A kinematic view of protein loop flexibility, with applications to conformational exploration

Timothée O'Donnell

UNIVERSITÉ CÔTE D'AZUR

ÉCOLE DOCTORALE
SCIENCES ET TECHNOLOGIES DE L'INFORMATION ET DE LA COMMUNICATION

THÈSE DE DOCTORAT

Une approche cinématique de la flexibilité des boucles protéiques, avec application à l'exploration conformationnelle

Timothée O'Donnell

ABS, Inria Sophia Antipolis Méditerranée

Présentée en vue de l'obtention du grade de docteur en Informatique d'Université Côte d'Azur

Directeur: Frédéric Cazals

Co-directeur: Bernard Delmas

Soutenue le: 01/06/2022

Devant le jury, composé de:

Frédéric Cazals, Directeur de Recherche, Inria
Bernard Delmas, Directeur de Recherche, INRAe
J. Cortés, Directeur de recherche, CNRS
R. Dunbrack, Professeur, Temple University
M. Blackledge, Directeur de recherche, CNRS
C. H. Robert, Directeur de recherche, CNRS
A. Lopes, Enseignant chercheur, Université Paris-Saclay
H. Berry, Directeur de recherche, Inria

Une approche cinématique de la flexibilité des boucles protéiques, avec application à l'exploration conformationnelle

Jury:

Directeur

F. Cazals	Directeur de Recherche, Inria	Directeur
B. Delmas	Directeur de recherche, INRAe	co-Directeur

Rapporteurs

J. Cortés	Directeur de recherche, CNRS
R. Dunbrack	Professeur, Temple University

Examinateurs

M. Blackledge	Directeur de recherche, CNRS
C. H. Robert	Directeur de recherche, CNRS
A. Lopes	Enseignant chercheur, Université Paris-Saclay

Invité

H. Berry	Directeur de recherche, Inria

Une approche cinématique de la flexibilité des boucles protéiques, avec application à l'exploration conformationnelle

Cette thèse introduit le premier modèle paramétrique global d'une boucle de protéine, qui soit passible de stratégies d'échantillonnage de type Hit-and-Run. Quatre contributions sont présentées.

Partant du problème classique de fermeture cinématique d'une boucle tripeptidique par Coutsias et al (Tripeptide Loop Closure ou TLC), la première présente une analyse géométrique de TLC utilisant un espace angulaire de dimension 12. Des conditions nécessaires assez strictes sont développées, afin que TLC admette des solutions.

En utilisant une base de données exhaustive de tripeptides extraite de la Protein Data Bank (PDB), la seconde contribution étudie les reconstructions produites par TLC. En utilisant des statistiques de Ramachandran, il est montré que ces solutions sont géométriquement diverses, et ont par ailleurs des énergies potentielles favorables.

Afin d'échantillonner des conformations d'une boucle contenant plus de trois acides aminés, la troisième contribution développe une stratégie d'échantillonnage basée sur les solutions individuelles associées aux tripeptides formant la boucle, tripeptides dont la géométrie est conditionnée par les positions des corps rigides les connectant. Les conditions nécessaires évoquées ci-dessus sont utilisées pour échantillonner des conformations, en utilisant un algorithme randomisé de type Hit-and-Run.

Enfin, pour aller au delà de la seule géométrie du backbone, la dernière contribution présente le premier algorithme robuste du calcul de la moyenne de Fréchet/du centre de masse sur le cercle S^1, une statistique clef pour l'étude des conformations de chaînes latérales.

Mots-clés: flexibilité des protéines, exploration conformationnelle, boucles flexibles, coordonnées internes, cinématique.

A kinematic view of protein loop flexibility, with applications to conformational exploration

This thesis introduces the first global parametric model of protein loops amenable to effective sampling strategies a-la Hit-and-Run, making four contributions.

Starting with the classical kinematic view of loop closure developed by Coutsias et al, the first one resides in a geometric analysis of the Tripeptide Loop Closure (TLC) problem in terms of 12 angular coordinates describing the tripeptide geometry. Tight necessary conditions in this angular space are derived for TLC to admit solutions.

Using an exhaustive database of tripeptides from the Protein Data Bank, the second contribution studies TLC reconstructions. Using Ramachandran statistics, we show that TLC solutions are geometrically more diverse than tripeptide structures, and also exhibit favorable potential energies.

To sample protein loop conformations beyond tripeptides, the third contribution develops a loop sampling strategy based on solutions of individual tripeptide reconstructions, conditioned to motions of the peptide bodies connecting them. The aforementioned necessary conditions are used to sample conformations using a randomized algorithm reminiscent from Hit-and-Run.

Finally, to go beyond the sole backbone geometry, the last contribution proposes the first robust calculation of the Fréchet mean/center of mass on the flat torus, a key requirement to compute statistics on protein side chains.

Keywords: protein flexibility, conformational exploration, flexible loops, internal coordinates, kinematics.

Acknowledgments

I would like to thank a number of people for contributing to this work. My supervisor at Inria Frédéric Cazals for the time spent on helping me completing this thesis particularly when it came to writing where he went beyond what I could ask of him. My supervisor at INRAe Bernard Delmas for the understanding and patience. My colleagues who contributed to improve the work environment. In the context of the work produced I would like to thank Théo Roncalli for his contribution in the form of figures and Maximilien Martin for his help with algebra.

I would like to thank Juan Cortés and Roland Dunbrack for the time taken out of their busy schedule to review my work. On the same note I would like to thank the remaining members of the jury Anne Lopes, Charles Robert, Martin Blackledge, and Hugues Berry.

Surtout je voudrais remercier ma mère pour m'avoir permis d'arriver la ou j'en suis aujourd'hui malgré des circonstances difficiles. Pour toutes les choses blessantes que j'ai pu te dire par maladresse ou colère mal placée sache que je les regrette et que je t'aime.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Loops in computational structural biology

1.1.1 Computational Structural Biology at a glance

Computational structural biology involves two related main endeavors, namely analyzing structures solved experimentally, and making predictions of observables, be they structural, thermodynamic, or kinetic [BKP88, Fer99, KKW12]. Both tasks rely on a geometric representations of molecules, and in fact, the previous two activities are only meant to understand to what extent biophysics biases geometry. In other words, the main goal is to understand how the physical rules bias the geometric representations observed in nature. To understand this perspective, one may consider the classical and most natural representation of an n-atom molecule, based on its $3n$ Cartesian coordinates. As we shall see, this representation is not the best one to understand the specific interactions of atoms which are (or not) covalently bonded, and a more efficient representation to do so consists of using so-called internal coordinates [Fie99, BMRW01].

A shear difficulty inherent to the study of molecular representations is the high dimensionality of molecular systems–the aforementioned $3n$ Cartesian coordinates, and the multi-scale of biomolecular processes. Molecular motions are indeed known to span ~ 15 and ~ 4 orders of magnitude in time and amplitude respectively [AM06]. Despite intensive efforts over the past fifty years or so, developing methods able to exploit this multi-scale structure has remained elusive.

The structure - dynamics - function paradigm stipulates that it is the structure and dynamics of biomolecules which account for their function.

Three broad classes of methods have been developed. The first one relies on Newton's equations, whose numerical solution uses time steps of the order of femto-seconds [FS02]. Alas, unless massive simulations are used [SMLL$^+$10] such tiny time steps are prohibitive for simulations with large systems , or systems undergoing large amplitude conformational changes. The second one, encompassing Monte Carlo based methods and basin-hopping like methods [FS02, Wal03], require *movesets* to propose novel conformations, which are accepted or not to further the simulation. The last one is the framework of energy landscapes [Wal03], which decouples structure (identifying meta-stable states), thermodynamics (computing statistical weights of such states), and dynamics/kinetics (modeling transitions using say Markov state models). The latter two classes of methods are appealing since arbitrarily large spatial steps may be used, a shear difficulty is to avoid steric clashes and retain low (potential) energy conformations. This latter constraint is a strong incentive to work in internal coordinates, favoring dihedral angles which are *softer* coordinates than bond lengths and valence angles.

In developing molecular representations, a central goal is therefore to find sparse representations giving access to those degrees of freedom accounting for important properties, *i.e.* observables.

Finally, the study of molecular representations encompasses two aspects, generic and specific: the former deals with representations which are valid throughout chemistry, while the latter deals with models which are specific to biomolecules and/or to proteins. The very nature of polypeptide chains, namely a polymer of

amino acids, indeed calls for representations dedicated to the protein backbone and to the side chains.

1.1.2 Protein loops

Proteins are in general built from well structure 3D domains connected by linkers called *loops* [BT12]. Loops are structural components playing various roles in protein function. Enzymes typically involve conformational changes of loops for the substrate (resp. product) to enter (resp. leave) the active site [MAR10a]. Membrane transporters implement complex efflux mechanisms resorting to loops changing the relative position of (essentially) rigid domains [SBMVC21]. In the humoral immune response, the binding affinity of antibodies for antigens is modulated by the dynamics of loops called complementarity determining regions (CDRs) [SXK$^+$13]. In G-Protein-Coupled Receptors, extracellular loops binding to ligands trigger signal transduction inside the cell [HMK18].

From the experimental standpoint, these complex phenomena are studied using structure determination methods. However, the structural diversity of loops often results in a low signal to noise ratio, yielding difficulties to report complete polypeptide chains. As a matter of fact, a recent study on structures from the PDB showed that about 83% of structures solved at a resolution of 2.0Å or worse feature missing regions, which for 90% of them are located on loops or unstructured regions [DCC15].

1.1.3 Flexibility and dynamics of loops

While flexibility covers very different scenarios, two prototypical ones are of special interest for globular proteins involving loops. In the first scenario, which may be ascribed to structural changes, flexibility drives large amplitude conformation changes between meta-stable states involving rigid domains connected by linkers [LRH03], a process which is key for enzymatic function [QH09] or the efflux by complex membrane proteins [SBMVC21], to take two examples. In the second one, which may be ascribed to thermodynamics, more local fluctuations of loops contribute to statistical weights whence free energies, a classical implication being an enhanced binding affinity due to a lesser entropic penalty upon binding for pre-structured loops [SXK$^+$13]. A different realm is that of intrinsically disordered proteins (IDPs), whose structural plasticity is often linked to biological functions and diseases [DSUS08]. IDPs exist as an ensemble of rapidly inter-converting structures defining plateaus on the free energy landscape as opposed to the wells associated with stable structures [Uve13]. While differences with globular proteins in terms of Ramachandran distributions have been characterized [OSY$^+$12], predicting IDPs properties remains a challenge, and there has been recent awareness of the need for force field modifications (e.g. [Lem20]).

Predicting conformational changes for loops is in fact a hard problem, be it restricted to structure [MSD18] or thermodynamics [SXK$^+$13]. A core difficulty for such prediction methods is the inherent bias imposed by the datasets, extracted from the Protein Data Bank, used to calibrate general methods. By construction, experimentally resolved structures incur a bias towards stable structures, so that transient conformations are not accessible. We note in passing that in the aforementioned framework of energy landscapes, transient conformations are generally associated with saddle point regions on the potential energy surface, namely points whose identification requires numerical procedures [STH08].

1.1.4 Loop modeling strategies

While all atom simulations can naturally be used to explore the conformational variability of loops, their prohibitive cost prompted the development of simplified strategies, which we may ascribed to four tiers.

First, continuous geometric transformations can be used to deform loops, e.g. based on rotations of rigid backbone segments sandwiched between two C_α carbons. Such methods, which include `Crankshaft` [Bet05] and `Backrub` [DAIRR06, SK08], proved effective to reproduce motions observed in crystal structures. However, they are essentially limited to hinge like motions.

Second, a loop may be deformed using loop closure techniques solving an inverse problem which consists of finding the geometric parameters of the loop so that its endpoints obey geometric constraints. Remarkably, various such methods have been developed at the interface of structural biology and robotics [GS70, EM99,

CSRST04, NOS05, PRT$^+$07]. Using loop closure techniques, the seminal concept of *concerned rotations* was introduced long ago to sample loop conformations [DBT93]: first, the prerotation stage changes selected internal degrees of freedom (dof) and brakes loop connectivity; second, the postrotation step restores loop closure using a second set of dof. While early such strategies used solely dihedral angles only [DBT93], more recent ones use a combination of valence and dihedral angles [UJ03, BBEJ$^+$12]. The latter angles indeed provide a finer control on the the amplitude of angular changes in the postrotation stage, and therefore of atomic displacements. A specific type of loop closure playing an essential role is Tripeptide Loop Closure (TLC), where the gap consists of three amino, and loop closure is obtained using the six (ϕ, ψ) angles of the three C_α carbons [CSJD04, CS04, NOS05, MCK09].

Third, considering a loop as a sequence of protein fragments stitched together, high resolution structures from the protein data bank (PDB) can be used to sample its conformations [JT86, KGLK05]. These methods are greedy/incremental in nature, and the exponential growth of solutions results in a poorer sampling of residues in the middle of the loop. Also, they suffer from the bias inherent to the PDB structures, which favors meta-stable conformations. As a matter of fact, it has been shown recently using Ramachandran statistics that conformations found in the PDB are less diverse than those yielded by reconstructions in the rigid geometry model [ORC22].

Finally, several classes of methods may be combined. For example, exploiting structural data to bias the choices of angles used to perform loop closure yields a marked improvement in prediction accuracy [SK13]. More recently, a method growing the two sides of a loop by greedily concatenating (perturbed) tripeptides, before closing the loop using TLC has been proposed [BMV$^+$19].

Despite intensive research efforts, predicting large amplitude conformational changes, and/or predict thermodynamic quantities for long loops, say beyond 12 amino acids, remains a challenge [MSD18, BCC21]. These difficulties owe to the high dimensionality of loop conformational space, and also to the subtle biophysical constraints that must be obeyed.

Improving on such methods is precisely the goal of this thesis. These strategies are discussed in chapter 2 and we propose our own novel solution in chapter 5.

1.2 Contributions

This thesis presents five contributions.

1.2.1 Modeling proteins using internal coordinates: a survey

Context

Although molecular systems can be represented using Cartesian coordinates there are disadvantages. Superimposable sets of Cartesian coordinates will be considered different to one another for instance, causing issues in data analysis and efficiency loss in exploration using this type of coordinates. Seeking to impact mostly the *softer* softer diherdral angles is more in line with the nature of force fields used to model protein energies.

To do this the focus turns to so called Internal coordinates (IC) *i.e.* bond lengths, valence angles and dihedral angles defined by the molecular covalent graph. These enable more efficient exploration of conformational diversity.

As such internal coordinates are ubiquitous in the manipulation of molecular structures, be it in the context of the analysis of structures from the Protein Data Bank, or in simulation packages. This diversity of methods and models developed on the subject calls for a thorough survey of this field of research.

Contribution

In Chapter 2, we propose a survey of the state of the art concerning the modeling of proteins using internal coordinates.

This survey covers the representation of the protein backbone and side chains, (Sec. 2.3), statistical and geometric analysis in angular spaces (Sec. 2.4), rotamer libraries (Sec. 2.5), side chain conformational sampling (Sec. 2.6) and backbone conformation sampling and its important connexions to inverse problems in robotics (Sec. 2.7).

1.2.2 Tripeptide Loop Closure and steric constraints

Context

As noted above, a key building block in this context is the celebrated Tripeptide Loop Closure (TLC) algorithm [PRT$^+$07, CSJD04, NOS05]. In TLC, one consider three consecutive amino acids, with two types of constraints. The first one stipulates that the segments (also called *anchors* or *legs* $N_iC_{\alpha;i}$ and $C_{\alpha;i+2}C_{i+2}$ are fixed in a given reference frame. The second one imposes that all internal coordinates are fixed, except the six rotatable bonds / dihedral angles $\{(\phi_i, \psi_i)\}_{i=1,2,3}$ found before / after the three C_α carbons. Remarkably, TLC admits at most 16 solutions. This property sorts of discretizes the search space when running a simulation, in a manner akin to rotameric degrees of freedom for side chains.

Using TLC in the context of backbone simulations raises a difficulty, though. To see which, consider $m(> 1)$ consecutive tripeptides. To use TLC on each tripeptide and mix their solutions if any, one needs to fix the position of all anchors. For a given tripeptide, this raises a novel question which is to study the existence of solutions to TLC when the second anchors moves relatively to the first one.

Contribution

In Chapter 3, we study the TLC problem when the legs of the tripeptide are free to move.

More specifically, consider a tripeptide in which all internal coordinates (but the 6 dihedral angles) hold canonical values. Assuming that the two segments $N_1C_{\alpha;1}$ and $C_{\alpha;3}C_3$ are free to move with respect to one another, we aim at finding necessary conditions on these two segments for TLC to admit solutions. A tripeptide yielding solutions is termed *embeddable*. As we can assume without loss of generality that the first segment is fixed in a reference frame, this problem is posed in a five dimensional configuration space: the position of $C_{\alpha;3}$ enjoys 3 Cartesian coordinates, and that of C_3 two spherical coordinates w.r.t. $C_{\alpha;3}$. The question becomes to find out which positions of $C_{\alpha;3}$ and C_3 yielding embeddable tripeptides.

In chapter 3, our contribution is made in an attempt to answer which constraints yield embeddings when using TLC. To answer this, we exploit the limit values for angle σ_{i-1} and τ_i defined in [CSJD04] and combine them with a constraint associated with each $C_{\alpha;i}C_{\alpha;i+1}$ edge. This allows us to derive an Inter-angular constraint, a necessary condition for the existence of solutions.

1.2.3 Tripeptide Loop Closure and associated solutions

Context

The TLC problem is also closely related to the study of Ramachandran distributions, which characterize the coupling between ϕ and ψ angles along the protein backbone [Fer99]. There are four main types of Ramachandran plots: glycine – an amino acid without side chain, proline – whose cycle induces specific constraints, pre-proline – residues preceding a proline, and the remaining amino acids, whose C_β carbon induces specific constraints. In our work, we illustrate this latter class with ASP. Four main regions are occupied in the Ramachandran diagram: β-sheets (βS), polyproline II (βP; left-handed helical structure whose angles are characteristic of β-strands); α-helical (αR); and left handed helix (αL). These regions were characterized using a combination of five steric constraints between four atoms defining the so-called Ramachandran tetrahedron [STM$^+$77].

More recently the diagonal shape of level set curves in the occupied regions was explained using dipole-dipole interactions, distinguishing the generic case and proline [HTB03], and glycine and pre-proline [HB05]. The characterization of neighbor dependent Ramachandran distributions has also been studied [TWS$^+$10].

From a statistical standpoint, the Ramachandran distributions of two specific residues can be compared using say f-divergences such as Kullback-Leibler, Hellinger, etc.

Contribution

In chapter 4, we perform a careful assessment of reconstructions to TLC problems, with a particular emphasis on the comparison between distributions in angular spaces, between data from the PDB on the one hand, and TLC reconstructions on the other hand. We present a detailed analysis of reconstruction, from the geometric, statistical, and biophysical standpoints. We also present a robust implementation of TLC, showing the role of multiprecision in ensuring the existence and the accuracy of reconstructions. We also discuss some possibilities to exploit such reconstructions.

1.2.4 Enhanced conformational exploration of protein loops

Context

Generating diverse conformations of loops requires sampling the conformational space. Because dihedral angles are in general softer than bond lengths and valence angles, methods of choice are those restricting the sampling to the former. Narrowing down the focus further, the tripeptide loop closure problem (TLC) consider the six dihedral angles ϕ, ψ found in the context of three consecutive C_α carbons. The TLC problem has a long history in robotics and molecular modeling, see e.g. [GS70, PC94, CS04, PRT$^+$07, CSWD06, CLW$^+$16]. Mathematically, consider a tripeptide whose internal coordinates (bond lengths $\{d_i\}$, valence angles $\{\theta_i\}$, and dihedral angles $\{\phi_i, \psi_i, \omega_i\}$) have been extracted. The TLC problem consists of finding all geometries of the tripeptide backbone compatible with the internal coordinate values $\{d_i, \theta_i\}$. Solving the problem requires finding the real roots of a degree 16 polynomial, which also means that up to 16 solutions may be found [PRT$^+$07, CSJD04, NOS05].

Over time, TLC has proven to be a key building block to reconstruct and sample loop conformations, as shown by the following two examples. In `Rosetta`, a so-called *KIC move* consists of closing a backbone segment using TLC upon sampling six dihedral angles associated with three C_α carbons, using residue specific distributions [MCK09]. This method was subsequently evolved to the *next-generation KIC* based on three sampling strategies meant to optimize internal coordinates, while still using TLC to close the loop [SK13]. More recently, TLC has been used to generate conformations based on a backbone segment decomposition into tripeptides [BMV$^+$19]. In a nutshell, the method grows the two sides of a loop by greedily concatenating (perturbed) tripeptide geometries to the chains being elongated, and closes the loop by solving a TLC problem. Two key steps of the method are the perturbation and sampling from a database of tripeptides used (derived from SCOP), and the final TLC step.

Contribution

In chapter 5 we put forward a new paradigm to explore the conformational space of flexible protein loops, able to deal with loop length that were out of reach. The framework is reminiscent from the Hit-and-Run (HAR) Markov chain Monte Carlo technique.

While it also relies on the tripeptide loop closure, it is, to the best of our knowledge, the first one exploiting a global continuous parameterization of the conformational space on the loop studied. The algorithm uses a decomposition of the loop into tripeptides, and exploits the rigidity of peptide bodies (the four atoms $C_\alpha - C - N - C_\alpha$). Denoting m the number of tripeptides, the algorithm works in an angular space of dimension $12m$. In this space, the hyper-surfaces associated with the necessary conditions developed in chapter 3 are used to run a HAR-like sampling technique.

1.2.5 Fréchet mean for angular values and generalizations

Context

The celebrated center of mass of a point set P in a Euclidean space is the (a) point minimizing the sum of squared Euclidean distances to points in P. The center of mass plays a key role in data analysis at large, and in particular in principal components analysis since the data are centered prior to computing the covariance matrix and the principal directions. Generalizing these notions to non Euclidean spaces is an active area of research. Motivated by applications in structural biology (molecular conformations), robotics (robot conformations), and medicine (shape and relative positions of organs), early work focused on direct generalizations of Euclidean notions. Analysis tailored to the unit circle and sphere were developed under the umbrella of directional statistics [AJ91, MT93, MJ09]. In a more abstract setting, generalizations of the center of mass in general metric spaces were first worked out – the so-called Fréchet mean [Fré48], followed by a generalization to distributions on such spaces – the so-called Karcher mean [GK73, AM14, Pen18].

In fact, previous works span two complementary directions. On the one hand, efforts have focused on mathematical properties of spaces generalizing affine spaces, so as to provide statistical summaries of ensembles in terms of geometric objects of small dimension. On the other hand, algorithmic developments have been performed proposed to compute such objects. The case of the unit circle S^1 provides the simplest compact non Euclidean manifold to be analyzed. Despite its simplicity, this case turns out to be of high interest since S^1 encodes angles, a particularly important case e.g. to describe molecular conformations.

Finally, the last thing to consider in the context of co;puting the center of mass on S^1 are numerical issues, more specifically the necessity of using the Exact Geometric Computation (EGC) paradigm. The EGC relies on so-called *exact predicates* and *constructions*. A predicate is a function whose output belongs to a finite set, used to compare numbers together for instance while a construction computes a continuous value. Using exact predicates guaranties an exact result when comparing two numbers together, enabling the development of more robust algorithms.

Contribution

Three contributions can be found in chapter 6 regarding p-means the point minimizing the sum of distances exponent p of a finite point set. First, we show that the function F_p is determined by a very simple combinatorial structure, namely a partition of S^1 into circle arcs. Second, we give an explicit expression for F_p, deduce that the problem is decidable, and present an algorithm computing p-means. Third, we present an effective and robust implementation using EGC, based on multi-precision interval arithmetic.

Chapter 2

A survey on models using internal coordinates

2.1 Introduction

As mentionned in the introduction (Sec. 1.1.1) the high dimensionality of molecular systems, and the multi-scale of biomolecular processes makes molecular representations a central issue. Knwoing so, we compiled a survey of the state of the art with a particular focus on geometry and algorithms.

The structure of this chapter is as follows:

- Internal coordinates and conversions with Cartesian coordinates are introduced in section 2.2
- An introduction to the representation of the protein backbone and side chains is provided in section 2.3
- Statistical and geometric analysis in angular spaces are accounted for in section 2.4
- Rotamer libraries are surveyed in section 2.5
- Side chain conformational sampling is deal with in section 2.6
- Backbone conformation sampling and its important connexions to inverse problems in robotics is surveyed in section 2.7

2.2 Molecular geometry in internal coordinates

2.2.1 Overview

A system of n atoms can naturally be represented by $3n$ Cartesian coordinates $\{x_i, y_i, z_i\}_{i=1,\dots,n}$. Two observations, though, pledge for using the so-called internal coordinates (IC), which are distances and angles.

The first one is the fact a rigid motion (translation and/or rotation in 3D) modifies CC. This demands a representation intrinsically independent from rigid motions, based on $3n-6$ degrees of freedom–since the group of rigid motions $SE(3)$ has dimension six. The second one is the nature of force fields used to model protein energies (Sec. 2.3), since distances and angles defined by atoms come with very different energies and forces.

This elementary section introduces molecular coordinates, as well as conversions between them.

2.2.2 The three components of internal coordinates

Bond lengths, valence angles, dihedral angles

Bonds are defined by two points connected in the molecular covalent graph (Fig. 2.1(A)).

Valence angles are defined by around a particle participating in two such bonds, thereby defining an angle (Fig. 2.1(B)).

The last type of internal coordinate corresponds to the angle between the two planes defined by the first and last three particles in a path of four (Fig. 2.2(A)). These angles are called dihedral or torsion angles they are the main degrees of freedom. What is used in internal coordinates are so called *proper dihedral angles* (Fig. 2.2(B)), improper angles being used in potential energy computations but not as a part of classical internal coordinate systems. We will get back on these coordinates in the context of potential energy models – Sec. 2.3.1.

Figure 2.1 Internal coordinates: bond length and valence angle. (A) Bond length (B) Valence angle, The black lines represent covalent bonds in a given molecular graph.

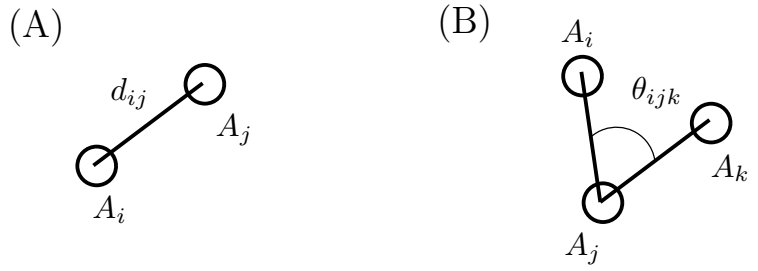
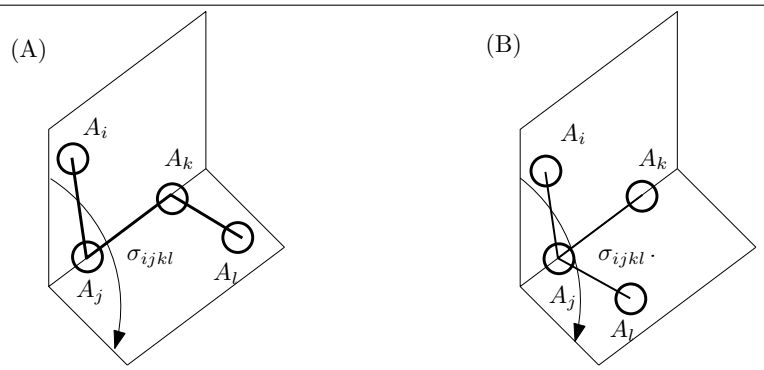


Figure 2.2 Internal coordinates: dihedral angles. The black lines represent covalent bonds in a given molecular graph. **(A)** Proper dihedral angle: defined by a path of four successive particles. **(B)** Improper dihedral angle defined by three particles connected to a common one: this *out-of-plane* measurement is meant to keep a planar structure planar.



2.2.3 Representations using internal coordinates

The set of all internal coordinates is in general redundant, due to the presence of cycles. To remedy this fact and serve in particular algorithms used for energy minimization, several representations using bond lengths, valence and dihedral angles have been designed.

Primitive internal coordinates

Although fictive edges are sometimes used in the absence of a connected graph internal coordinates are generally defined using the covalent graph of the molecule. These coordinates however suffer from redundancy if all are taken, with $3n-6$ coordinates being sufficient to uniquely define an embedding. We refer to such redundant internal coordinates as primitive internal coordinates.

Their advantage over a non redundant set of coordinates is that they are unambiguously defined. Deciding a non redundant set of coordinates does not admit a unique solution. To answer this different criteria were put forward for non-redundant representations.

Natural internal coordinates

While searching for the best representation for molecular force fields using internal coordinates in small chemical compounds, natural internal coordinates where defined [PFPB79]. They seek to reduce the coupling, both harmonic and anharmonic, between internal coordinates. The algorithms to compute them from a covalent graph remain complex, especially on larger molecules they were not designed for. Local coordinates systems exploiting the symmetry of small rings are used in parallel when necessary to include corresponding constraints. These systems are called ring deformation coordinates or deformational symmetry coordinates and vary according to ring size.

Delocalized internal coordinates

In order to define a complete, non-redundant set of coordinates which can be generated in a simple and straightforward manner for essentially any molecular topology Delocalized internal coordinates were introduced [BKD96].

Wilson B matrix. Considering the vector q of n primitive internal coordinates and the vector X of the n_{cc} cartesian coordinates for a given topology. Using d the differential used in multivariate calculus, the $n \times n_{cc}$ Wilson B matrix is defined as follows:

$$dq = BdX \tag{2.1}$$

The delocalized internal coordinates are the eigenvectors associated to strictly positive eigenvalues obtained from $G = BB^{\mathsf{T}}B$ [BKD96]. Typically the number of eigenvectors selected in this manner equals the number of degrees of freedom $3n_{particles} - 6$ in the absence of cycles.

2.2.4 Conversion from IC and CC

Conversion from IC to CC requires first choosing an arbitrary reference frame. The second step involves embedding particles as the graph is traversed. This can be done efficiently by the `SN-NeRF` algorithm (for *Self-Normalizing Natural Extension Reference Frame*) [PHR+05], which generalizes the `NeRF` method.

The operation which consists of computing the Cartesian coordinates of one atom is called the *embedding* step. This operation requires a *context*, that is 3 atoms already embedded, with respect to which the new atom is positioned.

Given the embedding operator, the algorithm `SN-NeRF` consists of an initialization to define the first context, followed-up by the iterative embedding of the remaining atoms.

