitable for our task:

$$\mathcal{L}(P,\phi,\theta) = \frac{1}{2}\chi^2[P,\theta] + \alpha D_{KL}[P|P_\phi] + \beta R_1[P_\phi|P_0] + \gamma R_2[\theta|\theta_0] \tag{3.8}$$

where α, β, γ are hyperparameters regulating the strength of the corresponding regularization terms, P_ϕ is the canonical ensemble determined by applying the force-field correction ϕ to the reference ensemble P_0, and R_1, R_2 are customizable regularizations for the force-field and the forward-model corrections, respectively. In the previous Chapter 2, we focused on the case in which the regularization R_1 is given by the Kullback-Leibler divergence $D_{KL}[P_\phi|P_0]$. When considering multiple molecular systems, the χ^2 and D_{KL} terms (including those coming from R_1) are summed over all the systems.

The minimization of this loss function can be interpreted as a Pareto (or multi-objective) optimization [104], solved through a linear scalarization, with $\alpha, \beta, \gamma \geq 0$, $\alpha \neq 0$ the relative weights of the objective functions. This loss function is quite adaptable to different aims by formally setting some of the hyperparameters to $+\infty$. For example, if only α is finite we recover ensemble refinement (subsection 1.4.1), while if only β is finite we recover force-field refinement (subsection 1.4.2). For $\alpha < +\infty$, $\beta < +\infty$, and $\gamma = +\infty$ we fully trust the original forward-model coefficients getting the same loss function in 2 [126] (Eq. 2.1). Similarly, for $\alpha < +\infty$, $\beta = +\infty$, and $\gamma < +\infty$ we recover the same refinement scheme introduced in subsection 1.4.3 [1].

The `loss_function` is minimized by the `minimizer` routine. In this way, passing to `minimizer` the `data` object, the `regularization` dictionary and the hyperparameters, we get an output object `Result` containing the optimal solution (P, ϕ, θ) and its properties. By default, the hyperparameters α, β, γ are set to $+\infty$, implying no refinement in the corresponding directions.

In the `minimizer` routine, devoted to minimize the loss function, we exploit the same dimensionality reduction already used in subsection 2.1.3, based on the λ coefficients. Making the dependence $\lambda^*(\phi,\theta)$ explicit in the loss function $\mathcal{L}(P,\phi,\theta)$ we get

$$\mathcal{L}_1(\phi,\theta) = \mathcal{L}(\lambda^*(\phi,\theta),\phi,\theta) = -\alpha\Gamma(\lambda^*(\phi,\theta);\phi,\theta) + \beta R_1(\phi) + \gamma R_2(\theta) \tag{3.9}$$

and the optimal coefficients are determined by

$$(\phi^*,\theta^*) = \arg\min_{\phi,\theta} \mathcal{L}_1(\phi,\theta) \quad \text{and} \quad \lambda^* = \arg\min_{\lambda} \Gamma(\lambda;\phi^*,\theta^*) \tag{3.10}$$

(see Appendix B.1-C.1 for further details). Notice that the dimensionality reduction with λ parameters holds whenever the loss function has the form $\mathcal{L}[P] = \ell(\langle g_i(x)\rangle_P) + \alpha D_{KL}[P|P_0]$ with $\ell(\langle g_i(x)\rangle_P$ a generic function of the average values. Still, the derivation of the $\tilde{\Gamma}$ function in Eq. 1.16, enabling a more efficient minimization, requires ℓ to be a quadratic function of its arguments[1]. When minimizing Γ, experimental values for upper/lower bounds (as in Eq. 3.6) can be included by employing boundaries in the λ space ($\lambda_i \geq 0$ for upper limits `ref = '<'`, $\lambda_i \leq 0$ for lower limits `ref = '>'`).

3.1.4 Searching for optimal hyperparameters via cross validation

Suitable ranges of the hyperparameters α, β, γ can be determined by searching for an intermediate range between underfitting and overfitting. This can be done by splitting the data set into training and validation set and then searching for the hyperparameters that minimize the χ^2 on the validation set (cross validation). This splitting can be done in several ways. The way we recommend is to select some frames and some observables to belong to the validation set, leaving them out from the search of the optimal coefficients, which is done using only the training set. Keeping frames in the validation set makes sure the corrections can be generalized to a new (or longer) trajectory simulated for the same system, whereas keeping observables in the validation set strengthen the transferability of the correction to new observables [33]. Before splitting the dataset into training and validation set, it is advisable to put aside a batch of the dataset that will be used as a test set, to report the final model performance. This external splitting should be done in the same way as the one of the internal splitting between training and validation set (nested cross-validation).

The function `split_dataset` performs these partitions. Passing the `data` object, this is split into two sets, the first used in training `data_train` and the second in test (external splitting). After that, we can run again `split_dataset` several times passing `data_train`: in this way, the training dataset will be iteratively split into (internal) training and validation sets, the second one used to optimize the hyperparameters. The default ratios of the validation frames `frames_fraction` and validation observables `obs_fraction` are of 20% on the full number of frames and (cross-validation) observables for each molecular system. Alternatively, one can also pass two dictionaries `valid_frames` and `valid_obs` with user-defined values. The input variable

[1]This is because the Legendre transform of a quadratic function $f(\langle g_i\rangle_P)$ is another quadratic function $\tilde{\Gamma}(\lambda_i)$.

`random_state` (an integer number) allows for fixing the seed used in the random splitting, thus making it reproducible. By default, the validation observables are calculated only on validation frames, however they can be computed on all the frames just by setting `if_all_frames = True`.

The `validation` function, called by `minimizer`, includes in the output object `Result` also properties referring to the validation set, such as the χ^2, both for training observables on validation frames and for validation observables.

With the implementation described so far, one can scan over different values of the hyperparameters, minimize the loss function on the training set and evaluate the χ^2 on the validation set, to identify values of the hyperparameters that maximize transferability of the parameters. Whereas this procedure allows for determining suitable values for the hyperparameters, it suffers from the curse of dimensionality: while an exhaustive search is feasible for 1 or 2 dimensions, making a scan in 3 dimensions or more is cumbersome. To overcome this issue, we implemented an automatic search of the optimal hyperparameters [148].

This search is performed by randomly splitting the full data set into training and validation set several times (as indicated by `random_states`) and minimizing the average χ^2 on the validation set, evaluated at the optimal solution found in training. Since both ϕ and θ appear in the minimization, in the following we will use $\mu = (\phi, \theta)$ as a shortcut. The optimal coefficients depend on the hyperparameters via

$$\begin{aligned} \lambda^*(\mu^*; \alpha) &= \arg\min_{\lambda} \Gamma(\lambda; \mu^*, \alpha) \\ \mu^*(\alpha, \beta, \gamma) &= \arg\min_{\mu} \mathcal{L}_1(\mu; \alpha, \beta, \gamma), \end{aligned} \tag{3.11}$$

therefore by the chain rule the derivatives of the $\chi^2(\lambda^*, \mu^*)$ with respect to the hyperparameters result

$$\frac{\partial \chi^2}{\partial \alpha} = \frac{\partial \chi^2}{\partial \mu_i}\frac{\partial \mu_i^*}{\partial \alpha} + \frac{\partial \chi^2}{\partial \lambda_i}\left(\frac{\partial \lambda_i^*}{\partial \mu_j}\frac{\partial \mu_j^*}{\partial \alpha} + \frac{\partial \lambda_i^*}{\partial \alpha}\right) \tag{3.12}$$

$$\frac{\partial \chi^2}{\partial (\beta, \gamma)} = \frac{\partial \chi^2}{\partial \mu_i}\frac{\partial \mu_i^*}{\partial (\beta, \gamma)} + \frac{\partial \chi^2}{\partial \lambda_i}\frac{\partial \lambda_i^*}{\partial \mu_j}\frac{\partial \mu_j^*}{\partial (\beta, \gamma)} \tag{3.13}$$

(see Appendix B.1, C.1 for the complete derivation).

This optimization is performed by `hyper_minimizer`, which requires the `data` object, the starting values of the hyperparameters (by default constrained to $+\infty$, so that one has to select the refinement by setting finite values to some of them), the `regularization` and the `random_states`. One can

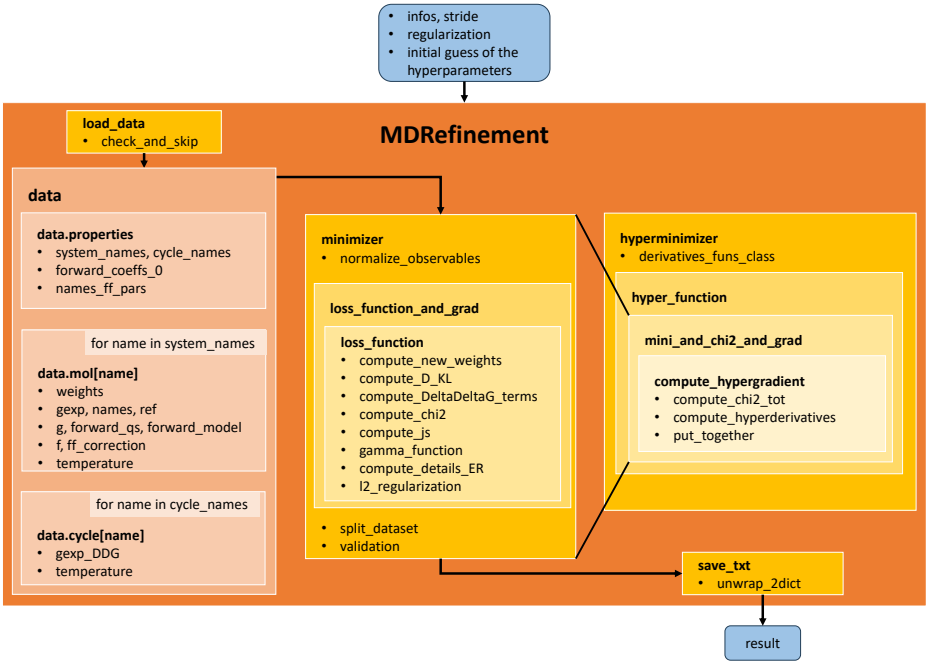


Figure 3.1: Software architecture: `MDRefinement` contains two main blocks `minimizer` and `hyperminimizer`, besides `load_data` and `save_txt`. The input variables of `MDRefinement` include `infos` (dictionary with information about the data set to be loaded), user-defined `regularization` terms and initial guess for the hyperparameters (selected refinement).

also choose among minimizing the χ^2 for training observables and validation frames (`which_set = 'valid_frames'`) or validation observables and frames (`which_set = 'validation'`). Since the minimization acts on orders of magnitude (namely, in logarithmic scale), the optimal hyperparameters are returned as $\log_{10}(\alpha, \beta, \gamma)$.

3.1.5 The main function: `MDRefinement`

The main function of the package is named `MDRefinement` and integrates most of the routines defined above, as illustrated in Fig. 3.1. Its flow chart is shown in Fig. 3.2. It loads both the `data` (from input files according to `infos`) and the `regularization`, searches for the optimal hyperparameters in cross validation and returns the optimal solution `Result` on the full data set. The output values are also saved as `csv` files in a folder named `results`, including `input`, `min_lambdas`, `result` object, the steps in the hyperparameter search `hyper_search` and the optimized weights.

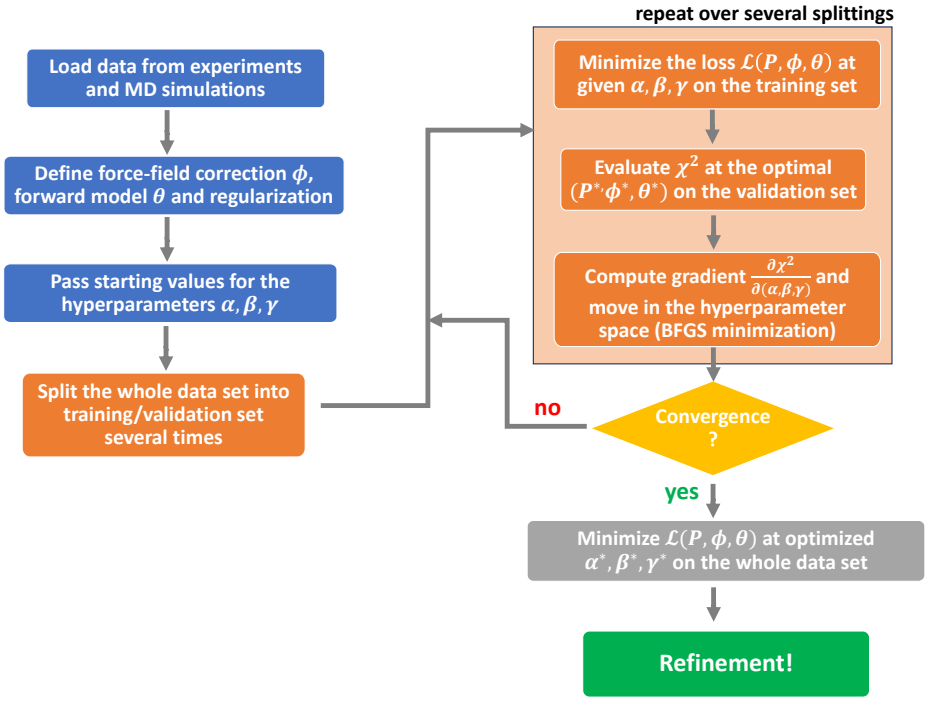


Figure 3.2: Flow chart of `MDRefinement`: after loading the data and defining the functions (blue blocks), the cross-validation section (in orange) determines the optimal hyperparameters, which are then used (gray block) to get the most suitable refinement on the data set.

3.1.6 Limitations and good practices

Based on reweighting pre-existing MD trajectories, `MDRefine` requires to make some assumptions, whose validity can be verified both before and after the refinement itself. First of all, we can assess if a *meaningful* reweighting does exist, given the datasets from the MD simulations and the experimental values. By this, we mean to check if there exists a set of weights $\{w_j\}$ for which the corresponding weighted average of the observables (sampled during the MD trajectory) $\langle \vec{g}_j \rangle_w$ is in good agreement with the experimental measurements $\{g_{i,exp} \pm \sigma_{i,exp}\}$. If such a reweighting does not exist, `MDRefine` can still be applied, but this might result in transferring all the statistical weight to very few frames, the closest ones to the experimental region, with a very low effective sample size (high relative entropy). This happens when the experimental region $\{g_{i,exp} \pm \sigma_{i,exp}\}$ lies outside (and far from) the convex hull defined by the values of the observables that have been sampled. Notice that the existence of such a reweighting does not imply that the MD simulation has visited the region highlighted by the experimental measures, as it is expected since the experiments only report the average values. In order to find such set of weights, we can minimize the χ^2 with a very soft relative-entropy regularization (by using the `minimizer` function with hyperparameter α very small), and then analyse the χ^2 itself. If, even in the presence of such a very small regularization term, the χ^2 is significantly larger than the n. of observables, then it is not possible to reweight the ensemble to match the experiments. In these cases, alternative methods based on enforcing the experiments on-the-fly [19, 116, 136, 149–151] might be more effective in generating an ensemble overlapping with the experimental measurements.

Once the refinement has been performed, one should check the discrepancy between reference and reweighted ensembles (quantified by the Kullback-Leibler divergence, for instance; see Appendix A). This quantity should be "small enough" to ensure that a significant overlap with the original MD simulation does exist. Rules of thumb for this value have been suggested in previous works [14]. If this condition is not fulfilled, reweighting procedures would be statistical inefficient and further MD simulations should be required before the refinement takes place. The strong disagreement among experimental data and reference ensemble might be due to poor / inefficient sampling, or sampling with inaccurate force field. In this second case, the application of `MDRefine` will give anyway a suggestion on how to modify the force field / ensemble in order to improve the agreement with the experiments.

Regarding the regularization for the force-field correction, using only the relative entropy might not be a good choice. This is because some force-field parameters might be "uninformative", namely, a region of the parameter space

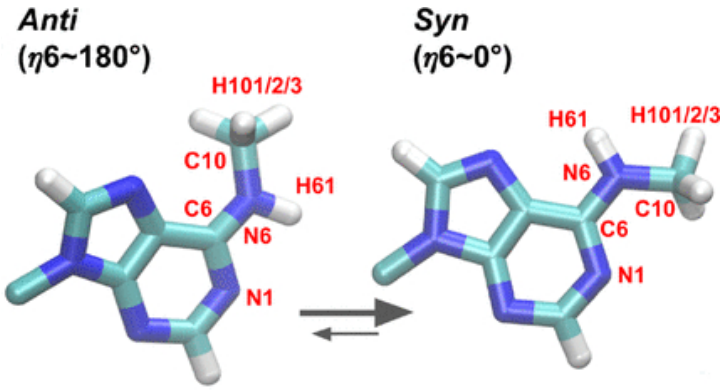


Figure 3.3: N6-methyladenosine (m6A) nucleobase adopts both anti (less stable) and syn (more stable) conformations. Image adapted from [119].

could correspond to almost the same correction to the sampled trajectory, due to steric hindrance for instance. When this happens, including an $L2$ regularization might be useful [118]. We also advice on the importance to perform new MD simulations after the force-field refinement has been determined.

More details about the strengths and limitations of the various implemented methods can be found in the original papers. To facilitate access, `MDRefine` provides a list of relevant references based on the selected options. Users are encouraged to read these references carefully and to cite them if they use the methods described therein.

3.2 Results

To showcase the capabilities of the `MDRefine` package, we present analyses of several precomputed trajectories. Some of this data reproduces results from previous studies [1, 119, 126], while new analyses highlight the enhanced features introduced in this package. Complete Python notebooks for reproducing these analyses are available at `https://github.com/bussilab/MDRefine`. The datasets required to run the tutorials are available at `https://zenodo.org/records/14956459` and were obtained as indicated in the notebooks `load_data_oligomers` and `load_data_alchemical`.

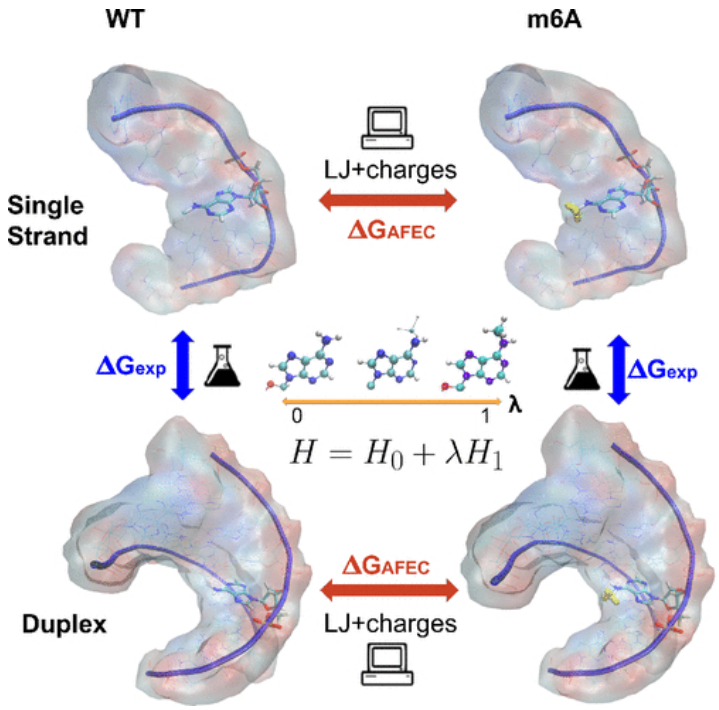


Figure 3.4: Thermodynamic cycle. Alchemical free-energy calculations (AFECs) enable the computation of ΔG by integrating along an alchemical coordinate λ, which describes the transformation of a non-methylated adenosine into m6A. This is achieved by gradually turning on the non-bonded interactions of specific selected atoms. The relative free-energy change introduced by the modification can be estimated as the $\Delta\Delta G$ between AFECs performed on the duplex and the corresponding single strand. This quantity can be directly compared to experimental measurements of thermodynamic-stability differences between single-strand and duplex, both in the methylated and wild-type structures, as obtained from denaturation experiments. Image adapted from [119].

3.2.1 Force-field refinement with denaturation experiments

In this section we reproduce force-field corrections derived in [119] and discuss the effect of using a different regularization strategy based on the relative entropy. This is performed in the notebook `Tutorial_1c.ipynb`. The main intent of this refinement is to improve the description of N6-methyladenosine (m^6A), one of the most common post-transcriptional modifications of RNA, consisting in the methylation of the standard adenosine at position N6. Notably, m^6A can exist in two possible conformations, depending on the orientation of the methyl group with respect to the rest of the nucleobase (see Fig. 3.3): the *syn* and *anti* isomers, which are distinguished by the value of the torsional angle η_6 defined by the atoms N1-C6-N6-C10. The *syn* conformation is known to be the most stable for unpaired m^6A, whereas the *anti* isomer is the one expected when m^6A is Watson-Crick paired with uracil in an internal position of a double-stranded RNA.

As a starting parametrization we used the modrna08 force-field [37], developed by Aduri and coworkers using the standard AMBER protocol based on quantum-mechanical calculations and ReSP (Restrained Electrostatic Potential) fitting [152]. This parametrization does not predict the correct *syn*/*anti* balance for unpaired m^6A. We then apply a correction to the force-field using the same prescription outlined in [119]. This correction includes a sinusoidal term acting on the η_6 torsional angle $\Delta V(\eta_6) = -V_\eta \cos\eta_6$ (positive values of V_η favour syn conformations over anti) and a reparametrization of the partial charges close to the methyl group (C6, N6, H61, N1, C10, H101/2/3), with the constraint on the total charge to be constant $\sum_i \Delta Q_i = 0$ and on the charges of the three equivalent hydrogens (H101/2/3) to be identical. Here ΔQ_i represents the variation of the partial charge of atom i when compared with the starting parametrization. This results in 6 free parameters: V_η and 5 partial charges ΔQ_i.

The total energy depends quadratically on the charge perturbations ΔQ_i, hence, following [119], it can be expressed as

$$\Delta V(x) = \sum_{i=1}^{5} K_i(x)\Delta Q_i + \sum_{i=1}^{5}\sum_{j=1}^{5} K_{ij}(x)\Delta Q_i \Delta Q_j - V_\eta \cos\eta_6(x). \quad (3.14)$$

The 20 coefficients K_i and K_{ij} can be determined through linear algebra operations starting from 20 linearly independent sets of charge perturbations, allowing the computation of the total energy for any choice of the charges ΔQ_i without the need to repeat the computationally expensive electrostatic-energy calculation.

These 6 free parameters (ΔQ_i, V_η) were fitted by comparing the reweighted predictions of MD simulations with a number of denaturation experiments measuring free-energy differences ΔG. The molecules used in the training phase include: m^6A (*syn*/*anti* difference), $\underset{AUGCUGAC}{UACG6CUG}$, $\underset{GCUAUCCA}{CGAU6GGU}$, $\underset{GCG}{6CGC}$, $\underset{CGC}{GCG6}$ where 6 stands for the m^6A nucleoside. This comparison is made by realizing a thermodynamic cycle through alchemical calculations (see subsection 3.1.1 and references [119, 146, 153] for further details). The corresponding loss function includes a generic regularization term $R(\Delta Q, V_\eta)$, which encodes the prior knowledge on the force field:

$$\mathcal{L}(\Delta Q, V_\eta) = \frac{1}{2} \sum_{i=1}^{N_{cycles}} \left(\frac{\Delta\Delta G_i(\Delta Q, V_\eta) - \Delta\Delta G_{i,exp}}{\sigma_{i,exp}} \right)^2 + \beta R(\Delta Q, V_\eta). \quad (3.15)$$

Strictly speaking, the $L2$ regularization requires dimensionally homogeneous parameters, which is not the case here. This issue can be solved by introducing an additional hyperparameter to regulate the strength of the $L2$ penalty across V_η and the set of partial charges ΔQ_i. However, in [119] it has been suggested that, for this system, it suffices to regularize the partial charges. We follow the same protocol here, resulting in a $L2$ regularization that reads:

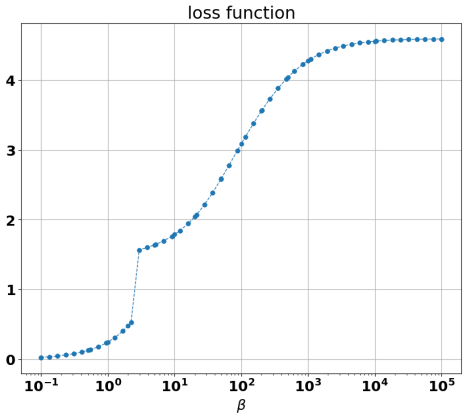
$$R(\Delta Q, V_\eta) = \sum_{i=1}^{6} \Delta Q_i^2 = \sum_{i=1}^{5} \Delta Q_i^2 + \left(\sum_{i=1}^{5} \Delta Q_i + 2\Delta Q_{H101} \right)^2. \quad (3.16)$$

As an alternative procedure, we here examine the possibility of regularizing using the relative entropy $R(\Delta Q, V_\eta) = D_{KL}[P(\Delta Q, V_\eta)||P_0]$ [118], summed over all the systems. The results are shown in Figs. 3.5 and 3.6.

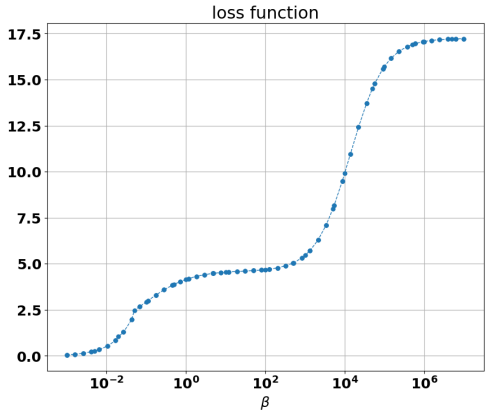
In Fig. 3.5 we report the minimum value of the loss function, the χ^2 for the different thermodynamic cycles and the average Kish size ratio (KSR) as a function of the hyperparameter β. The overfitting transition is quite evident in the case with $L2$ regularization, while it is smoother in the second case. For the sake of clarity, in Appendix A, we report the definitions of Kish size ratio and relative Kish size, introduced in previous works [1, 119], and highlight their relationship with the entropy or the relative entropy.