Embedding one atom given a context. Given a set of points A_i with $i \in \{1,2,3,4\}$ with known embeddings for the first three and the relative position of the fourth(d_3, θ_2, τ_1 Fig. 2.3), the aim is to embed A_4. The first operation consists of placing A_4 as follows:

$$A_4^* = (d_3 \cos\theta_2, d_3 \cos\tau_1 \sin\theta_2, d_3 \sin\tau_1 \sin\theta_2) \tag{2.2}$$

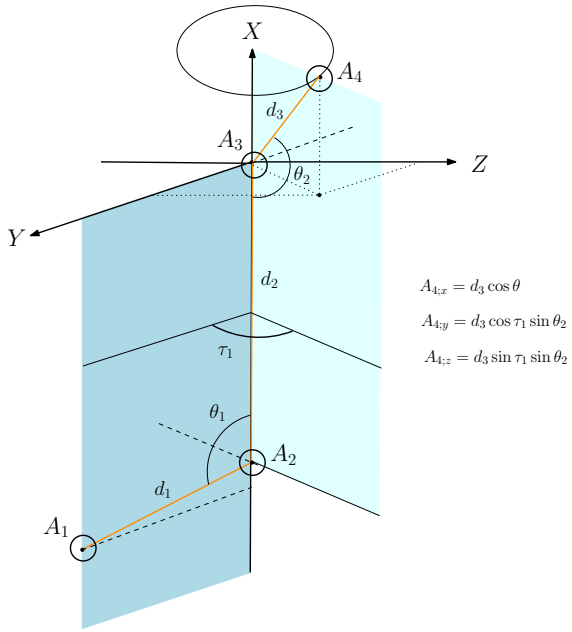
Then A_4 is obtained trough the following transformation:

$$A_4 = RA_4^* + A_3 \tag{2.3}$$

Considering $\hat{A}_{2-3} = \frac{A_2A_3}{|A_2A_3|}$ and $\hat{n} = \frac{A_1A_2 \times \hat{A}_{2-3}}{|A_1A_2 \times \hat{A}_{2-3}|}$ we obtain R:

$$R = [\hat{A}_{2-3}, \hat{n} \times \hat{A}_{2-3}, \hat{n}] \tag{2.4}$$

Figure 2.3 Cartesian embedding of a point given three other. Adapted from [PHR+05].



Initialization. The initialization consists of embedding three particles connected in a path, using two distances and an angle in an arbitrary Cartesian reference frame. In our case (Fig. 2.4):

- $A_1(0, 0, 0)$
- $A_2(0, 0, d_1)$
- $A_3(0, d_2 \sin\theta_1, d_1 - d_2 \cos\theta_1)$

Iterative embedding of the remaining particles. The embedding of the remaining particles in the coordinate system defined by the first three is computed while performing a traversal of the molecular covalent graph.

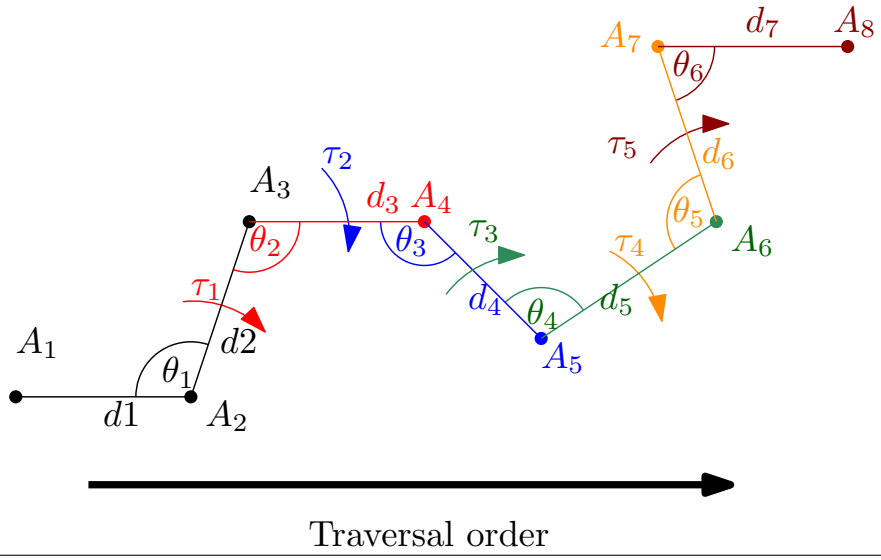
The traversal is performed using two stacks: one corresponding to the points to be embedded next; the second one refers to the contexts (one for each atom to be embedded).

Note that the initialization makes it possible to stack all the neighbors of the first three atoms – their context is defined by these three atoms.

Then, the algorithm proceeds iteratively as follows:

- The particle on the top of the stack is popped, and is embedded using its context. The particles linked to it by an edge in the covalent graph and not already visited are stacked, together with their context.
- Each time an embedded particle is stacked it is tagged to avoid processing twice.

Figure 2.4 Conversion from internal to cartesian coordinates. Adapted from [PHR+05]. In this illustration the graph is traversed from left to right and each colored particle is embedded using the previous three as context and the three internal coordinates with the same color.



The process terminates when the stacks are empty.

Remark 2.1. *For graphs with multiple connected components (c.c.), the process is iterated for each c.c.*

Remark 2.2. *The difference between* `NeRF` *and* `SN-NeRF` *is that when computing* $\hat{n} \times \hat{A}_{2-3}$ *no normalization is used as it equals 1 by construction. Also the norm* $|A_2A_3|$ *used when embedding a point is stored as the points of the covalent graph are embedded such that it is not recomputed.*

Remark 2.3. *The* `p-NeRF` *method [AlQ19] is a parallelized version of* `SN-NeRF`*.*

2.3 Modeling conformations of backbones and side chains

Conformations of protein backbones and side-chains are fundamental concepts to study protein structures. Having recalled the fundamental role of the potential energy, internal coordinates are used to introduce Ramachandran diagrams and rotamer libraries.

2.3.1 Potential energy models

In the realm of molecular mechanics, one decouples the nuclei of atoms and their electron clouds, so that the potential energy of the system obeys the general equation [Fie99, DB10, Zuc10]:

$$V = V_{\text{bond}} + V_{\text{angle}} + (V_{\text{proper}} + V_{\text{improper}}) + V_{\text{vdw}} + V_{\text{electro}}, \tag{2.5}$$

with contributions V_{bond} for covalent bonds (requires two atoms), V_{angle} for valence angles (requires three atoms), V_{proper} for proper dihedral angles (requires four atoms), V_{improper} for improper dihedral angles (requires four atoms), V_{vdw} for van der Walls interactions (typically a Lennard-Jones potential; requires two atoms), and V_{electro} for electrostatic interactions (requires two atoms). Note that the first four terms correspond to bonded terms, while the latter two correspond to non covalent interactions.

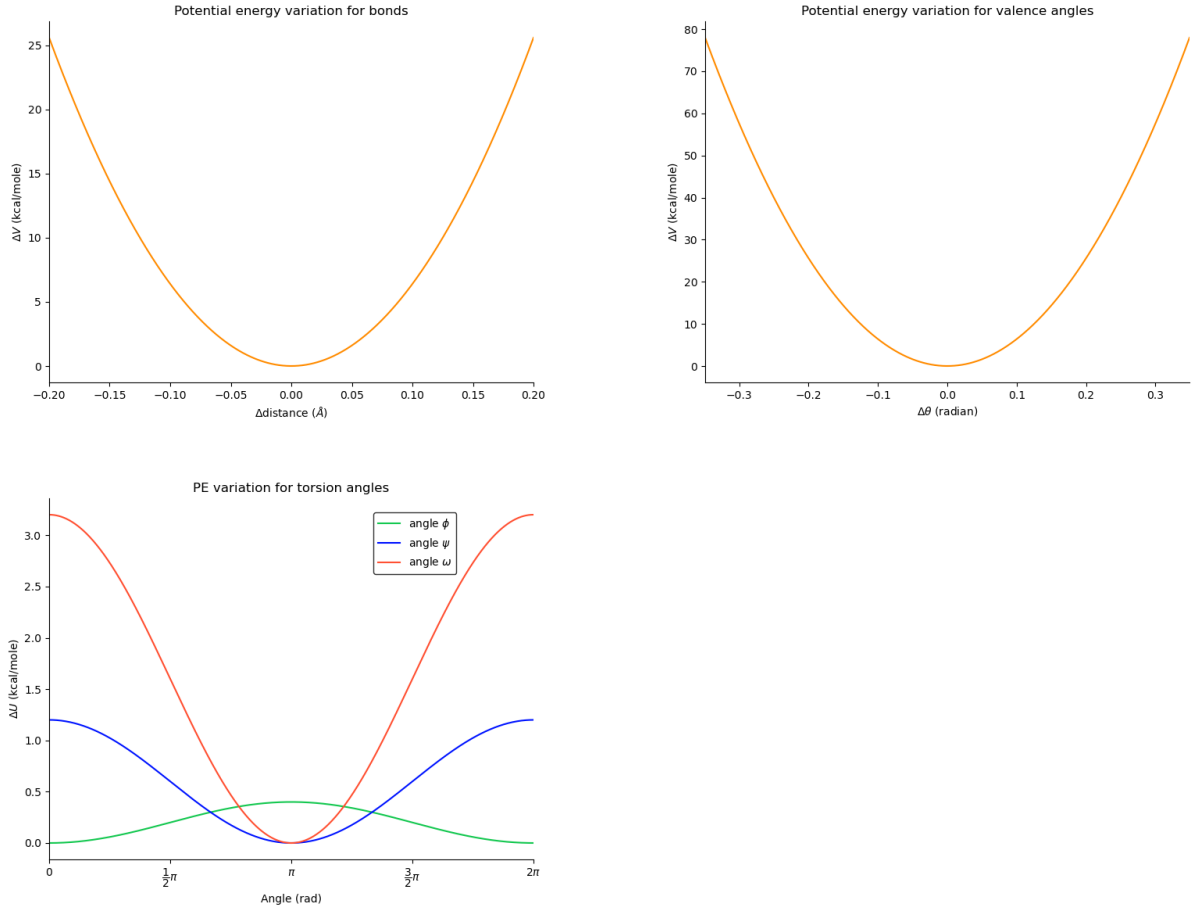
It is instructive to summarize a force field by all its parameters, resulting in the following count of unique parameters $S_u = (B, A, PD, ID, LJ, E)$. Such parameters are usually fitted to reproduce chemical / physical properties of organic molecules [WMP14].

Out of the many force fields available, one may cite:

- `AMBER`, `http://ambermd.org/`: $S_u = (73, 133, 112, 3, 14, 758)$ *i.e.* 1093 unique parameters.
- `CHARMM`, `http://www.charmm.org`: $S_u = (85, 152, 209, 13, 33, 1)$ *i.e.* 493 unique parameters.
- `MARTINI`, `http://cgmartini.nl`: $S_u = (16, 4, 0, 2, 21, 3)$ *i.e.* 46 unique parameters.

The different terms found in a force field have different sensitivity to variations of the corresponding internal coordinates. Taking the example of the `CHARMM` 36 force field [BZS+12], it can be see that the torsion angles are associated to a more tame variation of the potential energy (Fig. 2.5).

Figure 2.5 `CHARMM` force field: variation of potential energies as a function of the internal coordinate type. The various plots were obtained as follows, using the `CHARMM` 36 force field: bond lengths and valence angles: quadratic potential plot using the median spring constant for the whole force field; torsion angles: parameters associated with the angles depicted.



2.3.2 Backbone and Ramachandran diagrams

Ramachandran diagrams. The TLC problem is also closely related to the study of Ramachandran distributions, which characterize the coupling between ϕ and ψ angles along the protein backbone [Ram63, Fer99]. There are four main types of Ramachandran plots: glycine – an amino acid without side chain, proline

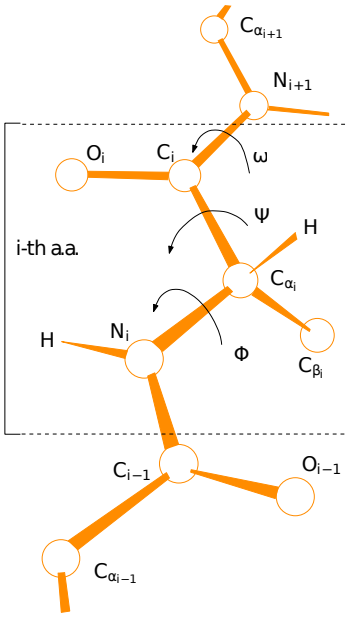
– whose cycle induces specific constraints, pre-proline – residues preceding a proline, and the remaining amino acids, whose C_β carbon induces specific constraints. In this work, we illustrate this latter class with ASP.

Four main regions are occupied in the Ramachandran diagram: β-sheets (βS), polyproline II (βP; left-handed helical structure whose angles are characteristic of β-strands); α-helical (αR); and left handed helix (αL).

Shape of occupied regions. These regions were characterized using a combination of five steric constraints between four atoms defining the Ramachandran tetrahedron ([STM+77], Fig. 2.7). (We note in passing that the 6th edge of this tetrahedron, between O_i and $N_i i+1$, was not used in defining the steric constraints, likely due to the fact that this edge corresponds to a valence angle – a constraint stronger than that associated with the other edges.) In this work, the curves delimiting the occupied regions are termed the Ramachandran *template*. More recently the diagonal shape of level set curves in the occupied regions was explained using dipole-dipole interactions, distinguishing the generic case and proline [HTB03], and glycine and pre-proline [HB05]. The characterization of neighbor dependent Ramachandran distributions has also been studied [TWS+10]. From a statistical standpoint, the Ramachandran distributions of two specific residues can be compared using say f-divergences such as Kullback-Leibler, Hellinger, etc.

In the case of move sets in large proteins and excepting sequence structure prediction, generating conformations for the backbone is often formulated as a loop closure on a subset of the whole structure. This is done as either steric conflict or unrealistic internal coordinates arise when manipulating many atomic coordinates simultaneously. In general either a part of an experimental protein structure is missing or a "hole" is created to sample conformation space. This turns backbone conformation sampling into a loop closure problem. The specificity of loop closure is that the freedom of movement will be concentrated on *phi* and *psi* backbone dihedral angles. This makes the backbone loop closure a geometric problem first and foremost with proposal conformations being scored by a sum of potential energy terms down the line.

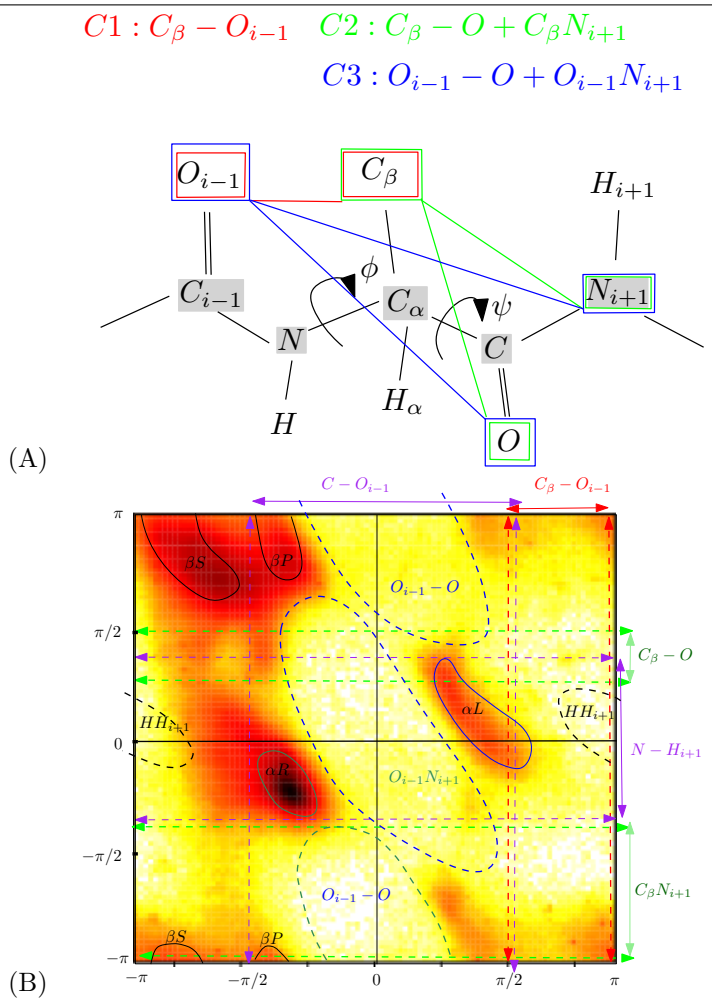
Figure 2.6 Conformations of a protein backbone



ϕ, ψ angular representation

As already mentioned, the backbone is often thought as having two main degrees of freedom. When manipulating the embedded molecular covalent graph of the backbone, the bond lengths and bond angles do not

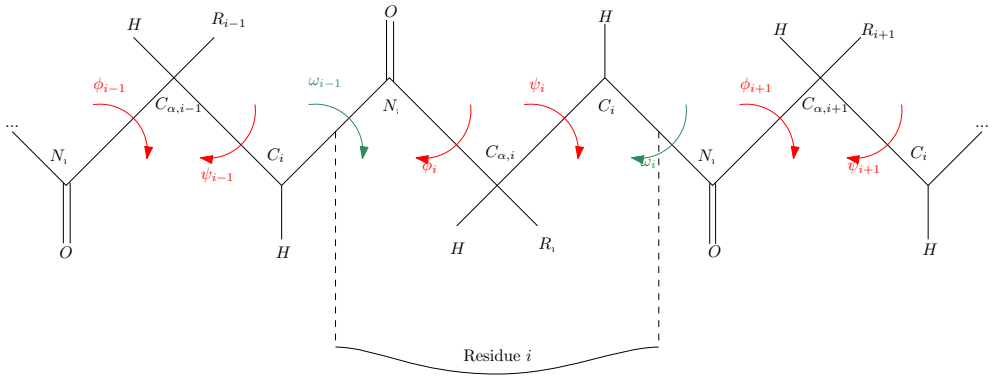
Figure 2.7 Ramachandran diagrams: distance constraints and occupied regions. (A) The Ramachandran *tetrahedron* and its five distance constraints – adapted from [HTB03, HB05]. Note that the four atoms define a tetrahedron: five of its edges are constrained; the last one (ON_{i+1}) corresponds to a valence angle, and is not constrained. **(B)** Main regions occupied in the Ramachandran space, with associated steric constraints, materialized by dashed lines/curves, involving vertices of the Ramachandran tetrahedron. The background distribution was obtained using all amino acids in the structure files used in this study (loops and SSE). The partition of the Ramachandran space illustrates the location of the classical SSE: β-sheets (βS), polyproline II (βP; a left-handed helical structure whose angles are characteristic of β-strands), α-helical (αR), and left handed helix αL.



vary as a first approximation. Among the dihedral angles the torsion ω around the peptide bond can also be considered to be fixed at $+\pi$ or $-\pi$. Most of the variations in the backbone are contained in dihedral angles ϕ and $\psi \in [-\pi, +\pi)$ around the c_α carbon.

We also note that ϕ, ψ dependencies on both the type of a residue and those of its neighbor have been studied [TWS+10], Technically, these analysis use a non parametric Bayesian model based on hierarchical Dirichlet processes (HDP) to produce mixture models. Such analysis are of special interest when defining rotamers (Sec. 2.5), and also for backbone sampling algorithms (Sec. 2.7).

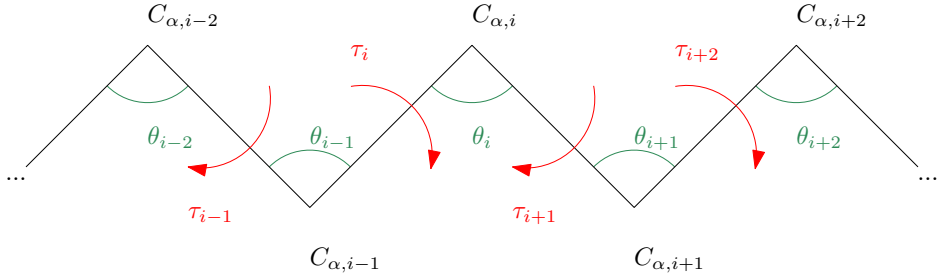
Figure 2.8 ϕ ψ **representation.** In red the continuous dihedral degrees of freedom ϕ and $\psi \in [-\pi, +\pi)$. in green the discrete dihedral $\omega \in \{\pi, +\pi\}$



C_α representation.

A simplified representation, which is of special interest for structures of medium resolution, consists of retaining C_α atoms only (Fig. 2.9). assuming the distances between C_α atoms are constant, such a model enjoys two types of internal coordinates: the pseudo-bond angle $\theta \in [0, \pi)$ defined by the two pseudo-bonds connected to a given C_α and the pseudo-torsion $\tau \in [-\pi, \pi)$ around each pseudo-bond. Equivalently, the model is encoded by the sequence of unit vectors connecting the consecutive C_α atoms. It is therefore represented $n-1$ points on the unit sphere S^2 – assuming n C_α atoms. A parametric model suitable in this case is the 5-parameter Fisher-Bingham function (FB5), which generalizes the Gaussian distribution on the sphere – see Section 2.4 and [Bin74].

Figure 2.9 Backbone: simplified C_α representation. The vertices in black do not represent covalent bonds but fictive pseudo-bonds between $C\alpha$ carbons. The degrees of freedom are (i) the pseudo-band angles $\theta \in [0, +\pi)$, and (ii) the torsion angles around such bonds $\tau \in [0, 2\pi)$.



2.3.3 Rotameric and non rotameric dihedral angles χ, rotamer libraries

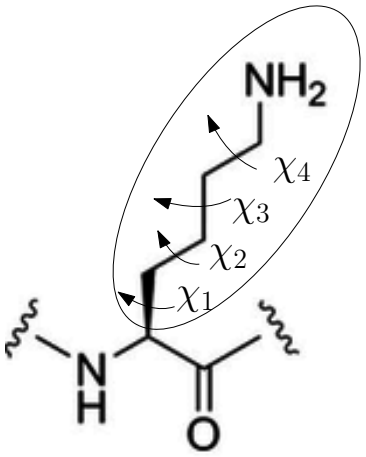
Rotameric and non rotameric degrees of freedom. Except for the case disulfide bonds and proline residues, side chains are connected to the backbone only through the alpha carbon present in the same residue. Omitting the particular cases of GLY and ALA, this makes it practical to model their conformational diversity for a given backbone. As early recognized [JWLM78, MT93], such analysis hinge on dihedral angles of side chains, denoted χ_i. (Nb: in the sequel, an angle χ is called a degree of freedom or dof.) (Fig. 2.10). The analysis of experimental data yielded so-called *rotamers* or rotational isomers. In the simplest case, such an angle has a distribution which sharp peaks/local maxima (Fig. 2.11(A)). Such peaks correspond to high energy barriers. In that case, one defines one rotamer for each peak, which is characterized by three numbers namely the mean μ_i, stdev σ_i, and propensity – the sum of all propensities equals one. In the

simplest case, one represents a rotamer by its model or mean value – and the associated weight. As a more elaborate representation, one can use a a mixture of 1D von Mises distributions.

However, selected χ angles do not match the aforementioned mixture model (Fig. 2.11(B)). Such cases typically correspond to degrees of freedom at the end of the side chain with little constraints so that they may take any value. Physically, polar side chains make electrostatic interactions, while aromatic side chains can face steric clashes.

Summarizing, there are two main difficulties to deal with: modeling non rotameric dof; taking into account the coupling between rotamers and backbone conformations. We shall get back to these issues in Sec. 2.5.

Figure 2.10 Conformations of protein side chains: dihedral degrees of freedom in a Lysine residue. The four dihedral angles χ_i are rotameric degrees of freedom containing most of the diversity for Lysine side chain conformations.



LYS

2.3.4 Notes

As we will see in section 2.5 and 2.6, rotamers play a pivotal role in several applications, including (i) quality check of protein structures and outlier detection, (ii) generative models (for energy landscape exploration and docking), and (iii) computational protein design.

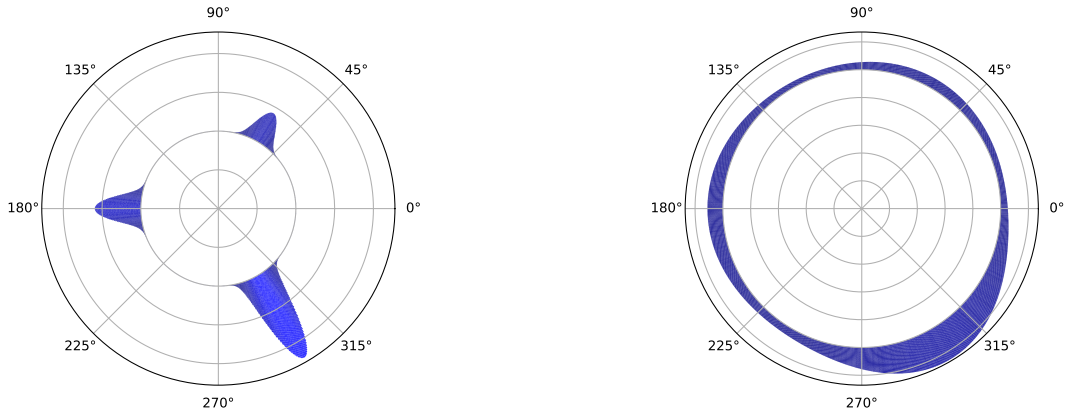
2.4 Statistical and geometric analysis: methods

This section presents (geometric, statistical) methods in the realm of directional statistics [MJ09]. These methods will be of special interest in the design of rotamer libraries (Sec. 2.5), and for side chain sampling (Sec. 2.6).

2.4.1 Circular means and centers of masses

In Euclidean geometry, the center of mass of a set of data points is the point minimizing the sum of squared distances to these data points. This centering operation is central to statistical analysis, e.g. PCA. Performing an equivalent centering appeared as an early need while analyzing angular values [MT93].

Figure 2.11 Dihedral angles of side chains: rotameric versus non-rotameric angles χ. (A) A rotameric dof exhibits a multimodal, sharply peaked distribution on the unit circle. **(B)** A rotameric dof has a distribution which cannot be modeled as a simple mixture.



(A) **(B)**

Circular mean. A classical way to define the circular mean of a set of angles is the *resultant* or *circular mean*, defined as follows [MJ09]:

$$\overline{\theta} = \text{atan2}(\sum_i \sin\theta_i/n, \sum_i \cos\theta_i/n). \tag{2.6}$$

The circular mean does not minimize the sum of squared distances along the unit circle, but minimizes instead [JS01, Section 1.3]:

$$\overline{\theta} = \arg\min \sum_{i=1,\dots,n} d(\theta_i, \theta), \text{ with } d(\alpha, \beta) = 1 - \cos(\alpha - \beta). \tag{2.7}$$

Fréchet mean. Generalizing the center of mass in a general metric spaces (the so-called Fréchet mean, [Fré48]), and for distributions (the so-called Karcher man, [GK73]) has a long history. Surprisingly, the calculation of the Fréchet mean for circular data was only finalized recently [CDO21] (Fig. 2.12). The solution involves two ingredients. First, the decomposition of the sum of squares into a simple polynomial expression, so that one needs to Second, delicate numerical analysis calculations, as transcendental numbers are dealt with. Interestingly, using the Fréchet mean instead of the circular can yield a significant variance reduction [CDO21].

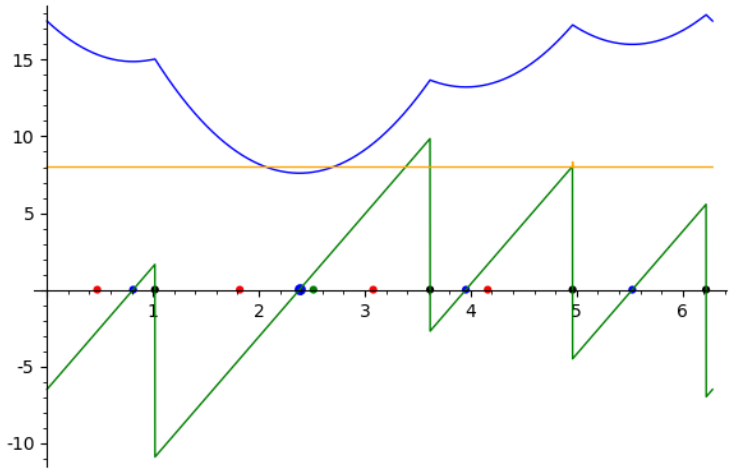
2.4.2 Parametric models

In the following, we introduce several parametric models / functions, suitable to represent (angular, vector) data for proteins. In spirit, these functions are meant to generalize Gaussian distribution for compact / non Euclidean spaces. The reader is referred to [HMFB12, Chapters 6 and 7] and references therein for more details.

On the unit circle: univariate von Mises distribution. The classical 1D von Mises distributions is the analogous of the Gaussian distribution on the unit circle S^1. As such, it is parameterized by a mean value $\overline{\theta}$ and a concentration parameter κ:

$$f(\theta) = \frac{1}{2\pi I_0(\kappa)} \exp(\kappa \cos(\theta - \overline{\theta})). \tag{2.8}$$

Figure 2.12 Fréchet mean of four points on S^1 (Functions) blue: function F_2; green: derivative $F_2^{'}$; orange: second derivative $F_2^{''}$ **(Points)** red bullets: data points; black bullets: antipodal points; blue bullets: local minima of the function; large blue bullet: Fréchet mean θ^*; green bullet: circular mean Eq. 6.14.



In this equation, $I_0(\cdot)$ stand for the modified Bessel function of the first kind and order 0 [ASR88].

On the torus $\mathbb{T}^2$: bivariate von Mises distribution. This is used as the equivalent of a Gaussian model in a space with two angles ranging the full unit circle S^1:

$$f(\phi,\psi) \propto \exp(k_1\cos(\phi-\overline{\phi}) + k_2\cos(\psi-\overline{\psi}) + \tag{2.9}$$

$$(\cos(\phi-\overline{\phi}), \sin(\phi-\overline{\phi}))A(cos(\psi-\overline{\psi}), sin(\psi-\overline{\psi}))^T) \tag{2.10}$$

The normalization constant reads as an infinite series, see [Mar10b]. This model has eight parameters: $\overline{\phi}, \overline{\psi}$ which be described as the mean values, the concentrations k_1, k_2, and the 2×2 matrix A encoding a coupling between the two angles. the parameters are known to be redundant for high concentrations, which leads to difficulties in fully interpreting the meaning of the parameters. Restricting matrix A by fixing the off diagonal elements to zero (the remaining values being denoted α and β) yields four simpler models [HMFB12, Chapter 6]: the sine model with $\alpha = 0, \beta = \lambda$; the cosine model with positive interaction $\alpha = \beta = -k_3$; the cosine model with negative interaction $\alpha = -k_3' = -\beta$; and the hybrid model.

On the sphere S^2: Fisher-Bingham. A coarse protein representation based solely on C_α atoms yields a model parameterized by two angles, namely $\theta \in [0,\pi)$ and $\tau \in [-\pi,\pi)$ (Fig. 2.9). This parameterization is akin to spherical coordinates on the unit sphere S^2. Consider the usual parameters of a d-dimensional multivariate Gaussian distribution μ and Σ. To model a set of points on S^d, one can use the Fisher-Bingham distribution [Bin74]:

$$f(X) \propto \exp(-\frac{(X-\mu)^\mathsf{T}\Sigma^{-1}(X-\mu)}{2}), X^\mathsf{T}X = 1. \tag{2.11}$$

For the particular case $d = 2$, the model has five parameters [HMFB12, Chapter 7]: the means values $(\mu_\theta, \mu_\tau)^T$, the three terms $\sigma_{11}, \sigma_{22}, \sigma_{12} = \sigma_{21}$ of the covariance matrix.

Further details can be found in [KGJH14] for the computation of the normalization constant, and in [KS18] for the maximum likelihood estimation of the parameters.

2.4.3 Notes

A variety of statistical data analysis techniques have been used to deal with the models just mentioned.

Parametric models. Classical approaches from Bayesian statistics are typically used [Mur12]. In a nutshell, consider a model whose parameters are stored in a vector Θ. These parameters typically define a parameterized family of probability distributions, say f_Θ. Also assume that a dataset X is available. The goal is to estimate the parameters Θ to explain the evidence X.

$$\mathbb{P}[\Theta \mid X] = \frac{\mathbb{P}[X \mid \Theta]\,\mathbb{P}[\Theta]}{\mathbb{P}[X]}, \tag{2.12}$$

which reads as follows

$$\text{posterior} = \frac{\text{likelihood} \times \text{prior}}{\text{evidence}}. \tag{2.13}$$

To obtain pointwise estimates of the parameters, on resorts to Maximum likelihood yields ($\hat{\Theta}_{\text{MAP}}$) or Maximum a Posteriori (MAP) yields ($\hat{\Theta}_{\text{ML}}$) estimates–both methods neglect the denominator of Eq. (2.12). Dealing with full posterior models is possible using conjugate priors, i.e. $\mathbb{P}[\Theta]$ and $\mathbb{P}[\Theta \mid X]$ have the same distribution. Otherwise, criteria such as the Bayesian Information Criterion (BIC), which assess the model based on its fit to the data and its complexity, can be used. Yet another criterion is the Akaike Information Criterion.

Non parametric models. Dealing with non parametric models, say for density estimation, uses different techniques [BD15, DGL96]. Of particular interest are kernel density estimates and kernel regression techniques. The mathematical expression used to model a response variable, say Y, is by means of a the conditional expectation $\mathbb{E}[Y \mid X] = m(X)$. The estimate $m(x)$ is obtained by averaging the response values y_i – denoting h the bandwidth and $K_h(\cdot)$ the kernel used

$$\hat{m}(x) = \frac{\sum_i K_h(x - x_i) y_i}{\sum_i K_h(x - x_i)}. \tag{2.14}$$

Another non parametric techniques often used to analyse structural data is clustering. A reference method is `k-means++`, a variant of `k-means` with smart seeding coming with a guarantee of the expectation of the `k-means` functions [AV07]. Another important clustering technique is density based clustering, where one assigns a cluster for each local maximum of the estimated density. This techniques is especially effective once coupled to topological persistence so as to identify the significant clusters [CGOS13]. Finally, we may also cite useful techniques to compare clustering, see e.g. [CMTW19] and references therein.

2.5 Rotamer libraries

2.5.1 Overview

As noticed in Section 2.3.3, rotamers play a key role to discretize and simplify the study of side chain conformations. In this section, we review recent rotamer libraries, which is especially informative since these libraries differ in three main respects:

- Goal pursued: quality check - outlier detection, generative models for landscape exploration or docking, computational protein design.
- Biophysical specification: generic rotamers, rotamers backbone dependent, rotamers sequence dependent.
- Methodology: mathematical model used for rotamers, and the associated model selection procedure.

In describing the libraries, we focus in each case on (i) Rationale and goals, (ii) Model, and (iii) Assessment.

2.5.2 Smoothed Backbone-Dependent Rotamer Library – 2011

Rationale - goals. We have mentioned in Section 2.3.3 the difficulties to design rotamer libraries, due in particular to the duality rotameric - non-rotameric dihedral angles χ. In [SDJ11] – see also `https://dunbrack.fccc.edu/retro/bbdep2010/Tutorial.php`), the dichotomy used is as follows:

- 10 a.a. with rotameric dof only: ARG, ILE, LEU, LYS, MET, SER, THR, VAL CYS (combines statistics for CYH:cysteine reduced free sulfhydryl + CYD: cysteine oxidized disulfide-bonded) PRO (combines statistics for TPR: trans-proline + CPR: cis-proline)
- 8 a.a. with non rotameric dof: terminal dof for ASN, ASP, GLU, and GLN; aromatic residues, PHE, TYR, HIS, and TRP whose c2 angles are more broadly distributed than rotameric dof.

(Nb: this yields a total of 22 a.a.: 18 + 2 for CYS + 2 for PRO.) A possibility to handle non rotameric dof is to use bins, with one rotamer per bin. However, this ad hoc options involves the arbitrary choice of the number of bins, and does not provide a compact representation.

Model for rotameric dof. The goal is to compute $\mathbb{P}[r \mid \phi, \psi]$, with the constraint that

$$\sum_r \mathbb{P}[r \mid \phi, \psi] = 1. \tag{2.15}$$

The solution from [SDJ11] uses two main ingredients:

- $\rho(\phi, \psi \mid r)$: the Ramachandran distribution for rotamer r;
- $\mathbb{P}[r]$: backbone independent observed frequency of rotamer r.

Using these and Bayes's formula, one obtains the following conditional probability for a given rotamer, conditioned on the backbone geometry:

$$\mathbb{P}[r \mid \phi, \psi] = \frac{\rho(\phi, \psi \mid r)\mathbb{P}[r]}{\sum_{r'} \rho(\phi, \psi \mid r')\mathbb{P}[r']} \tag{2.16}$$

•*Ingredient 1: adaptive kernel density estimate for* $\rho(\phi, \psi \mid r)$. The Ramachandran distribution for rotamer r is defined as the sum over the N_r data points of rotamer of type r, of the product of two 1D von Misses kernels (Eq. 2.8):

$$K_h(\phi - \phi_i, \psi - \psi_i) = \sum_{i=1,\ldots,N_r} \frac{1}{4\pi^2 I_0(\kappa)^2} exp(\kappa(\cos(\phi - \phi_i) + \cos(\psi - \psi_i))) \tag{2.17}$$

The previous uses a bandwidth parameter h. To reduce the role of outliers and cope with the local density of data, this parameter is adapted locally, yielding **Adaptive kernel density estimation** (AKDE). This parameter is tuned for each each residue type, chosen using cross validation.

The result is then used to obtain the backbone-dependent rotamer probabilities – Eq. (2.16).

•*Ingredient 2: adaptive kernel regression for rotameric angles and variances.* The second main ingredient consists, for each of the 22 residue types and each χ angle of: the population mean (μ_i) and the stdev (σ_i). That is, angle χ_i is modeled as follows:

$$\chi_i = m(\phi_i, \psi_i \mid r) + \nu^{1/2}(\phi_i, \psi_i)\varepsilon_i, \tag{2.18}$$

with m the regression function, ν the variance, and ε_i an error term normally distributed (mean zero, unit variance). The variance of the observation is expected to depend on the particular values of ϕ, ψ – the model is called heteroscedastic (*different dispersion*).

$$m(x, y \mid r) = \mu(\chi \mid \phi = x, \psi = y, r) = \mathbb{E}\left[\chi \mid \phi = x, \psi = y, r\right] \tag{2.19}$$

$$\nu(x, y \mid r) = \sigma^2(\chi \mid \phi = x, \psi = y, r) = \mathbb{E}\left[\chi \mid \phi = x, \psi = y, r\right] \tag{2.20}$$

$$\tag{2.21}$$

These two terms are estimated using kernel regression. In this case, a kernel which is adaptive based on the query point rather than the data point is used.

Model for non rotameric dof. We now consider the case on non rotameric dof, which typically exhibit broad distributions, also depending on the geometry of the backbone. For such cases, denoting r_{-n} the set of χ_i angles preceding that of interest, the goal is to estimate the probability density

$$\rho(\chi_n \mid r_{-n}, \phi, \psi). \tag{2.22}$$

Using a kernel based on the 2D von Mises distribution (Eq. 2.9), define the following weight:

$$w_i(\phi, \psi) = \frac{K_{h(\phi,\psi)}(\phi - \phi_i, \psi - \psi_i)}{\sum_{i=1,\dots,N_r} K_{h(\phi,\psi)}(\phi - \phi_i, \psi - \psi_i)} \tag{2.23}$$

The density is estimated as follows based on the 1D von Mises kernel $K_h(\cdot)$ (Eq. 2.8):

$$\rho(\chi_n \mid \phi, \psi), r_{-n}) = \sum_{i=1,\dots,N_r} w_i(\phi, \psi) K_h(\chi_i)(\chi_n - \chi_i). \tag{2.24}$$

Nb: a query point dependent kernel is used.

Results and assessment. Two assessments are presented. The first one is the side chain prediction problem, where a χ angle is correctly predicted if the angular difference wrt the experimental value is less than 40 degrees. Using the program `SCWRL4` [KSDJ09] with the novel rotamer library yields a better prediction rate. The second one is the ability of `Rosetta`'s energy minimization scheme to exploit the rotamer library. These protocols (ClassicRelax, FastRelax) are Monte Carlo explorations of the energy landscape, using a move set which uses a small deformation of the backbone and a consistent choice of rotamers. For this second task too, an improvement is observed.

2.5.3 The Dynamic Rotamer library – 2016

Rationale - goals. Using data from the PDB is the classical route to obtain rotamer libraries. Yet, using static structures from the PDB faces several limitations.

First, such libraries tend to overlook flexibility and motion since flexible regions are seldom reported in crystal structures, and atoms with large B factors are filtered out. Second, despite remediation projects, errors may remain in PDB structures (e..g chirality and cis peptide bond errors), setting aside ambiguities in electron density maps. Second, using PDB structures induces a bias on Ramachandran regions populated [TRVD16, Fig. 4]. Third, the paucity of sampling in selected regions requires a more advanced statistical processing to obtain meaningful statistics and models of commensurable complexity [SDJ11].

Model. To circumvent these limitations, the approach promoted in [TRVD16] uses MD simulation data, obtained from 807 proteins covering 97% of known autonomous protein folds. The definitions of rotamers use classical angular ranges [TRVD16, Fig. 7A]. For example, for tetrameric carbons, one defines three rotamers: $g^+ : 0 - 120, t : 120 - 240, g^- : 240 - 360$. The amount of data collected made it possible to obtain individual probabilities from raw counts. For the backbone dependent library developed, rotamer probabilities were obtained using bins of 10×10 degrees in (ϕ, ψ) space. The approach pursued has two main merits. First, there is less bias on the structures and a better coverage of the Ramachandran space [TRVD16, Fig. 4], which is especially useful to design backbone dependent libraries. Second, the rotameric states covered are much more comprehensive: using 10 degrees bin for ϕ, ψ, 97% of the Ramachandran domain is covered, as opposed to 53% using the dataset from [SDJ11]. easier statistical processing due to the size of datasets.

Results and assessment. As expected, the rotamers obtained differ from those associated with static analysis, in particular for more mobile chains on the surface of proteins.

2.5.4 Checking the integrity of conformations: `Molprobity`, 2016

Rationale - goals. [HLRR16]. design of a rotamer library primarily focusing on the validation of side chain conformations in protein models. based on a careful selection of 8,000 high resolution structures.

Model. The main ingredients are as follows:
•*KDE estimates in multi-dim space.* uses adaptive KDE to estimate the density in the multi-dimensional χ angles space. uses a cosine kernel with varying width depending on the location queried (width is larger is less populated regions). KDE estimates are stored at grid points – the spacing varies depending on the number of χ angles.

The score of a grid point provides the assessment using the classical thresholds for Ramachandran distributions: $< 0.03\%$: outlier; $0.3\%, 2\%$: allowed; $> 2\%$: favored. These values correspond to iso-surfaces in the χ angle space.
•*Definition or rotamers.* Consider now the data within a level set surface associated with a threshold. Each connected components defines one rotamer. For each rotamer, the following pieces of information are reported: mean for the valence angles, weighted center of mass for the χ angles – requires circular statistics.
•*Rotamer assignment.* The score of a side chain is a mix (average) from its χ angles. The value of a given χ angle is obtained by interpolating over the nearest grid vertices.

Results and assessment. Assessment:

- plus: multi dim space, adaptive KDE, kernel with compact support
- minus: formula com false; no dependency to backbone and sequence

2.5.5 Sequence dependent rotamer libraries – 2021

Rationale - goals. Rotamers clearly depend on the a.a. flanking a given residue. However, statistical studies on side chain conformations for 20^3 tripeptides face paucity of structural data available. To fudge around this difficulty, molecular simulation has been used to define Sequence Dependent Rotamer Libraries (SDRL) [DW].

Model. Each a.a. is studied in the context of all possible flanking a.a. yielding peptides of the form ACE-XXX-YYY-ZZZ-NME. (Nb: ACE and NME are the usual caps corresponding to an acetyl group and a methyl group.) The rationale is to allow for spatial effects without interferences with a whole protein fold.

Using the 18 naturally occurring a.a. (excluding alanine and glycine), the tripeptides are simulated using Basin-Hopping [LS87], using the `AMBER` force field and an implicit solvent model. For a given tripeptide, the low free energy minima are retained – using a harmonic model for the free energy. These minima represent various combinations of the ϕ, ψ angles, so that the rotamer library generated is backbone independent.

To obtain the rotamers of the central amino acid, a hierarchical clustering procedure is applied to local minima, using the Euclidean distance in dihedral angle space – that is the periodic distance on the flat torus. A cutoff of 40° is then used to obtain the clusters from the dendogram. One rotamer per cluster is then defined. This rotamer is obtained by computing the weighted center of mass of each dihedral angle – the weight of a conformation being given by Boltzmann's factor of the free energy. Note that the computation of the center of mass was recently solved in [CDO21]. The importance of the local sequence is evidenced by the probability of a given rotamer.

Results and assessment. The library is also used to assess coverage of rotamers found in experimental structures, with performances comparable to non sequence depending libraries. (Nb: a side chain conformation is covered/represented by a rotamer provided that each of its χ angles is within 40 degrees.)

2.5.6 Notes

Various other methods have been developed to predict rotamers and side-chain conformations. A notable one is that based on neural networks (using back propagation and a sigmoid activation function) [HL95].

2.6 Side chain conformational sampling

2.6.1 Overview

This section focuses on sampling methods for side chains, using rotamers. Two main classes of methods are presented: combinatorial methods mainly used for computational protein design, and generative models based on dynamic Bayesian networks.

2.6.2 Group rotations

Side chain conformational sampling can be done by choosing values for the χ angles at random, , and applying Rodrigues' formula to rotate the portion of the side-chain found downstream the bond. This strategy, used very early to pack side chains [LS91], has recently been termed *group rotations* [MWS$^+$14]. The strategy can be refined using the information associated with rotamers. represented as a mixture of 1D von Mises distributions. For non rotameric dof specified using a smooth distribution, this strategy only requires sampling this distribution.

2.6.3 Computational Protein Design with continuous rotamers – 2012 and 2017

Computational Protein Design and classical approaches. Computational Protein Design (CPD) consists of finding a sequence of a.a. that fold into a specific structure. The classical approach uses two main ingredients: an energy model which in general is pairwise decomposable; and a library of rotamers. In other words, the algorithm searches over a discrete set of conformations obtained by combining specific conformations (rotamers) of the a.a used/selected. Recently, the classical notion of rigid rotamer has been evolved into a notion of continuous rotamer, amenable to energy minimization.

The dead-end elimination (DEE) algorithm [DDMHL92] reduces the search space of the problem iteratively by removing rotamers that can be provably shown to be not part of the global lowest energy conformation (GMEC). DEE is typically coupled to the A* branch-and-bound algorithm [HNR68] to maintain a lower bound on the partial trees to explore and extent the most promising one [LL98].

The simplest kind of rotamer lib consists of using rigid rotamers – in which case the associated GMEC is termed the rigidGMEC. Unfortunately, χ angles in protein structures may significantly differ from the modal values [JWLM78], so that steric clashes arising using rigid models cannot be fixed. On the other hand, given a continuous energy model, it is possible to minimize the energy of conflicting rotamers. The use of continuous rotamers in protein design is explored in [GRD12]. In this context, the GMEC obtained is called minGMEC.

Using continuous rotamers. In the *continuous rotamer model*, the χ space of a side chain is decomposed into voxels called *Residue Conformation* (RC) – that is a voxel corresponds to a continuum of side chain conformations. A conformation of the whole protein is therefore a vector of RCs [GRD12, HKD13, Fig. 1]. More precisely, to describe the conformational space of a single dihedral angle χ, let $Decomp(S^1)$ be a decomposition of the unit circle S^1 into intervals. If the i-th a.a. has n_i dihedral angles, its conformation space is the product $\prod_{i=j,\dots,n_i} Decomp(S^1)$. An element in this product is a voxel or RC.

Minimizing a side chain in a voxel creates a domino effect on neighboring side chains. The MinDEE algorithm handles this coupling by combining MinDEE and A* [GRD12]: rotamers which are not part of MinGMEC are pruned by MinDEE; A* is used to sort rotamers by increasing lower bound – yielding the processing order. Note that A* requires minimizing each rotamer vector – the minimization of a rotamer

takes place in the cartesian product of the single-residue voxels. (It is stated that RC are small enough so that minimization yields an optimum within the voxel [GRD12, HKD13].)

LUTE. The bottleneck in [GRD12] is the minimization process. While this step can be accelerated using a compact/polynomial representation of the (potential) energy surface [HGD15], the solution lags behind CPD based on rigid rotamers. The goal of LUTE [HJD17] is to handle the domino effect by processing tuples of side chains directly, using a representation suitable for DEE/A*. This representation uses two main ingredients: first, a global quadratic model for the energy; second, tuples (pairs, triples) of side chains in close vicinity to make a significant contribution to the total energy.

Consider a chain with n residues, denoting n_i the number of dihedral angles of the i-th residue. Using bins of 18 degrees, the configuration space of this side chain is decomposed into $(360/18)^{n_i} = 20^{n_i}$ *side-chain voxels*. The rationale of this bin width is to obtain one local minimum per voxel – even though this cannot be guaranteed. Taking the Cartesian product of side-chain voxels yields a decomposition of the configuration space of all side chains into *global voxels*.

The LUTE methods is based on two steps: training and learning. Training consists of minimizing the energy model used within a global voxel. When this optimization problem is well poised, one obtains one minimum, encoding one conformation for each side chain, and the associated energy. Applying least-squares (LS) to the training data–the list of (minima, energy) associated to the global voxels minimized, learning consists of fitting a (unique) quadratic model for all global voxels.

This model can then be used to compute the energy of a tuple (pair, triple) of locally interacting side chains. To do so, one restricts the global LS model to those dihedral angles present in the tuple. Note that in doing so, one obtains an energy for this tuple without effectively minimizing the energy model for this tuple. Also, since the energy is computed for the tuple in the absence of other side chains in the direct environment, one typically obtains a lower bound – steric constraints with neighbors being omitted.

2.6.4 TorusDBN– 2008

The TorusDBN model is not concerned with side chains, but instead the local structure of the backbone. Still, we present it briefly to make the presentation of BASILISK self contained.

Rationale - goals. To model local protein structure in a spirit analogous to HMM for sequences, a method using a Dynamic Bayesian Network (DBN) is proposed in [HBP$^+$10]. Recall that a Bayesian network (BN) is meant to represent a joint distribution of a set of random variables (RV) using a directed acyclic graph (DAG) [Gha97]. The adjective dynamics refers to the indexing provided by the sequence – analogous in spirit to a time series.

Model. The particular DBN used is a path hidden nodes (Fig. 2.13). A given hidden node represents a residue at a specific chain position, and can adopt 55 different states (see below). Each state emits a four tuple of values denoted (d, a, s, c), the distributions of the individual RV being *state* dependent. The four values are:

- (i) $d = (\phi, \psi)$ dihedral angles : modeled by a bivariate von Mises distribution,
- a: amino acid type,
- s: secondary structure,
- c: cis ($\omega = 0$) /trans ($\omega = \pi$) conformation for the peptide bond.

Note that due to the linear structure of the DBN, each four tuple (d, a, s, c) can be used as input or output.

Sampling from this model is a two step process:

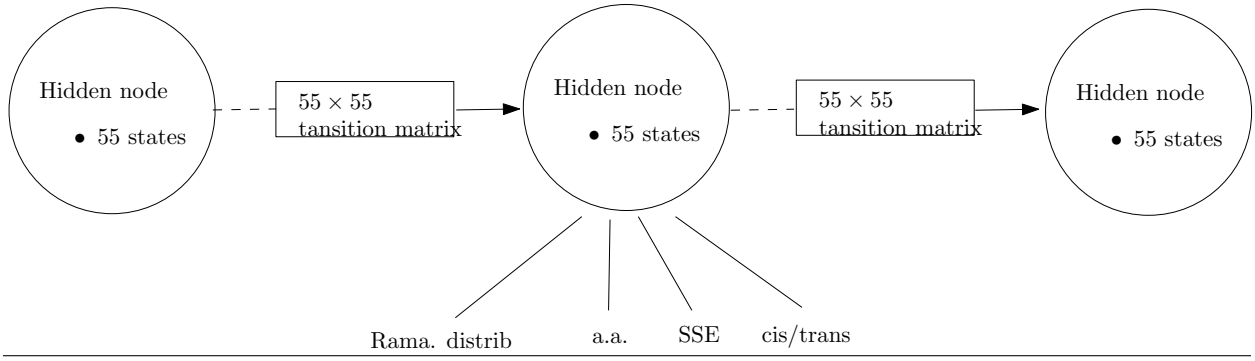
- Step 1: sampling a hidden node sequence (using observed nodes if any) – assigning a state to each of them. This is done using the so-called forward-backtrack algorithm. (Nb: Some input information may also be used, e.g. amino acids at specific positions.)

- Step 2: sampling the remaining pieces of information, using the conditional probability distributions of the nodes.

The number of states for a hidden note (55) is a hyperparameter optimized using the BIC criterion [HBP$^+$10, SI], peaking at a value ~ 55.) The probability distributions associated with hidden nodes are learned using Monte Carlo Markov Chain (Gibbs sampling) [PH10]. DB used consists of 1647 protein structures.

Results and assessment. Several assessment of the model are provided, including the analysis of (i) angular preferences, (ii) length 4 amino acid generated (as such motifs have been studied in detail in the structural bioinformatics literature) (iii) structures compared against the native states.

Figure 2.13 The Dynamic Bayesian Network generative model for protein structure: a path of hidden nodes. Adapted from [HBP$^+$10]. A length n sequence is represented by a sequence of n hidden nodes, each emitting backbone angles, an amino acid type, a SSE type, and a cis/trans conformation for the peptide bond.



2.6.5 BASILISK– 2010

Rationale - goals. The goal of BASILISK is to offer a a compact/continuous/generative model for backbone dependent conformations of side chains. The adjective *compact* refers to the fact that all a.a. are processed at once, and that for a given a.a., the dihedral angles of the backbone and those from the side chains are inherently coupled. The adjective *continuous* indicates that the conformational space modeled is continuous – with possibly sharply peaked distributions. The adjective *generative* refers to the fact that the model is a generator of conformations.

Model. The name BASILISK stands for *Bayesian network model of side chain conformations estimated by maximum likelihood*. BASILISK is also based on a DBN (Dynamic Bayesian Network). The DBN has two slices for the (ϕ, ψ) angles of the backbone, and four more slices to accommodate the maximum number of χ angles (Fig. 2.14). Each slice has an index to specify the angle, a hidden node, and a von Mises distribution for that specific angle – its parameter depend on the value in the hidden node. The hidden nodes introduce the required coupling between all angles.

Consider the following three vectors: $\overline{\chi}$: sequence of χ angles, $\overline{A}$: vector of angle info (a.a. type), $\overline{H}$: vector of hidden node values.

Also recall the following decomposition of the joint probability, say for four random variables:

$$\mathbb{P}[WXYZ] = \mathbb{P}[W] \frac{\mathbb{P}[WX]}{\mathbb{P}[W]} \frac{\mathbb{P}[WXY]}{\mathbb{P}[WX]} \frac{\mathbb{P}[ZWXY]}{\mathbb{P}[WXY]} \tag{2.25}$$

$$= \mathbb{P}[W] \mathbb{P}[X \mid W] \mathbb{P}[Y \mid WX] \mathbb{P}[Z \mid WXY]. \tag{2.26}$$

The joint probability encoded by the network reads as follows

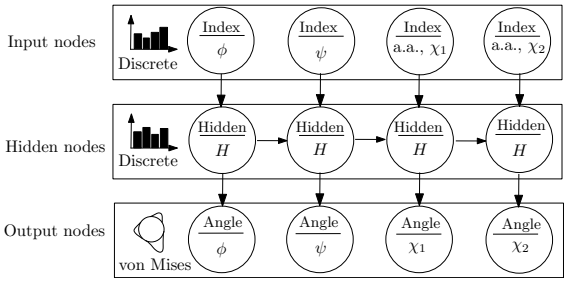
$$\mathbb{P}\left[\overline{A}, \overline{H}, \overline{\chi}, \phi, \psi\right] = \mathbb{P}\left[\overline{A}\right] \mathbb{P}\left[\overline{H} \mid \overline{A}\right] \mathbb{P}\left[\overline{\chi}, \phi, \psi \mid \overline{A}, \overline{H}\right]. \tag{2.27}$$

Using Eq. (2.25), one obtains the following expression for the conditional distribution of interest:

$$\mathbb{P}\left[\overline{\chi} \mid \phi, \psi, \overline{A}\right] = \frac{\mathbb{P}\left[\overline{\chi}, \phi, \psi, \overline{A}\right]}{\mathbb{P}\left[\phi, \psi, \overline{A}\right]} = \frac{\sum_{\overline{H}} \mathbb{P}\left[\overline{\chi}, \phi, \psi, \overline{A}, \overline{H}\right]}{\sum_{\overline{H}} \mathbb{P}\left[\phi, \psi, \overline{A}, \overline{H}\right]} = \frac{\sum_{\overline{H}} \mathbb{P}\left[\overline{\chi}, \phi, \psi \mid \overline{H}\right] \mathbb{P}\left[\overline{H} \mid \overline{A}\right]}{\sum_{\overline{H}} \mathbb{P}\left[\phi, \psi \mid \overline{H}\right] \mathbb{P}\left[\overline{H} \mid \overline{A}\right]}. \tag{2.28}$$

Note that $\mathbb{P}\left[\chi \mid H\right]$ is modeled using a 1D von Mises distribution.

Figure 2.14 The Dynamic Bayesian Network used in BASILISK for a single amino acid. Adapted from [HBP+10]. The input nodes specify the a.a. type and the angles; the output nodes specify the dihedral angles. The number of slices is equal to 2 (backbone angles) + 4 (maximum number of dihedral angles.). The parameters of the von Mises distribution used in a slice depend on the value stored in the hidden node.



Results and assessment. A twofold assessment is presented. The first one is a direct comparison to backbone independent rotamer libraries. First consider the frequencies P_{data} of rotamers observed in a reference dataset. For a reference (backbone independent) library Q of rotamers, assume one has computed $\mathbb{P}\left[\overline{\chi} \mid R_A\right]$, the probability of the vector of angles $\overline{\chi}$ given the rotamer R_A. The quality of this particular library is given by $D_{\mathrm{KL}}\left(P_{data} \| P_Q\right)$. Similarly, the quality of the rotamers provide by BASILISK is given by $D_{\mathrm{KL}}\left(P_{data} \| P_{\mathrm{BASILISK}}\right)$. The comparison between q and BASILISK is provided by the difference $D_{\mathrm{KL}}\left(P_{data} \| P_Q\right) - D_{\mathrm{KL}}\left(P_{data} \| P_{\mathrm{BASILISK}}\right)$. In general, BASILISK captures the conformational preferences of a.a. more accurately than other libraries. The same KL based analysis can be used to compare BASILISK with and without backbone dependency. As expected, the former is better at capturing conformational preferences.

The second assessment is concerned with the generation of high quality side chain conformations – side chain placement for a fixed backbone. The energy model, a plain 6-12 Lennard-Jones model, is used via Boltzmann's factor to assign a probability to each structure. This energy (multiplied by the probability assigned by BASILISK to the side chain conformation in a second experiment) is used in the Metropolis-Hastings criterion. At each iteration, three new side chain conformations at random sequence positions are proposed. After 500,000 MCMC iterations, the lowest energy (highest probability) structure is retained. A χ angle is termed correct if the difference with the value in the crystal structure is less than 20 degrees. In using the BASILISK likelihood as a pseudo-energy component, results on par with specialized programs such as SCWRL4 are obtained.

As a conclusion, using a single model for all a.a. reduces the number of parameters to be estimated, and makes it possible to transfer information, which is relevant for a.a. with similar properties. But it also faces the risk of mitigating properties for a.a. from different groups.

2.6.6 Notes

The problem of packing side chains was originally studied using simulated annealing [LS91]. Using mean field theory, libraries of rotamers have be used to predict side chain conformations in a protein, as well as their

conformational entropy [KD94]. For docking, a hybrid strategy combining rigid body docking optimization of the partners and (via quenching/gradient descent in the space of rigid motions) and side chain packing (using Monte Carlo search) has been developed [GMW$^+$03]. Evolutionary approaches mixing discrete and continuous global optimization have also been considered [YTH$^+$02].

2.7 Backbone conformation sampling

2.7.1 Overview

Loops and their importance. Flexible loops (segments along the backbone) in proteins are commonplace, and play a key role in various processes. One may for example consider the role of complementarity determining regions (CDR) in antibodies, the role of enzyme loops moving functional domain around, linkers connecting two essentially) rigid bodies, flexible regions in intrinsically disordered proteins, etc. Modeling loop has long been identified as a key problem [FD$^+$00], with a particular interest for homology models since templates for flexible loops are in general not available. After two decades of intense research, modeling flexible loops remains a challenge, especially for highly diverse loops [SK12, MSD18]. As a matter of fact, as of today, it is considered that modeling loops beyond 12 amino acids is hard or beyond reach [BCC21].

Beyond isolated loops, molecular modeling also needs to cope with multi loop systems. Closed loops associated to the covalent graph are found due to disulfide bonds in proteins, and naturally due to multiple cycles in poly-cyclic molecules–such as steroids for example. Closed loops are also found beyond covalent bonds, due in particular to hydrogen bonding. The complexity of such situations is described, from the purely topological standpoint, by the so-called cyclomatic number (in graph theory), or equivalently the first Betti number β_1 in algebraic topology [EH10].

Goals: structure versus thermodynamics versus kinetics. When working with loops, it is important the keep final goal in mind, given the trichotomy structure - thermodynamics - kinetics mentioned in Introduction. For example, selected works solely target the reconstruction of loops observed in crystal structures. Such works belong to the realm of structure. On the other hand, other works addressed the problem of sampling sense loop conformations in the thermodynamics realm, for example an NVT ensemble. Such ensembles of conformation are usually meant to compute observables, e.g. a heat capacity of binding affinity. Finally, loops sampling methods can also be designed in the context of kinetics, to compute transition rates and the stability of meta-stable states.

In the sequel, we focus is deliberately on structural / geometric questions.

Modeling loops: methods. From the methods standpoint, the methods developed to study loops are remarkably diverse, and we may ascribe them to three tiers.

First, the loop can be deformed by some continuous strategy. In this vein, various methods were designed by rotating rigid bodies about rotation axis defined by C_α carbons delimiting a backbone segment. Such methods include `Crankshaft` [Bet05], as well as `Backrub` [DAIRR06, SK08]. These methods proved successful to reproduce motions observed in crystal structures, but hey are essentially limited to hinge like motions. Second, the other hand a piece of the loop may be deformed in such a way that the connectivity is disrupted, which requires performing a *loop closure* step. As we shall see, these methods, which lie in the lineage of [GS70], give rise to a remarkably rich body of work. Finally, one may also consider a loop as a sequence of protein fragments stitched together. Give the high resolution structures from the PDB, it is therefore natural to reconstruct loops using existing structures [JT86]. As we shall see, such approaches can be combined with the previous two.

Section overview. The goal of this section is to present the fundamental aspects of loop reconstruction and sampling, especially for loop closure and database approaches. In doing so, the ambition is to shed light on the trade-off between the complexity of methods, and the diversity of solutions generated. That is to say,

our focus is deliberately on structural rather than thermodynamic aspects–which is why detailed balanced, Metropolis-Hastings criteria and the like are omitted. The section is organized as follows:

- Section 2.7.2 introduces background concepts.
- Section 2.7.3 deals with *ab initio methods* based in particular on inverse kinematics, a class of methods to reconstruct backbones based solely on geometric constraints, in the absence of any external knowledge.
- Section 2.7.4 present hybrid methods, *i.e.* methods combining calculation of the inverse kinematics (IK) type, and also exploiting data from the PDB. A natural strategy to handle (long) loops indeed consists of splitting the reconstruction problem into easier sub-problems. Such strategies typically combine *ab initio* approaches based on IK, and database approaches exploiting existing fragments available in the Protein Data Bank.
- Section 2.7.5 discusses partially rigid geometry methods, exploiting a relaxation of the constraint on fixed bond lengths and valence angles.