In both cases (Fig. 3.6 a, b), starting from a high value of the hyperparameter β and decreasing it, the parameters change abruptly below a certain threshold. This can be interpreted as a transition to overfitting the experimental data. Fig. 3.6c reports the relationship between the discrepancy from the reference ensemble and the discrepancy from the experiment, using both regularization flavors [104]. In both cases, a smaller discrepancy from experiment (χ^2) can only be obtained at the price of a larger perturbation

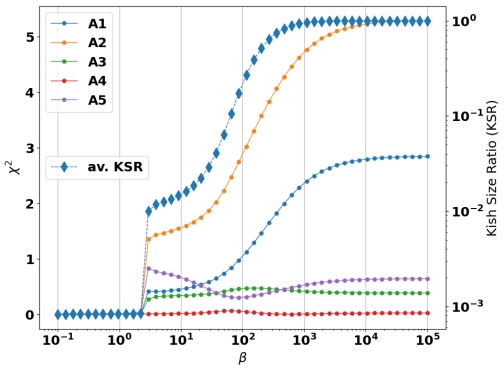
(a) Min. loss ($L2$ regularization)



(b) Min. loss (S_{rel} regularization)



(c) χ^2 and av. KSR ($L2$ regularization)



(d) χ^2 and av. KSR (S_{rel} regularization)

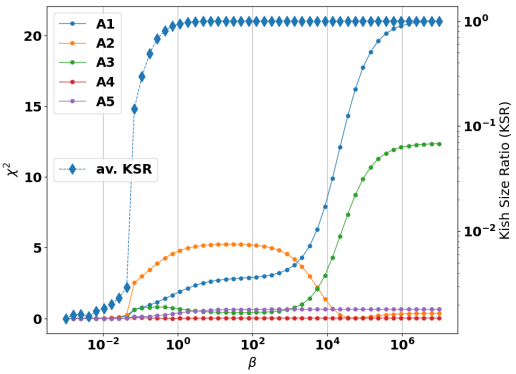


Figure 3.5: Loss function, average Kish size ratio and χ^2 for each thermodynamic cycle at the optimal solution found at given hyperparameter β and regularization ($L2$ on charges or relative entropy). The overfitting transition is quite evident in the case of $L2$ regularization, while it is smoother for the relative entropy regularization.

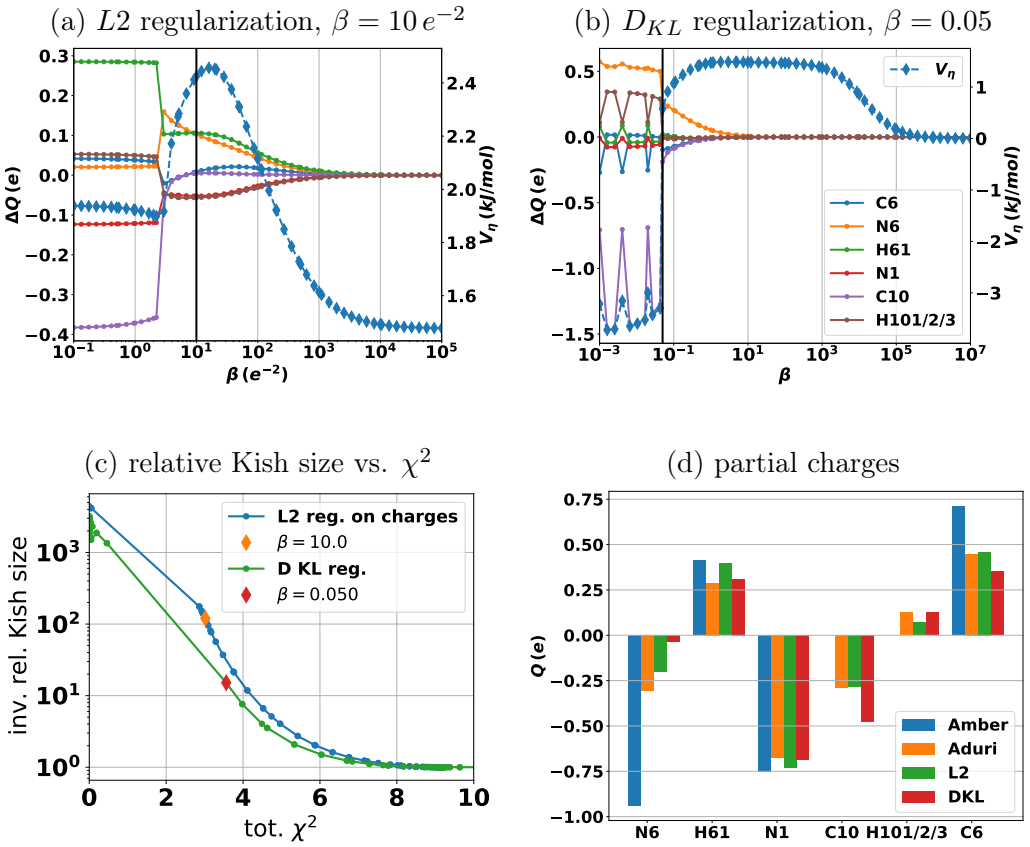


Figure 3.6: Refinement of charges and V_η for alchemical calculations; comparison between $L2$ (a) and relative entropy (b) regularization. (c) Inverse of the relative Kish size $K^{-1} = \sum_x w(x)^2/w_0(x)$, measuring the deviation from the reference ensemble, as a function of the total χ^2 of the $\Delta\Delta G$ on the thermodynamic cycles, at constant steps in the $\log\beta$ scale. (d) Partial charges $Q(e)$ for different parametrizations: Amber force field for wild-type adenosine, Aduri force field for methylated adenosine, $L2$ and relative entropy regularization.

of the ensemble. As expected, relative-entropy regularization results in less perturbed ensembles than $L2$ regularization for the same χ^2 value. Interestingly, even if the β values are uniformly spaced in logarithmic scale, an abrupt change to the overfitting regime is observed, leading to a sharp decrease in χ^2.

Based on this, we choose hyperparameter values slightly higher than this threshold: $\beta = 10\,e^{-2}$ (the same as in Ref. [119]) and $\beta \simeq 0.5$ for $L2$ and D_{KL} regularizations, respectively. These values are marked with bold vertical line in Fig. 3.6 a, b. Having an objective criterion to choose the hyperparameter is critical in this case because it allows us to compare the results obtained with the two regularization strategies, even if the value of the hyperparameters are not directly comparable. An alternative strategy for choosing hyperparameters that we did not examine here is cross validation, described in subsection 2.1.5 and employed in the following investigations.

In Fig. 3.6d we report the partial charges obtained for the chosen hyperparameters, and compare them with other sets of partial charges for the same system. There are important differences when comparing the refinement with $L2$ or relative entropy regularization. Most importantly, the D_{KL} regularization leaves relatively unconstrained C6 and N6 charges (see Fig. 3.3 for atom numbering). This is due to the fact that these atoms can not form close contacts with other atoms, and thus their charge can be modified without a significant perturbation of the ensemble. The strengthening of Watson-Crick hydrogen bonds that when using $L2$ regularization was obtained by making N1 and H61 slightly more polar, is here obtained by making N6 virtually apolar, transferring its charge largely to C10. Indeed, the $L2$ regularization keeps the final parameters close to those chosen based on the quantum mechanical calculations in Ref. [37], whereas D_{KL} considers the effect of the parameters on the resulting ensemble.

3.2.2 Simultaneous refinement of forward model and structural ensembles

In this section we illustrate the usage of `MDRefine` for the case of simultaneous refinement of the Karplus equations (used in Chapter 2) and structural ensembles. The computations have been performed in the Jupyter notebook `Tutorial_3.ipynb`, with extensive calculations run on HPC cluster. Given the loss function in Eq. 3.8, this refinement corresponds to setting $\beta = +\infty$, without further flexibility in the force-field parameters.

This benchmark case reproduces results from a previous work [1]. The molecular systems include several RNA oligomers: AAAA, CAAU, CCCC,

GACC, UUUU, UCAAUC. MD trajectories and experimental observables are the same used in Chapter 2 (see subsection 2.3.1).

The forward models here consist in the Karplus equations with empirical parameters A, B, C, relating the dihedral angle ϕ to the ${}^3J$ scalar coupling:

$$ {}^3J(\phi) = A\cos^2\phi + B\cos\phi + C. \tag{3.17} $$

The dihedral angles for which experimental data are available include β, ε, γ along the backbone and the sugar puckering angles ν_1, ν_2, ν_3. We use as reference coefficients the values reported in [5–7]. The same coefficients were used for β and ε dihedral angles, both corresponding to H-C-O-P atom types. The regularization of the forward model here adopted is the square deviation between new and reference ${}^3J$ couplings averaged over the angle, which can be obtained from the parameters as:

$$ \begin{aligned} R(A,B,C) &= \frac{1}{2\pi}\int_0^{2\pi} d\phi\,({}^3J(\phi) - {}^3J_0(\phi))^2 \\ &= \frac{3}{8}(A-A_0)^2 + \frac{1}{2}(B-B_0)^2 + (C-C_0)^2 + (A-A_0)(C-C_0), \end{aligned} \tag{3.18} $$

This expression is identical to the one used in Ref. [1], except for an arbitrary prefactor 2.

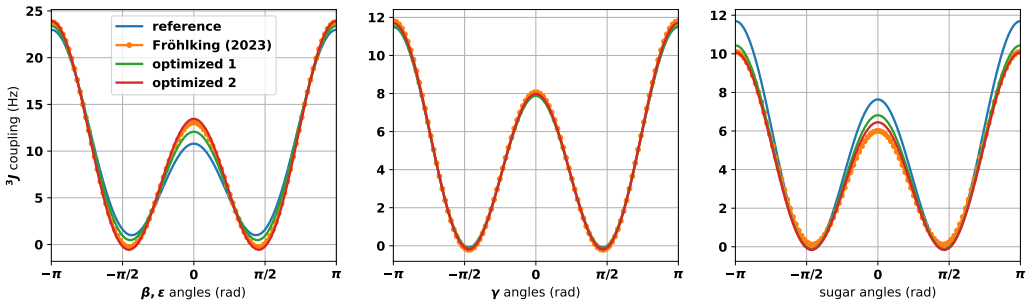


Figure 3.7: Refinement of the empirical coefficients for the Karplus equations with Ensemble + Forward Model Refinement; comparison between *optimized 1* and *optimized 2* (see Table 3.1).

name	A	B	C
β, ε			
reference	15.3	-6.1	1.6
Fröhlking	18.34	-5.39	0.11
optimized 1	16.80	-5.66	0.94
optimized 2	18.91	-5.23	-0.21
fully combined	18.13	-5.29	0.072
γ			
reference	9.7	-1.8	0
Fröhlking	10.07	-1.87	-0.13
optimized 1	9.77	-1.83	-0.057
optimized 2	9.95	-1.86	-0.11
fully combined	9.92	-1.76	-0.12
sugar			
reference	9.67	-2.03	0
Fröhlking	7.81	-2.05	0.25
optimized 1	8.68	-1.81	-0.049
optimized 2	8.34	-1.81	-0.069
fully combined	6.99	-1.64	0.67

Table 3.1: Empirical coefficients A, B, C (in Hz) of the Karplus equations. Comparison between: reference values; coefficients determined by Fröhlking et al. in [1] (hyperparameters $\alpha = 174.33288$, $\gamma = 6.86664/2\,Hz^{-2}$ – factor $1/2$ because of different regularization prefactor); coefficients determined in this work (optimized 1, i.e., with same hyperparameters as in [1]; optimized 2, i.e., with optimal hyperparameters found in this work $\alpha = 46.31$, $\gamma = 0.96\,Hz^{-2}$); coefficients determined in this work with the fully combined approach ($\alpha = 959.86$, $\beta = 167.94$, $\gamma = 0.42\,Hz^{-2}$).

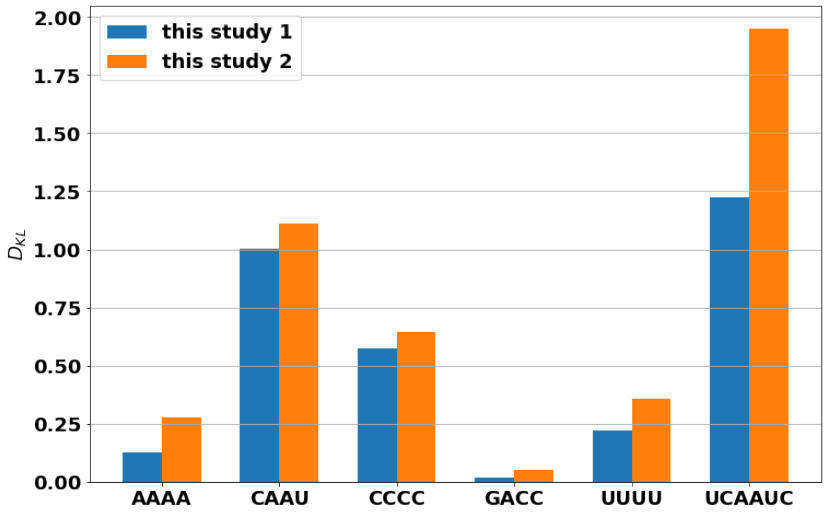


Figure 3.8: Refinement of Karplus equations: Kullback-Leibler divergences of the refined ensembles with respect to the original ones (optimized 1 with same hyperparameters as in [1]; 2 with optimized hyperparameters).

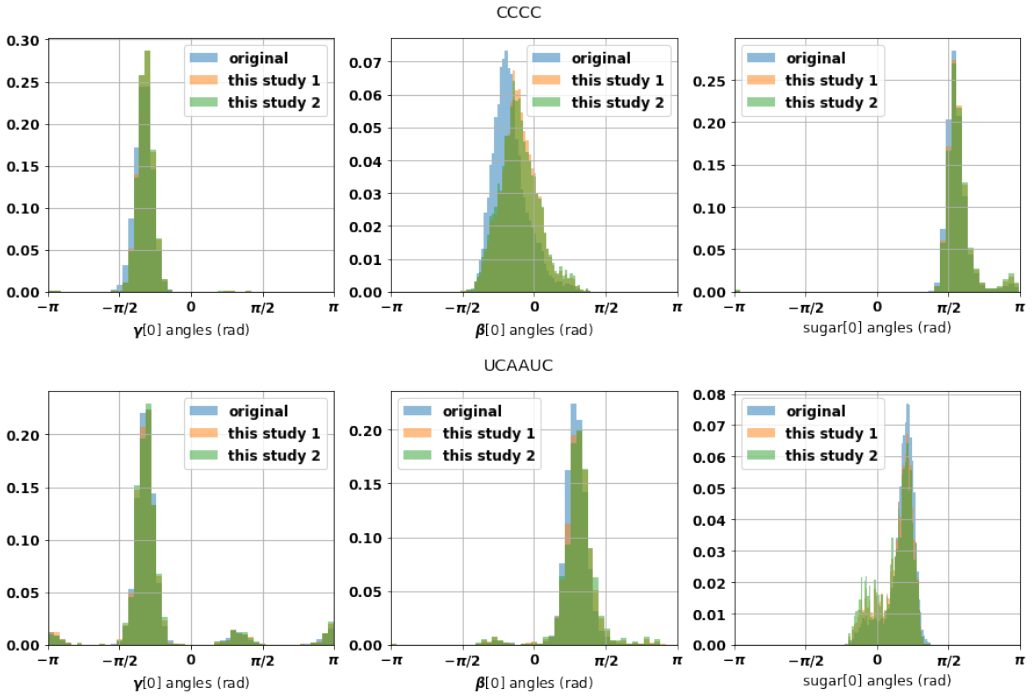


Figure 3.9: Weighted histograms of some dihedral angles of CCCC and UCAAUC oligomers, as examples (comparison between original ensemble and reweighted ones).

We then perform a simultaneous optimization of the parameters A, B, and C (three for each of the three Karplus equations), and of the structural ensembles, as in [1]. To determine the optimal hyperparameters, we perform an automatic search with the `hyper_minimizer` tool described above through minimization of the χ^2 on validation observables. Notice that the cross-validation performed in [1] employed a slightly different choice for the training/validation set, hence we do not expect to get exactly the same hyperparameter values as the optimal ones. In particular, the values corresponding to those reported in [1] are $\alpha = 174.33$, $\gamma = 6.87\,Hz^{-2}$, whereas those obtained in this work are $\alpha = 46.31$, $\gamma = 0.96\,Hz^{-2}$. Then, we compare the parameters found for these two sets of optimal hyperparameters with both the reference values and those determined in [1] (Fig. 3.7 and Table 3.1). Overall, our results adjust the reference empirical coefficients of the Karplus equations in the same direction suggested in [1]. Moreover, the refinement performed with hyperparameters optimized by `MDRefine` is closer to the results proposed in [1]. Hence, we report the refinement with this second choice of coefficients in Fig. 3.10.

Remarkably, these refinements minimally modify the structural ensembles: what matters most is the inaccuracy in the forward model. Indeed, the Kullback-Leibler divergence between reweighted and original ensembles is quite small, and also the histograms of the angles are not significantly modified (see Figs. 3.8-3.9). Hence, we conclude that most of the discrepancy between MD trajectories and experiments is a consequence of forward-model errors rather than of genuine structural discrepancies.

3.2.3 Fully combined refinement

In this section we describe an application of the fully combined refinement, which adds the adjustment of the force-field coefficients to the simultaneous refinement of ensembles and forward models shown in the previous section. This is obtained by minimizing the full loss function in Eq. 3.8. In this way, we are able to smoothly interpolate between these three different refinements (of force field, ensembles, and forward model), casting in a single framework several independent works [1, 119, 126]. The benchmark case discussed here is fully described in `Tutorial_3.ipynb`, and builds on the refinement described in the previous section adding some flexibility in the force-field coefficients. In particular, following Chapter 2 [126], we focus on a sinusoidal correction on α dihedral angles

$$\Delta V(\alpha) = \phi_1 \cos\alpha + \phi_2 \sin\alpha. \tag{3.19}$$

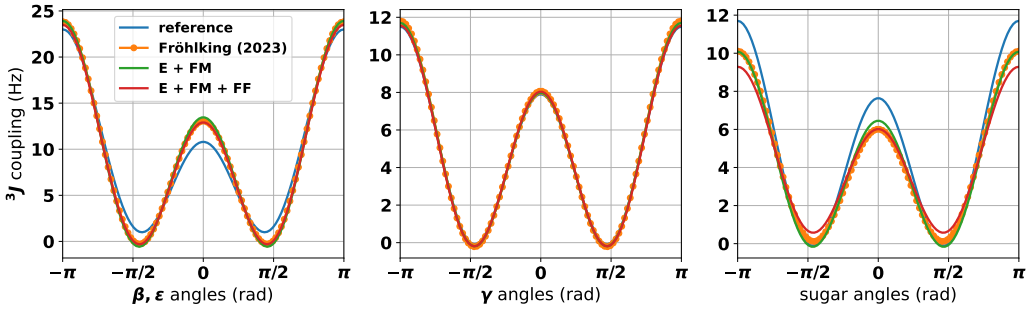


Figure 3.10: Refinement of Karplus equations. Comparison between: reference values (Lankhorst [6] for β, ϵ angles, Davies [5] for γ angles, Condon [7] for sugar angles), refinement by Fröhlking et al. [1], this work with Ensemble + Forward-Model Refinement $E + FM$ and with the fully combined approach $E + FM + FF$.

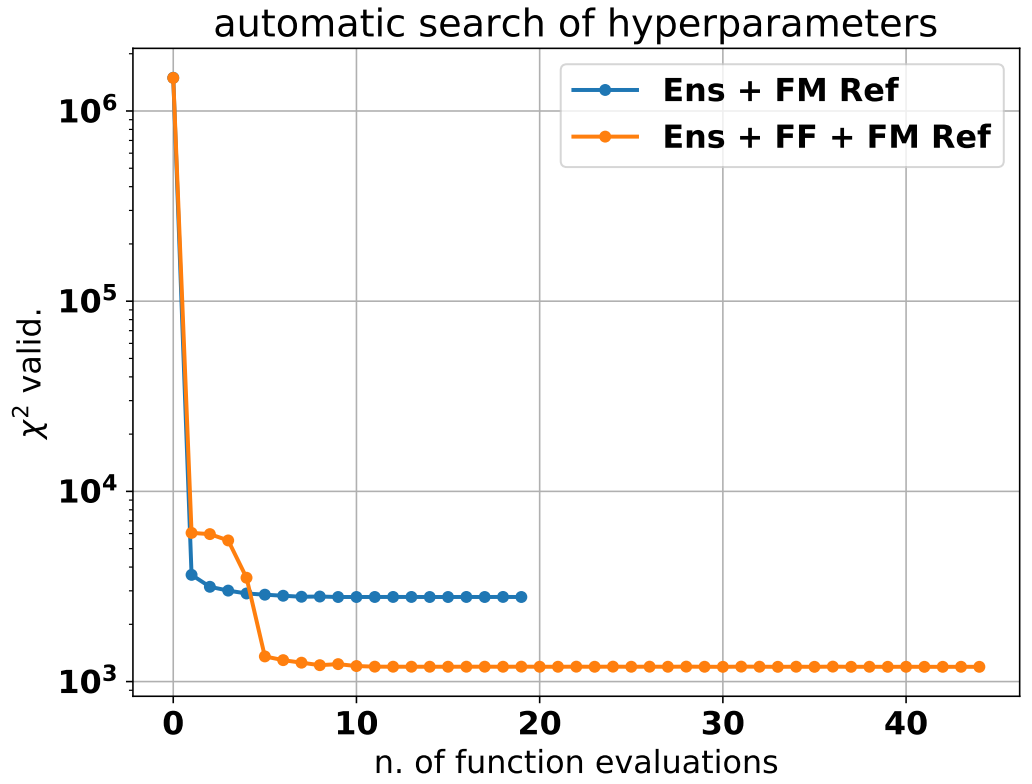


Figure 3.11: Average value of χ^2 on validating observables in the course of the automatic search for the optimal hyperparameters. Comparison among Ens + FM (Ensemble + Forward Model) Refinement and Ens + FF + FM (the fully combined) Refinement. Starting from the initial values $\alpha = \beta = 1$, $\gamma = 1\,Hz^{-2}$, we get the final values $\chi^2 = 2784.1$ for Ens+FM Ref. ($\alpha = 46.31$, $\gamma = 0.96\,Hz^{-2}$) and $\chi^2 = 1195.6$ for the fully combined approach ($\alpha = 959.9$, $\beta = 167.9$, $\gamma = 0.42\,Hz^{-2}$). The cross validation is performed on the full dataset (without test observables).

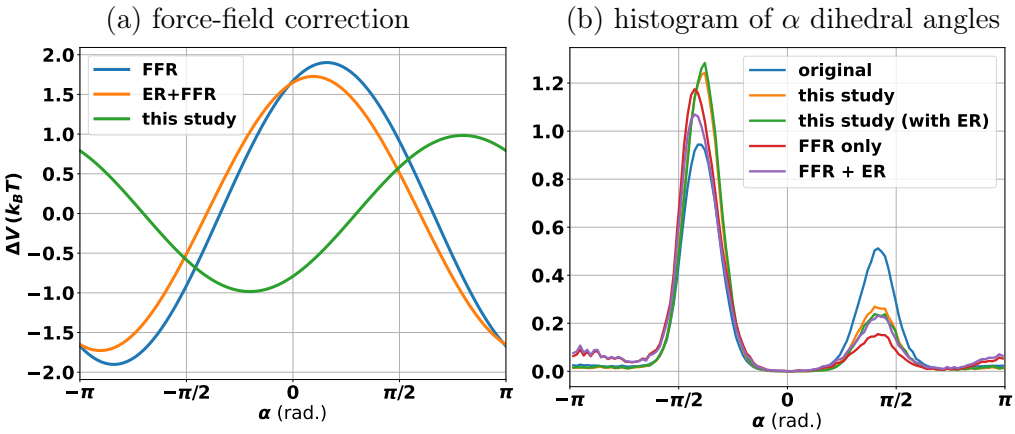


Figure 3.12: a) Force-field correction on α dihedral angles; comparison between this study (fully combined refinement) and previous [126] one (force-field refinement FFR only or combined with ensemble refinement FFR+ER). b) Average histogram of α dihedral angles; comparison between original one, this study (both reweighting with force-field correction only or including also contribution from ER) and previous study [126] (corresponding to the force-field corrections in Fig. 3.12a).

Then, we search for the optimal hyperparameters by using the `hyper_minimizer` routine of `MDRefine`. We perform the same splitting of observables and frames in training/validation set as in the previous cross validation (with 5 different choices) and then minimize the average χ^2 on cross-validated observables. We get as optimal hyperparameters $\alpha = 959.9$, $\beta = 167.9$, $\gamma = 0.42\,Hz^{-2}$. Importantly, a grid search in a three-dimensional hyperparameter space would have been extremely costly. In our case, the minimization algorithm converges in only 44 function evaluations, and is able to reach a cross-validation error very close to the minimum value in less than 10 evaluations (see Fig. 3.11).

The resulting refinement of the Karplus equations is reported in Fig. 3.10. It overlaps to that determined in [1] ("Fröhlking 2023" in the legend) very well, except for a slight deviation regarding sugar puckering angles close to $\pm\pi/2$ and π.