2.7.2 Loop closure and inverse kinematics: background

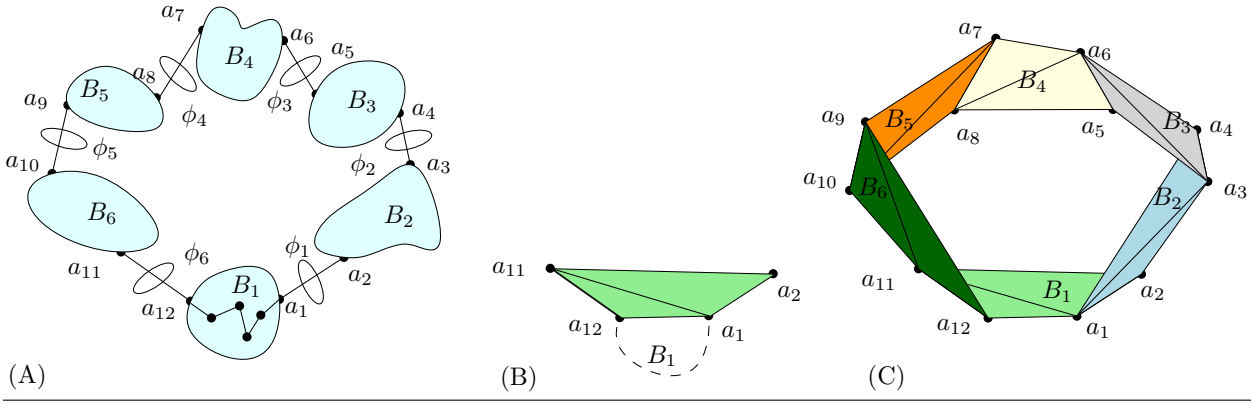
Molecular modeling and mathematical problems

Given a molecule represented in internal coordinates, a simplification consists of considering that only dihedral angles can vary [EH91]. This hypothesis is supported by the fact that in force fields, spring constants for these angles are much *softer* than for the other IC.

Using the idealized *rigid geometry* model, the backbone of the molecule can be viewed as a kinematic chain, a classical representation in robotics [Cra89, PC94]. Consider a robot consisting of an articulated arm connected by joints. The robot has a base and an end effector. The inverse kinematics problem seeks the possible geometries (angular values of the joints) compatible with a given position/orientation of the end effector. This model triggered a large body of work at the interface between chemistry, robotics, applied mathematics, and computer science, see e.g. [GS70, RR90, PC94, MC94, EM99, CSRST04, CS04, NOS05, PRT$^+$07, CSWD06, MLL08, CLW$^+$16] and references therein. In robotics, joints that allow two links to rotate *w.r.t.* one another at a fixed angle are called Rotator pairs or R-pairs. A molecule with 6 rotatable bonds is analogous to a 6R linkage, a case also referred to as the 6 rotors - 6 bars (6R-6B) system.

In the sequel, we present the classical methods.

Figure 2.15 A general 6 rotors - 6 bars (6R-6B) system. Adapted from [PRT$^+$07]. **(A)** Six rigid bodies linked by six linkers/bars, each endowed with a rotational degree of freedom. **(B)** The relative position of the two segments sandwiching a rigid body does not change, which is modeled by a rigid tetrahedron. **(C)** The distance geometry model associated to the 6R-6B system.



General loop closure using Denavit-Hartenberg local frames

Denavit-Hartenberg frames are classical to model kinematic chains in robotics [DH55, Cra89].

Denavit-Hartenberg local frames. We wish to define the DH frame associated with the i-th rotatable bond of a loop L. Let Z_i be a unit vector along the i-th rotatable bond b_i The DH local frame F_i is s defined as follows (Fig. 2.16):

$$\begin{cases} Z_i = & \text{unit vector along the i-th bond} \\ X_i = Z_{i-1} \times Z_i \\ Y_i = Z_i \times X_i \end{cases} \tag{2.29}$$

In addition, one defines the following parameters:

- u_i: the point on the line through b_i nearest to the line through b_{i-1}. Note that point u_i is supported by the line bi-orthogonal to $line(b_{i-1})$ and $line(b_i)$.
- distance d_i: distance along b_i from u_i to the closest point to u_{i+1} – the projection of u_{i+1} onto the line through b_i.
- offset a_i: distance from the line $line(b_i)$ to u_{i+1}.
- angle α_i: from Z_{i-1} to Z_i.
- torsion / dihedral angle θ_i: $\angle X_{i-1}, X_i$ about Z_{i-1}.

Remark 2.4. *When the bonds are consecutive – equivalently all backbone atoms are used: the u_i are the atomic positions, d_i are the bond lengths, α_i are the valence angles, and the θ_i dihedral angles.*

Two consecutive frames F_i and F_{i+1} and related by forward rigid motions $A_{i\to i+1}$. (Nb: one can also use the backward mapping defined by matrices $A_{i\leftarrow i+1}$.) To see how, denote $T_l(d)$ the translation of distance d along the vector l, and $R_l(\alpha)$ the rotation of angle α along the axis l. (Nb: these transformations are written in homogeneous coordinates.) The DH frames F_i and F_{i+1} satisfy:

$$F_{i+1} = A_{i\to i+1}F_i, \text{ with } A_{i\to i+1} = T_{Z_i}(d_i) \cdot R_{Z_i}(\theta_{i+1}) \cdot R_{X_{i+1}}(\alpha_{i+1}) \cdot T_{X_{i+1}}(a_i). \tag{2.30}$$

Indeed:

- $T_{X_{i+1}}(a_i)$: translation along X_{i+1} to compensate the offset a_i,
- $T_{Z_{i+1}}(d_i)$: translation along Z_{i+1} to compensate the distanced_i
- $R_{X_{i+1}}(\alpha_{i+1})$: rotation along X_{i+1} to bring Z_i onto Z_{i+1},
- $R_{Z_i}(\theta_{i+1})$: rotation along Z_i to bring X_i onto X_{i+1},

Consider now two fixed atoms u_0 and u_n, separated by n rotatable bonds. Using Eq. 2.30 iteratively, we obtain

$$F_n = A_{1\to n}F_1, \text{ with } A_{n-1\to n}F_{n-1} = A_{n-1\to n} \dots A_{1\to 2}. \tag{2.31}$$

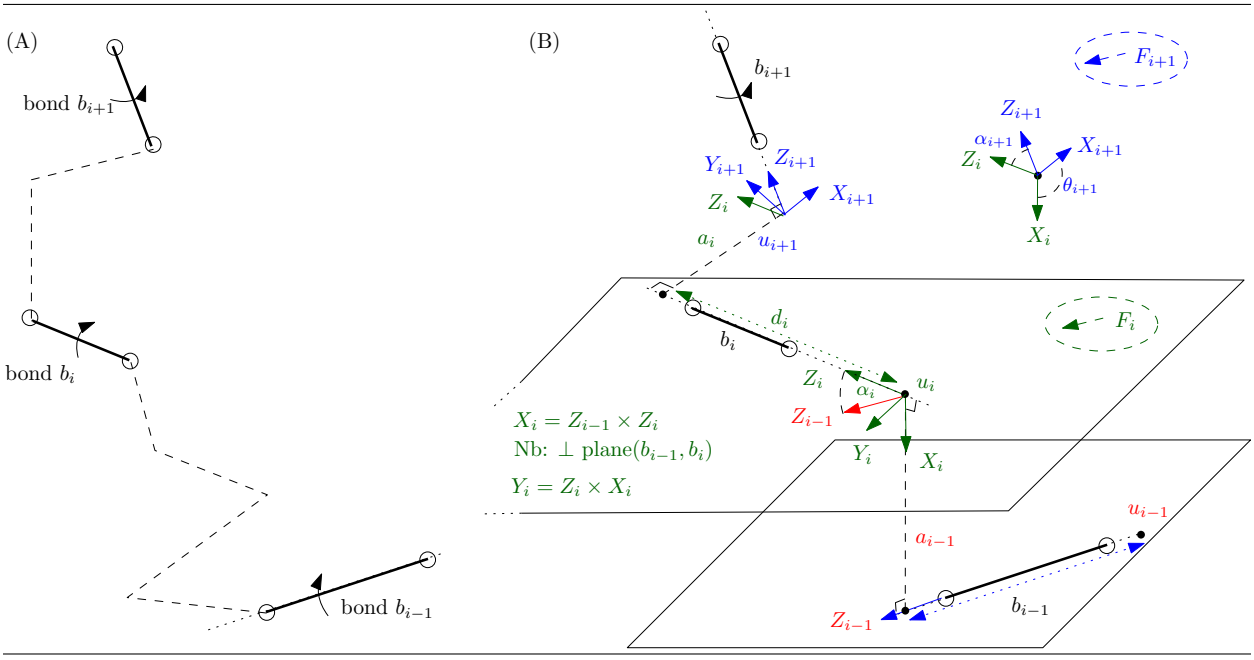
The product of Eq. 2.31 readily makes it possible to deform the loop L (Fig. 2.18). Consider two internal geometries of the loop L, defined by the parameters $I = \{(a_i, d_i, \alpha_i, \theta_i)\}$ and $I' = \{(a_i', d_i', \alpha_i', \theta_i')\}$. Consider the two transformations $A_{1\to n}(I)$ and $A_{1\to n}(I')$. The loop closure reads as

$$A_{1\to n}(I) = A_{1\to n}(I'). \tag{2.32}$$

The previous equation can be reduced to six equations describing the translation and rotation bringing F_1 onto F_n.

Applications to moveset will be treated in section 2.7.5.

Figure 2.16 Denavit-Hartenberg local frames F_i for possibly non consecutive rotatable bonds of a loop L. (A) Kinematic chain with n rotatable bonds $b_1, \ldots, b_n$. Three consecutive bonds b_i are represented. **(B)** The DH frames F_i defined for all rotatable bonds, in red/green/blue for $F_{i-1}/F_i/F_{i+1}$ respectively. Note that when the rotatable bonds are consecutive, the u_i are the atomic positions, d_i are the bond lengths, α_i are the valence angles.



Geometric solutions to IK problems and molecular conformations

For large enough systems *i.e.* sufficiently many degrees of freedom, one expects IK problems to admit a continuous solution set [Cra89, Lat12, LaV06]. However, ending up with an exhaustive description of such sets is in general difficult, setting aside the problem of sampling curved manifolds representing solutions. Therefore, particular cases of IK problems admitting finite solution sets are a route of choice. From the algebraic standpoint, such problems generally reduce to finding the real roots of a univariate polynomial. In molecular science, a root generally corresponds to one conformation, so that enumerating all solutions (a finite set by hypothesis) makes it possible to consider ensembles of conformations. This strategy is a route of choice when reconstructing flexible loops[], or when exploring conformations in the process of molecular simulations [].

Physical properties: rigid versus partially rigid models

Coming up with IK problems admitting finite solution sets requires imposing constraints on the geometric models. Mirroring these in molecular modeling requires considering the three classes of variables/internal coordinates which can be used for the loop/ring closure: bond length, bond angles, dihedral angles. Stiffness constants associated with the deformation of torsion angles are one or two orders of magnitude smaller than those associated with deforming bond lengths and bond angles []. The model in which bond lengths and valence angles are fixed, while torsion angles vary is termed the *rigid geometry model*. As we shall see, this model, in which one can only modify dihedral angles, is especially suited to define IK problems with finite solution sets. We note however that the rigidity of using only dihedral angles will typically force the appearance of unfavorable local structure in the chain. In the scope of Monte Carlo simulations, this is clearly detrimental since such conformations result in a high considering rates. This motivates less strict models, including the *Partially rigid geometry model*.

Figure 2.17 Denavit-Hartenberg local frames for a protein backbone: construction of the angle θ_i.

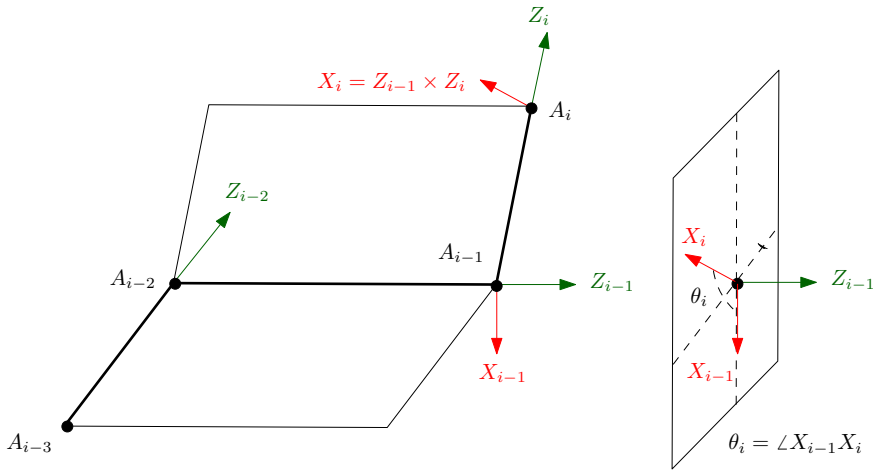
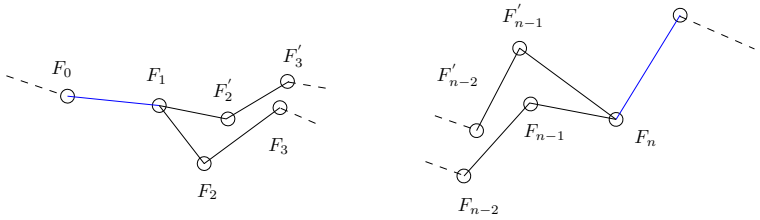


Figure 2.18 Loop closure using Denavit-Hartenberg local frames. The blue bonds are kept fixed.



2.7.3 Loop closure in the rigid geometry model

Single loop and the cyclic coordinate descent (CCD) algorithm

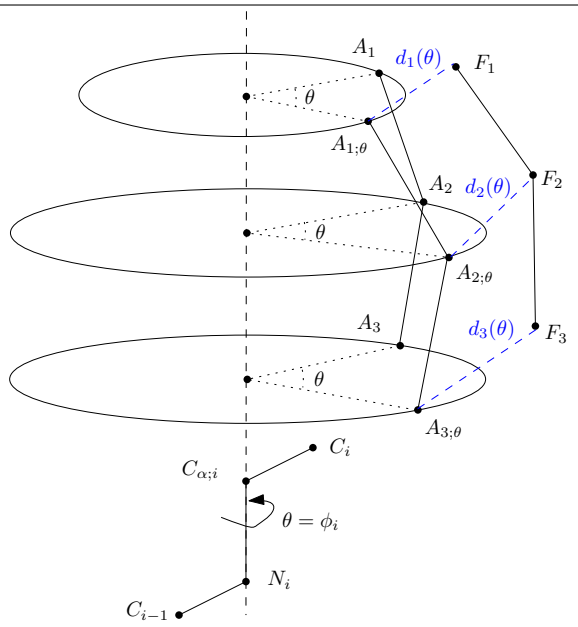
Consider a chain whose residues are numbered from 0 (left, N-ter) to n (right, C-ter). The goal is to find a conformation of this chain, whose left and right a.a. match two anchors (fixed amino acids). The algorithm takes as input a random initial configuration of all dihedral angles (ϕ, ψ) of the chain. These values may be picked from uniform distributions, or better, from Ramachandran distributions of the individual amino acids. The cyclic coordinate descent algorithm (CCD) [CDJ03] consists of iteratively changing the values of the backbone (ϕ, ψ) angles, to as to move last amino acid as close as possible of the target (the right anchor, C-ter). The elementary operation to do so deals with one angular value θ ($\theta = \phi$ or $\theta = \psi$) at a time. To see how, let us use the following notations: (i) $F_i, i = 1, 2, 3$ represent the N, C_α, C of the fixed amino-acid on the C-ter of the loop; (ii) $A_i, i = 1, 2, 3$ represent the same atoms on the last amino acid of the loop being reconstructed; (iii) $A_{i;\theta}, i = 1, 2, 3$ represent those atoms rotated by an angle θ about an axis corresponding to an ϕ or ψ angle (Fig. 2.19). Angle θ is chosen to minimize the sum of squared distances $d_1(\theta)^2 + d_2(\theta)^2 + d_3(\theta)^2$, with $d_i(\theta) = \|F_i A_{i;\theta}\|$. The optimal angular value is obtained from a simple analytical calculation.

For a given proposed change in θ, a change in ψ is proposed. This pair is accepted or rejected using a Metropolis criterion – based on the likelihood of the pair (ϕ, ψ) rather than some energy term. The process is iterated along the chain, until closure – or failure.

The 6 rotator-6 (6R-6B) bar problem

The local deformation and ring closure problems are formally introduced in [GS70]. While the ring closure name is self-supporting, the local deformation problem can be stated as follows. Consider a chain of $n + 1$

Figure 2.19 Single dihedral angle optimization in the cyclic coordinate descent (CCD) algorithm. Adapted from [CDJ03]. Angle θ is chosen to minimize the sum of squared distances between the fixed atoms of the anchor, and the atoms of the loop being deformed. See text for details. The CCD algorithm is based on the iterating of this process for all angles of the backbone.



atoms $a_0, \ldots, a_n$ connected by n bonds (Fig. 2.20(Left)). Keeping the bond lengths d_i and valence angles θ_i fixed, one wishes to find values for the dihedral angles $\omega_1, \ldots, \omega_n$ compatible with the position and orientation of the end atom a_n. The former problem is a particular case of the latter when one identifies endpoints. In this seminal paper, the authors show that there are $n-6$ independent angles. Intuitively, this number is best understood as follows. Consider the forward map assigning the position and orientation of a_n given the n angular values ω_i. Each constraint on the position and orientation of a_n can be written as an implicit equation on the n variables. Under genericity assumption, this yields a solution space which is a $n-1$ manifold. If the six solution spaces corresponding to the six constraints intersect transversely, one gets a solution space of dimension $n-6$. Observe in particular that when $n=6$, one obtains a zero dimensional set *i.e.* a finite number of solutions.

The problem can be generalized to the case where the portion of the molecule of interest has a rigid body (Fig. 2.20(Right)).

Deriving loop closure equations using line geometry. Loops closure for the 6R-6B mechanism can also be obtained using the geometry of lines [CLW+16]. Consider as usual a kinematic with rotor links b_i. Bonds lengths and valence angles are fixed, and only the rotor angles $(t_1, \ldots, t_n)$ associated with the n rotor links can vary (Fig. 2.21). Considering the i-th bond/bar, let R_i be its position in some reference frame, and Γ_i the unit vector along this bond/bar.

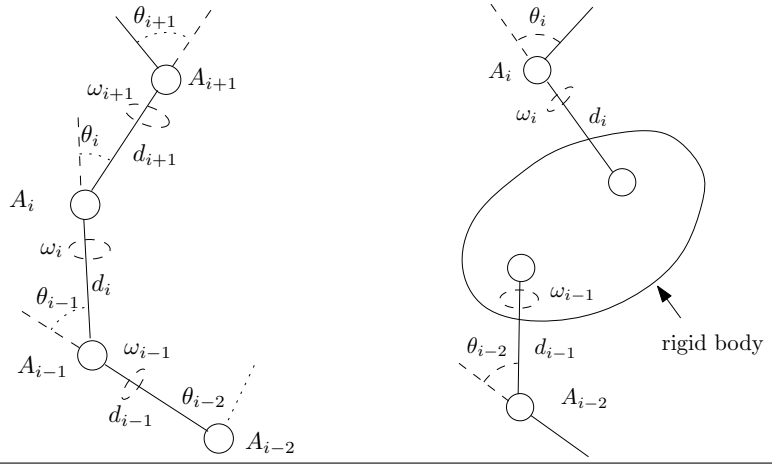
Consider a point R outside the region of interest, and therefore fixed when the vector $\boldsymbol{t}$ changes. An infinitesimal change dt_i for th i-th angle t_i yields the following change for point R

$$d\boldsymbol{R} = \Gamma_i \times (\boldsymbol{R} - \boldsymbol{R}_i) dt_i$$

But since $\boldsymbol{R}$ is fixed, aggregating the contributions of the n angles yields

$$d\boldsymbol{R} = 0 = \sum_i (\Gamma_i dt_i) \times (\boldsymbol{R} - \boldsymbol{R}_i) \tag{2.33}$$

Figure 2.20 Ring closure in a molecule with rigid portions. Adapted from [GS70].



or equivalently

$$(\sum_i \Gamma_i dt_i) \times \boldsymbol{R} - (\sum_i dt_i \Gamma_i \times \boldsymbol{R}_i) = 0 \tag{2.34}$$

Since this hold for all points R, the two sums vanish independently. For the i-th bond, consider the 6×1 vector obtained by pooling Γ_i and $\Gamma_i \times \boldsymbol{R}_i$. The previous equation yields

$$P \begin{pmatrix} dt_1 \\ \dots \\ dt_n \end{pmatrix} = 0 \text{ with } P \stackrel{Def}{=} \begin{pmatrix} \dots & \boldsymbol{P}_i & \dots \end{pmatrix} = \begin{pmatrix} \dots & \Gamma_i & \dots \\ \dots & \Gamma_i \times \boldsymbol{R}_i & \dots \end{pmatrix} \tag{2.35}$$

When the $6 \times n$ matrix P has full rank, the implicit function Thm guarantees that 6 variables can be expressed as differentiable function of the remaining 6 ones. Call the free variables *drivers* and the dependent ones *pivots*. Upon re-indexing, consider the following notations $\{q_k := t_{i_k}\}_{k=1,\dots,n-6}$ for the drivers, and $\{p_k := t_{i_k}\}_{k=1,\dots,6}$ for the pivots. One obtains the differential of the pivot variables as a function of the differential of the drivers

$$d\boldsymbol{p} = -J^{-1}\boldsymbol{Q}d\boldsymbol{q}, \text{ with } J := \begin{pmatrix} \dots & \boldsymbol{P}_i & \dots \end{pmatrix} \tag{2.36}$$

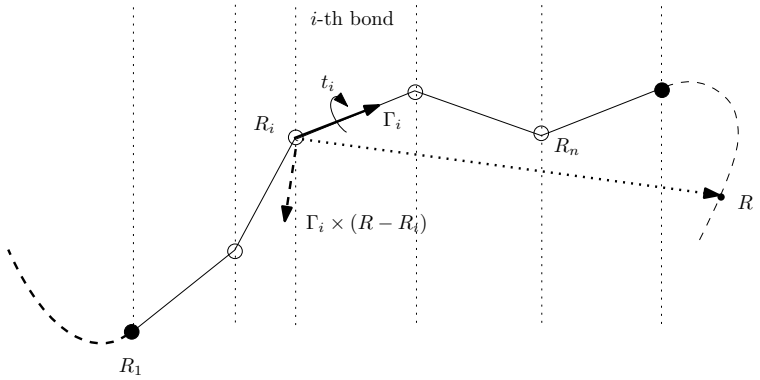
The columns of the Jacobian J are the so-called Plücker coordinates of the pivot axes [HHH78]. If the determinant of the Jacobian in non null, the IK problem is termed well posed, and faces a singularity otherwise. Equation 2.36 can be used to derive the closure equations, which are polynomials in the sines and cosines of the pivot values p_{i_k}.

Tripeptide/Triaxial loop closure

The original loop closure problem can be specialized by considering 6 dihedral angles corresponding to three peptide bonds (consecutive or not), a case study first considered for cyclic peptides [GS78]. This is a particular case of that studied in the previous section, since bars come into pairs which share an endpoint. This particular case is termed the *Tripeptide/Triaxial loop closure* (TLC). In the sequel, we follow [CSJD04, CSWD06] and outline the elegant and powerful solution to TLC, based on a degree 16 polynomial. The reader may also consult [ORC22], where the method is explained in detail.

In considering three consecutive a.a., the six rotatable bonds / dihedral angles $\{(\phi_i, \psi_i)\}_{i=1,2,3}$ are found before / after the C_α carbons (Fig. 2.22(A)). In using these six dihedral angles, the atoms $N_i, C_{\alpha;i}, C_{\alpha;i+2}, C_{i+2}$ are fixed in the global coordinate system (frame). This observation is important, since the calculation will be carried out in a different frame, in which these four atoms are not fixed even though they relative position remains so.

Figure 2.21 Derivation of loop closure equations: construction from [CLW$^+$16]. The segment of interest consists of n rotatable bonds between points R_1 and R_{n+1}. Γ_i is a unit vector along the i-th bond.



More precisely, the method consists of modeling the geometry of the system by three rigid bodies rotating around the three axis $C_{\alpha;i}C_{\alpha;i+1}$, $C_{\alpha;i+1}C_{\alpha;i+2}$ and $C_{\alpha;i+2}C_{\alpha;i}$. In doing so, the six angles $\{(\phi_i, \psi_i)\}_{i=1,2,3}$ get replaced by three rotation angles $\tau_i, i = 1, 2, 3$ (Fig. 2.22(B)). The conservation of the valence angles θ_i at the C_α carbons imposes three constraints–recall that valence angles and bond lengths are fixed in the rigid geometry model. To accommodate these, one defined thee local frames (one for each C_α) whose z axis are colinear and perpendicular to the place of the C_α triangle. In these frames, writing the conservation of the θ_i angles yields three biquadratic (quadratic in two variables) polynomials in the three variables $u_i = \tan \tau_i/2$. The elimination of two of these variables yields the degree 16 polynomial. Every real root of this polynomial defined an embedding for the three rigid bodies. Moving back that associated to the four atoms $N_i, C_{\alpha;i}, C_{\alpha;i+2}, C_{i+2}$ into its original positions defines a valid reconstruction.

Remark 2.5. *Interestingly, the idea of using rigid bodies for a tripeptide already appears in [DAIRR06].* `Crankshaft` *indeed performs a rotation around $C_{\alpha;i}C_{\alpha;i+2}$, and proceeds with rotations around $C_{\alpha;i}C_{\alpha;i+1}$ and $C_{\alpha;i+1}C_{\alpha;i+2}$. However, valence angles are not preserved, which is the main difficult in TLC.*

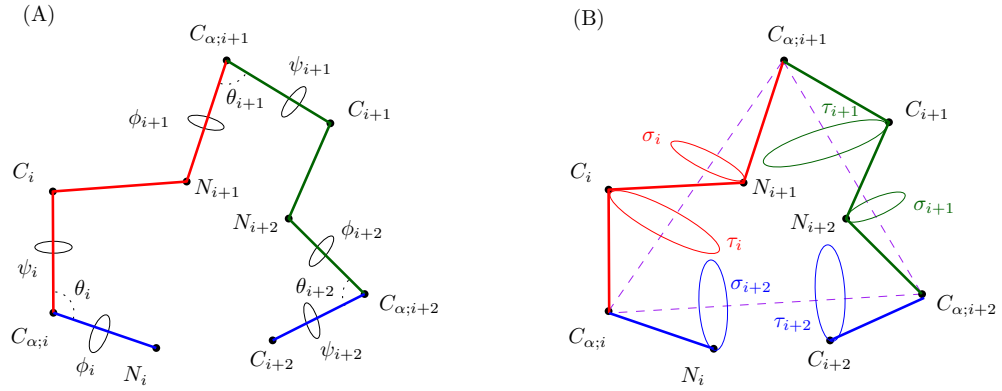
Remark 2.6. *A generalization making it possible to change other dihedral angles is also proposed [CSJD04], yielding a continuous problem. The reader is also referred to [PRT$^+$07] for the historical notes on the quest for the 16 real solutions in general and for the six-atoms ring. The same paper also survey such problems, also providing a general method in the realm of real algebraic geometry.*

Application: concerted rotations – `Conrot` and `Conrot-CRA`

One of the earlier example implementing loop closure using only dihedral angles is the concerted rotation algorithm called `Conrot` [DBT93]. Consider the problem of changing the conformation of a backbone fragment away from the chain endpoints – that is one must respect closure constraints. In the first step, a dihedral angle is perturbed at random. In the second step, six dihedral angles are used to obtain loop closure. In case multiple solutions exist, one is chosen at random. Overall, this moveset therefore alters seven dihedral angles in the chain.

The concerted rotations strategy gave rise to a fruitful lineage. The algorithms `Conrot-CRA` [UJ03] and `Conrot-CRISP` [BBEJ$^+$12] obtain loop closure using three valence angles and three dihedral angles. A more direct application of the initial `Conrot` is the method from [DABV$^+$18], where the six (ϕ, ψ) angles of a tripeptide are used to restore loop closure upon perturbing a particular dihedral angle. These algorithms will be detailed in the sequel.

Figure 2.22 The Triaxial (or Tripeptide) Loop Closure Problem. Adapted from [CSJD04]. The three colors correspond to the three rigid bodies involved in the loop closure. **(A)** The original problem involves six rotations corresponding to the angles $\{(\phi_i, \psi_i)\}_{i=1,2,3}$ found before/after the C_α carbons. **(B)** The solution to TLC uses (i) three rotation angles τ_i corresponding to three rigid bodies around the three axis $C_{\alpha;i}C_{\alpha;i+1}$, $C_{\alpha;i+1}C_{\alpha;i+2}$ and $C_{\alpha;i+2}C_{\alpha;i}$; and (ii) three constraints stating that the valences angles θ_i must be conserved.



Nb: indices mod(3), e.g., $\sigma_{i+2} = \sigma_{i-1}$

2.7.4 Loop closure: database driven and combinatorial approaches

Incremental construction using fixed length fragments

For long protein backbone loops, a classical approach to reconstruct and/or sample conformations consists of mixing ab initio and database approaches, an approach initiated in [JT86]. . In the sequel, we outline two methods from [KGLK05], to reconstruct a loop involving n amino acids.

The first method consists of seeking whole candidate loops in the Protein Data Bank. Consider a loop L to be reconstructed, as well as the six amino acids flanking it – three to the left and three to the right. Focusing on the C_αs of these flanking residues yields a total of 15 distances. Consider now all contiguous backbone fragments of length $n + 6$ found in structures from the PDB. For such a fragment, one can also compute the aforementioned 15 distances. The method consists of retaining those fragments with a Root Mean Square $RMS(L, S) < 1$Å. These candidate reconstructions for L may also be filtered out using additional criteria, based on chemical considerations, but also to filter out the geometric redundancy.

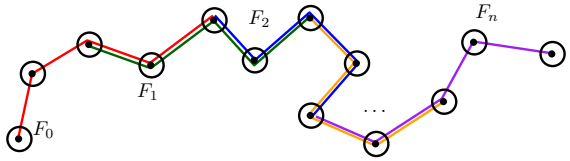
This method appears to be well suited for small loops, as long loops are not so frequent in structures from the PDB.

The second method is a greedy/incremental reconstruction, which comes in two guises: unidirectional (bidirectional) incremental construction. Both are similar and we only describe the former. The strategy consists of iteratively concatenating fragments of length l ($l = 5$ in [KGLK05]) at the end of the loop under construction. To do so, the first three C_α of the fragment are superimposed onto the last three C_α of the loop under construction – the optimal rigid motion to do so must be computed. Because a new fragment yields $f - 3$ new amino acids, a total of $\lceil l/(f-3) \rceil + 1$ fragments are required – the last one being the *staple* ensuring compatibility. Assuming there are L fragments in the DB, the total number of reconstruction is thus

$$N = L^{\lceil l/(f-3) \rceil + 1} \tag{2.37}$$

This approach appears to be well suited to loops of moderate length, but faces a combinatorial explosion when the size of the database increases.