The resulting force-field corrections are reported in Fig. 3.12a. The correction proposed here appears completely different from the one suggested in the previous Chapter 2 [126]. However, this can be explained by comparison with the populations of the dihedral angle, averaged over all systems (Fig. 3.12a). Due to steric hindrance, only some regions for the α dihedral angles are allowed to be populated. In spite of the significant difference in the force-field corrections, the resulting populations are quite similar. Even though the corrections represented in Fig. 3.12a appears different, their effect on the

test χ^2_{red} original	test χ^2_{red} E+FM	test χ^2_{red} E+FM+FF
537.4	1.62	2.13

Table 3.2: Reduced χ^2 (namely, χ^2 divided by the n. of observables) on the test set; original set of weights (before the refinement), refinement of ensemble and forward model (E+FM), fully combined refinement (E+FM+FF).

accessible regions sampled by MD (see histograms in Fig. 3.12b) is comparable. By comparing the histograms in Fig. 3.12b, we conclude that this refinement shifts the population of α dihedral angles in the same direction as the one determined in the previous work [126] through force-field refinement only or combined with ensemble refinement. It is however more conservative due to the combined refinement of the forward model. Further, by comparing the two relative entropies $S[P_\phi|P_0]$ and $S[P|P_\phi]$ (Fig. 3.13), we argue the contribution of flexible ensemble refinement does not further modify the ensembles in a significant way. This means that the force-field correction alone is able to capture a significant fraction of the required correction.

In order to compare the performances of the two different refinement methods (with or without the force-field correction), we put aside some observables and repeat analogous calculations as above by applying `MDRefinement` on the reduced dataset (nested cross-validation). We recover results compatible to those reported in the previous plots Fig. 3.10 and 3.12 – see Appendix C.2 for further details. Then, we evaluate the corresponding χ^2 on the test set, reported in table 3.2. The improvement with respect to the original ensembles is noticeable, however the inclusion of force-field fitting does not exhibit a significant impact. Hence, we conclude that the forward-model correction by itself could be able to explain the discrepancy with the experimental values. A greater dataset might allow for a more precise assessment of the difference between the two approaches.

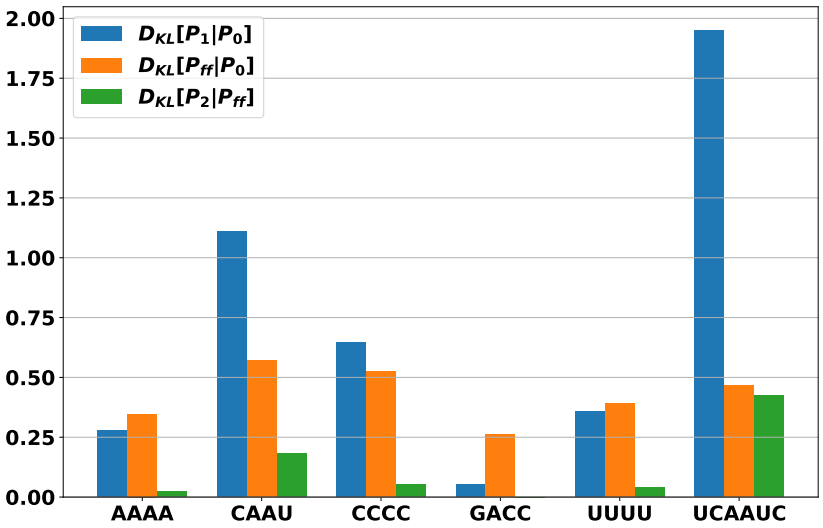


Figure 3.13: Comparison of the Kullback-Leibler divergences for different molecular systems when performing E+FM refinement (P_1) or the fully combined approach, with P_{ff} the force-field correction ensemble and P_2 the final ensemble in the loss function (Eq. 2.1).

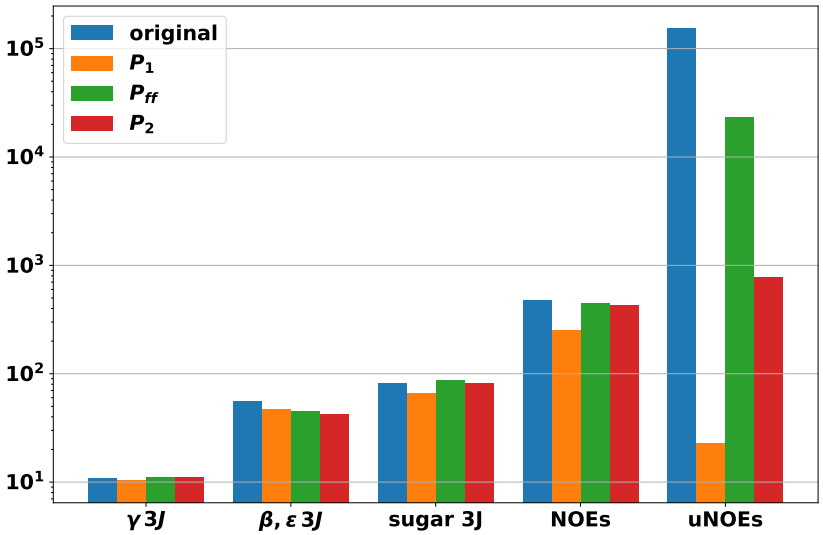


Figure 3.14: Comparison of the χ^2 for different experimental measures when performing E+FM refinement (P_1) or the fully combined approach, with P_{ff} the force-field correction ensemble and P_2 the final ensemble in the loss function (Eq. 2.1) (sum over different molecular systems).

3.3 Discussion

The integration of MD-generated trajectories with experimental measures is useful to both improve the accuracy of MD simulations and to provide a microscopic interpretation to experimental data. Specifically, ensemble-refinement methods can be used to minimally modify MD-generated ensembles to match experiments that report averaged information. Several variants of this idea have been proposed and used in the past years, based on the idea of refining either ensembles, force-field parameters, or forward models used to back-calculate experimental data. However, a general tool to perform these types of analysis was still lacking.

We here developed the Python package `MDRefine`, which can be downloaded and installed from https://pypi.org/project/MDRefine and provides a flexible implementation of the methods above. `MDRefine` consists of a number of functions, starting from standard calculations (relative entropy, χ^2) to more elaborate ones (minimization of the loss function and automatic search of optimal hyperparameters). It has been developed with a modular structure. The full input data is represented by an object `data`, that has two main attributes: `data.properties`, containing global properties, and `data.mol`, a dictionary whose values are instances of `data_class`. Each `data_class` instance stores all the information related to a specific molecule. `MDRefine` is fairly customizable: the user can define functional forms for the force-field corrections and/or the forward models which are best suitable for his own cases of interest. The output is then returned as a folder with several files including optimal hyperparameters and coefficients, χ^2 and relative entropies, refined weights, and other relevant quantities.

We showed how to use `MDRefine` to reproduce data from recently published works [1, 119, 126], where analyses were performed using custom scripts that are more difficult to adapt to new systems. These examples include fitting of partial charges of a modified nucleobase using alchemical simulations and adjusting structural ensembles of RNA oligomers using various combinations of force-field refinement, ensemble refinement, and forward-model refinement. Jupyter notebooks that can be used both to reproduce these analyses and as starting points to analyze new systems are available on GitHub https://github.com/bussilab/MDRefine.

In addition to reproducing existing analyses, we generalized them taking advantage of the flexibility of this new implementation. Specifically, we compared the effect of different regularization strategies, and introduced the idea of tuning the hyperparameters by directly minimizing the validation error. We consider this to be especially valuable in applications where the number of hyperparameters is larger than two and an exhaustive search is

unfeasible.

It is important to recall that `MDRefine` is only designed to analyse and reweight existing trajectories, and not to generate new ones. This results in an important assumption, namely that the modifications required to match experimental measurements are small (gentle) refinements [96, 104]. This assumption holds whenever the original ensemble has a significant overlap with the experiments. For instance, in the case of force-field refinement, as soon as the deviations become higher than a certain threshold (fixed by the relative entropy or the effective sample size, for instance), it is advisable to perform a new simulation with the updated parameters [14, 113, 117, 118, 154, 155]. Also ensemble-refinement strategies based on the maximum-entropy principle can be iterated in this way, by running new simulations where linear restraints are used as biases [156–158]. When the relative-entropy regularization is used, large variational freedom is left to non-informative parameters (namely, those which do not significantly affect the ensemble, in the visited regions). To avoid this effect, the $L2$ regularization can be used as a prior additional to the D_{KL} one [118]. Alternatively, one could also add a relative-entropy regularization on previously simulated molecular systems, even if no experimental data is available for those systems, as proposed in Ref. [159]. Parameters might also be improved on-the-fly, resulting in a much larger statistical significance, at the price of a more complex implementation [19, 116, 136, 149–151]. Finally, replica-based methods where the time average is replaced with a replica average, might also be used [140, 141, 160, 161].

`MDRefine` provides an off-the-shelf implementation of complex state-of-the-art algorithms which allow to refine structural ensembles for complex molecular systems. We hope that proving an open-source and well-documented tool will facilitate the use of these algorithms in the molecular-dynamics community.

Chapter 4

Bayesian sampling of the posterior distribution

In this Chapter, we address the issue of defining a normalizable Bayesian posterior for refinement and ensuring its consistent sampling. This content is largely adapted from our manuscript *Bayesian ensemble refinement revisited: the entropy matters*, which is currently in its final stages of preparation.

As widely described in the previous Chapters, structural biology increasingly relies on integrative methods, where optimal solutions are sought by minimization of suitable loss functions. Casted in a Bayesian framework, minimizing the loss function corresponds to identifying the ensemble that maximizes the posterior distribution (its mode). However, restricting attention to this single ensemble is limiting, since alternative, less probable models may still carry valuable information, first of all about the uncertainty of the optimal ensemble. To this end, a fully Bayesian treatment would enable sampling of the entire posterior distribution.

A notable step in this direction was presented in the *Bayesian Energy Landscape Tilting* (BELT) framework [98]. Yet, attempting to sample the ensemble-refinement posterior as described there reveals a critical issue: the distribution is not normalizable. Here, we explain the origin of this issue and propose (based on [162, 163]) the use of the Jeffreys uninformative prior as an entropic prior, which restores normalizability and enables consistent sampling of the posterior. The need for an entropic prior in order to correctly tackle the sampling of a hyperensemble (an ensemble defined on a set of probability distributions) is discussed in the work by G. Crooks [164]. We here show its usage in the context of integrative modeling and we provide

We show this problem and its resolution using both a simple toy model and a realistic case study on one of the RNA oligomers discussed in subsection 2.3.1. Finally, we incorporate the computation of Bayesian a posteriori averages with

the Jeffreys prior into the recently developed `MDRefine` package, described in Chapter 3 and in Ref. [142], making this approach directly applicable to practical case studies of ensemble refinement and force-field fitting.

4.1 Methods

When refining structural ensembles or force-field models, we are faced with the minimization of a loss function[1] that is typically given by

$$\mathcal{L}(P) = \frac{1}{2}\chi^2(P) - \alpha S[P||P_0] \tag{4.1}$$

as described in Eqs. 1.12 and 1.17. The ensemble P may be let free to vary among all the possible solutions (for ensemble refinement) or constrained to some functional form (in the case of force-field fitting). We recall that, in ensemble refinement, the optimal solution takes a specific functional form P_λ, which allows the minimization of the loss function $\mathcal{L}$ to be performed on the restricted set of ensembles parametrized by $\vec{\lambda}$ (see Eqs. 1.8 and 1.15). While this restriction is sufficient for the minimization of $\mathcal{L}$, it excludes many ensembles that would otherwise contribute when sampling the full posterior distribution.

In the context of Bayesian inference, such a loss function is interpreted as the (negative) log-posterior, where the relative-entropy term $\alpha S[P||P_0]$ corresponds to the log-prior and the χ^2-term ($-\frac{1}{2}\chi^2$) represents the log-likelihood, assuming the errors to be modeled by a Gaussian distribution. Consequently, $e^{-\mathcal{L}[P]}$ is proportional to the posterior probability of an ensemble P given the initial assumption P_0 and the experimental data:

$$\mathcal{P}(P|P_0,\mathcal{D}) \propto \mathcal{P}(\mathcal{D}|P)\,\mathcal{P}(P|P_0) \propto e^{-\frac{1}{2}\chi^2(P)}\, e^{\alpha S[P||P_0]} = e^{-\mathcal{L}(P)} \tag{4.2}$$

where we have used calligraphic $\mathcal{P}$ for the probability of an ensemble P to distinguish it from the ensemble itself, and $\mathcal{D}$ for the set of experimental data. Hence, minimizing the loss function $\mathcal{L}(P)$ corresponds to determine the max-a-posteriori (MAP) ensemble on the basis of the new experimental measurements.

Whereas in most of the application cases one focuses solely on the ensemble that maximizes the posterior distribution, this is inherently limiting, since other models – even though less probable – are still possible. Going beyond MAP estimate, our aim is to sample the full posterior distribution. This allows

[1] Here, and throughout the following, we use round brackets for $\mathcal{L}(P)$ instead of square brackets, despite the functional dependence, to emphasize the parallelism with $\mathcal{L}(P_\lambda)$, depending on a set of coefficients $\vec{\lambda}$.

us to get the Bayesian expected value and uncertainty for any a-posteriori chosen observable. Formally, this requires computing a sum over the set of all the possible ensembles, each weighted with its probability $\mathcal{P}(P)$:

$$\mathbb{E}[\langle O\rangle_P] = \sum_P \mathcal{P}(P)\langle O\rangle_P = \frac{1}{\mathcal{Z}}\sum_P e^{-\mathcal{L}(P)}\langle O\rangle_P. \tag{4.3}$$

In this way, one can estimate the inherent uncertainty associated to the MAP optimal ensemble within a fully Bayesian treatment, and verify if significant deviations from it are present. Notice that, in the case of ensemble refinement, the loss function $\mathcal{L}(P)$ is convex (see section B.2), so the posterior distribution is unimodal, nonetheless alternative ensembles distinct from the MAP solution are still possible.

Consistently performing this sampling is, at our knowledge, an open challenge, due to the huge variational freedom within the ensemble space, and no unique strategy has yet been established. One possible approach is to sample the posterior over the weights, subject to normalization, as done in the *BioEn* work by J. Köfinger *et al.* [99]. However, this treats changes in weights uniformly; for example, shifting a weight from 0.4 to 0.5 is considered equally significant as changing it from 0.01 to 0.11, which may be considered not appropriate. In addition, Bayesian sampling of the weights suffers from a fundamental limitation: the posterior estimates (including means and variances) are sensitive to how finely the trajectory is discretized.

4.1.1 BELT (Bayesian energy landscape tilting)

A hybrid approach has been proposed in the *Bayesian Energy Landscape Tilting* (BELT) method, by K. A. Beauchamp *et al.* [98]. In this article, it has been shown that, under certain general assumptions (described in the Supporting Information of Ref [98]), one can restrict to a specific functional form, namely, the set $\{P_\lambda\}$ of ensembles parametrized by $\vec{\lambda}$ as described in Eqs. 1.8 and 1.15. In this way, a single ensemble P for each set of average values $\langle\vec{g}\rangle_P$ of the observables will be eventually sampled, the one maximizing the relative entropy from the reference ensemble P_0 at given $\langle\vec{g}\rangle_P$. Afterwards, they propose to sample the posterior $\mathcal{P}(P_\lambda) \propto e^{-\mathcal{L}(P_\lambda)}$ as a function of $\vec{\lambda}$.

While this approach may appear straightforward, it actually reflects a specific choice for how the posterior averages are calculated: namely, by sampling $\mathcal{P}(P_\lambda)$ uniformly as a function of $\vec{\lambda}$. This implies interpreting $e^{-\mathcal{L}(P_\lambda)}$ as proportional to the probability density of ensembles within an infinitesimal volume $d\vec{\lambda}$ around $\vec{\lambda}$. This approach does not capture the correct ensemble density, introducing an inconsistency in the sampling. Indeed, at very high

values of $|\vec{\lambda}|$, the loss function, and thus the posterior, does not change significantly, reaching an asymptotic plateau value, rather than diverging. Consequently, the posterior is not normalizable because the ensembles show minimal variations across these regions, and so we basically end up repeatedly sampling the same ensemble multiple times. This is clear when focusing on a simple toy model, as the quadratic potential shown below.

In this simple case, we can perform an analytical estimate to quantify the difference between the minimum value of the loss function and its lowest asymptotic value. We consider a quadratic potential $V(x) = \frac{x^2}{2\sigma^2}$ where we enforce $\langle x \rangle = x_{exp} \pm \sigma_{exp}$. A numerical example is reported in subsection 4.2.1. The point of minimum is given by eq. 1.15, which, in the limit of high number of frames, is well approximated by the continuum-limit value

$$\lambda^* = -\frac{x_{exp}}{\sigma^2 + \alpha\sigma_{exp}^2} \tag{4.4}$$

(derived from eq. 1.15 by substituting $\langle x \rangle_\lambda = -\frac{\partial}{\partial\lambda} \log Z_\lambda$ with Z_λ the normalization factor defined in Eq. 1.8, approximated by the continuum-limit value $Z_\lambda \to e^{\frac{1}{2}\lambda^2\sigma^2}$). Moreover, the plateau values are given by the frame(s) at which the observable takes its maximum (minimum) value $\bar{x}$, respectively in the two limits $\lambda \to \pm\infty$. In this case, the relative entropy takes the limit value $\log P_0(\bar{x}) \simeq -\log N$, being $P_0(\bar{x})$ the probability to be in frames with $\bar{x}$ for the reference ensemble with N the number of frames, and $\chi^2 \to (\bar{x} - x_{exp})^2/\sigma_{exp}^2$. The expected value for $\bar{x}$, the maximum value among N ones extracted from a normal distribution $\mathcal{N}(0;\sigma)$, can be estimated as

$$x_{max}(N;\sigma) = \sqrt{2}\,\sigma\,\mathrm{erf}^{-1}(1 - 2/N) \simeq \sigma\sqrt{2\log N}, \tag{4.5}$$

so the limit value of the loss function is expected to be

$$\mathcal{L}_{lim} = \frac{1}{2}\left(\frac{x_{max}(N;\sigma) - x_{exp}}{\sigma_{exp}}\right)^2 + \alpha \log N \simeq [(\sigma/\sigma_{exp})^2 + \alpha] \log N. \tag{4.6}$$

Low asymptotic values arise when a small number of frames N is available or the confidence in the prior distribution (α) and in the experimental measurement (σ/σ_{exp}) is low, in short all the cases in which the statistics is not robust enough. So, sampling the posterior is ill-defined and in practice, if the asymptote is low enough, one might end up in the plateau regions, repeatedly sampling very similar ensembles multiple times.

4.1.2 Jeffreys uninformative prior

The key issue underlying the ill-defined sampling in BELT is that the density of the ensembles is not captured in the same way independently on the

$\vec{\lambda}$ parametrization. This problem was already recognized in the Bayesian inference literature by Jaynes [163], but has received little attention in the context of methods to integrate molecular simulations and experiments. In particular, if $|\vec{\lambda}|$ is large, it would be convenient to count fewer ensembles for the same step in the $\vec{\lambda}$ space, rather than uniformly with $\vec{\lambda}$, since we are focusing on similar ensembles where all the probability is localized in a few frames (due to the discreteness of the sampling). A natural way to account for the distance between ensembles is through the calculation of the relative entropy. This quantity is already used in the ensemble-refinement framework and has been recently introduced as a regularization for the force-field refinement [118]) for quantifying the closeness between distributions. In the following, we will use it to measure distances between two arbitrarily close ensembles P and $P' = P + \delta P$. This choice has been proposed by Crooks with the definition of thermodynamic length in [164], which addresses the issue of sampling a hyperensemble – namely, an ensemble defined over multiple probability distributions. In this way, the similarity between two very close ensembles P and $P + \delta P$ is

$$\begin{aligned} D_{KL}[P + \delta P \,||\, P] &= \sum_x [P(x) + \delta P(x)] \log \frac{P(x) + \delta P(x)}{P(x)} \\ &= \frac{1}{2} \sum_x \frac{\delta P(x)^2}{P(x)} + O(\delta P^3) \end{aligned} \tag{4.7}$$

where we have used the Taylor expansion of the logarithm to the second order and the fact that $\sum_x \delta P(x) = 0$ due to normalization.

As established in the literature, this choice leads to the Jeffreys "uninformative" prior [162]; specifically, the second-order expansion – defining the squared infinitesimal distance $(ds)^2$ – yields

$$\begin{aligned} (ds)^2 &= \sum_x \frac{dP(x)^2}{P(x)} = \sum_x P(x) \Big(\frac{dP(x)}{P(x)} \Big)^2 = \sum_x P(x) (d \log P(x))^2 \\ &= \sum_x P(x) \sum_{j,k} \frac{\partial \log P(x)}{\partial \lambda_j} \frac{\partial \log P(x)}{\partial \lambda_k} d\lambda_j d\lambda_k = \sum_{j,k} F_{jk}(\vec{\lambda})\, d\lambda_j d\lambda_k \end{aligned} \tag{4.8}$$

where

$$F_{jk}(\vec{\lambda}) = \Big\langle \frac{\partial \log P(x)}{\partial \lambda_j} \frac{\partial \log P(x)}{\partial \lambda_k} \Big\rangle_{P_\lambda} \tag{4.9}$$

is the Fisher information matrix, corresponding to the variance-covariance matrix $C(\vec{\lambda})$ of the observables in the case of ensemble refinement (introduced in Eq. 2.6). Notice that the derivation in eq. 4.8 is quite general and can also

be applied to force-field refinement, where the Fisher matrix is given by the variance-covariance matrix of the derivatives of the force-field correction with respect to its free coefficients $f_j(\vec{\phi}) = \frac{\partial V(\vec{\phi})}{\partial \phi_j}$. In this second case, a bijective (one-to-one) correspondence between the set of parameters and the ensembles has to be enforced to map sampling over P_ϕ onto sampling over $\vec{\phi}$. This is ensured as long as the matrix C is positive definite. This requirement is instead always satisfied in ensemble refinement with independent observables.

Taking equal amount of ensembles for equal length ds as defined in Eq. 4.8, we get the local number of ensembles around $\vec{\lambda}$:

$$dn(\vec{\lambda}) = \sqrt{\det F(\vec{\lambda})}\, d\lambda_1 \dots d\lambda_N \tag{4.10}$$

where the density $\sqrt{\det F(\vec{\lambda})}$ is precisely the Jeffreys "uninformative" prior. Since this is purely a way to count the ensembles, it does not affect the MAP estimate. Rather, it only enters as an entropic contribution when sampling the loss function:

$$\int d\mathcal{P}[P_\lambda] \propto \int e^{-\mathcal{L}[P_\lambda]}\, dn(\lambda) = \int e^{-\mathcal{L}[P_\lambda]} \sqrt{\det F(\lambda)}\, d\lambda \tag{4.11}$$

corresponding to an effective loss function

$$\mathcal{L}'[P_\lambda] = \mathcal{L}[P_\lambda] - \log \sqrt{\det F(\lambda)}. \tag{4.12}$$

For this reason, and to distinguish it from the prior $\alpha D_{KL}[P||P_0]$ – which instead affects the MAP estimate – we refer to this as an *entropic* prior. Note that the flat sampling over $\vec{\lambda}$ is an alternative entropic prior, as it implicitly prescribes a rule for ensemble counting. Two further entropic priors will be introduced in the following section. The sampling of the ensembles with entropic prior has been implemented in the Python function `local_density`, which is provided in the module `bayesian.py` as part of the Python package `MDRefine`.We notice that the usage of the Jeffreys prior has been proposed in [98] with a deeply different meaning, since they intended it as an alternative to the maximum relative entropy principle and not as a purely entropic prior.

4.1.3 Importance of a restricted functional form

In all cases, imposing a functional form is essential to prevent results from depending on the discretization (i.e., the number of frames) of the trajectory. For instance, when focusing on full space of ensembles, without any selected functional form, posterior averages in Eq. 4.3 can be computed as a function

of the weights $w_1 \dots w_N$ as [99]

$$\frac{1}{\mathcal{Z}} \int dw_1 \dots dw_N \, \delta(w_1 + \dots + w_N - 1) \, e^{-\mathcal{L}[\vec{w}]} \, \langle O \rangle_{\vec{w}}. \tag{4.13}$$

where the δ-function enforces the normalization $\sum_i w_i = 1$ and $\mathcal{Z}$ is the corresponding normalization constant. This case corresponds to a flat entropic prior over the unit simplex, namely a Dirichlet distribution $\mathrm{Dir}(\alpha = 1)$. However, if the trajectory is sampled twice as frequently, the number of weights doubles, and this artificially reduces the variance of the observables.

The same happens with the Jeffreys uninformative prior when applied without focusing on any restricted functional form. It is defined by the metric $ds^2 = \sum_x \left(\frac{dP(x)}{\sqrt{P(x)}} \right)^2$ which leads to the integral

$$\int \frac{dw_1}{\sqrt{w_1}} \dots \frac{dw_N}{\sqrt{w_N}} \, \delta(w_1 + \dots + w_N - 1) \, e^{-\mathcal{L}[\vec{w}]} \, \langle O \rangle_{\vec{w}}, \tag{4.14}$$

corresponding to a Dirichlet distribution with parameter $\mathrm{Dir}(\alpha = \frac{1}{2})$.

In both cases, posterior estimates (averages and variances) are sensitive to the discretization of the trajectory. To avoid this artifact, one can restrict to a specific functional form, such as the exponential family P_λ in the case of ensemble refinement or a correction P_ϕ parametrized by some coefficients in force-field fitting.

4.1.4 Alternative ways for ensemble counting

Two further ways to account for the ensemble density have been proposed in the literature. In [112], T. J. Lane *et al.* suggest to sample the posterior uniformly with respect to the average values of the observables $\langle \vec{g} \rangle_\lambda$. This work extends the BELT method, so we refer to this choice of the entropic prior as "BELT2". Somewhat surprisingly, this work has only been published on the arXiv. In addition, one can take inspiration from the *BioEn* method [99] by J. Köfinger *et al.* in order to sample the posterior $\mathcal{L}(P_\lambda)$ by restricting the Euclidean $L2$ distance from the space of weights to the space of the parametrized ensembles. Both these two approaches are discussed in the following, highlighting their strengths and limitations.