Figure 2.23 Incremental construction of a loop, by adding superimposed fragments of a fixed length. Adapted from [KGLK05]. The left and right fragments (F_0 and F_n) are fixed. The incremental elongation consists of choosing from a database a fragment F_i whose first three C_α carbons define a geometry compatible with the last three C_α from F_{i-1}. In the unidirectional construction, F must be compatible with the right anchor F_n. In the bidirectional constructions, the chains elongated independently from the left and right must meet in the middle.



Kinematic loop closure (KIC) combined to Ramachandran sampling

The TLC (for possibly non contiguous C_α carbons) can be used in the context of information provided by Ramanchandran distributions [MCK09], to model whole loops (backbone and side chains) with sub-angstrom accuracy.

Consider an n residue chain, in which three C_α carbons are identified as *pivots*. Dihedral angles can be randomly sampled from residue specific distributions, resulting in the opening of the loop. TLC can then be applied to check whether the whole chain can be closed. Up to 16 solutions may be obtained, the result being called a KIC move. To model an atomic model of a loop, KIC moves have been combined to two Monte Carlo based minimization protocols, respectively optimizing a coarse grain model (side chain represented by one pseudo-atom), and the full atomic model. The lowest energy model is finally reported [MCK09].

This method was subsequently evolved to the *next-generation KIC* [SK13] based on three sampling strategies, in the scope of `Rosetta`[1]. The first uses taboo search to improve the diversity of low resolution models obtained. The second resort to ω angle sampling as well enhanced (ϕ, ψ) sampling based on neighbor dependent Ramachandran distributions [TWS$^+$10], Finally, the third uses a ramping strategy to gradually change the terms in the `Rosetta` energy function, so as to overcome high energy barriers. (Nb: a classical ramping strategy consists of changing the terms in the VdW energy, which is sensitive to small inter atomic distances.)

Hybrid methods using energy functionals

The Ramachandran sampling just used is one step towards using biophysical properties. The following two methods take one step beyond, combining local loop closure heuristics, and energy based functions.

Algorithm LEAP is hybrid method, based on three steps [LZZ14]. First, conformations of the backbone are generated with the CCD algorithm [CDJ03]. Second, side chains are built using an energy based potential (OSCAR). Last, the top selected models are further optimized using the OSCAR potential for flexible side chain rotamers, and the `CHARMM` bond potential energies.

The method Sphinx is similar in spirit, but combines four different ingredients [MNK$^+$17]. Assume one wishes to build a loop from N-ter to C-ter. The first step consists of assembling a database of fragments, using several criteria (i) fragment length (ii) matching of the geometry of the left two flanking residues (iii) sequence similarity. The second one is the elongation step, which consists of incrementally adding fragments. The third one is the loop closure, performed with the CCD algorithm [CDJ03]. Finally, the last step performs a decoy selection, and the addition of side chains–using an energy minimization based on `Rosetta`.

[1] See also `https://www.rosettacommons.org/docs/latest/application_documentation/structure_prediction/loop_modeling/loopmodel-kinematic`

Incremental construction using tripeptides and reinforcement learning

An elegant reconstruction method combining two key ingredients s presented so far was recently proposed in [BMV+19]: the first ingredient is the decomposition of the loop processed into small fragments, tripeptides here (the loop length is assumed to be a multiple of three); the second one is the ability to reconstructed solutions using the TLC algorithm. The method comes into two guises, namely without and with reinforcement learning (RL).

To present these methods, we assume that a database of tripeptides. The DB is assembled by sliding a window of size 3 along polypeptide chains from the PDB, keeping only tripeptides void of intersection with an alpha-helix and a beta strand. Also, tripeptides collected are put into buckets labeled with their type (the three amino acids). A tripeptide is represented by its *state* that is the three angles ϕ, ψ, ω.

Consider first the method without RL. The method grows segments of length $p - 1 (\geq 1)$ (left hand side of the loop) and $n - p (\geq 1)$ (right hand side of the loop), and attempts to bridge the gap using the TLC algorithm. Along the way, an important ingredient is a random perturbation of the geometry of the tripeptides used, both during the elongation steps, and the closure with TLC.

While the method is effective for loops of moderate length (≤ 10), it also faces the combinatorial explosion already mentioned for [KGLK05]. This difficulty motivates the RL step, meant to prune infeasible regions of the search space. In a nutshell, define the signature of a tripeptides as the vector joining the N and C atoms. An octree is built to decompose the space of signatures and store tripeptides into homogeneous groups, as a function of their involvement in successfully reconstructed loops. More precisely, when incrementally building the loop, an octree is used for each tripeptide slot in the reconstruction. To sample a state for a candidate tripeptide, all leaves of the current tree are scored (using a score based on a *learning rate*), and the leaf providing effectively the tripeptide used is chosen with a probability proportional to this normalized score [BMV+19]. The learning rate influences the speed for reconstruction vs the diversity of reconstructions. Each time a loop is reconstructed, statistics in all trees must be updated.

Overall, it is found that the RL based strategy yields a faster conformational sampling in most cases, the learning strategy being effective to *unlock* regions of the conformational space which are not accessible by the basic method.

2.7.5 Loop closure in the partially rigid geometry model

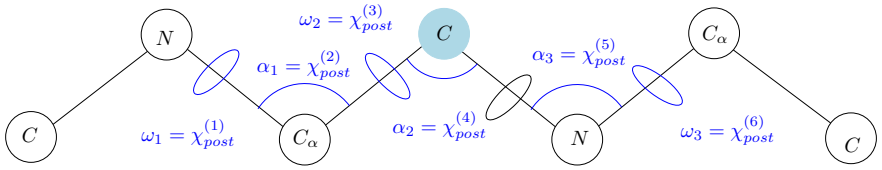
The efficiency molecular simulations heavily relies on the moveset used to generate novel conformations. While the rigid geometry assumption tames down the mathematical difficulties, it also drastically reduces the number of degrees of freedom. In turn, this may cause the appearance of unfavorable local structure in the chain, leading to an elevated potential energy and a high rejection rate using Boltzmann factor in the acceptance rule. In fact, allowing for small variations significantly increases the range of solutions [BK85].

Loop closure with 3 bond angles and 3 dihedral angles

`Conrot-CRA.` A difficulty inherent with `Conrot` [DBT93] is that the prerotation angles may induce a too large variation of the last atom moved–call it a, making impossible for the postrotation step to close the loop. To increase the probability of obtaining loop closure after the prerotation stage, and also to ease calculations, the `Conrot-CRA` algorithm [UJ03] utilizes three consecutive angles, three dihedral and three valence angles (Fig. 2.24). The prerotation angles are then biased to monitor the displacement of the aforementioned atom a. In the context of Monte Carlo sampling, the necessary correction factors for the Metropolis-Hastings criterion are also computed.

`Conrot-CRISP.` While `Conrot-CRA` [UJ03] minimizes the displacement of the last atom moved. But this strategy creates an imbalance in the magnitude of the prerotation versus postrotation angles, and the large changes of the latter atoms often yield steric clashes [BBEJ+12]. To remedy this problem, the `Conrot-CRISP` algorithm, which also uses the same set of 3+3 angles [BBEJ+12], couples the pre and postrotation steps

Figure 2.24 The six angles used to restore loop closure in `Conrot-CRA`. Adapted from [UJ03].

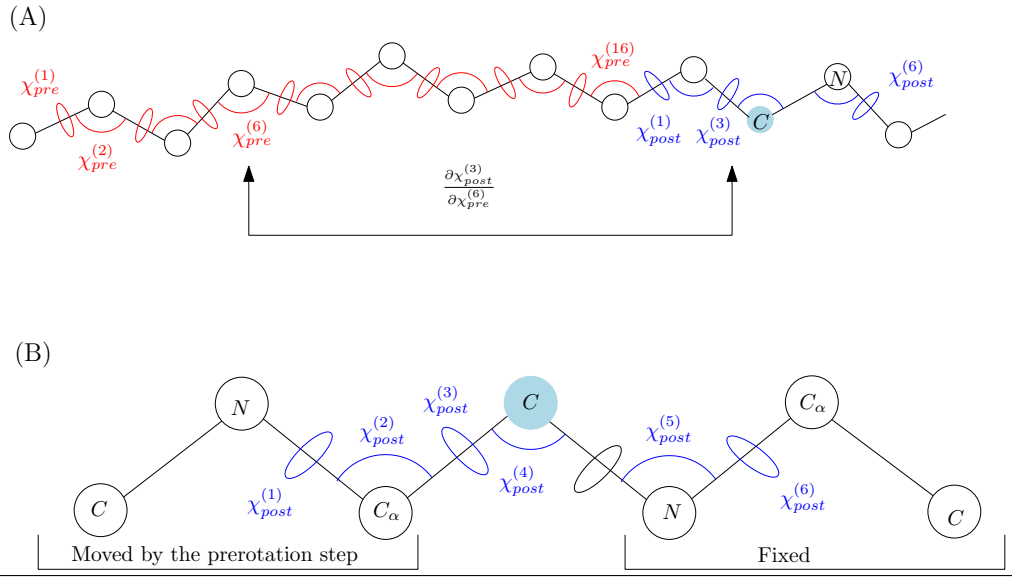


by pulling back the incidence of prerotation angles onto the postrotations ones into a unique probability distribution.

More precisely, assume that an arbitrary number of valence+dihedral angles are perturbed, braking the chain connectivity (Fig. 2.25(A)). These angles are denoted $\chi^{(i)}_{pre}, i = 1, \ldots, n$. In the second step, exactly six angles (three valence angles, three dihedral angles) are used to restore the connectivity (Fig. 2.25(B)). These angles are denoted $\chi^{(i)}_{post}, i = 1, \ldots, 6$. Overall, the method therefore alters the following set of n angles: $\boldsymbol{\chi} = \{\chi_1, \ldots, \chi_n\} = (\boldsymbol{\chi}_{pre}, \boldsymbol{\chi}_{post})$, We note in passing that these angles are inhomogeneous, since some are balance angles, and others dihedral angles. It is therefore useful to consider a diagonal scaling matrix C_n, to properly scale all angles as a function of their nature. (Nb: a diagonal term C_{ii} in matrix C_n is equal to a constant k for a bond or ω dihedral angle, and equal to one for a ϕ or ψ angle.)

The key advantage of `Conrot-CRISP` is that solving for the six angles $\chi^{(i)}_{post}$ yields calculations which are mush simpler than those related to the classical loop closure based on dihedral angles solely. These calculations make it possible to derive the analytical expression of the probability distribution with which the prerotation angles are chosen.

Figure 2.25 Moveset of the `Conrot-CRISP` method. Adapted from [BBEJ+12]. **(A)** The prerotation phase alters a number of valence and dihedral angles – in red, braking the integrity of the polypeptide chain. The postrotation phase modifies six angles (three valence, three dihedral) to rescue the integrity. One further computes the partial derivative of a postrotation angle as a function of a prerotation angle. **(B)** The analytical derivation of the values of the post-rotation angles $\chi^{(i)}_{post}$ is made via the placement of the C atom (blue atom).



Derivation of the probability density for prerotation angles. In the sequel, we assume the existence of analytical formulae for $\boldsymbol{\chi}_{post}$ as a function of $\boldsymbol{\chi}_{pre}$, see [BBEJ+12]. These formulae make it possible to compute the partial derivatives $\partial\chi^{(i)}_{post}/\partial\chi^{(j)}_{pre}$ can also be computed. Therefore, assembling the $6 \times n - 6$ dimensional Jacobian J of transformation gives the first order variation of the postrotation angles as a function of those of the prerotation angles:

$$\delta\boldsymbol{\chi}_{post} = J\delta\boldsymbol{\chi}_{pre}. \tag{2.38}$$

The previous calculation provides significant pieces of information. To see which, consider two conformations $\boldsymbol{\chi}$ and $\boldsymbol{\chi}'$. Also consider the (infinitesimal) variation of all angles between these two conformations, denoted $\delta\boldsymbol{\chi} = (\delta\boldsymbol{\chi}_{pre}, \delta\boldsymbol{\chi}_{post})$.

Assuming $\delta\boldsymbol{\chi}$ follows a multivariate Gaussian distribution, using the aforementioned C_n matrix and a parameter λ to code the desired level of locality, the probability of the move satisfies

$$\mathbb{P}\left[\delta\boldsymbol{\chi}\right] \propto \exp(-\frac{1}{2}\delta\boldsymbol{\chi}^{\mathsf{T}}\lambda C_n\delta\boldsymbol{\chi}). \tag{2.39}$$

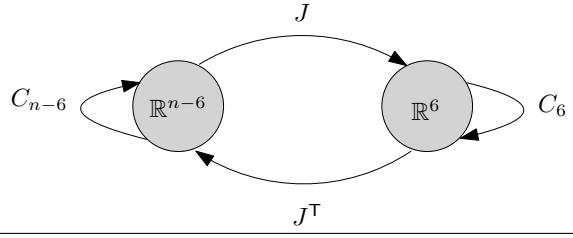
However, Eq. (2.38) provides the first order coupling between the pre and post angles. Using it, it can be shown that the previous formula can be written as a function of the prerotational angles only, namely:

$$\mathbb{P}\left[\delta\boldsymbol{\chi}\right] \propto \exp(-\frac{1}{2}\delta\boldsymbol{\chi}_{pre}\lambda(C_{n-6} + J^{\mathsf{T}}C_6J)\delta\boldsymbol{\chi}_{pre}), \tag{2.40}$$

where the diagonal matrix C_{n-6} controls the variations of the prerotational angles, and the (non diagonal) matrix $J^{\mathsf{T}}C_6J$ combines the Jacobian and the constraint matrix C applied to the postrotational angles (Fig. 2.26).

Equation (2.40) is the final probability distribution with which prerotation angles are sampled [BBEJ+12]. The `Conrot-CRISP` algorithm is available within the `PHAISTOS` software [BFH+13], a framework for Markov chain Monte Carlo sampling for simulation, prediction, and inference of protein structures. This framework has a number of ready move modules implemented as well as force field computations and probabilistic generative models.

Figure 2.26 `Conrot-CRISP`: linear transformations applied to the prerotational and postrotational angles. Adapted from [BBEJ+12]. Matrix J is the Jacobian of the transformation mapping angles $\boldsymbol{\chi}_{pre}$ onto angles $\boldsymbol{\chi}_{post}$. Transformations are applied as follows: (i) prerotations angles are scaled by matrix C_{n-6}; (ii) the Jacobian J is applied; (iii) postrotation angles are scaled by matrix C_6; (iv) the transformation given by J^{T} is applied.



Inverse kinematics using Denavit-Hartenberg frames

The formalism of Denavit-Hartenberg frames (Sec. 2.7.2) defines the loop closure based on all internal coordinates of the molecular system. This ability has been used as follows.

`Probik`. The general loop closure Eq. 2.32 can be used in several guises [NOS05].

First, in the case where the anchors are fixed and all internal coordinates of the loop are fixed except six rotation angles θ_i, this equation yields the usual degree 16 polynomial [RR90], for which stable numerical solutions can be obtained via a generalized eigenvalue problem [MC94].

Second, the same equation can be used to explore the benefits of changing other internal coordinates. Consider perturbing a specific parameter in a loop, called the *fuzz parameter* (bond length, valence angle, ω dihedral angle) [NOS05]. Varying this parameter amounts to changing the coefficients of the degree 16 polynomial into functions of this parameter. The real roots of this parameterized polynomial evolve, coalescing and/or disappearing. Mathematically, this behavior is studied by *continuation* methods.

Practically, one first generates an array of values $T = [\dots c_i, c_{i+1}, \dots]$ centered say on its default/average value of the fuzz parameter. Consider now all possible reconstructions obtained for two consecutive values c_i and c_{i+1}. Setting aside the vanishing and the appearance of solutions, one can *link* reconstructions, since those associated to c_{i+1} correspond to those for c_i up to *deformations* [SW$^+$05]. Linking solutions makes it possible to define *branches* of solutions. Furthermore, the trajectory of a given atom along such a branch can be modeled using a polynomial curve, so as to estimate the velocity of this atom when the fuzz parameter evolves.

Higher dimensional configuration spaces. The general loop closure Eq. 2.32 involving Denavit-Hartenberg frames can also be used to explore solution spaces beyond 0 dimensional ones. As opposed to the sampling based method `Probik` just discussed, the handling is made explicit in [ZRST15].

To see how, consider consider a set of DH frames parameterized by a set $I = \{(a_i, d_i, \alpha_i, \theta_i)\}$ of n parameters. Because Eq. 2.32 imposes six constraints, the solution space is (under suitable genericity assumptions) a $n-6$ dimensional manifold $\mathcal{M}$. The tangent space to this manifold can be exploited to define locally perturb a loop defined by the parameters I_0 as follows:

- Compute the tangent space $\mathrm{T}_{I_o}\mathcal{M}$ to $\mathcal{M}$ at I_0
- Pick a random vector V in $\mathrm{T}_{I_o}\mathcal{M}$ and define the configuration $I_{TS} = I_0 + \eta V$, with η a user defined parameter.
- Project I_{TS} onto $\mathcal{M}$ to obtain a new configuration. (Nb: the projection should exist, and be unique – which means that I_{TS} should not lie on the medial axis of $\mathcal{M}$.)

In [ZRST15], this strategy is shown to be effective when the solution space is one-dimensional. For example, one may use seven dihedral angles, in which case the solution space is a curve defined in the seven dimensional flat torus $\mathbb{T}^7$.

2.7.6 Multiple loops

Loop closure and multiple loops: global solutions using distance geometry

The IK problems just discussed can actually be cast in a very general algebraic model, based on distance geometry. As summarized in [PRT$^+$07], there are several cases involving 6 torsion angles, each of them corresponding to a particular robot: the general 6-torsion molecular loop, equivalent to the general 6R serial manipulator; the tripeptide loop closure, equivalent to the 6R serial manipulator with intersecting axes; the disulfide bond loop (where the S-S bond yields a fixed torsion), equivalent to the 4-4 parallel manipulator; the 7-atom loop, equivalent to the 4-3 parallel manipulator; and the 6-atom ring, equivalent to the 3-3 (octahedral) manipulator. Importantly, the framework of distance geometry is also used to propose a method covering all such cases and beyond, also able to handle multiple loops [PRT$^+$07].

To bridge the gap between molecular models and distance geometry, one consider the constraints associated to valence and dihedral angles Indeed, constant valence angles and dihedral angles translate into distance restraints (Fig. 2.27). The loop closure problem in the rigid geometry model thus amounts to finding embeddings compatible with distance restraints, which is a NP-hard problem in general [Sax79].

To translate the problem into algebraic terms, consider n atoms $A_1, \ldots, A_n$, and the associated Caylay-Menger determinant $D(1,n)$, based upon the $\binom{n}{2}$ squared pairwise distances. With $r_{i,j} = \|A_i - A_j\|^2$, the Caylay-Menger determinant of k points is the following $(k+1) \times (k+1)$ determinant:

$$D(1,k) = \begin{vmatrix} 0 & r_{1,2} & r_{1,3} & \ldots & r_{1,k} & 1 \\ r_{2,1} & 0 & r_{2,3} & \ldots & r_{2,k} & 1 \\ \ldots & \ldots & \ldots & \ldots & \ldots & \\ r_{k,1} & r_{k,_2} & r_{k,3} & \ldots & 0 & 1 \\ 1 & 1 & 1 & 1 & 1 & 0 \end{vmatrix} \tag{2.41}$$

Is it well known [] that the Caylay-Menger determinant and the squared volume V_{k-1}^2 of the $k-1$ simplex defined by the k points $A_1, \ldots, A_k$ satisfy

$$V_{k-1}^2 = \frac{(-1)^k}{((k-1)!)^2 2^{k-1}} D(1,k). \tag{2.42}$$

For example: with V_2 the surface area of the triangle A_1, A_2, A_3, one has $-16V_2^2 = D(1,3)$; with V_3 is the volume of the tetrahedron A_1, A_2, A_3, A_4, one has $288V_3^2 = D(1,4)$.

When trying to embed a molecular model, one deals with two types of (squared) distances: those which are known, since they are associated to constraints on the internal coordinates (fixed bond length, valence angles, dihedral angle); and those which are unknown. The latter are the variables of the problem, which must be determined. The outline of the strategy to find an embedding is as follows [Sax79].

Let R be a set of 4 atoms which can be embedded. Using CM determinants $CM(R,i), CM(R,j), CM(R,i,j)$, one can write constraints which guarantee the compatibility of the distance between points in R and other pairs $(A_,A_j)$. These conditions are necessary a and sufficient. One obtains algebraic equations whose unknowns are the unknown squared distances $r_{ij} = d_{ij}^2$. Efficient solvers can be developed in particular for zero dimensional systems. The solutions to these equations directly yield an embedding for the atoms.

¡

Loop closure and multiple loops: a general approach for local solutions

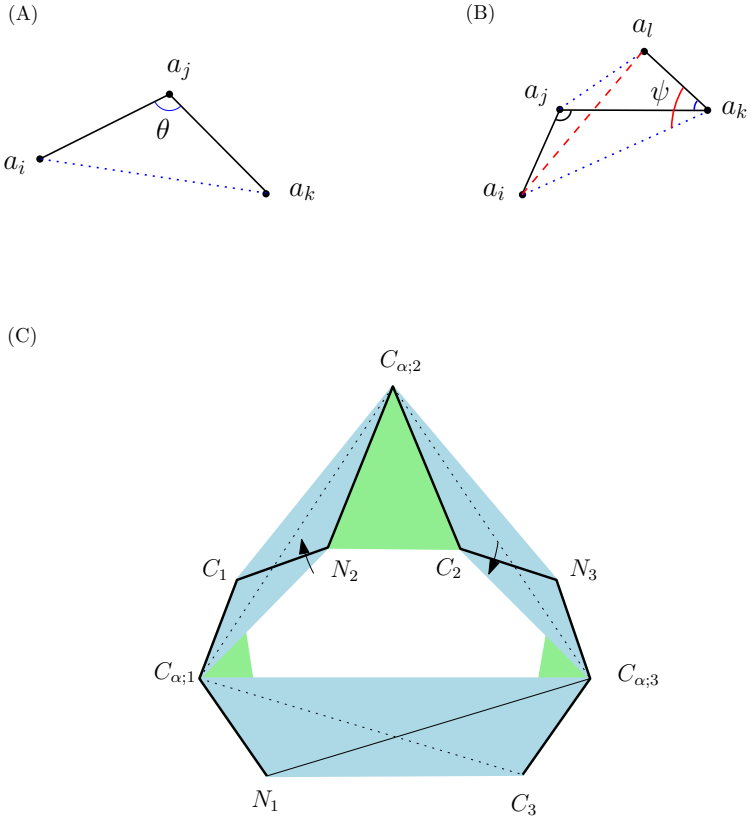
Algebraic methods based on the loop closure are of special interest particular for the solution set is zero dimensional [PRT$^+$07]. When this is not the case, it is beneficial to consider the solution space locally in some analytical form. The motivation for such cases comes from the need to model to non covalent interactions, as for example hydrogen bonds in proteins or nucleic acids [BLvdB15].

In the sequel, q refers to internal coordinates, and the embedding map returning a conformation for $q \in \mathbb{R}^n$ is denoted $f(q)$. The mathematical treatment of say p constraints can be envisioned in two ways. The first one consists of considering implicit equations for each of the p constraints, say $C_i(q) = 0, i = 1, \ldots, p$. If the implicit function theorem applies, the solution set $C_i^{-1}(0)$ is locally a $n-1$ manifold. Assuming the transverse intersection of the p solution sets, the intersection of the p constraints yields a $n-p$ manifold. For the second treatment, consider the gradient vector ∇C_i, which, under suitable conditions, is orthogonal to the solution set at point q. Form the corresponding $n \times p$ Jacobian matrix J, and denote r its rank. The complement of the span of the r vectors defines at position q the tangent space to the set respecting all the constraints, which has dimension $n-r$. When the p constraints are independent, one recovers a $n-p$ dimensional set.

This strategy is used to define deformations respecting H bonds in RNA structures [HBFdB17]. Consider the cycle associated to a H bond. Five constraints $C_i = 0$ are introduced: the first three–one for each cartesian coordinate x, y, z–stipulate that the midpoint donor-acceptor remains fixed; the remaining two stipulate that the angles defined by the donor-acceptor and the two covalent bonds flanking the H bond remain constant. Thus, m covalent bonds yield a total $p = 5m$ constraints, so that J has dimension $5m \times n$.

As an application, assume that one wishes to use these constraints to drive the interpolate between two RNA conformations, while respecting the m H bonds. Consider k atoms (or drivers) whose positions

Figure 2.27 Distance geometry models: from angular constraints to distance constraints. Adapted from [PRT+07]. **(A)** The angular constraint θ imposes the length of the opposite edge in the triangle. **(B)** Consider the dihedral angle ϕ defined by four consecutive atoms, also exhibiting valence angles constraints for a_j and a_k. The constraint on ϕ fixes the length of the edge $a_i a_l$, so that the tetrahedron is rigid. **(C)** Distance geometry model associated with the Tripeptide Loop Closure. The model involves three tetrahedra (light blue; two for the two peptide bonds, one for the rigid body involving the atoms $N_1 C_{\alpha;1} C_{\alpha;3} C_3$, and three triangles (light green) associated with the conservation of the valence angles at the C_αs.



$A_j^G (\in \mathbb{R}^3)$ are known. The algorithm is an iterative walk in the tangent space to the aforementioned solution set. At each step, a step δq is made to approach the vectors $A_j^G - f(q)_{|A_j}$ coding the discrepancy between the coordinates of the j-th anchor point and its target A_j^G. To avoid steric clashes between atoms getting too close, the authors also add temporary constraints to force the atoms the *slide* with one another.

Multiple loops: a greedy approach

To handle more general molecular systems, consider now a molecule involving several coupled cycles. Example such systems are steroids, which involved multiple cycles, and also proteins, where cycles arise do to disulfide bonds.

While the general algebraic method outlined in Sec. 2.7.6 for multiple loops can always be used, it may prove ineffective due to the complexity of the algebraic manipulations involved. In such cases, one may find solutions to the global problem, thanks to solutions to several 6R-6B sub-problems. To see how, assume that one has a toolbox with two algorithms: an algorithm solving the general 6R-6B problem; and another one tailored to the TLC (triaxial) loop closure. In any case, recall that the goal in solving such problems is to obtain an optimal degree 16 polynomial. Also recall the following terminology, from [CLW+16]: the $n-6$

free variables are called the drivers, and the 6 dependent variables the pivots.

The approach used in [CLW+16] to generate conformations of the multicycle system is a greedy one. To describe it, we distinguish the pre-processing step, and the geometry generation step.

At the pre-processing step, a continuous chain, the *backbone* is defined, so that it has at least one bond in common with each cycle. Then, a spanning tree is computed to order the cycles; in doing so, the challenge is to ensure that each cycle has at least six tortional DoF which have not been set by any previous ring closure. (This is not always possible, which may require sampling valence angles and/or bond lengths.) This data structure is then used to generate conformations by processing the cycles incrementally. Consider processing the i-th cycle. One may sample torsion angles which are not part of any of the cycles of index $1, \ldots, i-1$, and solve for the corresponding pivots. We note in passing that this requires taking into account improper dihedral angles at branching points between cycles. All the values obtained are then frozen for subsequent calculations.

A strength of this approach is to enumerate conformations in a greedy / hierarchical way. The relationship between these solutions is complex though, as different solutions may belong to different connected components of the admissible space. Also, the ordering matters, since setting dihedral angles of cycles with low rank determines the *fibers* explored above the corresponding solutions.

2.8 Conclusion

A remarkable amount of work has been carried out to model proteins and biomolecules using internal coordinates. These works encompass three main components, namely the geometric models, the statistical techniques, and the biophysical component. In terms of analysis of existing structures from the Protein Data Bank, various techniques and models have emerged to shed light on those populated regions of Ramachandran diagrams and side chain torsion angle spaces. These findings are essential for generative models, be they based on discrete libraries of rotamers, or hidden Markov models. In terms of conformational generation, a noteworthy body of work in the lineage of inverse kinematics and related techniques has been proposed. Coupled with machine learned properties of individual amino acids and/or tripeptides, these algorithms now make it possible to sample thoroughly backbone conformations.

Despite these achievements, we foresee important developments in two directions. The first one relates to a tight integration of models developed independently for the backbone and side-chains, as conformations generated for the former must currently be post-processed to remove nonphysical conformations featuring steric clashes in particular. The second one pertains to thermodynamics and dynamics. The integration of recent generative models with remarkable geometric properties into sampling techniques based on importance sampling might indeed make it possible to obtain reliable approximations for various thermodynamic/kinetic quantities. A first step in this direction would be to obtain guaranteed approximations of partition functions on a per (significant) basin basis, as this kind of information would make it possible to go beyond harmonic models when treating thermodynamics using Markov models.

Without a doubt, these developments will help unveil subtle properties of the dynamics of biomolecules, complementing the also astonishing body of work in protein science based on deep learning and related techniques.

Chapter 3

Geometric constraints within tripeptides

3.1 Introduction

We consider a tripeptide in which all internal coordinates (but the 6 dihedral angles) hold canonical values. Assuming that the two segments $N_1C_{\alpha;1}$ and $C_{\alpha;3}C_3$ are free to move with respect to one another, we aim at finding necessary conditions on these two segments for TLC to admit solutions. A tripeptide yielding solutions is termed *embeddable*. As we can assume without loss of generality that the first segment is fixed in a reference frame, this problem is posed in a five dimensional configuration space: the position of $C_{\alpha;3}$ enjoys 3 Cartesian coordinates, and that of C_3 two spherical coordinates w.r.t. $C_{\alpha;3}$ (Fig. 3.1). The question becomes to find out which positions of $C_{\alpha;3}$ and C_3 yielding embeddable tripeptides. To answer this question, our contributions are organized as follows:

- In Section 3.2, we present background material from [CSJD04].
- In Section 3.3, we derive C_α valence constraints at each $C_{\alpha;i}$ carbon to guarantee that the valence angle θ_i at this C_α is preserved. These constraints involve two angles denoted σ_{i-1} and τ_i.
- In Sec. 3.4, we exploit a constraint associated with each $C_{\alpha;i}C_{\alpha;i+1}$ edge, to derive an Inter-angular constraint on all σ_{i-1} and τ_i angles. This constraint is thus a necessary condition for the whole tripeptide.
- Section 6.4 provides illustrations of our constructions, showing the sharpness of our constraints in the aforementioned five dimensional space.

These contributions hinge on several interval types for the various angles involved in TLC (Fig. 3.2): IVI for Initial Validity Intervals, RVI for Rotated Validity Intervals, DVI for Depth-n/Deep Validity Intervals, RDVI for Restricted Deep Validity Intervals. In a sense, this work aims to understand the geometry of solutions of TLC in terms of necessary conditions on the six dihedral angles involved, expressed using these interval types.

3.2 Background on the Tripeptide Loop Closure

In this section, we review in detail the solution to TLC from [CSJD04].

3.2.1 Rotations and constraints

We consider the tripeptide $(N_1C_{\alpha;1}C_1)(N_2C_{\alpha;2}C_2)(N_3C_{\alpha;3}C_3)$.

Geometry of the C_α triangle of the tripeptide. Consider first the following four consecutive atoms $C_{\alpha;1}C_1N_2C_{\alpha;2}$ along the backbone, with C_1N_2 the peptide bond. Since the ω_1 angle of the peptide bond is fixed, the distance $\|C_{\alpha;1}C_{\alpha;2}\|$ is constant (Fig. 3.3(A)). This observation holds for the other edge $C_{\alpha;2}C_{\alpha;3}$. Consequently, the atom $C_{\alpha;2}$ is restrained to the circle defined by the intersection of two spheres: $S_1(C_{\alpha;1}, \|C_{\alpha;1}C_{\alpha;2}\|)$ and $S_2(C_{\alpha;3}, \|C_{\alpha;2}C_{\alpha;3}\|)$.

Finally, consider the base of this triangle. By hypothesis, atoms $N_1, C_{\alpha;1}, C_{\alpha;3}, C_3$ are fixed, so that the length of the base is fixed.

The geometry of the triangle $C_{\alpha;1}C_{\alpha;2}C_{\alpha;3}$ is therefore fixed. However, one has one rotating rigid body attached to each of its edges (Fig. 3.3(B)):

- The movement of the atom C_1 (resp. N_2) can be modeled by a rotation of angle τ_1 (resp. σ_1) about the axis $C_{\alpha;1}C_{\alpha;2}$;
- The movement of the atom C_2 (resp. N_3) can be modeled by a rotation of angle τ_2 (resp. σ_2) about the axis $C_{\alpha;2}C_{\alpha;3}$;
- The movement of atoms $N_1C_{\alpha;1}C_{\alpha;3}C_3$ can be modeled by a rotation of angle τ_3 about the axis $C_{\alpha;1}C_{\alpha;3}$.

Remark 3.1. *Using the dihedral angle δ_i defined by the four atoms $C_{\alpha;i}, C_i, N_{i+1}, C_{\alpha;i+1}$ (Fig. 3.3(B)) one has the relationship:*

$$\sigma_i = \tau_i + \delta_i. \tag{3.1}$$

Local C_α frames. The TLC from [CSJD04] defines one local frame to handle the rotation angles τ_i and σ_i. Using these frames and Eq. 3.1, TLC reduces the problem to three variables and three constraints:

- The three variables: the angles $\tau_i \forall i \in \{1, 2, 3\}$ which can be rotated.
- The three constraints: the valence angles $\theta_i \forall i \in \{1, 2, 3\}$ each at the corresponding $C_{\alpha;i}$, which which must be kept constant.

It is the coupling introduced by the θ_i angles onto the rotation angles τ_i that yields a degree 16 polynomial [CSJD04, NOS05].

We have recalled above that in TLC, the atoms $N_1, C_{\alpha;1}, C_{\alpha;3}, C_3$ are fixed. But in using the rotation angles $\{(\tau_i, \sigma_i)\}$, the segments $N_1C_{\alpha;1}$ and $C_{\alpha;3}C_3$ are moving. Therefore, once the atomic positions have been obtained using the local frames, all atoms are rotated such that these four are back into their original positions in the main frame.

Remark 3.2. *The valence angles around C_i and N_i atoms do not play a role, since they are fixed and internal to the two segments being rotated.*

3.2.2 Local coordinate system at $C_{\alpha;i}$

Three **unit** vectors $\hat{\mathbf{Z}}_\mathbf{i}, \hat{\mathbf{Y}}_\mathbf{i}, \hat{\mathbf{X}}_\mathbf{i}$ are defined to form an orthonormal coordinate system (Fig. 3.3(C)). These vectors are:

- (Unit vector aligned with one side of the triangle) $\hat{\mathbf{Z}}_\mathbf{i}$ is the unit vector anchored at $C_{\alpha;i}$ aligned with the edge of the C_α triangle. Note that $C_{\alpha;3}$ points to $C_{\alpha;1}$.
- (Unit vector perpendicular to the plane of the triangle) $\hat{\mathbf{Y}}_\mathbf{i} = \hat{\mathbf{Z}}_{\mathbf{i+2}} \times \hat{\mathbf{Z}}_\mathbf{i} / \left\| \hat{\mathbf{Z}}_{\mathbf{i+2}} \times \hat{\mathbf{Z}}_\mathbf{i} \right\|$. Intuitively, note that $\hat{\mathbf{Y}}_\mathbf{i}$ is obtained by taking the cross product between two of the unit vectors just mentioned: that *entering* the C_α carbon and that *exiting* it. (Nb: the vector $\hat{\mathbf{Y}}_\mathbf{i}$ does not depend on the index i, which can be dropped.)

- (Reference unit vector to define the rotation angles) $\hat{\mathbf{X}}_\mathbf{i} = \hat{\mathbf{Y}}_\mathbf{i} \times \hat{\mathbf{Z}}_\mathbf{i} = (\hat{\mathbf{Z}}_\mathbf{i} \cdot \hat{\mathbf{Z}}_{\mathbf{i+2}})\hat{\mathbf{Z}}_\mathbf{i} - \hat{\mathbf{Z}}_{\mathbf{i+2}}$. (For the latter, we use the double cross product formula $u \times (v \times w) = (u.w)v - (u.v)w$.)

Using the local frames, one also defines the angles α_i, η_i and ξ_i as follows (Fig. 3.3(D)):

$$\begin{cases} \alpha_i & = \angle \hat{\mathbf{Z}}_\mathbf{i} \hat{\mathbf{Z}}_{\mathbf{i-1}} \\ \xi_i & = \angle - \hat{\mathbf{Z}}_\mathbf{i} \hat{\mathbf{r}}^\sigma_\mathbf{i} \\ \eta_i & = \angle \hat{\mathbf{Z}}_\mathbf{i} \hat{\mathbf{r}}^\tau_\mathbf{i} \end{cases} \tag{3.2}$$

These variables will be used to handle the conservation of the valence angle at $C_{\alpha;i}$, which requires considering three atoms: N_{i-1}, $C_{\alpha;i}$, and C_i.

Remark 3.3. *Indices $i \in \{1, 2, 3\}$ are counted modulo 3; that is, $i-1$ is equivalent to $i+2$: e.g., $\hat{\mathbf{Z}}_{\mathbf{i+2}} = \hat{\mathbf{Z}}_{\mathbf{i-1}}$.*

Remark 3.4. *The following equalities are used throughout all calculations* [1]*:*

$$\begin{cases} \langle \hat{\mathbf{Z}}_\mathbf{i}, \hat{\mathbf{Z}}_{\mathbf{i-1}} \rangle & = \cos \alpha_i \\ \langle \hat{\mathbf{X}}_{\mathbf{i-1}}, \hat{\mathbf{X}}_\mathbf{i} \rangle & = \langle \hat{\mathbf{Y}} \times \hat{\mathbf{Z}}_{\mathbf{i-1}}, \hat{\mathbf{Y}} \times \hat{\mathbf{Z}}_\mathbf{i} \rangle = \cos \alpha_\mathbf{i} \\ \langle \hat{\mathbf{Z}}_\mathbf{i}, \hat{\mathbf{Y}} \rangle & = 0 \\ \langle \hat{\mathbf{Z}}_\mathbf{i}, \hat{\mathbf{X}}_{\mathbf{i-1}} \rangle & = \langle \hat{\mathbf{Z}}_\mathbf{i}, \hat{\mathbf{Y}} \times \hat{\mathbf{Z}}_{\mathbf{i-1}} \rangle - \langle \hat{\mathbf{Z}}_\mathbf{i}, \times \hat{\mathbf{Y}} \rangle = -\langle \hat{\mathbf{Z}}_\mathbf{i} \times \hat{\mathbf{Z}}_{\mathbf{i-1}}, \hat{\mathbf{Y}} \rangle = -\langle -\sin \alpha_\mathbf{i} \hat{\mathbf{Y}}, \hat{\mathbf{Y}} \rangle = \sin \alpha_\mathbf{i} \\ \langle \hat{\mathbf{Z}}_{\mathbf{i-1}}, \hat{\mathbf{X}}_\mathbf{i} \rangle & = \langle \hat{\mathbf{Z}}_{\mathbf{i-1}}, \hat{\mathbf{Y}} \times \hat{\mathbf{Z}}_\mathbf{i} \rangle = -\langle \hat{\mathbf{Z}}_{\mathbf{i-1}}, \hat{\mathbf{Z}}_\mathbf{i} \times \hat{\mathbf{Y}} \rangle = -\langle \hat{\mathbf{Z}}_{\mathbf{i-1}} \times \hat{\mathbf{Z}}_\mathbf{i}, \hat{\mathbf{Y}} \rangle = -\sin \alpha_\mathbf{i}. \end{cases} \tag{3.3}$$

3.2.3 Rotations of N_i and C_i

Consider $\hat{\mathbf{r}}^\sigma_{\mathbf{i-1}}$ the $C_{\alpha;i}N_i$ unit vector and $\hat{\mathbf{Z}}_\mathbf{i}$ the $C_{\alpha;i}C_i$ unit vector. The rotations of C_i and N_i are described as follows (Fig. 3.3(C,D)):

- atom N_i, angle σ_{i-1}: rotation of $\hat{\mathbf{r}}^\sigma_{\mathbf{i-1}}$ about $\hat{\mathbf{Z}}_{\mathbf{i-1}}$
- atom C_i, angle τ_i: rotation of $\hat{\mathbf{r}}^\tau_\mathbf{i}$ about $\hat{\mathbf{Z}}_\mathbf{i}$

The vectors $\hat{\mathbf{r}}^\sigma_{\mathbf{i-1}}$ and $\hat{\mathbf{r}}^\tau_\mathbf{i}$ are easily obtained using the local frames (Fig. 3.3(E)):

$$\text{Frame}(\hat{\mathbf{X}}_{\mathbf{i-1}}, \hat{\mathbf{Y}}, \hat{\mathbf{Z}}_{\mathbf{i-1}}) : \qquad \hat{\mathbf{r}}^\sigma_{\mathbf{i-1}} = -\cos \xi_{i-1} \hat{\mathbf{Z}}_{\mathbf{i-1}} + \sin \xi_{\mathbf{i-1}} (\cos \sigma_{\mathbf{i-1}} \hat{\mathbf{X}}_{\mathbf{i-1}} + \sin \sigma_{\mathbf{i-1}} \hat{\mathbf{Y}}) \tag{3.4}$$

$$\text{Frame}(\hat{\mathbf{X}}_\mathbf{i}, \hat{\mathbf{Y}}, \hat{\mathbf{Z}}_\mathbf{i}) : \qquad \hat{\mathbf{r}}^\tau_\mathbf{i} = \cos \eta_i \hat{\mathbf{Z}}_\mathbf{i} + \sin \eta_\mathbf{i} (\cos \tau_\mathbf{i} \hat{\mathbf{X}}_\mathbf{i} + \sin \tau_\mathbf{i} \hat{\mathbf{Y}}) \tag{3.5}$$

Using the previous two equations and the equalities from Eq. (3.3), one obtains the dot product between $\hat{\mathbf{r}}^\sigma_{\mathbf{i-1}}$ and $\hat{\mathbf{r}}^\tau_\mathbf{i}$:

$$\begin{aligned} \langle \hat{\mathbf{r}}^\sigma_{\mathbf{i-1}}, \hat{\mathbf{r}}^\tau_\mathbf{i} \rangle = & -\cos \xi_{i-1} \cos \eta_i \cos \alpha_i \\ & + \cos \xi_{i-1} \sin \eta_i \cos \tau_i \sin \alpha_i \\ & + \cos \eta_i \sin \xi_{i-1} \cos \sigma_{i-1} \sin \alpha_i \\ & + \sin \xi_{i-1} \sin \eta_i (\cos \sigma_{i-1} \cos \tau_i \cos \alpha_i + \sin \sigma_{i-1} \sin \tau_i) \end{aligned} \tag{3.6}$$

The conservation of the valence angle θ_i imposes the following *valence angle constraint*:

$$\langle \hat{\mathbf{r}}^\sigma_{\mathbf{i-1}}, \hat{\mathbf{r}}^\tau_\mathbf{i} \rangle = \cos \theta_i. \tag{3.7}$$

In TLC only the τ_i and σ_{i-1} terms vary, the value of this dot product can thus be represented depending on those two angles.

[1] Recall the triple product formula $\langle a, b \times c \rangle = \langle (a \times b), c \rangle$.

3.3 C_α valence angle constraints

In this section, we study constraints on σ_{i-1} and τ_i so as to guarantee that the valence angle θ_i is preserved. We first present derivation of intervals for σ_{i-1} and τ_i (Sec. 3.3.1), and proceed with the no-solution case, introducing a necessary condition for solutions to exist (Sec. 3.3.2).

3.3.1 Initial validity intervals for σ_{i-1} and τ_i

We first derive initial validity intervals, using boundary conditions at each C_α carbon.

Angle σ_{i-1}

We wish to define a *validity interval* for σ_{i-1}, namely $I_{\sigma_{i-1}} = [\sigma_{i-1;-}, \sigma_{i-1;+}] \subseteq [0, \pi]$.

Basic equation. We first wish to restrict the values of σ_{i-1} for which θ_i can be preserved, and use to this end the reference vector $\hat{\mathbf{Z}}_\mathbf{i}$. Using the expression of $\hat{\mathbf{r}}^\sigma_{\mathbf{i-1}}$ in the local frame of $C_{\alpha;i-1}$ (Eq. (3.4)), we get:

$$\langle \hat{\mathbf{r}}^\sigma_{\mathbf{i-1}}, \hat{\mathbf{Z}}_\mathbf{i} \rangle = \cos\sigma_{\mathbf{i-1}} \sin\xi_{\mathbf{i-1}} \sin\alpha_\mathbf{i} - \cos\xi_{\mathbf{i-1}} \cos\alpha_\mathbf{i}. \tag{3.8}$$

Because the vector $\hat{\mathbf{r}}^\tau_\mathbf{i}$ which makes an angle η_i with $\hat{\mathbf{Z}}_\mathbf{i}$ can only add or subtract the value η_i to the constraint θ_i, one must have:

$$\langle \hat{\mathbf{r}}^\sigma_{\mathbf{i-1}}, \hat{\mathbf{Z}}_\mathbf{i} \rangle = \cos(\theta_\mathbf{i} \pm \eta_\mathbf{i}) \tag{3.9}$$

This equation yields up to two solutions, namely $\sigma_{i-1;-}$ and $\sigma_{i-1;+}$.

Angle $\sigma_{i-1;-}$. The first limit case reads as:

$$\langle \hat{\mathbf{r}}^\sigma_{\mathbf{i-1};-}, \hat{\mathbf{Z}}_\mathbf{i} \rangle = \cos(\theta_\mathbf{i} - \eta_\mathbf{i}) \tag{3.10}$$

from which we obtain

$$\begin{cases} S^- = \frac{+\cos(\theta_i - \eta_i) + \cos\xi_{i-1}\cos\alpha_i}{\sin\xi_{i-1}\sin\alpha_i} \\ \sigma_{i-1;-} = \arccos S^- \end{cases} \tag{3.11}$$

When $S^- \to 1^-$ by properties of arccos, we have $\sigma_{i-1;-} \to 0^+$ (Fig. S3.9). Therefore, when

$$S^- \geq 1, \tag{3.12}$$

we set $\sigma_{i-1;-} = 0$, so that any value $\sigma_{i-1} \leq \sigma_{i-1;+}$ is valid.

Angle $\sigma_{i-1;+}$. The second limit case reads as:

$$\langle \hat{\mathbf{r}}^\sigma_{\mathbf{i-1};+}, \hat{\mathbf{Z}}_\mathbf{i} \rangle = \cos(\theta_\mathbf{i} + \eta_\mathbf{i}) \tag{3.13}$$

from which we obtain

$$\begin{cases} S^+ = \frac{+\cos(\theta_i + \eta_i) + \cos\xi_{i-1}\cos\alpha_i}{\sin\xi_{i-1}\sin\alpha_i} \\ \sigma_{i-1;+} = \arccos S^+ \end{cases} \tag{3.14}$$

When $S^+ \to -1^+$, by properties of arccos, we have $\sigma_{i-1;+} \to \pi^-$. Therefore, when

$$S^+ \leq -1, \tag{3.15}$$

we set $\sigma_{i-1;+} = \pi$, so that any value $\sigma_{i-1} \geq \sigma_{i-1;-}$ is valid.

Illustration. When considering the dot product $\langle \hat{\mathbf{r}}^\sigma_{\mathbf{i-1}}, \hat{\mathbf{r}}^\tau_\mathbf{i} \rangle$ as a function of the two variables τ_i and σ_{i-1}, the angles $\sigma_{i-1;-}$ and $\sigma_{i-1;+}$ correspond to planes orthogonal to the σ_{i-1} axis (Fig. 3.4(B,C)).

Angle τ_i

We also wish to set a *validity interval* for τ_i, that is $I_{\tau_i} = [\tau_{i;-}, \tau_{i;+}] \subseteq [0, \pi]$.

Basic equation. We proceed mutatis mutandis for the vector $\hat{\mathbf{r}}_{\mathbf{i}}^{\tau}$, using vector $\hat{\mathbf{Z}}_{\mathbf{i-1}}$ as landmark. Using the expression of $\hat{\mathbf{r}}_{\mathbf{i}}^{\tau}$ in the local frame of $C_{\alpha;i}$ (Eq. (3.5)), one obtains:

$$\langle \hat{\mathbf{r}}_{\mathbf{i}}^{\tau}, \hat{\mathbf{Z}}_{\mathbf{i-1}} \rangle = -\cos\tau_{\mathbf{i}} \sin\eta_{\mathbf{i}} \sin\alpha_{\mathbf{i}} + \cos\eta_{\mathbf{i}} \cos\alpha_{\mathbf{i}} \tag{3.16}$$

The vector $\hat{\mathbf{r}}_{\mathbf{i-1}}^{\sigma}$ can only add or subtract ξ_{i-1}, whence the following:

$$\langle \hat{\mathbf{r}}_{\mathbf{i}}^{\tau}, \hat{\mathbf{Z}}_{\mathbf{i-1}} \rangle = -\cos(\theta_{\mathbf{i}} \pm \xi_{\mathbf{i-1}}). \tag{3.17}$$

This equation yields up to two solutions, namely $\tau_{i;-}$ and $\tau_{i;+}$.

Angle $\tau_{i;-}$. The first limit case reads as:

$$\langle \hat{\mathbf{r}}_{\mathbf{i};-}^{\tau}, \hat{\mathbf{Z}}_{\mathbf{i-1}} \rangle = -\cos(\theta_{\mathbf{i}} - \xi_{\mathbf{i-1}}) \tag{3.18}$$

from which we obtain

$$\begin{cases} T^- = \frac{+\cos(\theta_i - \xi_{i-1}) + \cos\eta_i \cos\alpha_i}{\sin\eta_i \sin\alpha_i} \\ \tau_{i;-} = \arccos T^- \end{cases} \tag{3.19}$$

When $T^- \to 1^-$, by properties of arccos, we have $\tau_{i;-} \to 0^+$. Therefore, when

$$T^- \geq 1, \tag{3.20}$$

we set $\tau_{i;-} = 0$, so that any value $\tau_i \leq \tau_{i;+}$ is valid.

Angle $\tau_{i;+}$. The second limit case reads as:

$$\langle \hat{\mathbf{r}}_{\mathbf{i};-}^{\tau}, \hat{\mathbf{Z}}_{\mathbf{i-1}} \rangle = -\cos(\theta_{\mathbf{i}} + \xi_{\mathbf{i-1}}) \tag{3.21}$$

from which we obtain

$$\begin{cases} T^+ = \frac{+\cos(\theta_i + \xi_{i-1}) + \cos\eta_i \cos\alpha_i}{\sin\eta_i \sin\alpha_i} \\ \tau_{i;+} = \arccos T^+ \end{cases} \tag{3.22}$$

When $T^+ \to 1^+$, by properties of arccos, we have $\tau_{i;-} \to \pi^-$. Therefore, when

$$T^+ \leq -1, \tag{3.23}$$

we set $\tau_{i;+} = \pi$, so that any value of $\tau_i \geq \tau_{i;-}$ is valid.

Illustration. When considering the dot product $\langle \hat{\mathbf{r}}_{\mathbf{i-1}}^{\sigma}, \hat{\mathbf{r}}_{\mathbf{i}}^{\tau} \rangle$ as a function of the two variables τ_i and σ_{i-1}, the angles $\tau_{i-1;-}$ and $\tau_{i-1;+}$ correspond to planes orthogonal to the τ_i axis (Fig. 3.4,(B),(C)).

3.3.2 Necessary conditions for σ_{i-1} and τ_i

In deriving the lower and upper bounds of the initial validity intervals for σ and τ, we already processed four limit cases for the dot products (Eqs. (3.12), (3.15), (3.20), (3.23)). The remaining four yield the following:

Definition. 3.1. *(C_α valence constraints) The C_α valence constraints are the necessary validity conditions defined by :*

- *Angle $\sigma_{i-1;-}$: the condition $\sigma_{i-1;-} < \sigma_{i-1;+}$ requires*

$$S^- \geq -1. \tag{3.24}$$

- *Angle $\sigma_{i-1;+}$: the condition $\sigma_{i-1;-} < \sigma_{i-1;+}$ requires*

$$S^{+} \leq 1. \tag{3.25}$$

- *Angle $\tau_{i;-}$: the condition $\tau_{i;-} < \tau_{i;+}$ requires*

$$T^{-} \geq -1. \tag{3.26}$$

- *Angle $\tau_{i;+}$: the condition $\tau_{i;-} < \tau_{i;+}$ requires*

$$T^{+} \leq 1. \tag{3.27}$$

For the constraint to be verified all these conditions must be valid for all three $\{(\sigma_{i-1}, \tau_i)\}$ pairs.

Summarizing, when the previous equations are not verified, no validity interval can be defined for σ_{i-1} and/or τ_i (Fig. 3.5(E)).

3.3.3 Symmetry around the C_α triangular plane, and C_α valence constraints

The previous angles are defined in $[0, \pi]$. Due to the symmetry of the tripeptide with respect to the C_α plane, these angles have counterparts in $[\pi, 2\pi]$. We therefore define the following symmetric intervals:

- The *symmetric validity interval* for σ_{i-1} is defined by

$$I^{'}_{\sigma_{i-1}} = [\sigma^{'}_{i-1;-}, \sigma^{'}_{i-1;+}] \stackrel{Def}{=} [2\pi - \sigma_{i-1;+}, 2\pi - \sigma_{i-1;-}]. \tag{3.28}$$

- The *symmetric validity interval* for τ_i is defined by

$$I^{'}_{\tau_i} = [\tau^{'}_{i;-}, \tau^{'}_{i;+}] \stackrel{Def}{=} [2\pi - \tau_{i;+}, 2\pi - \tau_{i;-}]. \tag{3.29}$$

Using these, we can finally specify the valid intervals for the angles σ_{i-1} and τ_i must belong to:

Definition. 3.2. *(Initial validity intervals) The initial validity intervals for σ_{i-1} are defined by:*

$$\mathcal{I}_{\sigma_{i-1}} = I_{\sigma_{i-1}} \cup I^{'}_{\sigma_{i-1}} \tag{3.30}$$

Likewise, the initial validity interval for τ_i are defined by:

$$\mathcal{I}_{\tau_i} = I_{\tau_{i-1}} \cup I^{'}_{\tau_i}. \tag{3.31}$$

For σ_{i-1}, as long as the conditions of Eqs. (3.24) and (3.25) are satisfied, we have:

$$\sigma_{i-1} \in \mathcal{I}_{\sigma_{i-1}}. \tag{3.32}$$

For τ_i, as long as the conditions of Eqs. (3.26) and (3.27) are satisfied, we have:

$$\tau_i \in \mathcal{I}_{\tau_i}. \tag{3.33}$$

Combining the previous conditions yields the complete case analysis (Fig. 3.5).

3.4 Inter-angular constraints associated with the C_α triangle

3.4.1 Exploiting the coherence along a $C_{\alpha;i}C_{\alpha;i+1}$ edge

The constraints presented in the previous section focus on the three C_α carbons independently. On the other hand, for a given tripeptide, the angles τ_i and σ_i and the dihedral angle δ_i defined by the four atoms $C_{\alpha;i}C_iN_{i+1}C_{\alpha;i+1}$ satisfy $\sigma_i = \tau_i + \delta_i$ (Eq. (3.1)).

Given an interval of values for τ_i, the previous formula can be used to infer a projected interval for σ_i, and vice-versa (Fig. 3.2(B)). Whence the following definitions which exploit the previous formula along the two edges of the C_α triangle incident on $C_{\alpha;i}$:

Definition. 3.3. *(Rotated validity intervals) The rotated validity intervals for the angles σ_{i-1} and τ_i are defined by:*

- *for σ_{i-1}: $\mathcal{I}_{\sigma_{i-1}|\delta} = I_{\sigma_{i-1}|\delta} \cup I'_{\sigma_{i-1}|\delta}$ with:*
  - *$I_{\sigma_{i-1}|\delta}$: interval for σ_{i-1} obtained by applying Eq. (3.1) to $I_{\tau_{i-1}}$. (Nb: uses the edge $C_{\alpha;i}C_{\alpha;i-1}$ of the C_α triangle.)*
  - *$I'_{\sigma_{i-1}|\delta}$: interval for σ_{i-1} obtained by applying Eq. (3.1) to $I'_{\tau_{i-1}}$. (Nb: uses the edge $C_{\alpha;i}C_{\alpha;i-1}$ of the C_α triangle.)*
- *for τ_i: $\mathcal{I}_{\tau_i|\delta} = I_{\tau_i|\delta} \cup I'_{\tau_i|\delta}$ with:*
  - *$I_{\tau_i|\delta}$: interval for τ_i obtained by applying Eq. (3.1) to I_{σ_i}. (Nb: uses the edge $C_{\alpha;i}C_{\alpha;i+1}$ of the C_α triangle.)*
  - *$I'_{\tau_i|\delta}$: interval for τ_i obtained by applying Eq. (3.1) to I'_{σ_i}. (Nb: uses the edge $C_{\alpha;i}C_{\alpha;i+1}$ of the C_α triangle.)*

Summarizing, for each of the σ_{i-1} and τ_i angles, we have obtained 4+4 intervals using the θ_i angle constraint at $C_{\alpha;i}$ (Eq. 3.6):

- Four for the σ_{i-1} angle: $I_{\sigma_{i-1}}, I'_{\sigma_{i-1}}, I_{\sigma_{i-1}|\delta}, I'_{\sigma_{i-1}|\delta}$
- Four for the τ_i angle: $I_{\tau_i}, I'_{\tau_i}, I_{\tau_i|\delta}, I'_{\tau_i|\delta}$

3.4.2 Deep Validity Intervals: depth 1

We are now in position to combine two pieces of information:

- The conditions on σ_{i-1} and τ_i inherent to the conservation of the valence angles (Eq. (3.7)).
- The conditions exploiting rotated validity intervals, stemming from Eq. (3.1)

Since the previous intervals define necessary conditions, intersections between intervals for a given angle must be non empty. We therefore combine them as follows $(I_{\cdot_{i-1}}, I'_{\cdot_{i-1}}) \times (I_{\cdot_{i-1}|\delta}, I'_{\cdot_{i-1}|\delta})$, which yields *depth 1 validity intervals*:

Definition. 3.4. *(Depth 1 validity intervals) The depth 1 inter-angular interval set $\mathcal{J}^{(1)}_{\sigma_{i-1}}$ for σ_{i-1} is:*

$$\mathcal{J}^{(1)}_{\sigma_{i-1}} = (I_{\sigma_{i-1}} \cap I_{\sigma_{i-1}|\delta}) \cup (I_{\sigma_{i-1}} \cap I'_{\sigma_{i-1}|\delta}) \cup (I'_{\sigma_{i-1}} \cap I_{\sigma_{i-1}|\delta}) \cup (I'_{\sigma_{i-1}} \cap I'_{\sigma_{i-1}|\delta}) \tag{3.34}$$

Similarly, the depth 1 inter-angular interval set $\mathcal{J}^{(1)}_{\tau_i}$ for τ_i:

$$\mathcal{J}^{(1)}_{\tau_i} = (I_{\tau_i} \cap I_{\tau_i|\delta}) \cup (I_{\tau_i} \cap I'_{\tau_i|\delta}) \cup (I'_{\tau_i} \cap I_{\tau_i|\delta}) \cup (I'_{\tau_i} \cap I'_{\tau_i|\delta}) \tag{3.35}$$

Note that each depth 1 validity interval has up to four connected components.

Definition. 3.5. *(Depth 1 inter-angular constraint) The depth 1 inter-angular constraint for σ_{i-1} is:*

$$\mathcal{J}^{(1)}_{\sigma_{i-1}} \neq \{\emptyset\} \tag{3.36}$$

The depth 1 inter-angular constraint for τ_i is:

$$\mathcal{J}^{(1)}_{\tau_i} \neq \{\emptyset\} \tag{3.37}$$

For the constraint to be verified all these conditions must be valid for all three $\{(\tau_i, \sigma_{i-1})\}$ pairs.

As we shall see, this constraint is significantly more restrictive than the corresponding C_α valence constraint (Def. 3.1, Fig. 3.7)

The derivation of these inter-angular constraints can be seen as a bootstrap process (Fig. 3.2). The initialization consists of computing the initial validity intervals, using the boundary conditions imposed by Equations (3.11) (3.14) (3.19) (3.22). The second step consists of exploiting the coherence along each C_α edges, as imposed by Eq. (3.1).

It should also be noticed that transposing the initial validity intervals from τ_i to σ_i using Eq. 3.1 is equivalent to transposing from σ_i to τ_i using the same equation. Therefore the *depth 1 inter-angular constraint* for σ_i and the one for τ_i are redundant.

Remark 3.5. *The previous intersections naturally depend on the two anchor positions and the fixed internal coordinates.*

3.4.3 Deep Validity Intervals: arbitrary depth

The qualifier *depth 1* used in the previous section indicates that the dual process *specify validity intervals at each C_α* and *project along a C_α edge* can be repeated, moving from necessary conditions at depth j to necessary conditions at depth depth $j+1$.