BELT 2

In [112], the posterior is treated as a function of the average values of the observables $\mathcal{P}(\langle \vec{g} \rangle_{\vec{\lambda}})$, sampling it uniformly in the $\langle \vec{g} \rangle_{\vec{\lambda}}$ space. This is possible as long as the relation between $\langle \vec{g} \rangle_{\vec{\lambda}}$ and P_λ is one-to-one and so invertible,

which is the case for independent observables (see Eq. 1.15). Then, the infinitesimal distance squared between two ensembles with average values $\langle \vec{g} \rangle_\lambda$ and $\langle \vec{g} \rangle_\lambda + d\langle \vec{g} \rangle_\lambda$ is

$$\begin{aligned} ds^2 &= \sum_i d\langle g_i \rangle_\lambda^2 = \sum_j \Big(\sum_k \frac{\partial \langle g_j \rangle_\lambda}{\partial \lambda_k} d\lambda_k \Big)^2 = \sum_j \Big(\sum_k C_{jk}(\vec{\lambda}) d\lambda_k \Big)^2 \\ &= \sum_j \sum_{k,k'} C_{jk}(\vec{\lambda}) C_{jk'}(\vec{\lambda}) d\lambda_k d\lambda'_k \end{aligned} \tag{4.15}$$

where $C(\vec{\lambda})$ is the variance-covariance matrix of the observables, the same appearing within the Jeffreys prior. The local number of ensembles in a volume $d\vec{\lambda}$ centered around $\vec{\lambda}$ is given by the volume transformation as

$$dn(\lambda) = \sqrt{\det C^T(\vec{\lambda}) C(\vec{\lambda})}\, d\lambda_1 \dots d\lambda_N = \det C(\vec{\lambda})\, d\lambda_1 \dots d\lambda_N \tag{4.16}$$

and so we sample $\mathcal{P}(P_\lambda)$ by moving in the $\vec{\lambda}$ space as described by

$$\sum_{P_\lambda} \mathcal{P}(P_\lambda) = \int \mathcal{P}(P_\lambda)\, dn(\lambda) = \int \mathcal{P}(P_\lambda) \det C(\vec{\lambda})\, d\lambda_1 \dots d\lambda_N. \tag{4.17}$$

However, the bijective correspondence between the average values $\langle \vec{g} \rangle$ and the ensembles P holds for the restricted set $\{P_\lambda\}$ parametrized by $\vec{\lambda}$ as in Eq. 1.8, but not in general. For example, in force-field refinement, different values of the correction parameters $\vec{\phi}$ (and thus different ensembles P_ϕ) can correspond to the same average values $\langle \vec{g} \rangle_\phi$. This degeneracy introduces a critical issue when integrating over $d\langle g \rangle$, as it assigns zero measure to entire sets of distinct ensembles that share the same averages. While this choice for the local density of ensembles solves the issue of normalization of the posterior distribution, it does not capture the variability of the ensembles by itself, since it relies instead on the variability of the average values $\langle \vec{g} \rangle$.

Restricted Dirichlet

Finally, in the BioEn framework [99], the authors consider the posterior as a function of the weights of each frame $\mathcal{P}(w_1, \cdots, w_N)$ and then sample $\mathcal{P}$ uniformly with respect to them. This is an alternative way to measure the ensemble density, given by the Euclidean distance in the space of the normalized weights (a *unit simplex*). Within this counting of ensembles, the Dirichlet distribution $\mathrm{Dir}(\alpha = 1)$ is the uniform one, so we will refer to this choice as the "Dirichlet" one.

Restricting to the set of ensembles parametrized by the $\vec{\lambda}$ coefficients as in the previous cases (Eqs. 4.8, 4.15), the infinitesimal distance squared results

$$ds^2 = \sum_x (dP(x))^2 = \sum_x \Big(\sum_j \frac{\partial P(x)}{\partial \lambda_j} d\lambda_j \Big)^2 = \sum_{j,k} M_{jk}(\lambda)\, d\lambda_j d\lambda_k \tag{4.18}$$

where the matrix M is

$$\begin{aligned} M_{jk}(\lambda) = \sum_x \frac{\partial P_\lambda(x)}{\partial \lambda_j} \frac{\partial P_\lambda(x)}{\partial \lambda_k} = \sum_x \Big[& g_j(x) g_k(x) + \langle g_j \rangle_\lambda \langle g_k \rangle_\lambda \\ & - \left(\langle g_j \rangle_\lambda\, g_k(x) + \langle g_k \rangle_\lambda\, g_j(x) \right) \Big] P_\lambda(x)^2 \end{aligned} \tag{4.19}$$

in the case of ensemble refinement, while g_j has to be substituted with the $f_j(\vec{\phi})$ variables for force-field refinement. The main difference with the Jeffreys prior is that we are now focusing on absolute variations $\delta P(x)^2$ rather than relative ones $\frac{\delta P(x)^2}{P(x)}$, and the latter can better represent the variability of the ensembles. For instance, we expect that an increase of 0.1 in the probability to be in a specific frame has a different effect if the probability to be there is 0.01 or 0.6. For this reason, the matrix M_{jk} 4.19 differs from the Fisher matrix 4.9 by a factor $P_\lambda(x)$, which in the continuum limit (when the no. of frames is infinite and their sum has to be interpreted as an integral) makes $M_{jk}(\lambda) = 0$. Focusing on the uniform distribution $P(x) = 1/N$ where N is the no. of frames, the matrix M evaluated there depends on the no. of frames as $M_{jk} = F_{jk}/N$, where F is the Fisher matrix. Hence, it is $M_{jk}(\lambda) \simeq C_{jk}(0)/N \to 0$ for high number of frames N.

4.2 Results

The results discussed in the following sections are available in the GitHub folder `https://github.com/IvanGilardoni/Jeffreys_prior` (Jupyter Notebooks `main_notebook_1.ipynb`, `main_notebook_2.ipynb` and `main_notebook_3.ipynb`).

4.2.1 Quadratic potential

Here, we consider a model system with a single degree of freedom subject to a quadratic potential $V(x) = \frac{x^2}{2\sigma^2}$. Assuming the thermal energy $k_B T = 1$, this corresponds to x being drawn from a normal distribution $x \sim \mathcal{N}(0, \sigma^2)$. We then complement this model with an experimental measurement reporting $\langle x \rangle = x_{exp} \pm \sigma_{exp}$ (see Fig. 4.1a). According to the maximum-entropy principle, the resulting ensemble will also be a Gaussian with the same variance σ^2 and

an average that depends linearly on the λ coefficient, both in the discrete and continuous cases. When a reweighting procedure is applied to a finite sample from the initial ensemble, the average becomes bounded by the largest and smallest sampled values of x (see Fig. 4.1b). Interestingly, for very large absolute values of λ the resulting ensemble becomes independent of λ and collapse to a single point, either the highest or the lowest sampled value of x. The loss function $\mathcal{L} = \frac{\chi^2}{2} - \alpha S$ will also be bounded and display two plateau values in the limit $\lambda \to \pm\infty$ (see Fig. 4.1c). This behavior is not problematic for methods based on the loss-function minimization, with the exception of cases where the data are not compatible with the original trajectory (see subsection 3.1.6). However, the posterior distribution $\mathcal{P}(\lambda) \propto e^{-\mathcal{L}(\lambda)}$, sampled in the BELT method, does not converge to zero when $\lambda \to \pm\infty$ and therefore cannot be normalized (see Fig. 4.1d).

Several solutions have been tested in this work. Including Jeffreys uninformative prior results in an ensemble density that decreases when $\lambda \to \pm\infty$. As a consequence, the negative logarithm of the posterior distribution (Jeffreys, $\mathcal{L}_1$ in Fig. 4.1) diverges fast enough that the posterior distribution becomes normalizable. A similar, though not identical, effect can be obtained by counting the ensembles either through the average value of the observable (BELT2, $\mathcal{L}_2$) or based on the variability of the weights (Dirichlet prior, $\mathcal{L}_3$). The corresponding distributions are shown in Figs. 4.1c–d.

Whereas the exact choice of the ensemble density affects the posterior distribution in detail, a major effect arises when comparing cases where the posterior can or cannot be normalized. In Figs. 4.2a–c we report the results of a Monte Carlo sampling of either $P(\lambda) \propto e^{-\mathcal{L}(\lambda)}$ (BELT) or $P(\lambda) \propto e^{-\mathcal{L}(\lambda_1)}$ (Jeffreys prior). When using BELT, λ spends a significant amount of time in the plateau region with $\lambda < 0$, which corresponds to an average value $\langle x \rangle_\lambda$ equal to the maximum value of x observed in the initial trajectory. This region corresponds to a large loss value $\mathcal{L} \approx 4.7$. In contrast, when counting the ensemble density using the Jeffreys prior, λ explores a narrower region around zero, where the loss function $\mathcal{L}$ is typically smaller.

The ensemble density for the four tested priors is reported in Fig. 4.2d (and in Fig. 4.4 for multiple choices for the stride with which the MD trajectory is saved). Here it can be seen that the Jeffreys and BELT2 priors are related and have a maximum at $\lambda \approx 0$. The Dirichlet prior instead has a pronounced local minimum at $\lambda \approx 0$. This can be interpreted as a consequence of the nonlinear mapping between λ and the ensemble weights: although the Dirichlet prior with $\alpha = 1$ is uniform over the weight simplex, it induces a non-uniform distribution over λ, with a local minimum at $\lambda \approx 0$, as described in section 4.1.4. This implies that flat ensembles (close to the unbiased case) occupy a smaller volume in weight space, and are thus less likely under this

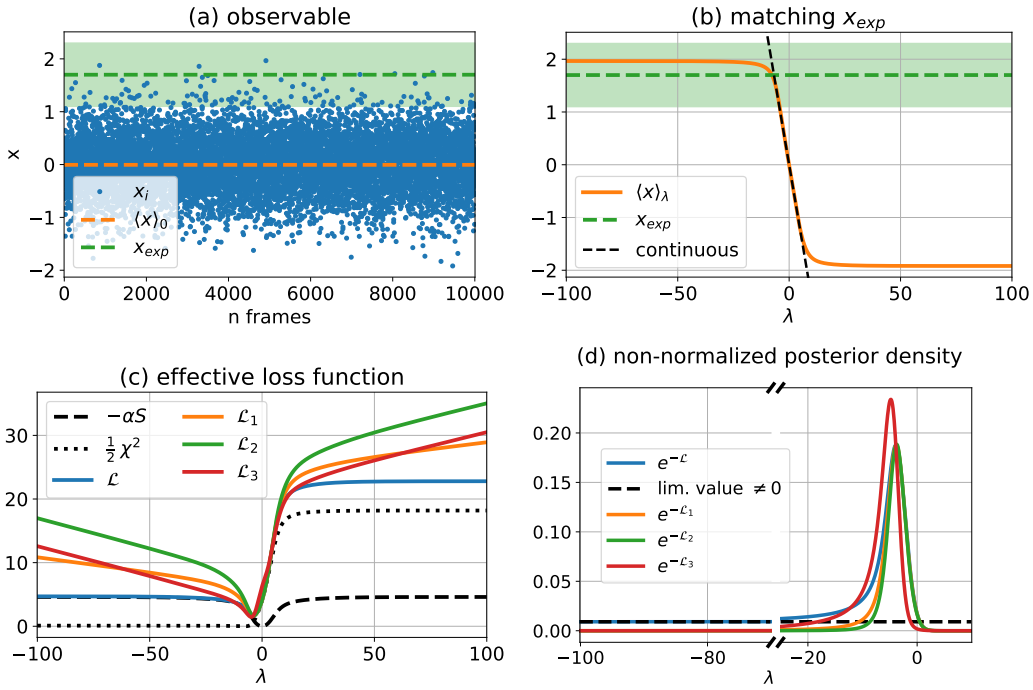


Figure 4.1: (a) Random values of the position sampled from a Gaussian distribution $\mathcal{N}(0; 0.5)$ and experimental value $x_{exp} = 1.7 \pm 0.6$. (b) Average position over the ensemble P_λ, matching the experimental value for $\lambda \simeq -7.7$. (c) Loss function $\mathcal{L}$ for $\alpha = 0.5$, given by $\mathcal{L} = \frac{1}{2}\chi^2 - \alpha S_{rel}$, the depth of the potential well is $\Delta\mathcal{L} \simeq 2.99$. We also include the effective loss function with the entropic contribution given by the entropic priors ($\mathcal{L}_1$ for the Jeffreys prior $-\log\sigma$, $\mathcal{L}_2$ for the BELT2 prior $-\log\sigma^2$, $\mathcal{L}_3$ for the Dirichlet prior). (d) Non-normalized density of the posterior: comparison among including or not any entropic contribution (notice that in the case of plain loss function the posterior has a non-zero plateau value marked by a dashed line).

prior.

Since the posterior distribution in the BELT method cannot be normalized, the exact result depends on the number of Monte Carlo steps performed on λ. For a given number of steps, the severity of this issue depends on the ratio between the posterior distribution in its maximum and in the highest of the two plateaus. The logarithm of this ratio can be interpreted as the height of an energy difference that should be overcome to reach the plateau region. Since the plateau region corresponds to weights where a single point of the initial trajectory contributes, the relative entropy between this ensemble and the uniform ensemble depends on the number of points sampled in the original trajectory. We here quantified this dependency on the quadratic

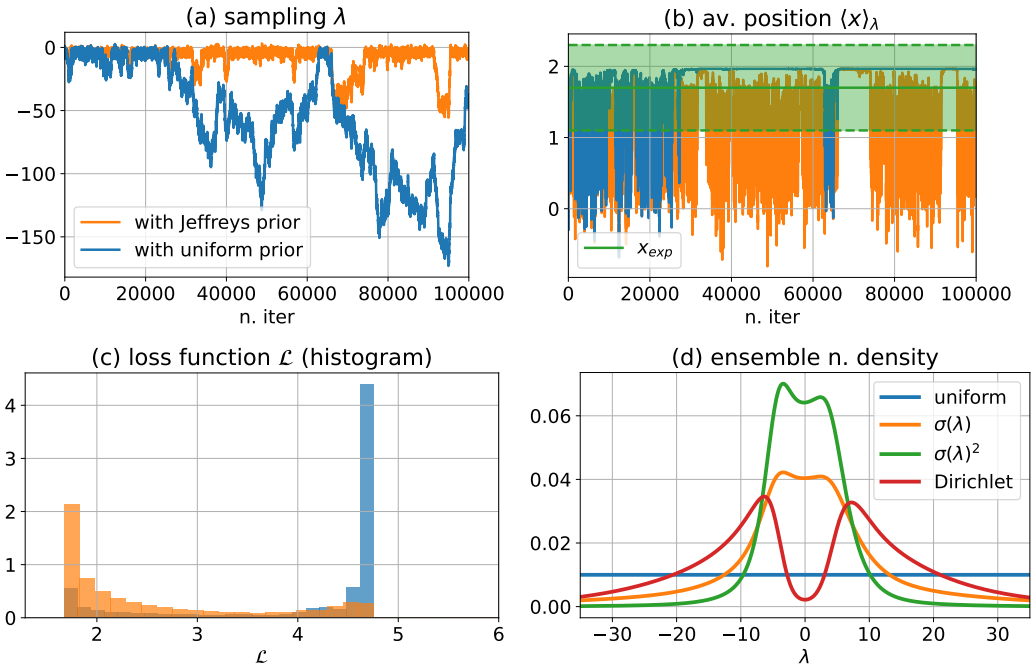


Figure 4.2: (a, b) Metropolis sampling of the posterior distribution reported in Fig. 4.1. Including or not the entropic contribution given by the Jeffreys prior changes the result significantly: without it, the sampling spends some time intervals stuck in the plateau region – horizontal lines in (b). This alters the Bayesian a-posteriori average of $\langle x \rangle_\lambda$. (c) Histogram of the loss function $\mathcal{L}$ (without entropic contribution) in the two samplings. (d) Density of the ensemble number for all the analysed entropic priors.

model introduced above (see eq. 4.6). Figure 4.3 reports how the contributions to the energy difference that should be overcome in the Monte Carlo procedure scale with the number of points sampled in the original trajectory. Both empirical values obtained sampling multiple trajectories from a Gaussian distribution and an analytical estimate derived in subsection 4.1.1 are reported. As it can be appreciated, the analytical estimate properly reproduces the empirical value of the energy difference, at least in this simplified case.

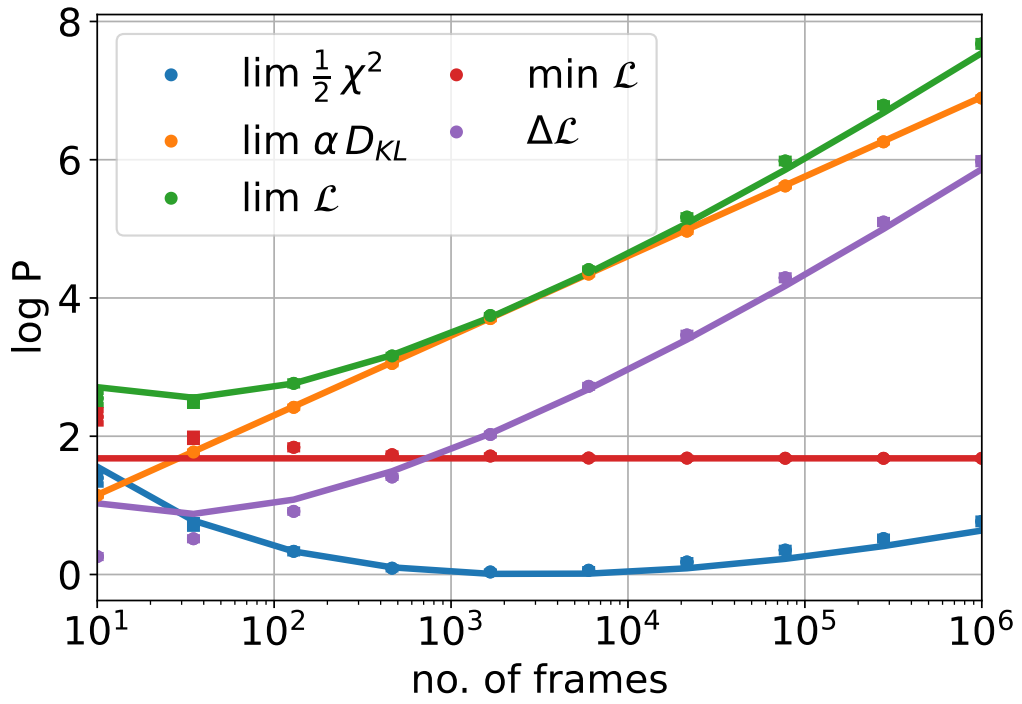


Figure 4.3: Scaling of the main quantities used to derive the depth $\Delta\mathcal{L}$ of the loss function with the number of frames, for the same choice of α, g_{exp}, σ_{exp} and distribution for $\{g_i\}$ used in Fig. 4.1. The dots are the averages over 50 random samplings of $\{g_i\}$, which interpolate well the corresponding lines given by the analytical scaling laws in Eqs. 4.4 and 4.6 for $\min\mathcal{L}$ and $\lim\mathcal{L}$. The discrepancies in $\min\mathcal{L}$ and $\Delta\mathcal{L}$ for low number of frames are because the continuum-limit approximation used for eq. 4.4 is not valid.

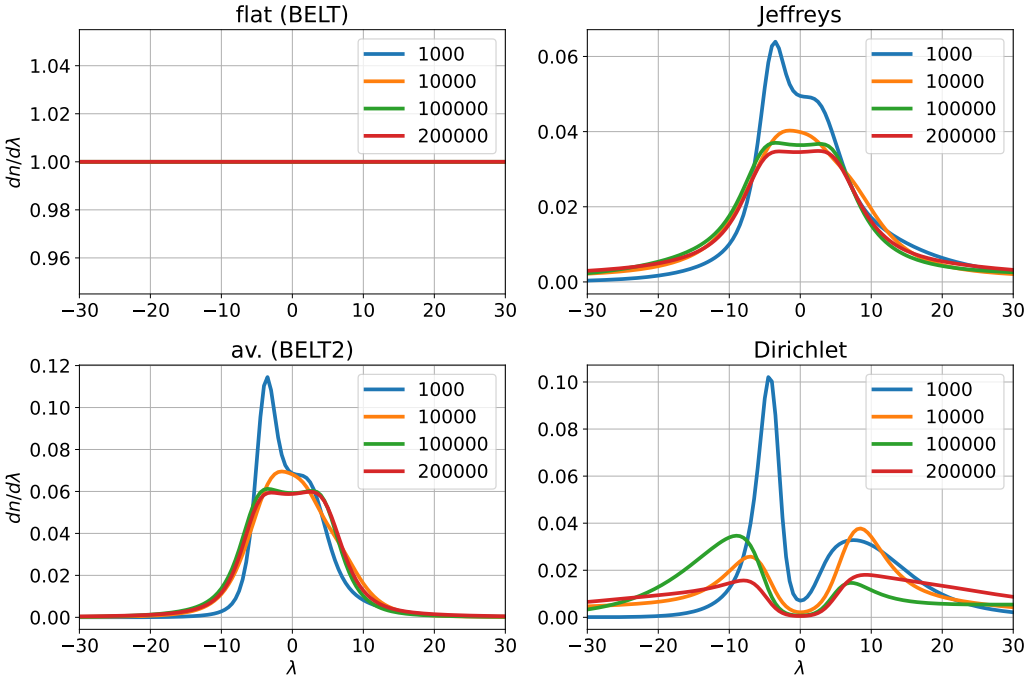


Figure 4.4: Entropic priors for the quadratic model with different lengths (no. of frames) in the analyzed time series, reported in the legend.

4.2.2 Realistic case: refinement of RNA oligomers

As a realistic case, we consider the refinement with maximum entropy of the structural ensemble of a (r)AAAA oligomer using ^3J scalar couplings, associated to the backbone $\beta, \gamma, \epsilon, \nu$ dihedral angles, for a total of 28 experimental values. We first restrict to enforcing only 2 experimental data points to better visualize the posterior distribution (two ^3J scalar couplings related to the γ dihedral angles, for instance). In Fig. 4.5(a–b) we report the posterior distribution for the two λ_i parameters using either the original BELT approach or the Jeffreys prior. In the first case, the posterior is not normalizable. The lowest asymptotic value can be reached for $\vec{\lambda}$ proportional to the vector approximately shown with a dashed line in Fig. 4.5, corresponding to $\Delta\mathcal{L} \simeq 78$ higher than the minimum value. This results in a Monte Carlo trajectory of $\vec{\lambda}$ that samples too much the region where $\lambda_0, \lambda_1 \gg 0$ (Fig. 4.5c). The Jeffreys prior instead samples a much narrower region (Fig. 4.5d). The different range of explored $\vec{\lambda}$ coefficients is then reflected in a different exploration for the average value of the observables (Fig. 4.5e–f and Fig. 4.6). We here focused on a value for the hyperparameter $\alpha = 10$ lying in the optimal range (determined in Chapter 2) so the effect of missing normalization is not as striking as in the case of a lower hyperparameter value, like $\alpha = 1$, for which we get the histograms shown in Fig. 4.7.

For the same system, we then perform a maximum-entropy reweighting where all the available 28 experimental values of the ^3J scalar couplings are simultaneously restrained. We repeat multiple Monte Carlo samplings of the posterior distribution with hyperparameter $\alpha = 10$ (same order of magnitude of the optimal α hyperparameter derived in Chapter 2 – see Fig. 2.2). We run different tests where the analyzed MD trajectory is sampled at different strides with all the four entropic priors described above. Then, we compute average values and variances of the observables and of the Kullback-Leibler divergence, removing an initial equilibration phase in the Monte Carlo procedure. The results are shown in Figs. 4.8 and 4.9–4.10.