To see how, we first note that an interval for σ_{i-1} or τ_i implies two intervals for the second angle – obtained by computing σ_{i-1} from τ_i, or vice versa. To see how, note that the dot product equation (3.6) can be written as:

$$K^{(1)}_{\sigma_{i-1}} \cos\sigma_{i-1} + K^{(2)}_{\sigma_{i-1}} \sin\sigma_{i-1} + K^{(3)}_{\sigma_{i-1}} = 0, \tag{3.38}$$

with

$$\begin{cases} K^{(1)}_{\sigma_{i-1}} & = \cos\tau_i \sin\xi_{i-1} \sin\eta_i \cos\alpha_i + \sin\alpha_i \cos\eta_i \sin\xi_{i-1} \\ K^{(2)}_{\sigma_{i-1}} & = \sin\xi_{i-1} \sin\eta_i \sin\tau_i \\ K^{(3)}_{\sigma_{i-1}} & = -\cos\xi_{i-1} \cos\eta_i \cos\alpha_i + \cos\xi_{i-1} \sin\eta_i \cos\tau_i \sin\alpha_i - \cos\theta_i \end{cases} \tag{3.39}$$

This latter equation makes it possible to obtain σ_{i-1} given τ_i. Dividing by $\sqrt{{K^{(1)}_{\sigma_{i-1}}}^2 + {K^{(2)}_{\sigma_{i-1}}}^2}$ and using the trigonometric identity $\cos(a-b) = \cos a \cos b + \sin a + \sin b$, the two values for σ_{i-1} $(\sigma^*_{i-1}, \sigma^{**}_{i-1})$ given τ_i are obtained as follows:

$$\begin{cases} \cos\left(\sigma^*_{i-1}(\tau_i)\right) & = \cos\left(\arccos\frac{K^{(1)}_{\sigma_{i-1}}}{\sqrt{{K^{(1)}_{\sigma_{i-1}}}^2+{K^{(2)}_{\sigma_{i-1}}}^2}} + \arccos\frac{-K^{(3)}_{\sigma_{i-1}}}{\sqrt{{K^{(1)}_{\sigma_{i-1}}}^2+{K^{(2)}_{\sigma_{i-1}}}^2}}\right) \\ \cos\left(\sigma^{**}_{i-1}(\tau_i)\right) & = \cos\left(2\pi + \arccos\frac{K^{(1)}_{\sigma_{i-1}}}{\sqrt{{K^{(1)}_{\sigma_{i-1}}}^2+{K^{(2)}_{\sigma_{i-1}}}^2}} - \arccos\frac{-K^{(3)}_{\sigma_{i-1}}}{\sqrt{{K^{(1)}_{\sigma_{i-1}}}^2+{K^{(2)}_{\sigma_{i-1}}}^2}}\right) \end{cases} \tag{3.40}$$

Remark 3.6. *Eq. (3.40) defines the cosine of $\sigma^*_{i-1}(\tau_i)$ and $\sigma^{**}_{i-1}(\tau_i)$. For each of them, two values are possible namely $arccos(\sigma_{i-1})$ and $arccos(2\pi - \sigma_{i-1})$. Each time only one of those values validates the valence angle constraint of Eq. (3.7).*

With these, we define:

Definition. 3.6. *(Restricted validity intervals) For each interval* $I \stackrel{Def}{=} [{\tau_i}^{\min}, {\tau_i}^{\max}] \in \mathcal{J}_{\tau_i}^{(j)}$*, consider the two intervals*

$$\begin{cases} I_\sigma^*(I) = [\min_{\tau_i \in I}(\sigma_{i-1}^*(\tau_i)), \max_{\tau_i \in I}(\sigma_{i-1}^*(\tau_i))], \\ I_\sigma^{**}(I) = [\min_{\tau_i \in I}(\sigma_{i-1}^{**}(\tau_i)), \max_{\tau_i \in I}(\sigma_{i-1}^{**}(\tau_i))]. \end{cases} \tag{3.41}$$

The restricted validity interval set is defined by:

$$\mathcal{K}_{\sigma_{i-1}}^{(j)} = \bigcup_{I \in \mathcal{J}_{\tau_i}^{(j)}} \{I_\sigma^*(I) \cup I_\sigma^{**}(I)\}. \tag{3.42}$$

One proceeds mutatis mutandis to obtain the values and intervals for τ_i (τ_i^*, τ_i^{**}) given σ_{i-1}, as well as $\mathcal{K}_{\tau_i}^{(j)}$.

Given the depth 1 validity intervals for σ_{i-1}, a set of intervals can be defined for τ_i. A temporary set of intersections between this set and the depth 1 validity intervals for τ_i can then be defined. Finaly using intersections between temporary sets and rotated temporary sets (Eq. 3.1) we can obtain *depth 2 validity intervals* (Fig. 3.2(C)). This iterative process is summarized in Algo. 1.

Algorithm 1 Computing depth-n validity intervals

Input: $\mathcal{J}_{\sigma_{i-1}}^{(j)}$ and $\mathcal{J}_{\tau_i}^{(j)}$, $i = 1, 2, 3$
Output: $\mathcal{J}_{\sigma_{i-1}}^{(j+1)}$ and $\mathcal{J}_{\tau_i}^{(j+1)}$, $i = 1, 2, 3$

for i $\in \{1, 2, 3\}$ **do**
  Step 1a Compute σ_{i-1} given τ_i and vice versa(Eq. (3.40))
  Step 1b Using $\mathcal{J}_{\sigma_{i-1}}^{(j)}$ and $\mathcal{J}_{\tau_i}^{(j)}$, define $\mathcal{K}_{\sigma_{i-1}}^{(j)}$ and $\mathcal{K}_{\tau_i}^{(j)}$ (Def. 3.6)
  Step 2a Define $\mathcal{J}_{\sigma_{i-1}}^{(tmp)}$, the intersections between $\mathcal{J}_{\sigma_{i-1}}^{(j)}$ and $\mathcal{K}_{\sigma_{i-1}}^{(j)}$.
  Step 2b Do the same for $\mathcal{J}_{\tau_i}^{(tmp)}$
  if $\mathcal{J}_{\sigma_{i-1}}^{(tmp)} = \{\emptyset\}$ or $\mathcal{J}_{\tau_i}^{(tmp)} = \{\emptyset\}$ **then**
    There will be no solutions for TLC
  Step 3: Project the $\mathcal{J}_{\sigma_{i-1}}^{(tmp)}$ and $\mathcal{J}_{\tau_i}^{(tmp)}$ along the edges of the C_α triangle using Eq. (3.1)
  Define $\mathcal{J}_{\sigma_{i-1}}^{(j+1)}$ as the intersections between $\mathcal{J}_{\sigma_{i-1}}^{(tmp)}$ and the projected $\mathcal{J}_{\tau_{i-1}}^{(tmp)}$.
  Do the same for $\mathcal{J}_{\tau_i}^{(j+1)}$ using $\mathcal{J}_{\tau_i}^{(tmp)}$ and the projected $\mathcal{J}_{\sigma_i}^{(tmp)}$
  if $\mathcal{J}_{\sigma_{i-1}}^{(j+1)} = \{\emptyset\}$ or $\mathcal{J}_{\tau_i}^{(j+1)} = \{\emptyset\}$ **then**
    There will be no solutions for TLC

Maximum number of deep validity intervals (DVI).

Lemma. 3.1. *Let* n_j *be the maximum number of intervals at depth* j *in* $\mathcal{J}_{\sigma_{i-1}}^{(j)}$ *or* $\mathcal{J}_{\tau_i}^{(j)}$*. This number satisfies*

$$n_1 = 4, n_{j+1} = 6n_j. \tag{3.43}$$

Proof. We first raise an observation for two sets of intervals $\mathcal{I}_1$ (of size n_1) and $\mathcal{I}_2$ (of size n_2) such that $I_1 \cap I_2 = \emptyset, \forall I_1 \in \mathcal{I}_1, \forall I_2 \in \mathcal{I}_1 \backslash I_1$. Assuming, without loss of generality, that $n_2 \geq n_1$, the maximum number of intervals determined by the intersections between intervals in these sets is $2 \times n_1 + n_2 - n_1$. To build this worst-case, we stab two intervals in $\mathcal{I}_1$ with an interval in $\mathcal{I}_2$, and squeeze the remaining $n_2 - n_1$ intervals from $\mathcal{I}_2$ inside intervals of $\mathcal{I}_1$.

To establish the lemma, we follow the steps of Alg0. 1, starting with the set $\mathcal{J}_{\sigma_{i-1}}^{(j)}$. If needed, we redefine the intervals in this set so that they are disjoint – to meet the hypothesis on the two sets $\mathcal{I}_1$ and $\mathcal{I}_2$ used in the observation above.

Considering the n_j intervals making up $\mathcal{J}_{\sigma_{i-1}}^{(j)}$ (or $\mathcal{J}_{\tau_i}^{(j)}$), we now inspect the following steps of Algo. 1:

- **Step 1b**) takes the intersection between $\mathcal{J}^{(j)}_{\sigma_{i-1}}$ and $\mathcal{K}^{(j)}_{\sigma_{i-1}}$ (Def. 3.6). By the observation above, this yields $2n_j \times 2$ restricted validity intervals.
- **Step 2a**) takes the intersection between $\mathcal{J}^{(j)}_{\sigma_{i-1}} \cap \mathcal{K}^{(j)}_{\sigma_{i-1}}$, which the observation above, yields $2n_j + (2n_j - n_j) = 3n_j$ intervals.
- **Step 3**) takes the intersection between the $3n_j$ intervals of the last step and the $3n_j$ from the δ projected intervals. This yields a maximum of $6n_j$ intervals.

□

3.5 C_α valence constraint and Inter-angular constraints: illustrations

3.5.1 Material: dataset of random instances

Our experiments use standard internal coordinates for bond lengths and valence angles [CSJD04]. The canonical values are available in our TLC implementation user manual (https://sbl.inria.fr/doc/Tripeptide_loop_closure-user-manual.html)

General dataset. We place our tripeptide in the reference frame using the first segment (Fig. 3.1). In this frame, we randomly generate $C_{\alpha;3}$ between two spheres around the origin. The inner radius $r_1 = 2$Å is smaller than the smallest value $\|C_{\alpha;3} - C_{\alpha;1}\|$ found in our exhaustive database of ~ 2.5 million tripeptides extracted from the PDB [ORC22]. The outer radius $r_2 = 2\,\|C_{\alpha;2} - C_{\alpha;1}\|$ respects the triangle inequality based on the distances between C_α carbons, and the canonical internal coordinates values. The position of $C_{\alpha;2}$ is generated uniformly in this volume. Atom C_3 is generated on sphere centered at $C_{\alpha;3}$, with a radius defined by the canonical bond length. The positions of $N_1C_{\alpha;1}$ and $C_{\alpha;3}C_3$ together with the canonical internal coordinate yield a TLC problem, which is fertile if embeddings/solutions are obtained. To each of those inputs correspond values for the input angles α_i, ξ_i, η_i for each $C_{\alpha;i}$.

Over $100,000$ instances, $24,076$ fertile ones were observed.

Planar dataset. A second similar dataset of the same size with $C_{\alpha;i+2}$ being positioned uniformly between two circles using the same radii. C_{i+2} is then also generated on a sphere around $C_{\alpha;i+2}$.

3.5.2 Validity intervals

Signatures. To assess the diversity of situations faced at C_α carbons, we introduce a signature based on the σ and τ interval types:

Definition. 3.7. *(Signature at C_α) Consider the endpoints of the validity intervals, in this order $\sigma_{i-1;-}, \sigma_{i-1;+}, \tau_{i;-}, \tau_{i;+}$. The* signature *of a TLC problem is a string in $\{N, P, Z\}^4$ – one letter for each each extreme angle, with the following convention:*

- *letter N for $cos(endpoint) < -1$,*
- *letter P for $cos(endpoint) > 1$,*
- *letter Z for $-1 < cos(endpoint) < 1$.*

Among the sample of embeddable anchor positions for TLC, only seven different signatures are found (Fig. 3.6). Not all combinations are possible as the the first and third letter cannot be N (Eq. (3.26)). Similarly the second and fourth cannot be P (Eq. (3.27)). If the first and second are ZZ (resp. third and fourth) then their counterpart will be PN. This illustrates that if the two endpoints are defined in $]0, \pi[$ for σ_{i-1} then it means that the whole circle is possible for τ_i (and vice versa).

3.5.3 Necessary conditions and Inter-angular constraints

General dataset. Using the general dataset, plotting the positions of $C_{\alpha;3}$ associated with solutions yields a bell shaped distribution with a hole on top (Fig. 3.7(A)).

Planar dataset. Due to the symmetry around the $N_1C_{\alpha;1}$, we consider the aforementioned planar data set (Fig. 3.7(B)). Consider a set of conformation, some fertile (TLC admits solutions), and some sterile. Fertile conformations naturally satisfy the necessary conditions defined by both the C_α *valence constraints* (Def. 3.1) and the Inter-angular constraints (Def. 3.5) for all σ and τ angles. However, we would like these constraints to be as tight as possible, retaining as few sterile configurations as possible. To assess this, for each constraint, we color code sterile configurations (Fig. 3.7(B), red points) using two colors: orange for sterile but failing the necessary test, and yellow for sterile passing the necessary test. Ideally, we would like as few yellow points as possible. It clearly appears that the *depth 1 inter-angular constraints* (Fig. 3.7(D)) are much tighter than the C_α *valence constraints* (Fig. 3.7(C)).

In terms of proportions, 24.08% of the points are instances where TLC yields solutions. In total 41.35% fulfill the C_α valence constraints. Finally 28.12% fulfill the *depth one inter-angular constraints*. They are included in the 41.35% and include the 24.08% just mentioned. The cases validating the Inter-angular constraint therefore represent $\sim 2/3$ of the cases fulfilling the θ_i linked equations. For the sake of clarity, it should be stressed that the gap between the 24.08% of instances with solutions, and the 28.12% fulfilling the depth one inter-angular constraints is not clearly visible using our projections in 3D and 2D (Fig. 3.7) since the whole configuration space is 5D.

Nevertheless, the of 2/3 gained can be used to scale the savings when it comes to explore the conformational space associated with m tripeptides defining a flexible region along a protein backbone. For various plausible values of m, one obtains: $m = 5 :\sim (3/2)^5 = 7.6$ $m = 10 :\sim (3/2)^{10} = 57.7$ $m = 15 :\sim (3/2)^{15} = 438$ $m = 20 :\sim (3/2)^{20} = 3325$. Therefore, when m increases, an exponential reduction is the size of the search space to be explored is gained. This strategy is used in the companion paper [OC22b]. Using depth n inter angular valence constraint with $n > 1$ can further reduce the search space as illustrated with the diminishing number of false positives (Fig. 3.8). Starting at depth 2 we have 18.12% fulfilling the constraint, then 16.57% for depth 3. There is no false positive in the planar dataset for depth 4 and onwards.

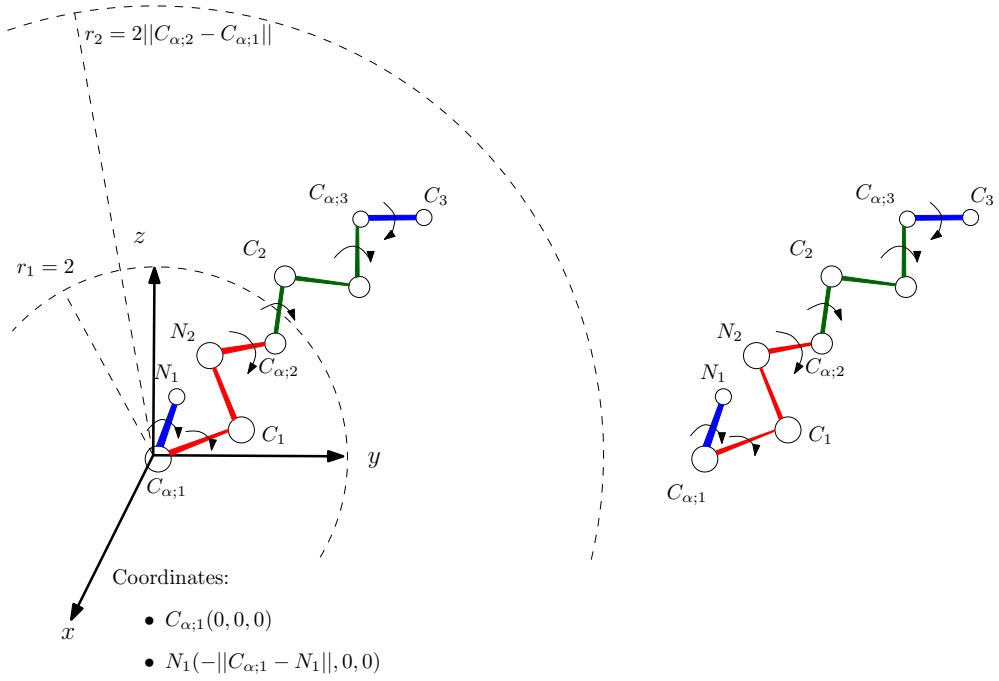
3.6 Outlook

The compact nature of folded proteins makes the exploration of their conformational space especially challenging. One must indeed avoid steric clashes and optimizes interactions, while dealing with complex coupled sub-problems involving the backbone and side-chains. A critical need in this context are so-called movesets able to propose plausible (low energy) configurations given a starting pause. Such movesets are indeed a corner stone in Monte Carlo based simulations at large.

The design of backbone movesets is a problem in itself, due to the necessity to handle loop closure constraints. The tripeptide loop closure provides an optimal solution to this problem, and therefore plays a crucial role to develop move sets. However, for loops involving several tripeptides, the question of combining solutions yielded by the individual tripeptides remains a challenging problem. Greedy approaches incrementally concatenating tripeptides have been developed, but these break the symmetry between the individual peptides, as the degree of freedom of those near the endpoints enjoy a finer sampling. On the way to processing all tripeptides in a sequence on an equal footing, this work studies (tight) necessary conditions on the first and last segment (bond) of a tripeptide, for TLC to yield solutions. As illustrated by our experiments, our conditions are rather tight, and yield an exponential saving in terms of the conformational space to be explored, when pooling several tripeptides. We leave the problem of improving the tightness of our constraints (notably using their iterated versions) as an open problem. Application-wise, the direct use of our constraints for the design of backbone movesets is presented in chapter **??**.

3.7 Artwork

Figure 3.1 Reference frame for tripeptide embeddings. We consider a tripeptide whose internal coordinates are fixed, except the six $\{(\phi, \psi)\}$ dihedral angles associated with the three C_α carbons. We assume that the segment $N_1 C_{\alpha;1}$ (first red line line segment) is fixed *i.e.* $C_{\alpha;1}$ is placed at the origin, and N_1 is placed at $(-\|N_1 - C_{\alpha;1}\|, 0, 0)$. We then aim at characterizing necessary conditions on the position of the last segment *i.e.* $C_{\alpha;3}C_3$ for the Tripeptide Loop Closure (TLC) algorithm to hold solutions.



3.8 Supporting information

Figure 3.2 Validity interval types and their relationships. (A) (IVI) Initial Validity Intervals. See Def. 3.2. **(B) (TVI) Rotated validity intervals.** See Def. 3.3. Obtained from the initial validity intervals (**(A)**). **(C) Depth-n/Deep Validity Intervals and their restrictions.** From $\mathcal{I}_\tau(i)$ and $\mathcal{I}_\sigma(i)$ we obtain $\mathcal{I}_{\tau|\delta}(i)$ and $\mathcal{I}_{\sigma|\delta}(i)$. From all of those we obtain intersections constituting $\mathcal{J}^{(1)}_{\tau_i}$ and $\mathcal{J}^{(1)}_{\sigma_i}$. This *depth one validity interval* set can be refined to depth n iteratively (Def. 3.4, Algo 1).

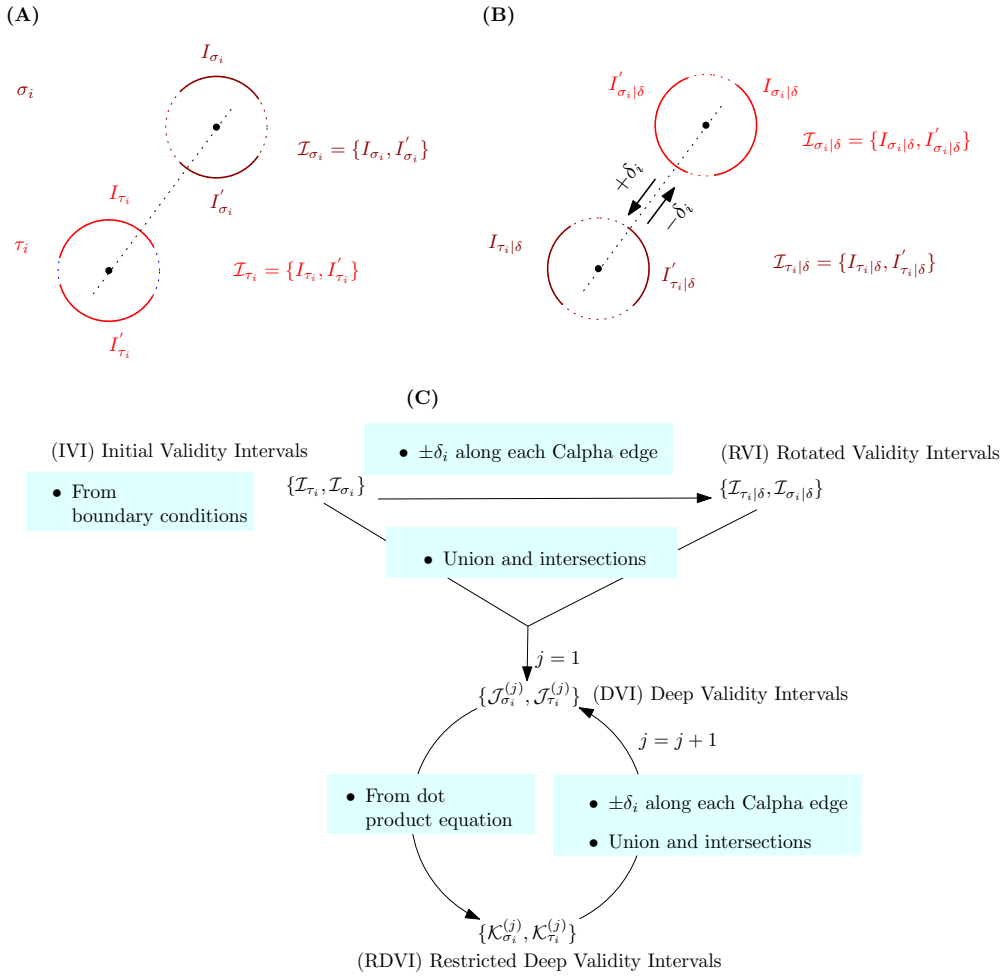


Figure 3.3 Tripeptide Loop Closure: main steps of the construction. Adapted from [CSJD04]. **(A)** Peptide bond linking two consecutive amino acids, and distance constraint induced on the line segment $C_{\alpha;i}C_{\alpha;i+1}$. The dihedral angle δ_i is defined by the three vectors $C_iC_{\alpha;i}, C_{\alpha;i}C_{\alpha;i+1}, C_{\alpha;i+1}N_{i+1}$. **(B)** The three rotations associated with the segments $C_{\alpha;1}C_{\alpha;2}$, $C_{\alpha;2}C_{\alpha;3}$ and $C_{\alpha;3}C_{\alpha;1}$. The rotation angles τ_i (resp. σ_i) concern atoms C_i (resp. N_i). But τ_i and σ_i satisfy $\sigma_i = \tau_i + \delta_i$. **(C)** Construction of the local orthonormal frame associated with $C_{\alpha;i}$ i.e. the C_α frame. **(D)** Introducing the variables α_i, η_i, ξ_i. **(E)** Modeling the constraint on valence angles at $C_{\alpha;i}$ carbons.

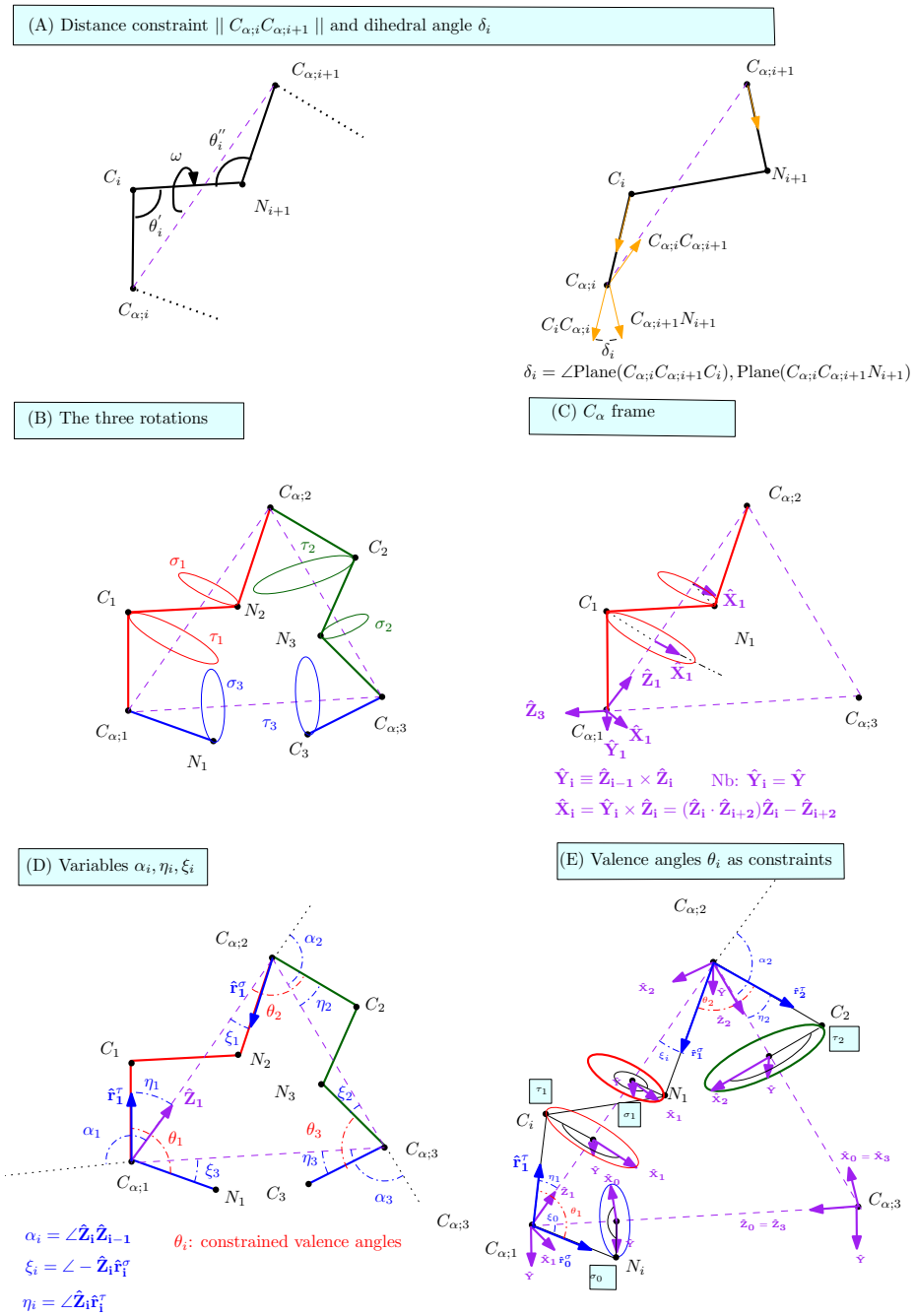
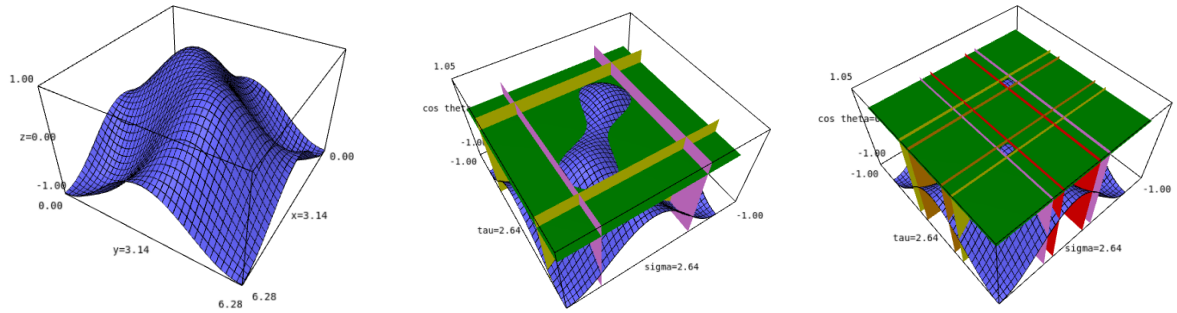
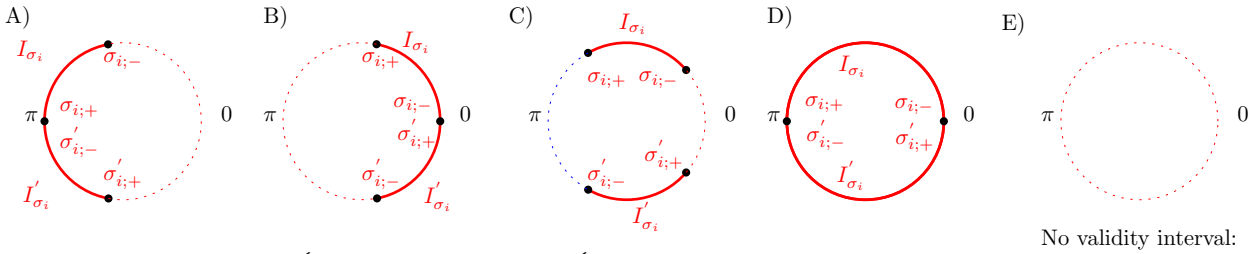


Figure 3.4 Example dot product surface and extreme angles $\sigma_{i-1;-}, \sigma_{i-1;+}, \tau_{i-1;-}, \tau_{i-1;+}$**.** TLC problem for the values $\alpha_i = 100, \chi_{i-1} = 50, \eta_i = 50$ **(A)** Whole surface **(B)** With horizontal plane $\cos\theta_i = \cos 9°$. Note the four vertical planes corresponding to the extreme angles. In this case, the intersection between the surface and the plane consists of a plane curve with two connected components. One component is enclosed by the four vertical planes. **(C)** With horizontal plane $\cos\theta_i = \cos 35°$. In this case, the intersection between the surface and the plane consists of a plane curve with one connected component.



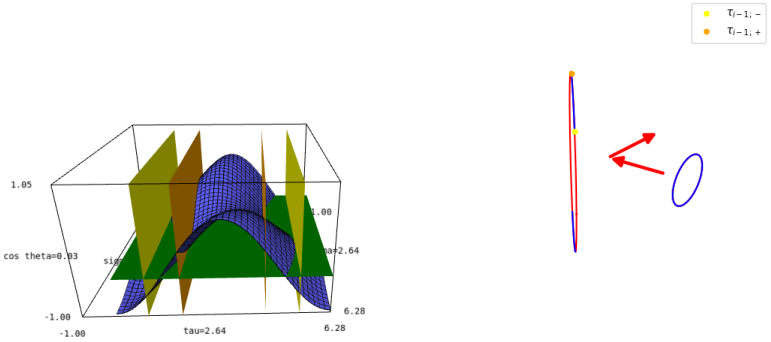
(A) (B) (C)

Figure 3.5 The four possible types of *initial/rotated* validity interval types for angle σ. The quantities of interest are defined in Eqs. (3.12) (3.15), (3.24), (3.25). Cases (A) to (D) stand for situations where $\sigma_{i;-}$ and/or $\sigma_{i;+}$ can be defined. In case (E), no validity interval can be defined.