In Fig. 4.8a we focus on a single observable, the no. 8, corresponding to the β dihedral angle C3′(0) – O3′(0) – P(1) – O5′(1) between the residues 0 and 1, for instance. This observable was chosen because the average value over the MD trajectory is significantly different from the experimental value (outside the experimental uncertainty). It is then representative of a case where the original trajectory is not in good agreement with the experimental value. While the average values computed with the Jeffreys or the BELT2 entropic priors (both derived from the variance-covariance matrix of the observables, as described in 4.1) are quite constant for choices of the stride over several orders of magnitude, this is not the case for the average values

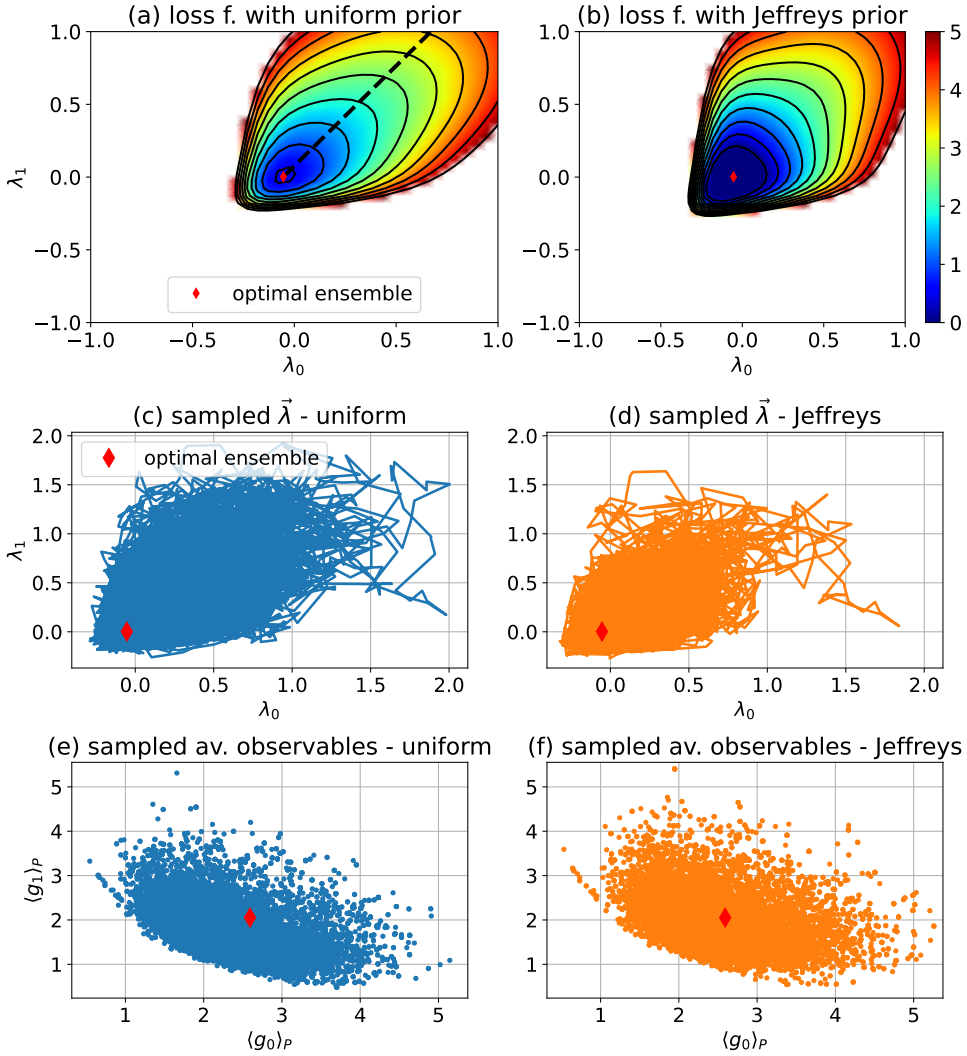


Figure 4.5: Realistic case of refinement for RNA oligomers (stride = 100, corresponding to 10170 frames in MD trajectory; hyperparameter $\alpha = 10$). Scatter plots of (a) the (plain) loss function $\mathcal{L}(\vec{\lambda})$ and (b) the loss function with Jeffreys prior $\mathcal{L}(\vec{\lambda}) - \frac{1}{2}\log\det C(\vec{\lambda})$; a dashed line in (a) indicates the approximate direction of the lowest plateau value. Notice that the optimal ensemble (red diamond) is always in the same position (the optimization does not depend on the ensemble counting). (c–d) An example for the Metropolis sampling in the $\vec{\lambda}$ space: including the Jeffreys prior allows to correctly take into account the ensemble density. (e–f) The observables corresponding to the sampling shown in (c–d); the red diamond indicates the average values of the observables for the optimal ensemble, determined by MAP, independent on the adopted entropic prior.

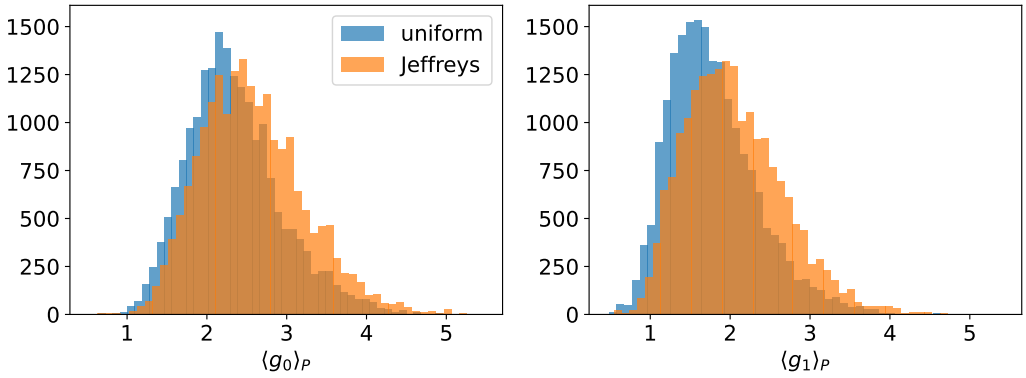


Figure 4.6: Histograms of the sampled average values of the observables corresponding to Fig. 4.5.

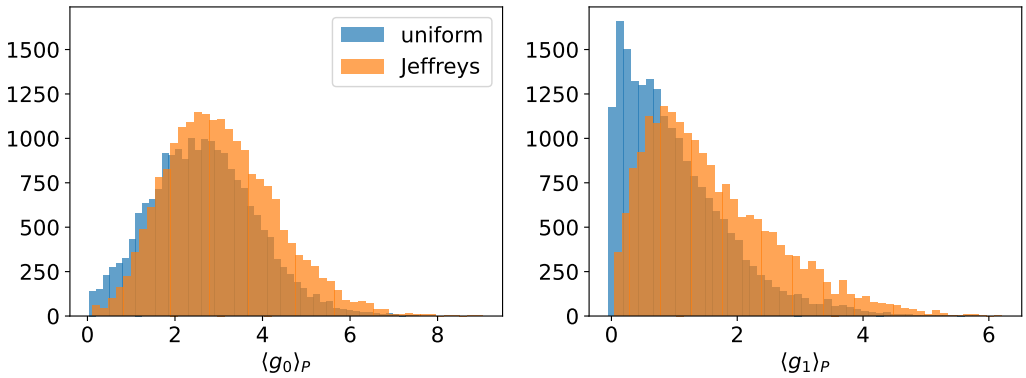


Figure 4.7: Histograms of the sampled average values of the observables corresponding to Fig. 4.5 for $\alpha = 1$ (same stride for the MD trajectory).

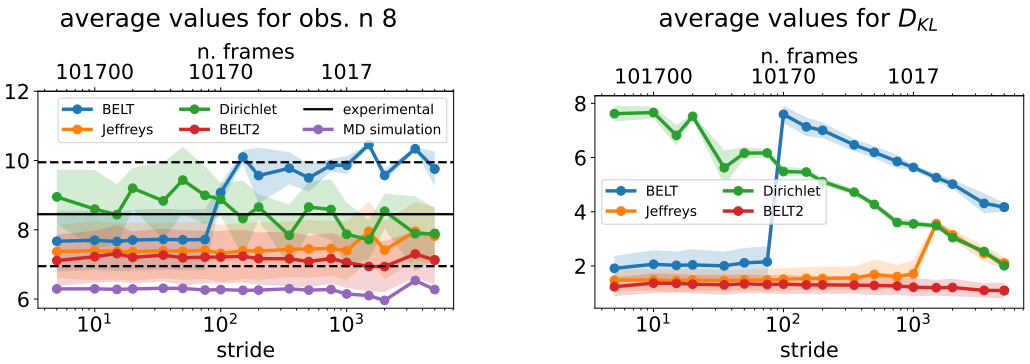


Figure 4.8: Average values of a given observable (no. 8) and of the Kullback-Leibler divergence computed from the sampling of the posterior distribution with multiple entropic priors at different strides of the MD trajectory.

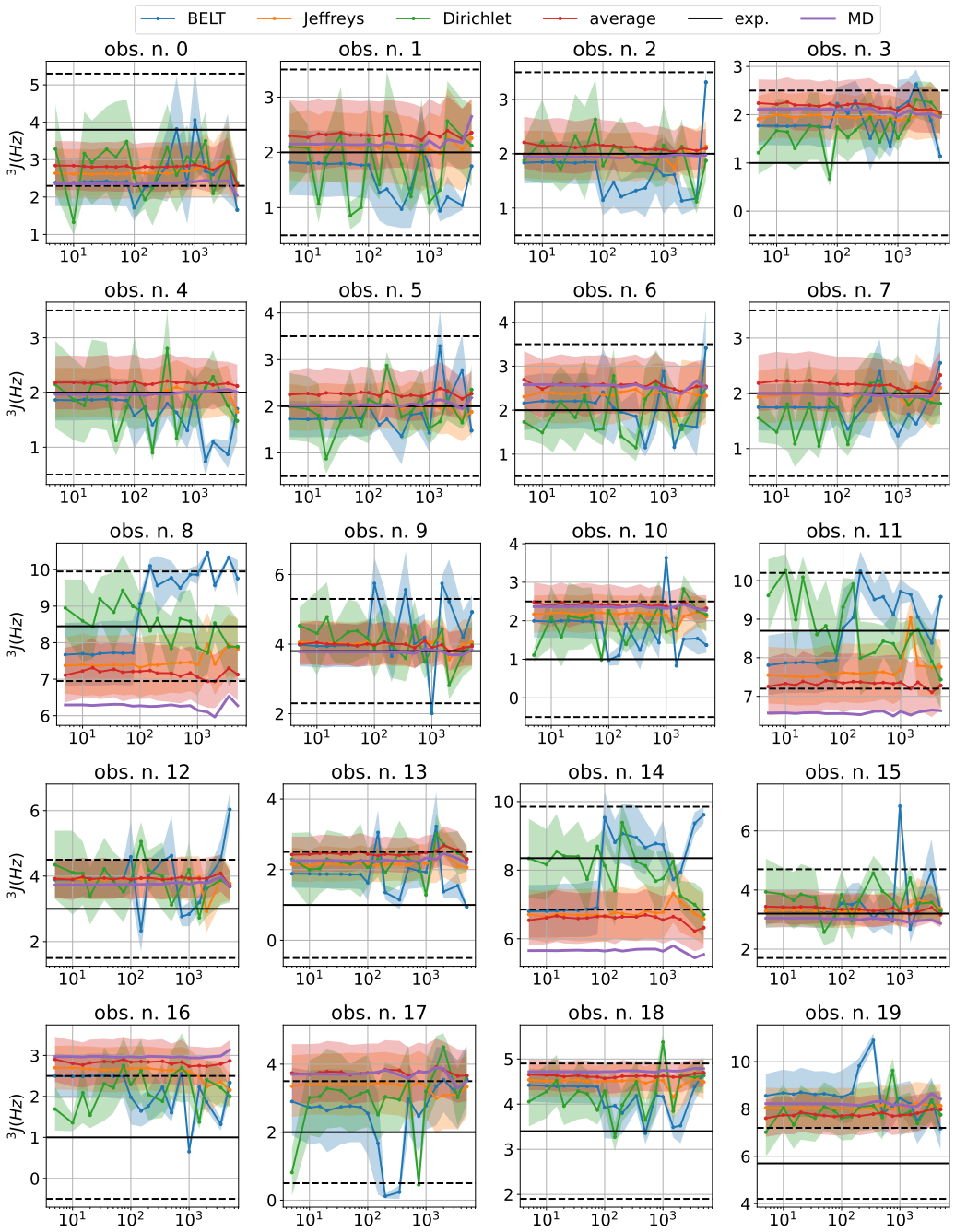


Figure 4.9: Plots corresponding to Fig. 4.8 left for all the sampled observables (first part).

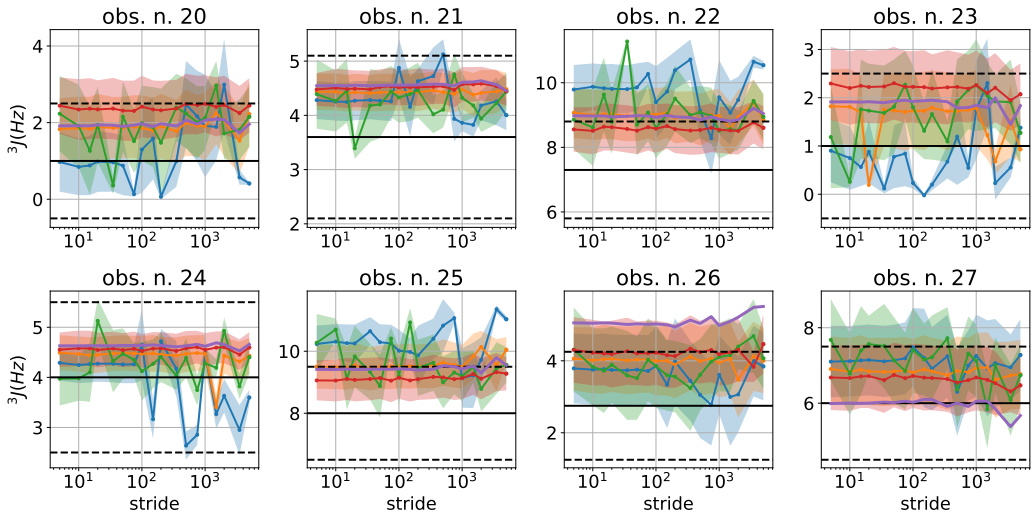


Figure 4.10: Plots corresponding to Fig. 4.8 left for all the sampled observables (second part).

computed with the flat (BELT) and Dirichlet priors. In the case of flat prior, the average value is quite constant and close to the Jeffreys / BELT2 ones for low strides (high number of frames), while it deviates from this value for high strides (moving towards the experimental value). An opposite situation is observed for the average values computed with the Dirichlet prior: in this case, these averages are closer to the experimental value for low strides, and move towards the Jeffreys / BELT2 average values for high strides. This behavior is observed also for other observables (see Fig. 4.9–4.10, observables no. 11, 14, 16, 17, whose MD average lies outside the experimental range), with overall instability effects in the case of Dirichlet and BELT priors.

The average Kullback-Leibler divergence D_{KL} (Fig. 4.8b) confirms the same behavior highlighted by the previous observables. In particular, the average values obtained with the Jeffreys and BELT2 entropic priors are remarkably stable over several orders of magnitude in the stride of the MD trajectory. For the Dirichlet prior, the averages move towards the Jeffreys / BELT2 values as the stride increases, while they are significantly far apart for large numbers of frames in the MD trajectory. This reflects the reduced sampling of the region near the reference ensemble $\vec{\lambda} = 0$ (MD simulation, corresponding to $D_{KL} = 0$), when the number of frames is large (see Fig. 4.4). In contrast, the average values computed with BELT prior exhibit an abrupt deviation from the Jeffreys / BELT2 values when the number of frames becomes sufficiently small to allow exploration of the plateau region in the posterior distribution.

In conclusion, these results show the Jeffreys and BELT2 (average) entropic priors to be quite stable and robust under decimation of the MD trajectory, which is not the case for the flat (BELT) nor the Dirichlet entropic priors.

4.3 Discussion

In summary, we compare different approaches to exploring the space of multiple solutions, going beyond maximum a posteriori (MAP) estimate in the specific context of ensemble refinement, easily extendible to force-field fitting. Most of the previous studies focused on getting the MAP ensemble by minimizing a loss function similar to that in Eq. 4.1. However, fully sampling the posterior distribution, which is a standard practice in Bayesian inference, allows one to estimate errors and correlations on the parameters and uncertainty on the predictions.As described in section 4.1, several sampling strategies have been proposed in the literature [98, 99, 112]. Despite their differences and limitations, no consensus exits on a single approach, leaving ambiguity about which entropic prior should be chosen for the posterior sampling.

Bayesian sampling of weights is one possible method [99], but the results depend on the number of discrete states (frames) used to represent the system. For example, saving trajectories at lower frequency increases the dimension of the weight vector, which tends to reduce the variance of observables in the posterior. This challenge is related to the issue of estimating the number of independent data points in an MD simulation that, to our knowledge, is still open. Notably, the Dirichlet prior tested here differs from that in [99], where weights on individual snapshots are sampled without the $\vec{\lambda}$ parametrization.

An alternative hybrid approach that does not suffer this problem is employed in BELT and BELT2 [98, 112]. These methods first select the distribution that maximizes the relative entropy under fixed average values of the observables, then perform Bayesian sampling of these optimal ensembles (one per each $\langle \vec{g} \rangle_P$), parametrized by the $\vec{\lambda}$ coefficients. However, choosing a suitable entropic prior in the $\vec{\lambda}$ space remains non-trivial. We here shown that the simplest choice, namely a uniform / flat prior on $\vec{\lambda}$ (as used in BELT [98]), yields a non-normalizable posterior, which could create numerical instabilities during sampling. This is because the loss function (Eq. 4.1) exhibits plateaus for large parameter values, an effect more pronounced in short MD trajectories. Alternatively, in BELT2 the sampling is performed uniformly over the average observable values $\langle \vec{g} \rangle_P$. While this guarantees a normalizable posterior, it does not directly capture ensemble variability, since distinct ensembles can share similar observable averages. This limitation is

somewhat mitigated here by restricting to the ensembles parametrized by $\vec{\lambda}$.

We propose here, for the first time in this context, the use of the Jeffreys prior. Like BELT2, this entropic prior remains robust under multiple strides of the MD trajectories. Unlike BELT2, however, it explicitly accounts for the ensemble density, making it directly applicable to force-field refinement without encountering further issues due to the missed variability of the observables.

We tested all these four entropic priors in reweighting a realistic MD simulation of RNA oligomers against NMR data, finding remarkable differences. BELT and Dirichlet methods led to results strongly dependent on the sampling frequency of the trajectory (namely, the number of frames). Conversely, methods where the ensemble density is estimated from the covariance matrix of the observables (BELT2 and Jeffreys) were more robust in this respect.

The probability of escaping from the main minimum in the negative log likelihood can be interpreted as an energy difference. This difference increases with the number of frames sampled from the analyzed MD trajectory. This fact might explain why the role of an appropriate entropic prior over $\vec{\lambda}$ has gone unnoticed in the past. Indeed, MD simulations typically result in a very large number of data points, making this transition difficult to observe, as illustrated by the dependence on trajectory stride in Fig. 4.8.

Moreover, ensemble refinement and force-field refinement share a strong conceptual analogy, which allows the extension of Bayesian posterior sampling from the former to the latter. In this context, the Jeffreys uninformative prior is particularly advantageous due to its invariance under non-linear parameter transformations and different choices of measured observables. Importantly, in force-field refinement, adopting the Jeffreys prior is significantly different from sampling uniformly with respect to the average values of the observables (BELT2), since distinct ensembles can yield similar average values of the observables. This approach has been implemented in the `MDRefine` Python package, enabling its direct application to practical case studies.

Conclusions and perspectives

Integrative methods for structural biology – combining experimental data with molecular simulations – have become an essential tool for resolving conformational landscapes of complex biomolecules. A major challenge in this field is the fact that errors arising from different sources (structural ensembles, force fields, and empirical forward models) have always been treated separately from each other, despite they mix together when comparing experiments with simulations. This Thesis addresses this issue by introducing a framework that enables their simultaneous optimization, thereby producing more robust and physically consistent models for interpreting experimental observations with MD simulations.

To this end, we developed `MDRefine`, a versatile Python package for the joint refinement of structural ensembles, force fields, and forward models in a customizable manner. This approach was demonstrated on benchmark applications, including RNA oligomers analyzed via NMR experiments and chemically-modified RNA molecules investigated with denaturation experiments and alchemical free-energy calculations. The framework supports user-defined force-field corrections and forward models. Also, its hyperparameter selection is automated through efficient cross-validation strategies that prevent overfitting and ensure generalizability with reduced computational cost. The modular and open-source nature of `MDRefine`, complemented by extensive documentations and tutorials, ensure that it is accessible and extensible for the broader computational biophysics community.

While this work focuses on maximum a posteriori (MAP) estimates, a natural next sep is full uncertainty quantification by sampling the posterior distribution. Leveraging the Jeffreys uninformative prior allows rigorous determination of confidence intervals for refined parameters and ensembles, providing deeper insight into the reliability of the refined models.

By developing and releasing a unified refinement framework for integrating experimental data into MD simulations, this Thesis advances the toolkit of integrative structural biology. This is achieved by explicitly taking into account different sources of error – whether inaccuracies in the force field,

structural ensembles, or empirical forward models – thereby enabling more accurate and reliable characterization of complex biomolecular systems.

Building on this foundation, several promising directions emerge:

- on-the-fly simultaneous refinement [19, 99, 116, 140, 141]: it enables dynamic updating of force fields and ensembles during MD simulations, moving beyond plain reweighting of precomputed MD trajectories (which is the main limitation of this Thesis); this remains a technically challenging but promising goal, with initial work performed in collaboration with Davide Di Vora (Università degli Studi di Trieste) as part of his master Thesis;

- application to larger systems: the method presented in this Thesis can be used to investigate complex biomolecular assemblies, such as ribozymes and intrinsically disordered proteins; integrating experimental data with MD simulations in these systems may uncover unexpected behaviors that are not observed in simpler cases like RNA oligomers;

- extension to a broader range of experimental data beyond ensemble averages, including multiple snapshots – as in the case of cryo-EM [165] – or kinetic observables [120, 121, 166, 167];

- advanced Bayesian inference techniques, incorporating more sophisticated model selection and uncertainty quantification approaches [168] to further improve predictive power and robustness.

Appendix A

Kish sample size vs. entropy

Let's consider a sample $\{x_1, ..., x_n\}$ (assumed to be uncorrelated) with corresponding normalized weights $\{w_1, ..., w_n\}$. If the weights are uniform, then the effective sample size is simply given by the number of points $n_{eff} = n$. Otherwise, we can obtain a good estimation of the effective sample size by using the Kish sample size, defined as the inverse of the average weight

$$n_{eff} = K[\{w_i\}] \coloneqq \langle w \rangle_w^{-1} = \Big(\sum_i w_i^2 \Big)^{-1}, \tag{A.1}$$

This definition of n_{eff} reduces to $n_{eff} = n$ if each sample contributes with uniform weight, while it returns the number of "dominant" frames if some of them count much more than the others. For example, if we have a non-normalized set of weights $\{1, ..., 1, N, ..., N\}$ with weights 1 and N repeated $n - m$ and m times, respectively, then the effective sample size is $n_{eff} \simeq m$ if $n - m \ll Nm$ (the subset of small weights $w_i = 1$ dominates over the other) and $n_{eff} \simeq n - m$ in the opposite limit. Clearly, it is always $n_{eff} \leq n$ and the equality holds only for uniform weights; it is also $n_{eff} > 1$ and small values are achieved when one frame contributes much more than the others.

The relation with the entropy S is straightforward. It suffices to change the weighted arithmetic average $\langle ... \rangle$ to the weighted geometric one in eq. (A.1) to get

$$e^{S[\{w_i\}]} = \Big(\prod_i w_i^{w_i} \Big)^{-1}, \tag{A.2}$$

which is always in between $1 < e^{S[\{w_i\}]} \leq n$. Hence, the effective number of frames can be estimated also through $e^{S[\{w_i\}]}$, in a way which is expected to be less sensitive to possible different orders of magnitude of the weights, being a geometric average. Moreover, by the Jensen's inequality, it is always $e^{S[\{w_i\}]} \geq K[\{w_i\}]$ (inequality of geometric and arithmetic average).

The relative entropy S_{rel} can be used to quantify the discrepancy of the distribution P, represented by the set of weights $\{w_i\}$, from P_0, represented by $\{w_{0i}\}$:

$$e^{S_{rel}[P||P_0]} = \Big[\prod_i \Big(\frac{w_i}{w_{0i}}\Big)^{w_i}\Big]^{-1}. \tag{A.3}$$

The relative entropy is always $S_{rel} \leq 0$ and can be very large in absolute value for P very different from P_0, with a lower bound depending on the number of frames[1]. Notice it is the geometric average (over w) of the ratio w/w_0; changing it to the arithmetic average we get the relative Kish size [33]

$$K_{rel}[P|P_0] = \Big\langle \frac{w}{w_0} \Big\rangle_w^{-1}. \tag{A.4}$$

If the reference ensemble is uniform ($w_0 = \frac{1}{n}$ constant), then the relative Kish size (and analogously $e^{S_{rel}[P|P_0]}$) reduces to the ratio n_{eff}/n.

Finally, the Kish size ratio (KSR) for an ensemble P reweighted from P_0 has been defined [119] as the ratio between the two Kish sample sizes of P and P_0, respectively:

$$KSR = \frac{\langle w \rangle_w^{-1}}{\langle w_0 \rangle_{w_0}^{-1}}. \tag{A.5}$$

By following the analogy between Kish size and entropy, which share the same definition but using arithmetic and geometric averages, respectively, the Kish size ratio can be considered analogous to the exponential of the entropy difference $e^{S[P]-S[P_0]}$.

A.1 Effects of coarse-graining

When clustering the configurations, as in the case of dimensionality reduction performed e.g. with histograms, the effective number of frames (estimated through the entropy or the Kish sample size) decreases, namely, the average weight increases. On the contrary, the average relative weight $\langle w/w_0 \rangle_w$ (both arithmetic or geometric average) decreases. This keeps happening at every step until we reach the maximum possible clustering, i.e., a single frame, when clearly the average weight is $\langle w \rangle = 1$ (max. possible value) as well as the average relative weight $\langle w/w_0 \rangle_w = 1$ (min. possible value).

Let's prove these two statements. Let's focus firstly on the effective (absolute) number of frames. In the trivial case, we cluster a uniform distribution by taking a constant number of frames M for each bin: we pass

[1] Taking for instance a uniform distribution over N frames as P_0, the furthest distribution P is the one with all the weight concentrated in a single frame, with $-S_{rel}[P||P_0] = \log N$.

from a set of weights $(\frac{1}{N}\cdots\frac{1}{N})$ to another one $(\frac{M}{N}\cdots\frac{M}{N})$, and so $n_{eff}=N$ to $n_{eff}=N/M$. In the most general case, we have a distribution $P(x)$ clustered to $Q(y)=\sum_{x\in\sigma(y)}P(x)$. Then, the entropy turns out to be $-S[Q]\geq -S[P]$. Indeed, it is

$$\begin{aligned}-S[Q]&=\sum_y Q(y)\log Q(y)=\sum_y\sum_{x\in\sigma(y)}P(x)\ \log\sum_{x'\in\sigma(y)}P(x')\\&\geq\sum_y\sum_{x\in\sigma(y)}P(x)\ \log P(x)=-S[P].\end{aligned}\tag{A.6}$$

The same happens for the inverse Kish size $K[Q]^{-1}\geq K[P]^{-1}$, since $(\sum_i w_i)^2\geq\sum_i w_i^2$.

Now, let's focus on the average relative weight. Let's show that the Kullback-Leibler divergence decreases when clustering: $D_{KL}[Q|Q_0]\leq D_{KL}[P|P_0]$. It results

$$\begin{aligned}D_{KL}[Q|Q_0]&=\sum_y Q(y)\log\frac{Q(y)}{Q_0(y)}=\sum_y\sum_{x\in\sigma(y)}P(x)\log\frac{\sum_{x'\in\sigma(y)}P(x')}{\sum_{x''\in\sigma(y)}P_0(x'')}\\&\leq\sum_y\sum_{x\in\sigma(y)}P(x)\log\frac{P(x)}{P_0(x)}=D_{KL}[P|P_0].\end{aligned}\tag{A.7}$$

This means, the variability between frames is given by the variability between clusters plus the variability between frames inside each cluster (similar to ANOVA). Indeed, it is:

$$\sum_{i\in\alpha}w_i\log\frac{w_i}{w_{0i}}=w_\alpha\log\frac{w_\alpha}{w_{0\alpha}}+w_\alpha D^\alpha_{KL}[w_i|w_{0i}]\tag{A.8}$$

where D^α_{KL} is the DKL for the normalized distribution inside cluster α. Indeed, it is:

$$\begin{aligned}\sum_{i\in\alpha}w_i\log\frac{w_i}{w_{0i}}-w_\alpha\log\frac{w_\alpha}{w_{0\alpha}}&=\sum_{i\in\alpha}w_i\Big[\log\frac{w_i}{w_{0i}}-\log\frac{\sum_{i\in\alpha}w_i}{\sum_{i\in\alpha}w_{0i}}\Big]\\&=\sum_{i\in\alpha}w_i\log\frac{w_i/\sum_{i\in\alpha}w_i}{w_{0i}/\sum_{i\in\alpha}w_{0i}}\end{aligned}\tag{A.9}$$

so

$$D_{KL}[w_i|w_{0i}]=D_{KL}[w_\alpha|w_{0\alpha}]+\sum_\alpha w_\alpha D_{KL}[w_i^\alpha|w_{0i}^\alpha]\tag{A.10}$$

where w_i^α is the normalized distribution of weights w_i inside cluster α. In a step of coarse-graining, when we pass from w_i to w_α, we lose the variability inside each cluster (2nd term in the equation). If the relative weights inside each cluster are not changed, it is $D_{KL}[w_i|w_{0i}] = D_{KL}[w_\alpha|w_{0\alpha}]$.

This inequality Eq. A.7 can be shown for each term summed over y by employing the Jensen's inequality in 2 dimensions. Since $P(x)\log\frac{P(x)}{P_0(x)}$ is a convex function $f(a,b)$ of the two variables $P(x)$, $P_0(x)$, then by the Jensen's inequality

$$f\Big(\frac{1}{M}\sum_i(a_i,b_i)\Big) \leq \frac{1}{M}\sum_i f(a_i,b_i), \tag{A.11}$$

the factors $1/M$ simplifies and it results to be

$$\sum_i P(x_i)\log\frac{\sum_i P(x_i)}{\sum_i P(x_{0i})} \leq \sum_i P(x_i)\log\frac{P(x_i)}{P(x_{0i})} \tag{A.12}$$

(sum over the frames in the same bin $x \in \sigma(y)$, with M their total number). The same happens for the relative Kish size, since $P(x)^2/P_0(x)$ is a convex function of $P(x)$, $P_0(x)$ and so, by simplification of $1/M$ factors, it is

$$\frac{\sum_i P(x_i)^2}{\sum_i P(x_{0i})} \leq \sum_i \frac{P(x_i)^2}{P(x_{0i})}. \tag{A.13}$$

Appendix B

Derivatives of the loss function

B.1 Computation of the derivatives

Let's remind $\mathcal{L}_1$, the loss function with the explicit dependence $\lambda^*(\mu)$

$$\mathcal{L}_1(\mu) = \mathcal{L}(\lambda^*(\mu), \mu) = -\alpha\Gamma(\lambda^*(\mu); \mu) + \beta R_1(\phi) + \gamma R_2(\theta) \tag{B.1}$$

where we introduced the shortcut $\mu = (\phi, \theta)$. The optimal parameters are

$$\mu^* = \arg\min_{\mu}\Big(-\alpha \min_{\lambda} \Gamma(\lambda; \mu) + \beta R_1(\phi) + \gamma R_2(\theta)\Big); \tag{B.2}$$

$$\lambda^* = \arg\min_{\lambda} \Gamma(\lambda; \mu^*). \tag{B.3}$$

In order to search for the optimal coefficients λ^*, μ^*, we rely on the BFGS algorithm (or L-BFGS-B if there are boundaries on λ, due to min, max in the χ^2, like for *unobserved NOEs* data) implemented by `Scipy`. This requires the computation of the derivatives, which we can carry out by leveraging the automatic differentiation performed by `JAX`. However, $\mathcal{L}_1$ depends on μ also through λ^*, namely through the full minimization algorithm employed for Γ. To overcome this issue, because λ^* is a point of minimum for $\mathcal{L}$, the derivative simplifies to

$$\frac{\partial \mathcal{L}_1(\mu)}{\partial \mu_j} = \frac{\partial \mathcal{L}(\lambda, \mu)}{\partial \lambda_k}\Big|_{\lambda^*(\mu)} \frac{\partial \lambda_k^*(\mu)}{\partial \mu_j} + \frac{\partial \mathcal{L}(\lambda, \mu)}{\partial \mu_j}\Big|_{\lambda^*(\mu)} = \frac{\partial \mathcal{L}(\lambda, \mu)}{\partial \mu_j}\Big|_{\lambda^*(\mu)} \tag{B.4}$$

and so it can be implemented via automatic differentiation of `loss_function` with given λ^* passed as `fixed_lambdas`.

Our aim is to compute how the χ^2 in validation, evaluated at the optimal parameters λ^* and μ^* found in training, varies with the hyperparameters α, β, γ, and to compute its derivatives. By employing the chain rule, this

task can be split into two parts: compute separately the derivatives of the χ^2 w.r.t. the optimal coefficients λ^*, μ^* and those of the optimal coefficients λ^*, μ^* w.r.t. the hyperparameters, then multiply opportunely the derivatives. For the sake of clarity, in the following we will use the Einstein notation (sum over repeated index is implicit).

Let's focus firstly on the second part and let's make the dependence on the hyperparameters explicit: our aim is to compute the derivatives of the optimal parameters λ^* and μ^* w.r.t. the hyperparameters α, β, γ. The optimal coefficients are determined by the equations

$$\lambda^*(\mu;\alpha) = \arg\min_{\lambda} \Gamma(\lambda;\mu,\alpha); \tag{B.5}$$

$$\mu^*(\alpha,\beta,\gamma) = \arg\min_{\mu} \mathcal{L}_1(\mu;\alpha,\beta,\gamma). \tag{B.6}$$

In absence of boundaries, the optimal parameters are interior to the domain of the loss function. In such points the gradient is zero:

$$\frac{\partial\Gamma(\lambda;\mu,\alpha)}{\partial\lambda}\Big|_{\lambda^*} = 0 \text{ and } \frac{\partial\mathcal{L}_1(\mu;\alpha,\beta,\gamma)}{\partial\mu}\Big|_{\mu^*} = 0. \tag{B.7}$$

Hence, we can exploit the implicit function theorem and get to the following relations:

$$\frac{\partial\lambda_i^*}{\partial(\mu_j,\alpha)} = -(H_{in}^{-1})_{ik}\frac{\partial^2\Gamma(\lambda;\mu,\alpha)}{\partial\lambda_k\partial(\mu_j,\alpha)} \tag{B.8}$$

$$\frac{\partial\mu_i^*}{\partial(\alpha,\beta,\gamma)} = -(H^{-1})_{ij}\frac{\partial^2\mathcal{L}_1(\mu;\alpha,\beta,\gamma)}{\partial\mu_j\partial(\alpha,\beta,\gamma)} \tag{B.9}$$

where H_{in} is the Hessian of the internal minimization (over λ)

$$H_{in\,ab} = \frac{\partial^2\Gamma(\lambda;\mu,\alpha)}{\partial\lambda_a\partial\lambda_b} \tag{B.10}$$

and H is the Hessian of the minimization over μ

$$H_{ab} = \frac{\partial^2\mathcal{L}_1(\mu;\alpha,\beta,\gamma)}{\partial\mu_a\partial\mu_b}. \tag{B.11}$$

The internal Hessian H_{in} can be directly computed through automatic differentiation of $\Gamma(\lambda;\mu,\alpha)$ by `JAX` (which can be equivalently done for `gamma_function` or for `loss_function`/α with `fixed_lambdas` not `None`), while for the second Hessian H a further calculation is required. Indeed, we have to consider also the dependence of $\mathcal{L}_1$ on μ through λ^*. Consequently, the Hessian results

$$\begin{aligned}\frac{\partial^2 \mathcal{L}_1}{\partial \mu_j \partial \mu_k} &= \Big[\frac{\partial}{\partial \lambda_l}\Big(\frac{\partial \mathcal{L}}{\partial \mu_j}\Big)\Big]_{\lambda^*} \frac{\partial \lambda_l^*}{\partial \mu_k} + \frac{\partial^2 \mathcal{L}}{\partial \mu_j \mu_k} \\ &= +\alpha \frac{\partial^2 \Gamma}{\partial \mu_j \partial \lambda_l} (H_{in}^{-1})_{lm} \frac{\partial^2 \Gamma}{\partial \lambda_m \partial \mu_k} + \frac{\partial^2 \mathcal{L}}{\partial \mu_j \partial \mu_k}\end{aligned} \tag{B.12}$$

where the first term has to be summed over the molecular systems used in cross validation, which are possibly different from those used in training. Similarly, it holds

$$\frac{\partial^2 \mathcal{L}_1}{\partial \mu_j \partial \alpha} = \frac{\partial}{\partial \alpha} \frac{\partial \mathcal{L}(\lambda^*(\mu,\alpha),\mu;\alpha,\beta,\gamma)}{\partial \mu_j} = \frac{\partial^2 \mathcal{L}}{\partial \mu_j \partial \lambda_k} \frac{\partial \lambda_k}{\partial \alpha} + \frac{\partial^2 \mathcal{L}}{\partial \mu_j \partial \alpha}. \tag{B.13}$$

Also these terms can be computed through automatic differentiation.

Then, let's focus on the first part, namely, the derivatives of the χ^2 with respect to the optimal parameters λ^*, μ^*. The full dependence is through the average value of the validating observables $\tilde{g}_i(\theta^*)$ (computed with the optimized forward model θ^*) on the ensemble corresponding to the optimal parameters λ^*, μ^*:

$$\langle \tilde{g}_i(\theta^*(\alpha,\beta,\gamma)) \rangle_{\lambda^*(\mu^*(\alpha,\beta,\gamma);\alpha),\, \mu^*(\alpha,\beta,\gamma)}. \tag{B.14}$$

These derivatives can be directly computed through automatic differentiation. Let's conclude our calculation by applying the chain rule:

$$\frac{\partial \chi^2}{\partial \alpha} = \frac{\partial \chi^2}{\partial \mu_i} \frac{\partial \mu_i^*}{\partial \alpha} + \frac{\partial \chi^2}{\partial \lambda_i} \Big(\frac{\partial \lambda_i^*}{\partial \mu_j} \frac{\partial \mu_j^*}{\partial \alpha} + \frac{\partial \lambda_i^*}{\partial \alpha} \Big) \tag{B.15}$$

$$\frac{\partial \chi^2}{\partial (\beta,\gamma)} = \frac{\partial \chi^2}{\partial \mu_i} \frac{\partial \mu_i^*}{\partial (\beta,\gamma)} + \frac{\partial \chi^2}{\partial \lambda_i} \frac{\partial \lambda_i^*}{\partial \mu_j} \frac{\partial \mu_j^*}{\partial (\beta,\gamma)}. \tag{B.16}$$

Now, each term on the right side of the equations can be computed through automatic differentiation and we just have to multiply them as indicated in order to get the derivatives of χ^2 w.r.t. the hyperparameters α, β, γ. This will facilitate the search for the optimal hyperparameters, without the need for a scan in 3 dimensions, which may be cumbersome. Lastly, the minimization can be sped up considering the fact that one needs to explore several orders of magnitude in the hyperparameter space. It is then convenient to change variable and perform the minimization considering as hyperparameters $\log \alpha$, $\log \beta$, and $\log \gamma$, using once again the chain rule for derivatives $\frac{\partial}{\partial \log \alpha} = \alpha \frac{\partial}{\partial \alpha}$, and analogous expressions for the other hyperparameters.

These calculations can be adapted to the presence of boundaries for the λ parameters (like in the case of *unobserved NOEs*). In these cases, when

the optimal solution is on the boundary $\lambda_j = 0$ for some j, the derivatives of these λ_j parameters w.r.t. α, μ is zero. To prove this statement, let's consider for simplicity the case with the constraint $\lambda_j > 0$ for some j. When the optimal solution is on the boundary $\lambda_j = 0$ it means that, if we could remove the boundary, the minimum of the loss function $\Gamma(\lambda; \alpha, \mu)$ would be inner to the forbidden region $\lambda_j < 0$. Hence, by continuity of the Γ function with respect to α (such as for μ), when varying α (or μ) by an arbitrarily small value, the minimum will still be in the forbidden region and the value of λ_j will still be zero, without changing. This holds for every function $f(\lambda; \alpha, \mu)$ which is continuous in its arguments, independently of its convexity or not. Notice that since the boundaries are in the form of linear inequalities on λ, the function Γ is expected to be convex also on the boundary.

B.2 Sign of the derivatives

In the following we provide a proof for the following claims.

1. the loss function $\mathcal{L}[P]$ in Eq. 1.12 is convex in terms of $P(x)$.
2. when increasing the hyperparameter α in the loss function $\mathcal{L}[P]$ (Eq. 1.12), the χ^2 evaluated at the optimal solution $P^*(\alpha)$ of $\mathcal{L}$ increases, the Kullback-Leibler divergence $D_{KL}[P^*(\alpha)||P_0]$ decreases, and the overall loss function $\mathcal{L}(P^*(\alpha))$ increases with derivative equal to $D_{KL}(\alpha)$; the same happens for force-field refinement (when acting on the hyperparameter β), even when combined with forward-model refinement (β, γ hyperparameters), but non-necessarily when combined with ensemble refinement.
3. the derivative of the χ^2 as a function of the regularization term is given by -2α.

Proof no. 1

Let's compute the Hessian of the loss function $\mathcal{L}[P]$ for an arbitrary number of frames $\mathcal{L}(\{P(x)\})$ and show it to be a positive-definite matrix. The first derivative results

$$\frac{\partial \mathcal{L}(P(x))}{\partial P(x)} = \sum_i \frac{\langle g_i \rangle_P - g_{i,exp}}{\sigma^2_{i,exp}} g_i(x) + \alpha \left[\log \frac{P(x)}{P_0(x)} + 1 \right] \tag{B.17}$$

so the Hessian is

$$H_{xx'} = \frac{\partial^2 \mathcal{L}(P(x))}{\partial P(x) \partial P(x')} = \sum_i \frac{g_i(x) g_i(x')}{\sigma^2_{i,exp}} + \alpha \frac{1}{P(x)} \delta_{x,x'} \tag{B.18}$$

that is positive definite because, by applying the definition, we have

$$\sum_{x,x'} v_x H_{xx'} v_{x'} = \sum_i \Big(\sum_x \frac{v_x g_i(x)}{\sigma^2_{i,exp}} \Big) + \alpha \sum_x \frac{v_x^2}{P(x)} > 0. \tag{B.19}$$

However, we have to restrict the function to the unit simplex determined by the normalization of weights. This preserves the Hessian positive definite and so the function convex (true for any function, provided that the restriction is linear). To proof this, let's focus on a generic function $\tilde{f}$ given by the restriction $\tilde{f}(x_1, \ldots, x_{N-1}) = f(x_1, \ldots, x_{N-1}, x_N(x_1, \ldots, x_{N-1}))$ where x_N is linearly dependent (so $\frac{\partial}{\partial x_\alpha}\frac{\partial x_\beta}{\partial x_j} = 0$); its Hessian is (by the chain rule)

$$\tilde{H}_{ij} = \frac{\partial^2 \tilde{f}}{\partial x_i \partial x_j} = \frac{\partial x_\alpha}{\partial x_i}\frac{\partial}{\partial x_\alpha}\Big(\frac{\partial x_\beta}{\partial x_j}\frac{\partial f}{\partial x_\beta}\Big) = \frac{\partial x_\alpha}{\partial x_i}\frac{\partial x_\beta}{\partial x_j} H_{\alpha\beta} = J^T{}_{i\alpha} H_{\alpha\beta} J_{\beta j} \tag{B.20}$$

so, when applying the definition of positive-definite matrix and the fact that H is positive-definite, we get

$$v_i \tilde{H}_{ij} v_j = v^T J^T H J v = (Jv)^T H (Jv) > 0, \tag{B.21}$$

completing our proof.

Proof no. 2

Let's focus firstly on ensemble refinement only. The derivative of the χ^2 at the optimal ensemble, represented by λ^*, with respect to the hyperparameter $\alpha \in \mathbb{R}^+$ is

$$\begin{aligned} \frac{d\frac{1}{2}\chi^2(\lambda^*(\alpha))}{d\alpha} &= \frac{\frac{1}{2}\partial\chi^2(\lambda)}{\partial\lambda_i}\Big|_{\lambda^*} \frac{d\lambda_i^*(\alpha)}{d\alpha} \\ &= \frac{\langle g_i\rangle_\lambda - g_{i,exp}}{\sigma_i^2} \frac{\partial\langle g_i\rangle_\lambda}{\partial\lambda_j}\Big[-(H^{-1}_{in})_{jk}\frac{\partial^2\Gamma}{\partial\lambda_k\partial\alpha}\Big] \\ &= \alpha\lambda_i C_{ij}(C_{ab} + \alpha\sigma_a^2\delta_{ab})^{-1}{}_{jk}\lambda_k\sigma_k^2 \geq 0 \end{aligned} \tag{B.22}$$

where $C_{ij} = \langle g_i g_j\rangle_\lambda - \langle g_i\rangle_\lambda\langle g_j\rangle_\lambda$ is the variance-covariance matrix of the observables $\vec{g}(x)$ on the optimal ensemble. So, $d\chi^2/d\alpha$ has the form $\lambda_i M_{ij}\lambda_j$ with $M_{ij} = C_{ik}H^{-1}_{kl}\sigma_l^2\delta_{lj}$. Since the two matrices C and H commutes with each other (and with the diagonal matrix $\sigma_l^2\delta_{lj}$), they have a common basis of eigenvectors; in addition, they are both positive definite. Hence, also their product is positive definite (the eigenvalues are the products of the eigenvalues), so we conclude that it is always $\frac{d\chi^2}{d\alpha} \geq 0$ (equivalently from

$\chi^2(\lambda^*(\alpha)) = \alpha^2 \sum_i \lambda_i^*(\alpha)^2 \sigma_i^2$). This does not happen for the validating observables, for which the new matrix C', coming from $\frac{\partial \chi^2}{\partial \lambda_i}$, in general does not commute with the Hessian H, and so the χ^2 may increase or decrease with α, as it does. Conversely, the derivative of the Kullback-Leibler divergence is

$$\frac{dD_{KL}(\lambda^*(\alpha))}{d\alpha} = \frac{\partial D_{KL}}{\partial \lambda_i} \frac{d\lambda_i^*(\alpha)}{d\alpha} = \lambda_j C_{ji} \frac{d\lambda_i^*}{d\alpha} = -\frac{1}{\alpha} \frac{d\frac{1}{2}\chi^2(\alpha)}{d\alpha} \leq 0. \tag{B.23}$$

Hence, increasing the hyperparameter α, we are moving away from P_0 (D_{KL} increases), searching for solutions according more and more with the measured values (although the latter have experimental errors).

Since the optimal value of the loss function depends on α via $\mathcal{L}^* = \mathcal{L}(\vec{\lambda}^*(\alpha), \alpha)$ with $\frac{\partial \mathcal{L}}{\partial \lambda_i}\Big|_{\lambda^*} = 0$, it follows

$$\frac{d\mathcal{L}(\vec{\lambda}^*(\alpha), \alpha)}{d\alpha} = \frac{\partial \mathcal{L}(\vec{\lambda}^*(\alpha), \alpha)}{\partial \alpha} = D_{KL}(\alpha) \geq 0 \tag{B.24}$$

as expected, which tends to zero for $\alpha \to \infty$, when the optimal ensemble approaches more and more the original one. As we have seen, the presence of boundaries does not affect these results.

These results can be extended to more general cases. If we have a single hyperparameter $\alpha \in \mathbb{R}^+$ with loss function $\mathcal{L}(x; \alpha) = f_1(x) + \alpha f_2(x)$ where f_1, f_2 are two general multi-variable smooth functions, then at the point of minimum $x^*(\alpha) = \arg\min_x \mathcal{L}(x; \alpha)$ it is $\frac{\partial \mathcal{L}(x;\alpha)}{\partial x_i}\Big|_{x^*} = 0$ and the Hessian $H_{ij} = \frac{\partial^2 \mathcal{L}(x;\alpha)}{\partial x_i \partial x_j}\Big|_{x^*}$ is a positive-definite matrix. Consequently, the min. value of the loss function $\mathcal{L}^*(\alpha) = \mathcal{L}(x^*(\alpha); \alpha)$ at given α has derivative

$$\frac{d\mathcal{L}^*(\alpha)}{d\alpha} = \cancel{\frac{\partial \mathcal{L}(x;\alpha)}{\partial x_i}\Big|_{x^*} \frac{dx_i^*(\alpha)}{d\alpha}} + \frac{\partial \mathcal{L}(x;\alpha)}{\partial \alpha}\Big|_{x^*(\alpha)} = f_2(x^*(\alpha)) \tag{B.25}$$

and also $\frac{df_1(x^*(\alpha))}{d\alpha} = -\alpha \frac{df_2(x^*(\alpha))}{d\alpha}$. It follows that $f_2(x^*(\alpha))$ decreases with α while $f_1(x^*(\alpha))$ increases (in our previous case, increasing α we move away from the exact experimental values towards the initial hypothesis, so the D_{KL} decreases and the χ^2 increases). Indeed, it is:

$$\frac{dx_i^*(\alpha)}{d\alpha} = -(H^{-1})_{ij} \frac{\partial^2 \mathcal{L}}{\partial x_j \partial \alpha}\Big|_{x^*(\alpha)} = -(H^{-1})_{ij} \frac{\partial f_2(x)}{\partial x_j}\Big|_{x^*(\alpha)} \tag{B.26}$$

$$\begin{aligned} \frac{df_2(x^*(\alpha))}{d\alpha} &= \frac{\partial f_2(x)}{\partial x_i}\Big|_{x^*(\alpha)} \frac{dx_i^*(\alpha)}{d\alpha} \\ &= -\frac{\partial f_2(x)}{\partial x_i}\Big|_{x^*(\alpha)} (H^{-1})_{ij} \frac{\partial f_2(x)}{\partial x_j}\Big|_{x^*(\alpha)} \leq 0 \end{aligned} \tag{B.27}$$

by definition of positive-definite matrix. It is also $\frac{df_1(x^*(\alpha))}{d\alpha} = -\alpha\frac{df_2(x^*(\alpha))}{d\alpha} \geq 0$.

This derivation can be repeated also for a loss function with expression

$$\mathcal{L}(x_1, \cdots, x_N; \alpha_1, \cdots, \alpha_N) = f_0(x_1, \cdots, x_N) + \sum_{i=1}^{N} \alpha_i f_i(x_i) \tag{B.28}$$

with $\alpha_i \in \mathbb{R}^+$ and f_i multivariable smooth functions (as before, $x_i \in \mathbb{R}^{n_i}$). This includes the cases of FMR+FFR $\mathcal{L}(\phi, \theta) = \frac{1}{2}\chi^2(\phi, \theta) + \beta R_1(\phi) + \gamma R_2(\theta)$ and ER with system-dependent hyperparameters α, but not the cases in which ER is "coupled" to FMR and/or FFR, since we have $\alpha D_{KL}(\lambda, \phi, \theta)$. So, we will refer to this loss function as the "uncoupled" one. Let's use the shortcut $z = (x_1, \cdots, x_N)$, so we have $z^* = \arg\min_z \mathcal{L}(z; \alpha_1, \cdots, \alpha_N)$, which means (provided they are smooth functions) $\frac{\partial \mathcal{L}(z;\vec{\alpha})}{\partial z_i}\Big|_{z^*} = 0$ and the Hessian $\frac{\partial^2 \mathcal{L}}{\partial z_i \partial z_j}\Big|_{z^*}$ is positive definite. Consequently, the total derivative of the optimal loss with respect to α_i is $\frac{\partial \mathcal{L}(z^*(\vec{\alpha});\vec{\alpha})}{\partial \alpha_i} = f_i(x_i^*(\alpha))$ and also

$$\frac{\partial z_i^*(\vec{\alpha})}{\partial \alpha_j} = -(H^{-1})_{ik}\frac{\partial^2 \mathcal{L}(z;\vec{\alpha})}{\partial z_k \partial \alpha_j}\Big|_{z^*(\alpha)} = -(H^{-1})_{ik}\frac{\partial f_j(z)}{\partial z_k}\Big|_{z^*(\alpha)}. \tag{B.29}$$

Now, we exploit the element-wise relation (x_{ij} is the j-th component of x_i, the coordinate specific of f_i and appearing in f_i and f_0 only)

$$\alpha_i \frac{\partial f_i}{\partial x_{ij}}\Big|_{z^*(\alpha)} = -\frac{\partial f_0}{\partial x_{ij}}\Big|_{z^*(\alpha)} \tag{B.30}$$

to infer the following sign of the derivatives for the optimal $f_i^*(\vec{\alpha})$ values:

$$\begin{aligned} \frac{\partial f_0(z^*(\vec{\alpha}))}{\partial \alpha_i} &= \frac{\partial f_0(z)}{\partial z_j}\Big|_{z^*}\frac{\partial z_j^*}{\partial \alpha_i} = \frac{\partial f_0(z)}{\partial z_j}\Big|_{z^*}(-H^{-1})_{jk}\frac{\partial f_i(z)}{\partial z_k}\Big|_{z^*} \\ &= \frac{1}{\alpha_i}\frac{\partial f_0(z)}{\partial z_j}\Big|_{z^*}(H^{-1})_{jk}\frac{\partial f_0(z)}{\partial z_k} \geq 0 \end{aligned} \tag{B.31}$$

and

$$\begin{aligned} \frac{\partial f_i(z^*(\vec{\alpha}))}{\partial \alpha_j} &= \frac{\partial f_i(z)}{\partial z_k}\Big|_{z^*}\frac{\partial z_k^*(\vec{\alpha})}{\partial \alpha_j} = \frac{\partial f_i(z)}{\partial z_k}\Big|_{z^*}(-H^{-1})_{kl}\frac{\partial f_j(z)}{\partial z_l}\Big|_{z^*} \\ &= -\frac{1}{\alpha_i \alpha_j}\frac{\partial f_0(z)}{\partial z_k}\Big|_{z^*}(H^{-1})_{kl}\frac{\partial f_0(z)}{\partial z_l}\Big|_{z^*} \leq 0. \end{aligned} \tag{B.32}$$

So, when combining FFR with FMR, and increasing either the hyperparameter β or γ, the χ^2 increases and the regularization terms decrease, as one could expected. However, this is not guaranteed to happen when combining also with ER, due to the "coupling" between the objective functions.

Proof no. 3

Both in the case of ensemble refinement and force-field fitting, we get a relation between the χ^2 and the regularization R (D_{KL} for instance) $\chi^2(R)$ just by plotting $\chi^2(\theta)$ vs. $R(\theta)$ where θ is the hyperparameter (α or β), defined as $\chi^2(\theta) = \chi^2(\lambda^*(\theta))$ and $R(\theta) = R(\lambda^*(\theta))$ where $\lambda^*(\theta) = \arg\min_\lambda \mathcal{L}(\lambda;\theta)$ with the loss function $\mathcal{L}(\lambda;\theta) = \frac{1}{2}\chi^2(\lambda) + \theta R(\lambda)$, so $\frac{\partial \mathcal{L}(\lambda;\theta)}{\partial \lambda_i} = 0$. Hence, the slope is

$$\begin{aligned} \frac{d\chi^2(R)}{dR} &= \frac{\chi^2(\theta + d\theta) - \chi^2(\theta)}{R(\theta + d\theta) - R(\theta)} = \frac{\frac{d\chi^2}{d\theta} d\theta}{\frac{dR}{d\theta} d\theta} \\ &= \frac{\left.\frac{\partial \chi^2}{\partial \lambda_i}\right|_{\lambda^*} \frac{d\lambda_i^*}{d\theta}}{\left.\frac{\partial R}{\partial \lambda_i}\right|_{\lambda^*} \frac{d\lambda_i^*}{d\theta}} = -2\theta \end{aligned} \tag{B.33}$$

because $\left.\frac{\partial \mathcal{L}}{\partial \lambda_i}\right|_{\lambda^*} = 0$ implies $\left.\frac{\partial \chi^2}{\partial \lambda_i}\right|_{\lambda^*} = -2\theta \left.\frac{\partial D_{KL}}{\partial \lambda_i}\right|_{\lambda^*}$ (see also the first plot in [104]).

Appendix C

Supplementary for `MDRefine`

C.1 Implementation of `minimizer`

To determine the optimal $\lambda^*(\phi, \theta)$ at given (ϕ, θ) we can employ the same `gamma_function` previously described. Notice `gamma_function` does not include either the computation of the observables through the forward model θ or the reweighting through the force-field correction ϕ (namely, the dependence of Γ on θ, ϕ), rather they should be externally computed and then passed to `gamma_function` as input observables and weights.

Regarding the full loss function, we implemented $\mathcal{L}_1(\phi, \theta)$ in `loss_function` by inserting the search for optimal $\lambda^*(\phi, \theta)$. Its inputs are the Numpy array of (ϕ, θ) coefficients, the `data` object, the `regularization` dictionary, the hyperparameters α, β, γ (by default $+\infty$, i.e., no refinement in that direction) and some other variables. The `regularization` dictionary has the two keys `force_field_reg`, `forward_model_reg`; the first item can be either a string (`KL divergence`, `plain l2`) or a user-defined function acting on ϕ; the second item is a user-defined function acting on θ and θ_0 (for example, an L^2 regularization).

The other inputs include: `fixed_lambdas` (by default `None`), `gtol_inn` and `bounds` (tolerance and boundaries for the inner minimization of λ; default values 10^{-3} and `None`, respectively), `if_save` (Boolean variable, by default `False`). For `if_save = True`, `loss_function` does not return just its numerical value, rather an object with all the quantities appearing in Eq. 9 of the Main text, such as the parameters (λ, ϕ, θ) (when defined), the new weights, the χ^2, the regularization terms.

Whenever the input variable `fixed_lambdas` is not `None`, `loss_function` implements the computation of $\mathcal{L}_1(\phi, \theta)$ through substitution of the given λ into $\Gamma(\lambda; \phi, \theta)$ in place of the inner minimization of Γ over λ (the two

computations are the same if α is infinite, namely, no ensemble refinement). This distinction turns out to be crucial when computing the gradient function $\frac{\partial \mathcal{L}_1(\phi,\theta)}{\partial(\phi,\theta)}$ `gradient_fun` with automatic differentiation (see section B.1).

The `loss_function` and its `gradient_fun` are evaluated together inside `loss_function_and_grad`, which is employed in `minimizer` in order to get the optimal solution. In this way, passing to `minimizer` the `data` object, the `regularization` dictionary and the hyperparameters (by default $\alpha, \beta, \gamma = +\infty$, i.e., no refinement in those directions), we get an output object `Result` containing the optimal solution (P, ϕ, θ) and its properties.

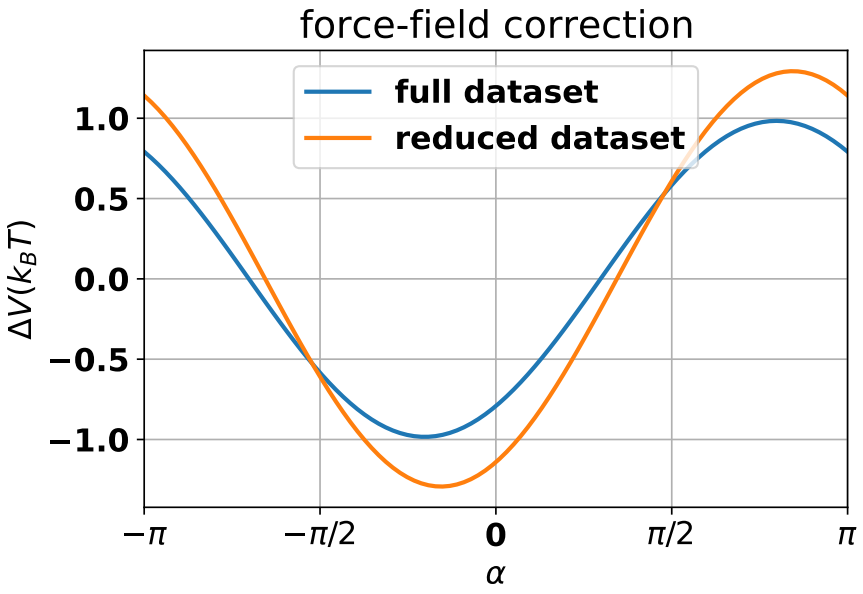
C.2 Fully combined refinement – comparison with test set

We set aside a test set (not used in cross-validation) in order to report the agreement of the refined models with experiments, to assess for model evaluation. Specifically, we have used `split_dataset` to select a test set and then repeated the minimizations with and without force-field fitting on the reduced dataset in the same way described in section 3.2 (namely, including the search for the optimal hyperparameters as performed by `MDRefinement`). The results we have determined on the subset are shown in Fig C.1 and Tables C.1, C.2. Once the optimal solution has been found on this reduced dataset, we evaluate the quality of the two refinement methods by reporting the χ^2 values (normalized to the total n. of observables) on the test set in Table C.3.

	α^*	β^*	γ^*	$\sin\alpha$	$\cos\alpha$
reference	$+\infty$	$+\infty$	$+\infty$	-	-
E + FM full	46.31	$+\infty$	0.96	-	-
E + FM + FF full	959.86	167.94	0.42	0.58	-0.79
E + FM reduced	14.77	$+\infty$	0.52	-	-
E + FM + FF reduced	37.24	1.89	0.35	0.61	-1.14

Table C.1: Optimal values of the hyperparameters $\alpha^*, \beta^*, \gamma^*$ and coefficients of the sinusoidal force-field on α dihedral angles (in k_BT). Comparison between: reference values; coefficients determined with/without including force-field fitting (E + FM or E + FM + FF) on the full dataset or on a reduced dataset (got by removing a test set).

(a) force-field correction on α dihedral angles



(b) refinement of the Karplus equations

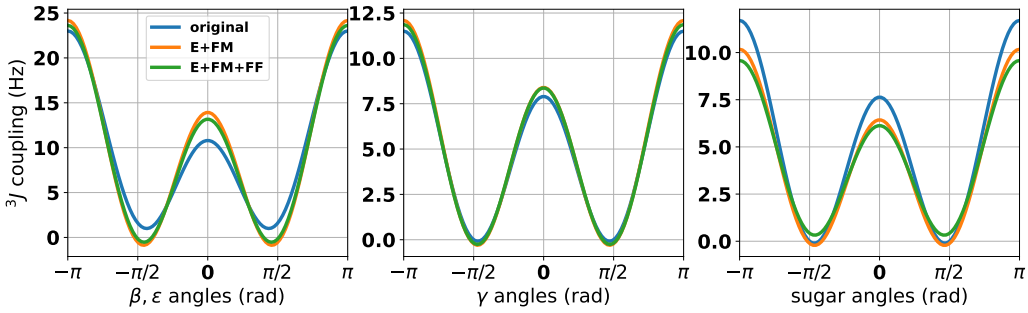


Figure C.1: Force-field correction (a) and refinement of the Karplus equations (b) derived by minimization of the loss function on a *reduced dataset*, leaving out a test set of 10% frames and 10% observables. In (a) we report also the force-field correction determined on the *full dataset*, which overlaps with the other in the visited regions, close to $\pm\pi/2$ (see also Fig. 5a in the Main text). In (b) we do not report the refinement derived with the full dataset (Main, Fig. 6) since it would completely overlap with the corresponding plotted refinements.

	A_β	B_β	C_β	A_γ	B_γ	C_γ	A_ν	B_ν	C_ν
reference	15.3	-6.1	1.6	9.7	-1.8	0	9.67	-2.03	0
E + FM full	18.91	-5.23	-0.21	9.95	-1.86	-0.11	8.34	-1.81	-0.069
E + FM + FF full	18.13	-5.29	0.072	9.92	-1.76	-0.12	6.99	-1.64	0.67
E + FM reduced	19.59	-5.12	-0.54	10.47	-1.84	-0.22	8.42	-1.87	-0.12
E + FM + FF reduced	18.57	-5.22	-0.17	10.30	-1.75	-0.19	7.43	-1.72	0.42

Table C.2: Empirical coefficients A, B, C (in Hz) of the Karplus equations. Comparison between: reference values; coefficients determined with/without including force-field fitting (E + FM or E + FM + FF) on the full dataset or on a reduced dataset (got by removing a test set).

test χ^2_{red} original	test χ^2_{red} E+FM	test χ^2_{red} E+FM+FF
537.4	1.62	2.13

Table C.3: Reduced χ^2 (namely, χ^2 divided by the n. of observables) on the test set; original set of weights (before the refinement), refinement of ensemble and forward model (E+FM), fully combined refinement (E+FM+FF).

Bibliography

[1] Thorben Fröhlking, Mattia Bernetti, and Giovanni Bussi. Simultaneous refinement of molecular dynamics ensembles and forward models using experimental data. *J. Chem. Phys.*, 158(21), 2023.

[2] Neocles B Leontis and Eric Westhof. Geometric nomenclature and classification of RNA base pairs. *Rna*, 7(4):499–512, 2001.

[3] Sandro Bottaro, Francesco Di Palma, and Giovanni Bussi. The role of nucleobase interactions in RNA structure and dynamics. *Nucleic acids research*, 42(21):13306–13314, 2014.

[4] MS Smyth and JHJ Martin. X ray crystallography. *Molecular Pathology*, 53(1):8, 2000.

[5] David B Davies. Conformations of nucleosides and nucleotides. *Prog. Nucl. Magn. Reson. Spectrosc.*, 12(3):135–225, 1978.

[6] PP Lankhorst, CAG Haasnoot, C Erkelens, HP Westennk, GA Van der Marel, JH Van Boom, and C Altona. Carbon-13 NMR in conformational analysis of nucleic acid fragments. 4. the torsion angle distribution about the C3'–O3' bond in DNA constituents. *Nucleic Acids Res.*, 13(3):927–942, 1985.

[7] David E Condon, Scott D Kennedy, Brendan C Mort, Ryszard Kierzek, Ilyas Yildirim, and Douglas H Turner. Stacking in RNA: NMR of four tetramers benchmark molecular dynamics. *J. Chem. Theory Comp.*, 11(6):2729–2742, 2015.

[8] Joachim Seelig and Anna Seelig. Protein stability analysis of heat and cold denaturation without and with unfolding models. *The Journal of Physical Chemistry B*, 127(15):3352–3363, 2023.

[9] Eva Nogales and Sjors HW Scheres. Cryo-EM: a unique tool for the visualization of macromolecular complexity. *Molecular cell*, 58(4):677–689, 2015.

[10] PS Singh. Small-angle scattering techniques (SAXS/SANS). In *Membrane Characterization*, pages 95–111. Elsevier, 2017.

[11] Jiang Wang, Simon Olsson, Christoph Wehmeyer, Adrià Pérez, Nicholas E Charron, Gianni De Fabritiis, Frank Noé, and Cecilia Clementi. Machine learning of coarse-grained molecular dynamics force fields. *ACS central science*, 5(5):755–767, 2019.

[12] Maciej Majewski, Adrià Pérez, Philipp Thölke, Stefan Doerr, Nicholas E Charron, Toni Giorgino, Brooke E Husic, Cecilia Clementi, Frank Noé, and Gianni De Fabritiis. Machine learning coarse-grained potentials of protein thermodynamics. *Nature communications*, 14(1):5739, 2023.

[13] Iryna Zaporozhets and Cecilia Clementi. Multibody terms in protein coarse-grained models: A top-down perspective. *The Journal of Physical Chemistry B*, 127(31):6920–6927, 2023.

[14] Stephan Thaler and Julija Zavadlav. Learning neural network potentials from experimental data via differentiable trajectory reweighting. *Nat. Commun.*, 12(1):6884, 2021.

[15] Stephan Thaler, Maximilian Stupp, and Julija Zavadlav. Deep coarse-grained potentials via relative entropy minimization. *The Journal of Chemical Physics*, 157(24), 2022.

[16] Thomas E Cheatham III, Piotr Cieplak, and Peter A Kollman. A modified version of the Cornell et al. force field with improved sugar pucker phases and helical repeat. *Journal of Biomolecular Structure and Dynamics*, 16(4):845–862, 1999.

[17] Alberto Pérez, Iván Marchán, Daniel Svozil, Jiři Šponer, Thomas E Cheatham, Charles A Laughton, and Modesto Orozco. Refinement of the AMBER force field for nucleic acids: improving the description of α/γ conformers. *Biophys. J.*, 92(11):3817–3829, 2007.

[18] Marie Zgarbová, Michal Otyepka, Jiří Šponer, Arnošt Mládek, Pavel Banáš, Thomas E Cheatham III, and Petr Jurecka. Refinement of the Cornell et al. nucleic acids force field based on reference quantum chemical calculations of glycosidic torsion profiles. *J. Chem. Theory Comp.*, 7(9):2886–2902, 2011.

[19] Andrea Cesari, Alejandro Gil-Ley, and Giovanni Bussi. Combining simulations and solution experiments as a paradigm for RNA force field refinement. *J. Chem. Theory Comp.*, 12(12):6192–6200, 2016.

[20] Sandro Bottaro, Pavel Banáš, Jiŕi Špöner, and Giovanni Bussi. Free energy landscape of GAGA and UUCG RNA tetraloops. *J. Phys. Chem. Lett.*, 7(20):4032–4038, 2016.

[21] Sandro Bottaro, Giovanni Bussi, Scott D Kennedy, Douglas H Turner, and Kresten Lindorff-Larsen. Conformational ensembles of RNA oligonu-

cleotides from integrating NMR and molecular simulations. *Science Adv.*, 4(5):eaar8521, 2018.

[22] Sandro Bottaro, Parker J Nichols, Beat Vögeli, Michele Parrinello, and Kresten Lindorff-Larsen. Integrating NMR and simulations reveals motions in the UUCG tetraloop. *Nucleic acids research*, 48(11):5839–5848, 2020.

[23] Christina Bergonzo, Alexander Grishaev, and Sandro Bottaro. Conformational heterogeneity of UCAAUC RNA oligonucleotide from molecular dynamics simulations, SAXS, and NMR experiments. *RNA*, 28(7):937–946, 2022.

[24] Xiang-Jun Lu and Wilma K Olson. 3DNA: a software package for the analysis, rebuilding and visualization of three-dimensional nucleic acid structures. *Nucleic acids research*, 31(17):5108–5121, 2003.

[25] Rajendra Kumar and Helmut Grubmüller. do_x3dna: a tool to analyze structural fluctuations of dsDNA or dsRNA from molecular dynamics simulations. *Bioinformatics*, 31(15):2583–2585, 2015.

[26] Jérôme Hénin, Tony Lelièvre, Michael R Shirts, Omar Valsson, and Lucie Delemotte. Enhanced sampling methods for molecular dynamics simulations. *arXiv preprint arXiv:2202.04164*, 2022.

[27] Vojtěch Mlỳnskỳ and Giovanni Bussi. Exploring RNA structure and dynamics through enhanced sampling simulations. *Current opinion in structural biology*, 49:63–71, 2018.

[28] Sereina Riniker. Fixed-charge atomistic force fields for molecular dynamics simulations in the condensed phase: An overview. *Journal of chemical information and modeling*, 58(3):565–578, 2018.

[29] Wendy D Cornell, Piotr Cieplak, Christopher I Bayly, Ian R Gould, Kenneth M Merz, David M Ferguson, David C Spellmeyer, Thomas Fox, James W Caldwell, and Peter A Kollman. A second generation force field for the simulation of proteins, nucleic acids, and organic molecules. *J. Am. Chem. Soc.*, 118(9):2309–2309, 1996.

[30] Junmei Wang, Piotr Cieplak, and Peter A Kollman. How well does a restrained electrostatic potential (RESP) model perform in calculating conformational energies of organic and biological molecules? *J. Comput. Chem.*, 21(12):1049–1074, 2000.

[31] Pavel Banás, Daniel Hollas, Marie Zgarbová, Petr Jurecka, Modesto Orozco, Thomas E Cheatham III, Jirí Sponer, and Michal Otyepka. Performance of molecular mechanics force fields for RNA simulations: stability of UUCG and GNRA hairpins. *Journal of Chemical Theory*

and Computation, 6(12):3836–3849, 2010.

[32] T Steinbrecher, J Latzer, and DA Case. Revised AMBER parameters for bioorganic phosphates. *J. Chem. Theory Comp.*, 8(11):4405–4412, 2012.

[33] Thorben Fröhlking, Vojtěch Mlỳnskỳ, Michal Janeček, Petra Kührová, Miroslav Krepl, Pavel Banáš, Jiří Šponer, and Giovanni Bussi. Automatic learning of hydrogen-bond fixes in an AMBER RNA force field. *J. Chem. Theory Comput.*, 18(7):4490–4502, 2022.

[34] Saeed Izadi, Ramu Anandakrishnan, and Alexey V Onufriev. Building water models: a different approach. *J. Phys. Chem. Lett.*, 5(21):3863–3871, 2014.

[35] In Suk Joung and Thomas E Cheatham III. Determination of alkali and halide monovalent ion parameters for use in explicitly solvated biomolecular simulations. *J. Phys. Chem. B*, 112(30):9020–9041, 2008.

[36] Pengfei Li and Kenneth M Merz Jr. Metal ion modeling using classical mechanics. *Chemical reviews*, 117(3):1564–1686, 2017.

[37] Raviprasad Aduri, Brian T Psciuk, Pirro Saro, Hariprakash Taniga, H Bernhard Schlegel, and John SantaLucia. AMBER force field parameters for the naturally occurring modified nucleosides in RNA. *J. Chem. Theory Comput.*, 3(4):1464–1475, 2007.

[38] Indrajit Deb, Joanna Sarzynska, Lennart Nilsson, and Ansuman Lahiri. Conformational preferences of modified uridines: comparison of AMBER derived force fields. *Journal of chemical information and modeling*, 54(4):1129–1142, 2014.

[39] Indrajit Deb, Rupak Pal, Joanna Sarzynska, and Ansuman Lahiri. Reparameterizations of the χ torsion and Lennard-Jones σ parameters improve the conformational characteristics of modified uridines. *Journal of Computational Chemistry*, 37(17):1576–1588, 2016.

[40] Travis Hurst and Shi-Jie Chen. Deciphering nucleotide modification-induced structure and stability changes. *RNA biology*, 18(11):1920–1930, 2021.

[41] Valerio Piomponi, Miroslav Krepl, Jiri Sponer, and Giovanni Bussi. Molecular simulations to investigate the impact of N6-methylation in RNA recognition: Improving accuracy and precision of binding free energy prediction. *J. Phys. Chem. B*, 128(37):8896–8907, 2024.

[42] Jörg Behler and Michele Parrinello. Generalized neural-network representation of high-dimensional potential-energy surfaces. *Physical review letters*, 98(14):146401, 2007.

[43] Jorg Behler. Four generations of high-dimensional neural network potentials. *Chemical Reviews*, 121(16):10037–10072, 2021.

[44] Tsz Wai Ko, Jonas A Finkler, Stefan Goedecker, and Jörg Behler. A fourth-generation high-dimensional neural network potential with accurate electrostatics including non-local charge transfer. *Nature communications*, 12(1):398, 2021.

[45] Emir Kocer, Tsz Wai Ko, and Jörg Behler. Neural network potentials: A concise overview of methods. *Annual review of physical chemistry*, 73(1):163–186, 2022.

[46] Albert P Bartók, Mike C Payne, Risi Kondor, and Gábor Csányi. Gaussian approximation potentials: The accuracy of quantum mechanics, without the electrons. *Physical review letters*, 104(13):136403, 2010.

[47] Albert P Bartók, Risi Kondor, and Gábor Csányi. On representing chemical environments. *Physical Review B – Condensed Matter and Materials Physics*, 87(18):184115, 2013.

[48] Giulio Imbalzano, Andrea Anelli, Daniele Giofré, Sinja Klees, Jörg Behler, and Michele Ceriotti. Automatic selection of atomic fingerprints and reference configurations for machine-learning potentials. *The Journal of chemical physics*, 148(24), 2018.

[49] Volker L Deringer, Albert P Bartók, Noam Bernstein, David M Wilkins, Michele Ceriotti, and Gábor Csányi. Gaussian process regression for materials and molecules. *Chemical Reviews*, 121(16):10073–10141, 2021.

[50] Volker L Deringer and Gábor Csányi. Machine learning based interatomic potential for amorphous carbon. *Physical Review B*, 95(9):094203, 2017.

[51] Patrick Rowe, Volker L Deringer, Piero Gasparotto, Gábor Csányi, and Angelos Michaelides. An accurate and transferable machine learning potential for carbon. *The Journal of Chemical Physics*, 153(3), 2020.

[52] Albert P Bartók, James Kermode, Noam Bernstein, and Gábor Csányi. Machine learning a general-purpose interatomic potential for silicon. *Physical Review X*, 8(4):041048, 2018.

[53] Volker L Deringer, Miguel A Caro, and Gábor Csányi. Machine learning interatomic potentials as emerging tools for materials science. *Advanced Materials*, 31(46):1902765, 2019.

[54] Albert P Bartók, Sandip De, Carl Poelking, Noam Bernstein, James R Kermode, Gábor Csányi, and Michele Ceriotti. Machine learning unifies the modeling of materials and molecules. *Science advances*, 3(12):e1701816, 2017.

[55] Tobias Morawietz, Andreas Singraber, Christoph Dellago, and Jörg Behler. How van der Waals interactions determine the unique properties of water. *Proceedings of the National Academy of Sciences*, 113(30):8368–8373, 2016.

[56] Bingqing Cheng, Edgar A Engel, Jörg Behler, Christoph Dellago, and Michele Ceriotti. Ab initio thermodynamics of liquid and solid water. *Proceedings of the National Academy of Sciences*, 116(4):1110–1115, 2019.

[57] Justin S Smith, Olexandr Isayev, and Adrian E Roitberg. ANI-1: an extensible neural network potential with dft accuracy at force field computational cost. *Chemical science*, 8(4):3192–3203, 2017.

[58] Kristof Schütt, Pieter-Jan Kindermans, Huziel Enoc Sauceda Felix, Stefan Chmiela, Alexandre Tkatchenko, and Klaus-Robert Müller. Schnet: A continuous-filter convolutional neural network for modeling quantum interactions. *Advances in neural information processing systems*, 30, 2017.

[59] P-J Kindermans and K-R Müller. Schnet – a deep learning architecture for molecules and materials. *The Journal of Chemical Physics*, 148(24), 2018.

[60] Johannes Gasteiger, Janek Groß, and Stephan Günnemann. Directional message passing for molecular graphs. *arXiv preprint arXiv:2003.03123*, 2020.

[61] Johannes Gasteiger, Shankari Giri, Johannes T Margraf, and Stephan Günnemann. Fast and uncertainty-aware directional message passing for non-equilibrium molecules. *arXiv preprint arXiv:2011.14115*, 2020.

[62] Justin Gilmer, Samuel S Schoenholz, Patrick F Riley, Oriol Vinyals, and George E Dahl. Neural message passing for quantum chemistry. In *International conference on machine learning*, pages 1263–1272. PMLR, 2017.

[63] Vıctor Garcia Satorras, Emiel Hoogeboom, and Max Welling. E(n) equivariant graph neural networks. In *International conference on machine learning*, pages 9323–9332. PMLR, 2021.

[64] Linfeng Zhang, Jiequn Han, Han Wang, Roberto Car, and Weinan E. Deep potential molecular dynamics: a scalable model with the accuracy of quantum mechanics. *Physical review letters*, 120(14):143001, 2018.

[65] Stefan Doerr, Maciej Majewski, Adrià Pérez, Andreas Kramer, Cecilia Clementi, Frank Noe, Toni Giorgino, and Gianni De Fabritiis. TorchMD: A deep learning framework for molecular simulations. *Journal of chem-*

ical theory and computation, 17(4):2355–2363, 2021.

[66] Philipp Thölke and Gianni De Fabritiis. Torchmd-net: equivariant transformers for neural network based molecular potentials. *arXiv preprint arXiv:2202.02541*, 2022.

[67] Furio Ercolessi and James B Adams. Interatomic potentials from first-principles calculations: the force-matching method. *Europhysics Letters*, 26(8):583, 1994.

[68] Brooke E Husic, Nicholas E Charron, Dominik Lemm, Jiang Wang, Adrià Pérez, Maciej Majewski, Andreas Krämer, Yaoyi Chen, Simon Olsson, Gianni De Fabritiis, et al. Coarse graining molecular dynamics with graph neural networks. *The Journal of chemical physics*, 153(19), 2020.

[69] Nicholas E Charron, Klara Bonneau, Aldo S Pasos-Trejo, Andrea Guljas, Yaoyi Chen, Félix Musil, Jacopo Venturin, Daria Gusew, Iryna Zaporozhets, Andreas Krämer, et al. Navigating protein landscapes with a machine-learned transferable coarse-grained model. *Nature chemistry*, pages 1–9, 2025.

[70] Antonio Mirarchi, Raúl P Peláez, Guillem Simeon, and Gianni De Fabritiis. Amaro: all heavy-atom transferable neural network potentials of protein thermodynamics. *Journal of Chemical Theory and Computation*, 20(22):9871–9878, 2024.

[71] John Jumper, Richard Evans, Alexander Pritzel, Tim Green, Michael Figurnov, Olaf Ronneberger, Kathryn Tunyasuvunakool, Russ Bates, Augustin Žídek, Anna Potapenko, et al. Highly accurate protein structure prediction with AlphaFold. *nature*, 596(7873):583–589, 2021.

[72] Mihaly Varadi, Stephen Anyango, Mandar Deshpande, Sreenath Nair, Cindy Natassia, Galabina Yordanova, David Yuan, Oana Stroe, Gemma Wood, Agata Laydon, et al. AlphaFold protein structure database: massively expanding the structural coverage of protein-sequence space with high-accuracy models. *Nucleic acids research*, 50(D1):D439–D444, 2022.

[73] Ruth Nussinov, Mingzhen Zhang, Yonglan Liu, and Hyunbum Jang. AlphaFold, artificial intelligence (AI), and allostery. *The Journal of Physical Chemistry B*, 126(34):6372–6383, 2022.

[74] David Baker and Andrej Sali. Protein structure prediction and structural genomics. *Science*, 294(5540):93–96, 2001.

[75] David E Kim, Dylan Chivian, and David Baker. Protein structure prediction and analysis using the Robetta server. *Nucleic acids research*,

32(suppl_2):W526–W531, 2004.

[76] Carol A Rohl, Charlie EM Strauss, Kira MS Misura, and David Baker. Protein structure prediction using Rosetta. In *Methods in enzymology*, volume 383, pages 66–93. Elsevier, 2004.

[77] Minkyung Baek, Frank DiMaio, Ivan Anishchenko, Justas Dauparas, Sergey Ovchinnikov, Gyu Rie Lee, Jue Wang, Qian Cong, Lisa N Kinch, R Dustin Schaeffer, et al. Accurate prediction of protein structures and interactions using a three-track neural network. *Science*, 373(6557):871–876, 2021.

[78] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

[79] John Jumper, Richard Evans, Alexander Pritzel, Tim Green, Michael Figurnov, Olaf Ronneberger, Kathryn Tunyasuvunakool, Russ Bates, Augustin Žídek, Anna Potapenko, et al. Applying and improving AlphaFold at CASP14. *Proteins: Structure, Function, and Bioinformatics*, 89(12):1711–1721, 2021.

[80] Kiersten M Ruff and Rohit V Pappu. AlphaFold and implications for intrinsically disordered proteins. *Journal of molecular biology*, 433(20):167208, 2021.

[81] Z Faidon Brotzakis, Shengyu Zhang, Mhd Hussein Murtada, and Michele Vendruscolo. AlphaFold prediction of structural ensembles of disordered proteins. *Nature Communications*, 16(1):1632, 2025.

[82] Vincent Schnapka, Tatiana Morozova, Samiran Sen, and Massimiliano Bonomi. Atomic resolution ensembles of intrinsically disordered and multi-domain proteins with AlphaFold. *bioRxiv*, pages 2025–06, 2025.

[83] Robert T McDonnell, Aaron N Henderson, and Adrian H Elcock. Structure prediction of large RNAs with AlphaFold3 highlights its capabilities and limitations. *Journal of molecular biology*, 436(22):168816, 2024.

[84] Fan Bu, Yagoub Adam, Ryszard W Adamiak, Maciej Antczak, Belisa Rebeca H de Aquino, Nagendar Goud Badepally, Robert T Batey, Eugene F Baulin, Pawel Boinski, Michal J Boniecki, et al. RNA-puzzles round V: blind predictions of 23 RNA structures. *Nature methods*, 22(2):399–411, 2025.

[85] Clément Bernard, Guillaume Postic, Sahar Ghannay, and Fariza Tahi. Has AlphaFold3 achieved success for RNA? *Biological Crystallography*, 81(2), 2025.

[86] Ivo L Hofacker, Walter Fontana, Peter F Stadler, L Sebastian Bonhoeffer, Manfred Tacker, Peter Schuster, et al. Fast folding and comparison of RNA secondary structures. *Monatshefte fur chemie*, 125:167–167, 1994.

[87] Ronny Lorenz, Stephan H Bernhart, Christian Höner zu Siederdissen, Hakim Tafer, Christoph Flamm, Peter F Stadler, and Ivo L Hofacker. ViennaRNA Package 2.0. *Algorithms for molecular biology*, 6:1–14, 2011.

[88] Chuong B Do, Daniel A Woods, and Serafim Batzoglou. CONTRAfold: RNA secondary structure prediction without physics-based models. *Bioinformatics*, 22(14):e90–e98, 2006.

[89] Andrew Martin Watkins, Ramya Rangan, and Rhiju Das. FARFAR2: improved de novo rosetta prediction of complex global RNA folds. *Structure*, 28(8):963–976, 2020.

[90] Hannah K Wayment-Steele, Wipapat Kladwang, Alexandra I Strom, Jeehyung Lee, Adrien Treuille, Alex Becka, Eterna Participants, and Rhiju Das. RNA secondary structure packages evaluated and improved by high-throughput experiments. *Nature methods*, 19(10):1234–1242, 2022.

[91] Edward J Merino, Kevin A Wilkinson, Jennifer L Coughlan, and Kevin M Weeks. RNA structure analysis at single nucleotide resolution by selective 2'-hydroxyl acylation and primer extension (SHAPE). *Journal of the American Chemical Society*, 127(12):4223–4231, 2005.

[92] Nathan A Siegfried, Steven Busan, Greggory M Rice, Julie AE Nelson, and Kevin M Weeks. RNA motif discovery by SHAPE and mutational profiling (SHAPE-MaP). *Nature methods*, 11(9):959–965, 2014.

[93] Raphael JL Townshend, Stephan Eismann, Andrew M Watkins, Ramya Rangan, Masha Karelina, Rhiju Das, and Ron O Dror. Geometric deep learning of RNA structure. *Science*, 373(6558):1047–1051, 2021.

[94] Bohdan Schneider, Blake Alexander Sweeney, Alex Bateman, Jiri Cerny, Tomasz Zok, and Marta Szachniuk. When will RNA get its alphaFold moment? *Nucleic Acids Research*, 51(18):9522–9532, 2023.

[95] Simone Orioli, Andreas Haahr Larsen, Sandro Bottaro, and Kresten Lindorff-Larsen. How to learn from inconsistencies: Integrating molecular simulations with experimental data. *Prog. Mol. Biol. Transl. Sci.*, 170:123–176, 2020.

[96] Ramya Rangan, Massimiliano Bonomi, Gabriella T Heller, Andrea Cesari, Giovanni Bussi, and Michele Vendruscolo. Determination of structural ensembles of proteins: restraining vs reweighting. *Journal of*

chemical theory and computation, 14(12):6632–6641, 2018.

[97] Jed W Pitera and John D Chodera. On the use of experimental observations to bias simulated ensembles. *J. Chem. Theory Comp.*, 8(10):3445–3451, 2012.

[98] Kyle A Beauchamp, Vijay S Pande, and Rhiju Das. Bayesian energy landscape tilting: towards concordant models of molecular ensembles. *Biophys. J.*, 106(6):1381–1390, 2014.

[99] Gerhard Hummer and Jürgen Köfinger. Bayesian ensemble refinement by replica simulations and reweighting. *J. Chem. Phys.*, 143(24):243150, 2015.

[100] David H Brookes and Teresa Head-Gordon. Experimental inferential structure determination of ensembles for intrinsically disordered proteins. *J. Am. Chem. Soc.*, 138(13):4530–4538, 2016.

[101] Riccardo Capelli, Guido Tiana, and Carlo Camilloni. An implementation of the maximum-caliber principle by replica-averaged time-resolved restrained simulations. *J. Chem. Phys.*, 148(18), 2018.

[102] Jürgen Köfinger, Lukas S Stelzl, Klaus Reuter, César Allande, Katrin Reichel, and Gerhard Hummer. Efficient ensemble refinement by reweighting. *J. Chem. Theory Comput.*, 15(5):3390–3401, 2019.

[103] Sandro Bottaro, Tone Bengtsen, and Kresten Lindorff-Larsen. Integrating molecular simulation and experimental data: a Bayesian/maximum entropy reweighting approach. *Structural bioinformatics: methods and protocols*, pages 219–240, 2020.

[104] Jürgen Köfinger and Gerhard Hummer. Encoding prior knowledge in ensemble refinement. *J. Chem. Phys.*, 160(11):114111, 2024.

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

[106] Edwin T Jaynes. Information theory and statistical mechanics. ii. *Physical review*, 108(2):171, 1957.

[107] Roger Fletcher. *Practical methods of optimization*. John Wiley & Sons, 2000.

[108] Jean-Paul Ryckaert, Giovanni Ciccotti, and Herman JC Berendsen. Numerical integration of the cartesian equations of motion of a system with constraints: molecular dynamics of n-alkanes. *Journal of computational physics*, 23(3):327–341, 1977.

[109] Andrea Cesari, Sabine Reißer, and Giovanni Bussi. Using the maximum entropy principle to combine simulations and solution experiments. *Computation*, 6(1):15, 2018.

[110] Bartosz Różycki, Young C Kim, and Gerhard Hummer. SAXS ensemble refinement of ESCRT-III CHMP3 conformational transitions. *Structure*, 19(1):109–116, 2011.

[111] Katrin Reichel, Lukas S Stelzl, Jürgen Köfinger, and Gerhard Hummer. Precision DEER distances from spin-label ensemble refinement. *The journal of physical chemistry letters*, 9(19):5748–5752, 2018.

[112] Thomas J Lane, Christian R Schwantes, Kyle A Beauchamp, and Vijay S Pande. Efficient inference of protein structural ensembles. *arXiv preprint arXiv:1408.0255*, 2014.

[113] Anders B Norgaard, Jesper Ferkinghoff-Borg, and Kresten Lindorff-Larsen. Experimental parameterization of an energy function for the simulation of unfolded proteins. *Biophys. J.*, 94(1):182–192, 2008.

[114] Da-Wei Li and Rafael Brüschweiler. Iterative optimization of molecular mechanics force fields from NMR data of full-length proteins. *J. Chem. Theory Comput.*, 7(6):1773–1782, 2011.

[115] Lee-Ping Wang, Todd J Martinez, and Vijay S Pande. Building force fields: an automatic, systematic, and reproducible approach. *J. Phys. Chem. Lett.*, 5(11):1885–1891, 2014.

[116] Andrew D White and Gregory A Voth. Efficient and minimal method to bias molecular simulations with experimental data. *J. Chem. Theory Comp.*, 10(8):3023–3030, 2014.

[117] Andrea Cesari, Sandro Bottaro, Kresten Lindorff-Larsen, Pavel Banáš, Jiří Šponer, and Giovanni Bussi. Fitting corrections to an RNA force field using experimental data. *J. Chem. Theory Comp.*, 15(6):3425–3431, 2019.

[118] Jürgen Köfinger and Gerhard Hummer. Empirical optimization of molecular simulation force fields by Bayesian inference. *Eur. Phys. J. B*, 94(12):245, 2021.

[119] Valerio Piomponi, Thorben Fröhlking, Mattia Bernetti, and Giovanni Bussi. Molecular simulations matching denaturation experiments for N6-methyladenosine. *ACS Cent. Sci.*, 8(8):1218–1228, 2022.

[120] Peter G Bolhuis, Z Faidon Brotzakis, and Bettina G Keller. Optimizing molecular potential models by imposing kinetic constraints with path reweighting. *J. Chem. Phys.*, 159(7), 2023.

[121] Felix Kümmerer, Simone Orioli, and Kresten Lindorff-Larsen. Fitting force field parameters to NMR relaxation data. *J. Chem. Theory Comput.*, 2023.

[122] Tone Bengtsen, Viktor L Holm, Lisbeth Ravnkilde Kjølbye, Søren R

Midtgaard, Nicolai Tidemand Johansen, Giulio Tesei, Sandro Bottaro, Birgit Schiøtt, Lise Arleth, and Kresten Lindorff-Larsen. Structure and dynamics of a nanodisc by integrating NMR, SAXS and SANS experiments with molecular dynamics simulations. *Elife*, 9:e56518, 2020.

[123] Samuel Schoenholz and Ekin Dogus Cubuk. Jax MD: a framework for differentiable physics. *Advances in Neural Information Processing Systems*, 33:11428–11441, 2020.

[124] Xinyan Wang, Jichen Li, Lan Yang, Feiyang Chen, Yingze Wang, Junhan Chang, Junmin Chen, Wei Feng, Linfeng Zhang, and Kuang Yu. DMFF: an open-source automatic differentiable platform for molecular force field development and molecular dynamics simulation. *Journal of Chemical Theory and Computation*, 19(17):5897–5909, 2023.

[125] Paul Fuchs, Stephan Thaler, Sebastien Röcken, and Julija Zavadlav. chemtrain: Learning deep potential models via automatic differentiation and statistical physics. *Computer Physics Communications*, page 109512, 2025.

[126] Ivan Gilardoni, Thorben Fröhlking, and Giovanni Bussi. Boosting ensemble refinement with transferable force-field corrections: Synergistic optimization for molecular simulations. *J. Phys. Chem. Lett.*, 15(5):1204–1210, 2024.

[127] Lee-Ping Wang, Jiahao Chen, and Troy Van Voorhis. Systematic parametrization of polarizable force fields from quantum chemistry data. *J. Chem. Theory Comput.*, 9(1):452–460, 2012.

[128] Thorben Fröhlking, Mattia Bernetti, Nicola Calonaci, and Giovanni Bussi. Toward empirical force fields that match experimental observables. *J. Chem. Phys.*, 152(23):230902, 2020.

[129] Thomas Dannenhoffer-Lafage, Andrew D White, and Gregory A Voth. A direct method for incorporating experimental data into multiscale coarse-grained models. *J. Chem. Theory Comput.*, 12(5):2144–2153, 2016.

[130] Jason D Tubbs, David E Condon, Scott D Kennedy, Melanie Hauser, Philip C Bevilacqua, and Douglas H Turner. The nuclear magnetic resonance of CCCC RNA reveals a right-handed helix, and revised parameters for AMBER force field torsions improve structural predictions from molecular dynamics. *Biochemistry*, 52(6):996–1010, 2013.

[131] Ilyas Yildirim, Harry A Stern, Jason D Tubbs, Scott D Kennedy, and Douglas H Turner. Benchmarking AMBER force fields for RNA: Comparisons to NMR spectra for single-stranded r(GACC) are improved by revised χ torsions. *J. Phys. Chem. B*, 115(29):9261–9270, 2011.

[132] Jianbo Zhao, Scott D Kennedy, Kyle D Berger, and Douglas H Turner. Nuclear magnetic resonance of single-stranded RNAs and DNAs of CAAU and UCAAUC as benchmarks for molecular dynamics simulations. *J. Chem. Theory Comp.*, 16(3):1968–1984, 2020.

[133] Jianbo Zhao, Scott D Kennedy, and Douglas H Turner. Nuclear magnetic resonance spectra and AMBER OL3 and ROC-RNA simulations of UCUCGU reveal force field strengths and weaknesses for single-stranded RNA. *J. Chem. Theory Comp.*, 18(2):1241–1254, 2022.

[134] Jes Frellsen, Ida Moltke, Martin Thiim, Kanti V Mardia, Jesper Ferkinghoff-Borg, and Thomas Hamelryck. A probabilistic model of RNA conformational space. *PLoS computational biology*, 5(6):e1000406, 2009.

[135] Bradley Efron and Robert Tibshirani. An introduction to the bootstrap. Chapman & Hall, New York, 1993.

[136] Alejandro Gil-Ley, Sandro Bottaro, and Giovanni Bussi. Empirical corrections to the AMBER RNA force field with target metadynamics. *J. Chem. Theory Comp.*, 12(6):2790–2798, 2016.

[137] Jun Chen, Hao Liu, Xiaochen Cui, Zhengxin Li, and Hai-Feng Chen. RNA-specific force field optimization with CMAP and reweighting. *J. Chem. Inf. Model.*, 62(2):372–385, 2022.

[138] Vojtěch Mlýnský, Petra Kührová, Tomáš Kühr, Michal Otyepka, Giovanni Bussi, Pavel Banáš, and Jiri Šponer. Fine-tuning of the AMBER RNA force field with a new term adjusting interactions of terminal nucleotides. *J. Chem. Theory Comp.*, 16(6):3936–3946, 2020.

[139] Tongye Shen and Donald Hamelberg. A statistical analysis of the precision of reweighting-based simulations. *J. Chem. Phys.*, 129(3), 2008.

[140] Andrea Cavalli, Carlo Camilloni, and Michele Vendruscolo. Molecular dynamics simulations with replica-averaged structural restraints generate structural ensembles according to the maximum entropy principle. *J. Chem. Phys.*, 138(9):094112, 2013.

[141] Massimiliano Bonomi, Carlo Camilloni, Andrea Cavalli, and Michele Vendruscolo. Metainference: A Bayesian inference method for heterogeneous systems. *Sci. Adv.*, 2(1):e1501177, 2016.

[142] Ivan Gilardoni, Valerio Piomponi, Thorben Fröhlking, and Giovanni Bussi. MDRefine: A Python package for refining molecular dynamics trajectories with experimental data. *The Journal of chemical physics*, 162(19):192501, 2025.

[143] Charles R Harris, K Jarrod Millman, Stéfan J Van Der Walt, Ralf Gommers, Pauli Virtanen, David Cournapeau, Eric Wieser, Julian Taylor, Sebastian Berg, Nathaniel J Smith, et al. Array programming with NumPy. *Nature*, 585(7825):357–362, 2020.

[144] Pauli Virtanen, Ralf Gommers, Travis E Oliphant, Matt Haberland, Tyler Reddy, David Cournapeau, Evgeni Burovski, Pearu Peterson, Warren Weckesser, Jonathan Bright, et al. SciPy 1.0: fundamental algorithms for scientific computing in Python. *Nat. Methods*, 17(3):261–272, 2020.

[145] James Bradbury, Roy Frostig, Peter Hawkins, Matthew James Johnson, Chris Leary, Dougal Maclaurin, George Necula, Adam Paszke, Jake VanderPlas, Skye Wanderman-Milne, and Qiao Zhang. JAX: composable transformations of Python+NumPy programs, 2018.

[146] Michael R Shirts, David L Mobley, and John D Chodera. Alchemical free energy calculations: ready for prime time? *Annual reports in computational chemistry*, 3:41–59, 2007.

[147] Robert W Zwanzig. High-temperature equation of state by a perturbation method. I. nonpolar gases. *J. Chem. Phys.*, 22(8):1420–1426, 1954.

[148] Jonathan Lorraine, Paul Vicol, and David Duvenaud. Optimizing millions of hyperparameters by implicit differentiation. In *Int. Conf. Artif. Intell. Stat.*, pages 1540–1552. PMLR, 2020.

[149] Andrew D White, James F Dama, and Gregory A Voth. Designing free energy surfaces that match experimental data with metadynamics. *J. Chem. Theory Comput.*, 11(6):2451–2460, 2015.

[150] Fabrizio Marinelli and José D Faraldo-Gómez. Ensemble-biased metadynamics: A molecular simulation method to sample experimental distributions. *Biophys. J.*, 108(12):2779–2782, 2015.

[151] Glen M Hocky, Thomas Dannenhoffer-Lafage, and Gregory A Voth. Coarse-grained directed simulation. *J. Chem. Theory Comput.*, 13(9):4593–4603, 2017.

[152] Christopher I Bayly, Piotr Cieplak, Wendy Cornell, and Peter A Kollman. A well-behaved electrostatic potential based method using charge restraints for deriving atomic charges: the RESP model. *J. Phys. Chem.*, 97(40):10269–10280, 1993.

[153] Caroline Roost, Stephen R Lynch, Pedro J Batista, Kun Qu, Howard Y Chang, and Eric T Kool. Structure and thermodynamics of N6-methyladenosine in RNA: a spring-loaded base modification. *J. Am.*

Chem. Soc., 137(5):2107–2115, 2015.

[154] Luca Giorgetti, Rafael Galupa, Elphège P Nora, Tristan Piolot, France Lam, Job Dekker, Guido Tiana, and Edith Heard. Predictive polymer modeling reveals coupled fluctuations in chromosome conformation and transcription. *Cell*, 157(4):950–963, 2014.

[155] Guido Tiana, Assaf Amitai, Tim Pollex, Tristan Piolot, David Holcman, Edith Heard, and Luca Giorgetti. Structural fluctuations of the chromatin fiber within topologically associating domains. *Biophys. J.*, 110(6):1234–1245, 2016.

[156] Bin Zhang and Peter G Wolynes. Topology, structures, and energy landscapes of human chromosomes. *Proc. Natl. Acad. Sci. U.S.A.*, 112(19):6062–6067, 2015.

[157] Bin Zhang and Peter G Wolynes. Shape transitions and chiral symmetry breaking in the energy landscape of the mitotic chromosome. *Phys. Rev. Lett.*, 116(24):248101, 2016.

[158] Christoph Müller-Hermes, Valerio Piomponi, Stefan Hilber, Sam Asami, Christoph Kreutz, Giovanni Bussi, and Michael Sattler. Unique conformational dynamics and protein recognition of A-to-I hyper-edited dsRNA. *Nucleic Acids Research*, 53(12):gkaf550, 2025.

[159] Dominic Rufa, Joshua Fass, and John D Chodera. Fine-tuning molecular mechanics force fields to experimental free energy measurements. *bioRxiv*, pages 2025–01, 2025.

[160] Kresten Lindorff-Larsen, Robert B Best, Mark A DePristo, Christopher M Dobson, and Michele Vendruscolo. Simultaneous determination of protein structure and dynamics. *Nature*, 433(7022):128–132, 2005.

[161] Benoît Roux and Jonathan Weare. On the statistical equivalence of restrained-ensemble simulations with the maximum entropy method. *J. Chem. Phys.*, 138(8):084107, 2013.

[162] Harold Jeffreys. An invariant form for the prior probability in estimation problems. *Proceedings of the Royal Society of London. Series A. Mathematical and Physical Sciences*, 186(1007):453–461, 1946.

[163] Edwin T Jaynes. Prior probabilities. *IEEE Transactions on Systems Science and Cybernetics*, 4(3):227–241, 1968.

[164] Gavin E Crooks. Measuring thermodynamic length. *Physical Review Letters*, 99(10):100602, 2007.

[165] Wai Shing Tang, David Silva-Sánchez, Julian Giraldo-Barreto, Bob Carpenter, Sonya M Hanson, Alex H Barnett, Erik H Thiede, and Pilar Cossio. Ensemble reweighting using cryo-EM particle images. *The*

Journal of Physical Chemistry B, 127(24):5410–5421, 2023.

[166] Z Faidon Brotzakis, Michele Vendruscolo, and Peter G Bolhuis. A method of incorporating rate constants as kinetic constraints in molecular dynamics simulations. *Proc. Natl. Acad. Sci. U.S.A.*, 118(2):e2012423118, 2021.

[167] Carl G Henning Hansen, Simone Orioli, and Kresten Lindorff-Larsen. A bayesian approach to interpret time-resolved experiments using molecular simulations. *bioRxiv*, pages 2025–07, 2025.

[168] Stephan Thaler, Gregor Doehner, and Julija Zavadlav. Scalable bayesian uncertainty quantification for neural network potentials: promise and pitfalls. *Journal of Chemical Theory and Computation*, 19(14):4520–4532, 2023.