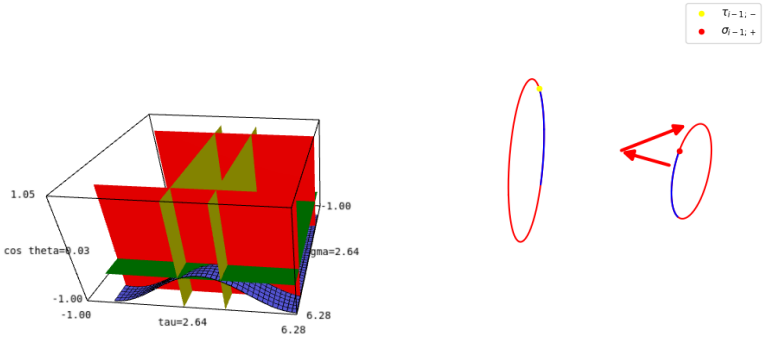


$\begin{cases} S^- < 1 \\ S^+ \leq -1 \end{cases}$ $\begin{cases} S^- \geq 1 \\ S^+ > -1 \end{cases}$ $\begin{cases} S^- < 1 \\ S^+ > -1 \end{cases}$ $\begin{cases} S^- \geq 1 \\ S^+ \leq -1 \end{cases}$

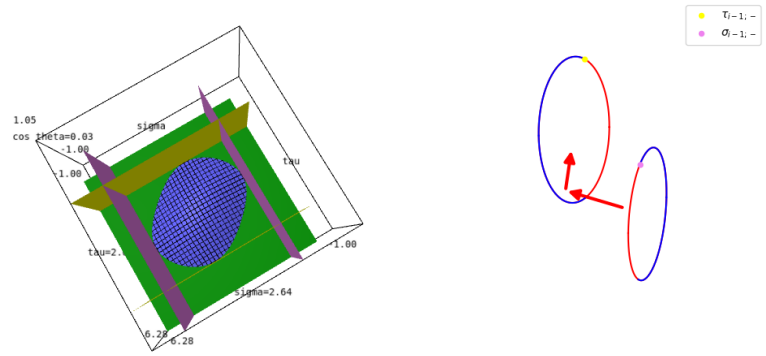
No validity interval:
$S^- < -1$
or
$S^+ > 1$



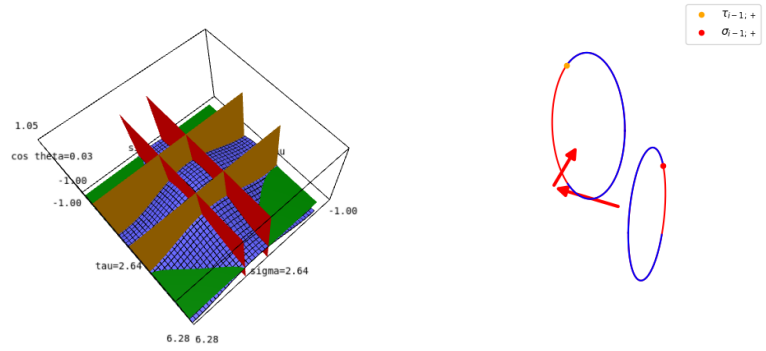
PNZZ



PZZN

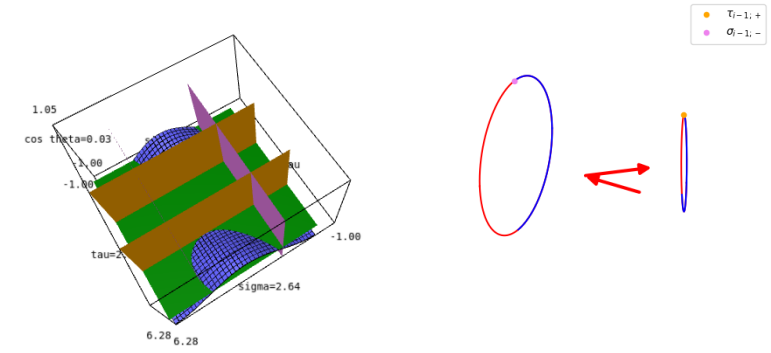


ZNZN

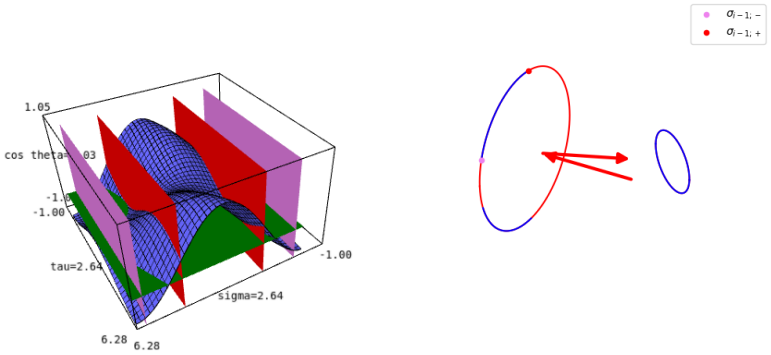


PZPZ

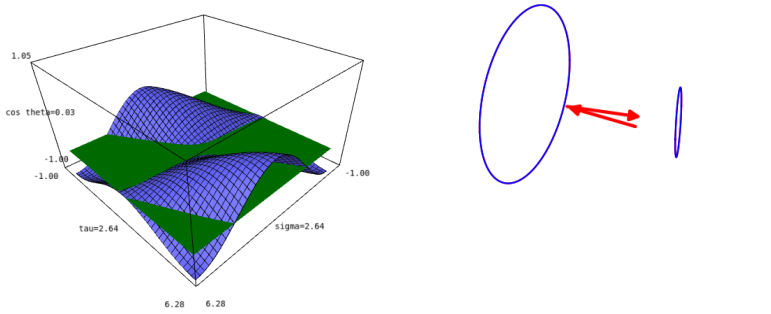
Figure 3.6 Dot surfaces and validity intervals for the dataset of random TLC instances. Color codes fir circle arcs: blue for valid intervals, black otherwise. Color code for circle arc endpoints: the colored bullets which indicate the angles. **(A)** The 7 signatures (Def. 3.7) in terms of extreme angles for the data set of random TLC instances. In all cases, the green plane corresponds to $\cos\theta_i = \cos 111.6°$. A signature reads as follows: N:negative ie dot product < -1; Z: zero ie dot product $\in [-1, 1]$; P: positive ie dot product > 1. **(B)** Validity intervals.



ZNPZ



ZZPN



PNPN

Figure 3.7 Embeddable tripeptides and necessary conditions: stringency of C_α valence constraints (Def. 3.1) versus depth 1 inter-angular constraints (Def. 3.5), illustrated on random instances projected into the reference frame of Fig. 3.1. (Nb: figures in 3D and 2D, while the configuration space is 5D.) (A) Blue (resp. red) points represent positions of $C_{\alpha;3}$ in instances when TLC yields at least one solution (resp. yields no solution). **(B)** A similar dataset generated uniformly on the of the sphere–gray equator in (A), color code as in (A). **(C) C_α valence constraints.** The $C_{\alpha;3}$ positions are depicted using three colors: blue points as in (A,B); orange points: points failing the C_α *valence constraints*; yellow points: points satisfying the C_α *valence constraints*, but for which TLC admits no solution. **(D) *Depth 1* inter-angular constraints.** Color code as in (C), using the depth 1 inter-angular constraints instead of the C_α valence constraints. Note the reduction of the yellow region.

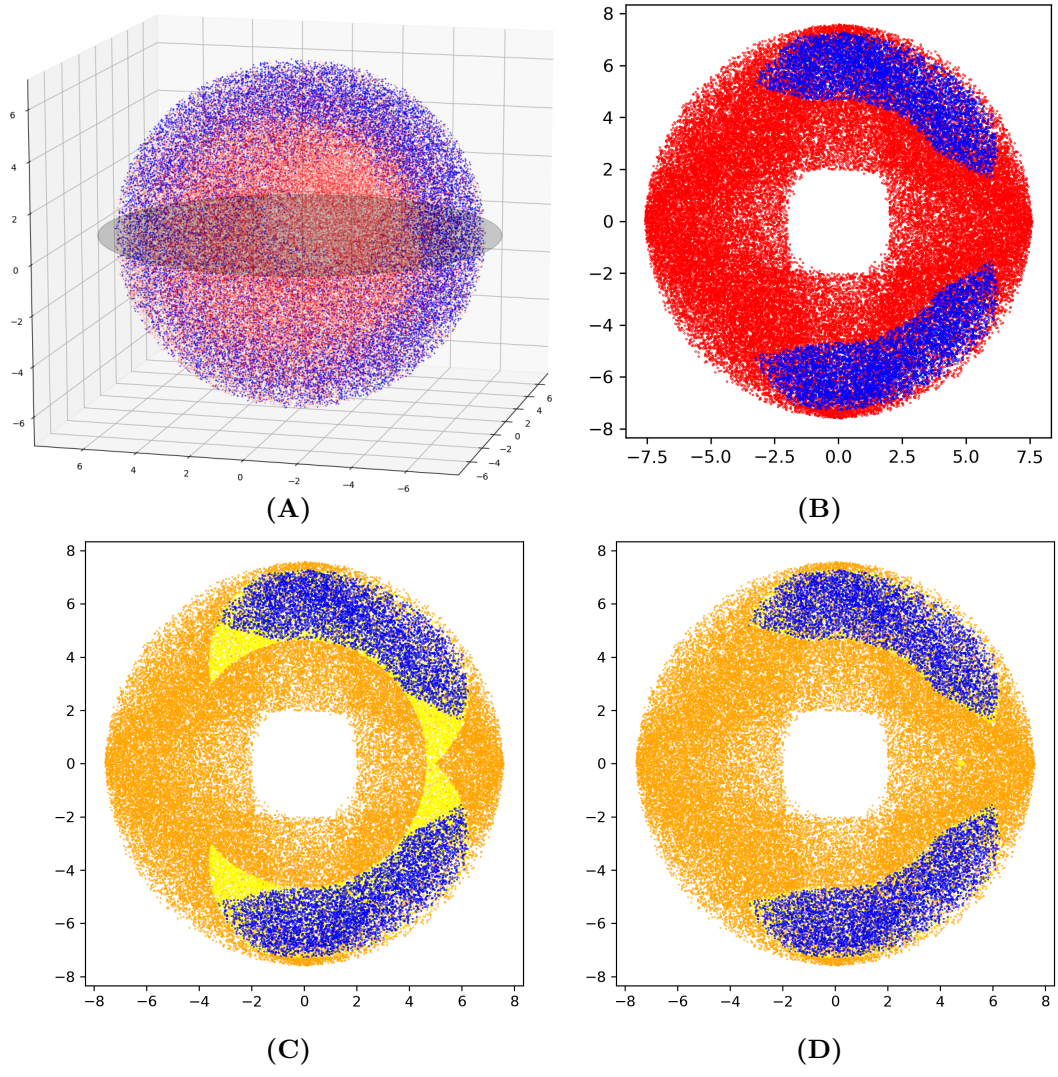


Figure 3.8 Proportion of false positives for C_α valence and depth n inter-angular constraints with n $\in \{1, 2, 3\}$ The proportion is defined as the number of false positives divided by the number instances when TLC yields no solution in the planar dataset (Sec. 3.5.1). The specific percentages are, C_α valence constraint: 18.24%, depth 1 inter-angular constraint: 4.01%, depth 2 inter-angular constraint: 1.89%, depth 3 inter-angular constraint: 0.02%.

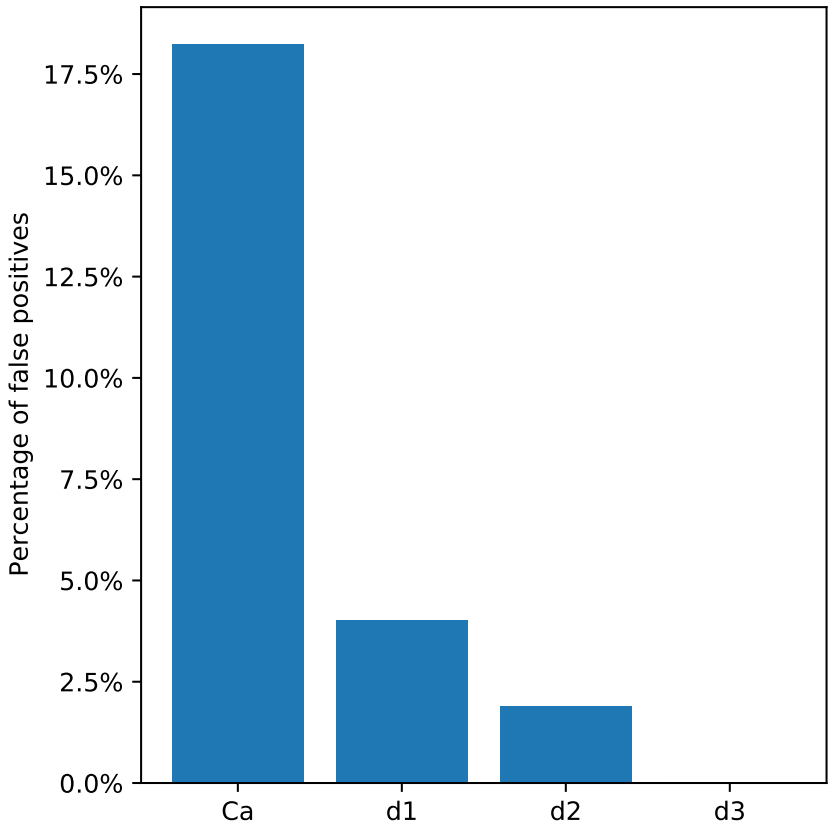
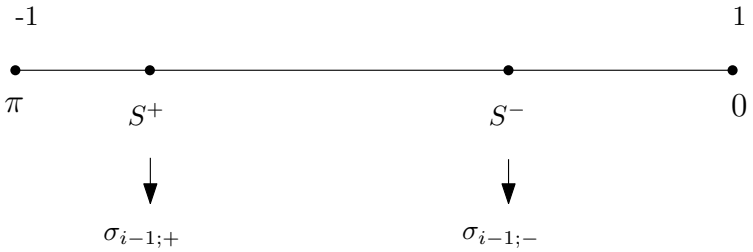


Figure 3.9 Conditions to define the four extreme angles: the case of σ_{i-1}.



Chapter 4

Analysis of tripeptide Loop Closure reconstructions

4.1 Introduction

The Tripeptide Loop Closure problem. The TLC problem has a long history in robotics and molecular modeling, see e.g. [GS70, PC94, CS04, PRT$^+$07, CSWD06, CLW$^+$16].

Consider, in a peptide chain, a tripeptide for wich we have all the bond length $\{d_i\}$, valence angle $\{\theta_i\}$ and ω_i dihedral angle values. As mentioned in chapter 3, TLC considers which combination of values for six consecutive ϕ and ψ dihedral angles close the gap given the positions first two backbone atoms in the tripeptide and the last two.

Mathematically, solving the problem requires finding the real roots of a degree 16 polynomial, which also means that up to 16 solutions may be found [PRT$^+$07, CSJD04, NOS05].

We note in passing that the sensitivity of atomic positions to fluctuations of a specific internal coordinate in a loop (a *fuzz* parameter) have been studied in [NOS05].

Over time, TLC has proven to be a key building block to reconstruct and sample loop conformations, as shown by the following two examples.

Ramachandran distributions. The TLC problem is also closely related to the study of Ramachandran distributions, which characterize the coupling between ϕ and ψ angles along the protein backbone [Fer99]. There are four main types of Ramachandran plots: glycine – an amino acid without side chain, proline – whose cycle induces specific constraints, pre-proline – residues preceding a proline, and the remaining amino acids, whose C_β carbon induces specific constraints. In our work, we illustrate this latter class with ASP. Four main regions are occupied in the Ramachandran diagram: β-sheets (βS), polyproline II (βP; left-handed helical structure whose angles are characteristic of β-strands); α-helical (αR); and left handed helix (αL). These regions were characterized using a combination of five steric constraints between four atoms defining the Ramachandran tetrahedron ([STM$^+$77], Fig. 2.7). (We note in passing that the 6th edge of this tetrahedron, between O_i and N_{i+1}, was not used in defining the steric constraints, likely due to the fact that this edge corresponds to a valence angle – a constraint stronger than that associated with the other edges.) In this work, the curves delimiting the occupied regions are termed the Ramachandran *template*. More recently the diagonal shape of level set curves in the occupied regions was explained using dipole-dipole interactions, distinguishing the generic case and proline [HTB03], and glycine and pre-proline [HB05]. The characterization of neighbor dependent Ramachandran distributions has also been studied [TWS$^+$10]. From a statistical standpoint, the Ramachandran distributions of two specific residues can be compared using say f-divergences such as Kullback-Leibler, Hellinger, etc.

Contributions. In this work, we perform a careful assessment of reconstructions to TLC problems, with a particular emphasis on the comparison between distributions in angular spaces, between data from the PDB on the one hand, and TLC reconstructions on the other hand.

First, we present a robust implementation of TLC, showing the role of multiprecision in ensuring the existence and the accuracy of reconstructions. Second, using tripeptides from the PDB as a reference, we present a detailed analysis of reconstruction, from the geometric, statistical, and biophysical standpoints. We also discuss some possibilities to exploit such reconstructions.

4.2 Material and Methods

4.2.1 Material: tripeptides from the PDB

We extract a database $\mathcal{D}$ of tripeptides found in high resolution structures (resolution better than 3Å) from the PDB (23rd of September 2020), having mutual sequence identity lower than 95%. A contiguous, gap-less region of a protein backbone qualifies as a tripeptide if the following two conditions hold: (C1) The highest Bfactor in backbone atoms of the tripeptide is less than 80 Å^2. (C2) The center of the tripeptide is separated by at least 3 amino acids from a stable secondary structure(SSE) on both ends, a condition meant to remove the constraint of SSE anchoring loops to the rest or the structure [TWS$^+$10]. Stable secondary structure (SSE, β folds and right handed α helices) are extracted from mmtf files. These files are annotated using the BioJava implementation [PYB$^+$12] of the DSSP program (Define Secondary Structure of Proteins [KS83]).

In order to compute the original values of the first ϕ and last ψ dihedral angles, the tripeptide at the end or beginning of a chain is excluded from our computation as the positions of the last atom of the previous residue and the first of the next one are necessary. Taken together these conditions result in the database $\mathcal{D}$ containing 2,495,095 tripeptides. We denote $\mathcal{A}_\mathcal{D}$ the corresponding encoding in the 6D space of dihedral angles, that is

$$\mathcal{A}_\mathcal{D} = \{\text{Angles}(t), t \in \mathcal{D}\}, \text{ with Angles}(t) = (\phi_1(t), \psi_1(t), \phi_2(t), \psi_2(t), \phi_3(t), \psi_3(t)). \tag{4.1}$$

We note in passing that the three pairs of angles are coupled–meaning for example that picking values independently for the three pairs would jeopardize loop closure. We qualify a tripeptide with its span (Euclidean distance between its endpoints the N_1 and C_3 atoms (Fig. S4.1). In computing the percentage of tripeptides containing a given amino acid at least once, Glycine is followed by Proline and Aspartic acid (29.6%, 24.7%, 21.6% of tripeptides respectively) (Table S4.1(A)). For the percentage of tripeptides containing a particular amino acid at least twice, this ordering remains the same, the relative gap between glycine and the following amino acids being wider (Table S4.1(B)).

4.2.2 The classical TLC problem

Data versus reconstructions. We consider the tripeptide loop closure with fixed bond lengths and angles, as well as ω dihedral angles. As mentioned previously both sides of this tripeptide are fixed (i.e. $C_O, N_1, C_{\alpha 1}$ and $C_{\alpha 3}, C_3, N_4$), meaning that the collective change in dihedral angles only affect the Cartesian embeddings of C_1, N_2, CA_2, C_2 and N_3 (Fig. 4.1(A)). For a TLC problem defined by a tripeptide t, the set of solutions and the ancestor of a solution (that is, the tripeptide yielding that solution) are denoted as follows (Fig. S4.6 for one example):

$$\begin{cases} Sol(t) = \{r_1, \ldots, r_k\}, \text{ with } k \leq 16. \\ \text{DataTripeptide}(r_i) = t, \forall i = 1, \ldots, k. \end{cases} \tag{4.2}$$

A TLC problem is expected to return the tripeptide it is defined from. (As we will see, this depends on the number type used.) In the solution set $Sol(t)$, we will therefore assume that r_1 is the reconstruction most similar to the data tripeptide t, in the RMSD in 3D space sense. (Setting aside numerical precision issues,

the data tripeptide should be exactly reconstructed, i.e. the RMSD should be zero.) We define accordingly the solution set minus the data tripeptide, that is

$$\overline{Sol}(t) = Sol(t)\backslash\{r_1\}. \tag{4.3}$$

Phrased differently, the set $\overline{Sol}(t)$ consists of *reconstructed only* geometries.

Angular spaces. Denote $d_{S_1}(\cdot,\cdot)$ the shortest angular distance between two points on the unit circle S_1, expressed in Radians. To compare tripeptides whose 6D dihedral coordinates are denoted $\text{Angles}(t) = (\tau_1,\ldots,\tau_6)$ and $\text{Angles}(t') = (\tau_1^{'},\ldots,\tau_6^{'})$ respectively, we use as distance the L_p-norm – in practice with $p = 1$:

$$d_p(t,t') = \big(\sum_{i=1}^{6} d_{S_1}(\tau_i,\tau_i^{'})^p\big)^{1/p}. \tag{4.4}$$

We also consider the following angular data associated with all reconstructions:

$$\mathcal{A}_{TLC} = \{\text{Angles}(l), l \in Sol(t), t \in \mathcal{D}\} \tag{4.5}$$

With the specific goal of analyzing reconstructions which differ from the original data, we consider the set of angles for all reconstructions except data:

$$\mathcal{A}_{\overline{\mathcal{D}}} = \mathcal{A}_{TLC}\backslash\mathcal{A}_{\mathcal{D}}. \tag{4.6}$$

For data in $\mathcal{A}_{\mathcal{D}}$ (resp. reconstructions in $\mathcal{A}_{\overline{\mathcal{D}}}$), the pairs of dihedral angles of the i-th tripeptide are denoted (ϕ_i,ψ_i) (resp. $(\overline{\phi}_i,\overline{\psi}_i)$) and the corresponding Ramachandran domain is denoted $\mathcal{R}_{\mathcal{D},i}$ (resp. $\mathcal{R}_{\overline{\mathcal{D}},i}$).

TLC and internal coordinates. Solving a particular TLC problem puts the focus on dihedral angles, so that that there are two options to handle the other internal coordinates (bond length and valence angles): *data internals* using those found in the tripeptide processed, and *canonical internals* using standard values for fixed internals, as done in the original version[CSJD04]. As we shall see, the former is beneficial in several respects.

4.2.3 TLC with gaps

A generalization of the classical TLC consists of considering three amino acid which are not contiguous along the backbone. This is of interest in the case of three linkers enclosing two rigid SSE. Mathematically, this is akin to the original problem, with the rigid blocks modeled as fictitious bonds separating the amino acid (Fig. 4.1(B)). Once the coordinates of all atoms not in these rigid blocks are embedded, the rigid blocks are then translated and rotated into their final positions (Fig. S4.7 for one example).

4.3 Results

4.3.1 Software

TLCG algorithm. This work is accompanied by our implementation of the tripeptide loop closure algorithm, in the Structural Bioinformatics Library ([CD17], `http://sbl.inria.fr`, `https://sbl.inria.fr/doc/Tripeptide_loop_closure-user-manual.html`). From the application standpoint, given a chain in a PDB file, together with the identification of the three a.a. defining the tripeptide (not necessarily contiguous), the application `sbl-tripeptide-loop-closure.exe` produces modified PDB-format files for each solution found, if any. The constraints for internal coordinates can be specified from the data (default), using standard values, or supplied in the form of a file.

Numerics. The numerical stability of an algorithm is key to its robustness [BC13]. For TLC, the precision used to represent the floating point numbers is expected to play a role.
The application `sbl-tripeptide-loop-closure.exe` makes it possible to specify the precision used for calculations. Internally, the number type used is `CGAL::Gmpfr`, a representation based on the `Mpfr` library [FHL+07] supplying a fixed precision floating point number type. Practically, this fixed precision is a multiple (> 1) of the default double precision: `TLCdouble`[-x1], called `TLCdouble` for short in this work, refers to the executable `sbl-tripeptide-loop-closure.exe` using the plain double precision; `TLCdouble`[-x2] (resp. `TLCdouble`[-x4]) refers to `sbl-tripeptide-loop-closure.exe` using a double double (resp. quadrice double) precision.

To assess the importance of using data-extracted (as opposed to standard) internal coordinates, we also evaluate `TLCCoutsias` [CSJD04], the original TLC algorithm using standard bond length and valence angles, with double precision for numerics.

4.3.2 Numerical analysis of the stability of the reconstruction

Rationale. Solving a TLC problem for a tripeptide l raises two questions.

The first question refers to the existence of a solution matching the data l itself. The response can be negative since numerical rounding errors during the calculation of the polynomial may yield, in particular for an ill-conditioned TLC polynomial, a situation with zero real solution [BC13]. In that case, we will say that the solution *evaporates*. If solutions are found, we define *the reconstruction* as the geometry most similar to l, using as distance the RMSD of the atoms in the tripeptide. Note that RMSD and not least-RMSD is used for this comparison as the orientations are fixed.

The second question is then the geometric distance between the data and the reconstruction. This distance, also measured by the RMSD, is expected to depend on the floating point number type used.

Results. We process all cases in the database $\mathcal{D}$. The fraction of TLC problems with no solution depends heavily on the option used for number types and internal coordinates other than dihedrals: `TLCCoutsias`: 8.1%; `TLCdouble`: 5 10^{-4}%; `TLCdouble`[-x2]: 2 10^{-5}%; `TLCdouble`[-x4]: 0.0. A similar conclusion holds for the RMSD between the data and the (best) reconstruction (Fig. 4.2(A, B)): `TLCCoutsias`: up to $\sim$ 3Å RMSD; `TLCdouble`: up to $\sim$ 1.2Å RMSD; `TLCdouble`[-x2]: very small values with one outlier at $\sim$ 0.65Å RMSD; `TLCdouble`[-x4]: all RMSDs smaller than $\sim$ 0.009Å RMSD.

Altogether, these observations stress the importance of using data-extracted internal coordinates, and to a lesser extent the role of numerical precision to avoid evaporation. While `TLCdouble` is sufficient to characterize distributions, `TLCdouble`[-x2] is preferable to process satisfactorily all individual cases. In the sequel, all results presented were obtained with `TLCdouble`[-x2].

4.3.3 Geometric analysis of solutions in 3D

Rationale. To assess solutions, we consider the reconstruction from $\overline{Sol}(t)$ most dissimilar to t, in the RMSD sense.

Results. For data extracted internals, the number of solutions of TLC problems is as high as 12 (Fig. S4.8). The analysis of the geometric diversity in terms of max RMSD as a function of the geometric span of the tripeptide (Euclidean distance between its endpoints) yields two interesting insights (Fig. 4.3). First, with only 5 displaced atoms, a significant RMSD is observed, up to 3.8Å. Second, the distribution is bimodal, but the two modes get closer (and even coalesce) when the span increases. This can be explained by the fact that the larger the gap, the straighter the solutions.

As a complementary analysis, consider the displacement of the 5 moving atoms in the tripeptide (Fig. 4.1). For each atom, we compare the generated position against the initial position. As expected, the displacement increases with the centrality of the atom, with displacements which can be very significant, namely up to 6 Å(Fig. S4.9).

4.3.4 Geometric analysis of solutions in 6D

Rationale. We wish to perform a geometric comparison of the two 6D point clouds $\mathcal{A}_{\mathcal{D}}$ and $\mathcal{A}_{\overline{\mathcal{D}}}$ coding all tripeptides in the data $\mathcal{D}$ and in all TLC solutions (minus the data tripeptides themselves) respectively (Section 4.2.2).

To see how, consider two set of points in 6D, say X and Y. For a point $x \in X$, using the distance from Eq. 4.4, we define the nearest neighbor in Y and the associated distance by

$$\forall x \in X : \begin{cases} nn_Y(x) \stackrel{Def}{=} \arg\min_{y \in Y} d_p(x,y); \\ d_p^{(Y)}(x) \stackrel{Def}{=} d_p(x, nn_Y(x)) \end{cases} \tag{4.7}$$

Phrased differently, $nn_Y(x)$ is the point in the database Y minimizing the distance $d_p(\cdot,\cdot)$ to x, and $d_p^{(Y)}(x)$ is the corresponding distance.

Remark 4.1. *We may need to restrict the search of the nearest neighbor of a tripeptide x to a certain class of tripeptides sharing a specific property with x – e.g. featuring a C_β. The corresponding operator is denoted $nn_Y^{Class}(x)$.*

Results. The distribution of $d_p^{(\mathcal{A}_{\mathcal{D}})}(x), \forall x \in \mathcal{A}_{\overline{\mathcal{D}}}$ has a sharp mode at zero, showing that $\sim 20\%$ of solutions (data tripeptides excluded) are highly similar to a tripeptide existing in $\mathcal{D}$ (Fig. S4.10(A)). Taking the reverse point of view, the distribution of $d_p^{(\mathcal{A}_{\overline{\mathcal{D}}})}(x), x \in \mathcal{A}_{\mathcal{D}}$ shows that the number of data tripeptides similar to a solution is $\sim 50\%$ (Fig. S4.10(B)). Interestingly, the span of values in these two histograms are circa 130 and 40 degrees respectively, showing that loop closure tripeptides are far more diverse than PDB peptides.

4.3.5 Analysis of Ramachandran distributions

Rationale. We complement the previous geometric analysis by studying the distributions in Ramachandran spaces. The focus in doing so is twofold: first, comparing the distributions in $\mathcal{R}_{\mathcal{D},i}$ versus $\mathcal{R}_{\overline{\mathcal{D}},i}$, and second, analyzing the patterns observed with respect to those known for classical Ramachandran plots (Fig. 2.7).

Results. We inspect individual Ramachandran distributions over the domains $\mathcal{R}_{\mathcal{D},i}$ versus $\mathcal{R}_{\overline{\mathcal{D}},i}$, considering three prototypical amino acids, namely ASP (Fig. S4.11), GLY (Fig. S4.12), and PRO (Fig. S4.13). Ramachandran distributions in the domains $\mathcal{R}_{\mathcal{D},i}$(left columns) are indistinguishable (also confirmed by the calculation of the Hellinger and Jensen-Shannon divergences, data not shown), a fact which is expected since tripeptides from the database $\mathcal{D}$ are obtained by sliding a window of size three along loops found in structures from the PDB.

On the other hand, the three Ramachandran distributions associated with the TLC domains $\mathcal{R}_{\overline{\mathcal{D}},i}, i = 1,2,3$ are rather different. Distributions in the domains $\mathcal{R}_{\overline{\mathcal{D}},1}$ and $\mathcal{R}_{\overline{\mathcal{D}},3}$ still exhibit isolated regions corresponding to classical regions, except that the distributions are much more uniform in the entire Ramachandran space. The middle distribution (space $\mathcal{R}_{\overline{\mathcal{D}},2}$) departs from these. The coverage of the entire space is more uniform, setting aside a central void surrounded by an *annulus* connecting the clusters corresponding to the classical structures (left and right handed α helices, β folds). The central void/eye corresponds to the steric constraint $O_{i-1}N_{i+1}$ in the reconstruction target(Fig. 2.7). It should be noticed, though, that the clear cut nature of this void results from the fact that data have been removed from the solutions set (Eq. 4.6). In plotting all pairs of angles (ϕ,ψ) from our database $\mathcal{D}$, one indeed obtains atypical conformations in this central region (background of Fig. 2.7). In any case, the superposition of the Ramachandran template onto the map shows that solutions partly fill the void (Fig. 4.4). Interestingly, the center of the void is preserved even though the distance constraints encoded in the Ramachandran tetrahedron are not used in the specification of the TLC problem – since N_{i+1} only is involved in the TLC problem (Fig. 2.7). The maps $\mathcal{R}_{\mathcal{D},i}$ and $\mathcal{R}_{\overline{\mathcal{D}},i}$ is can be compared in two ways. The first one consists of computing a *density difference map*, pixel-wise, for a discretization of the 2D Ramachandran space. Normalization for the difference map

is obtained by dividing the value of each bin by the sum of absolute differences of the compared maps. To indicate the map with larger values, density difference maps are visualized using two color maps. The second one consists of considering the *difference correlation map*, that is the density map for the differences $(\phi(t) - \phi(r_i), \psi(t) - \psi(r_i),$ for all $r_i \in \overline{Sol}(t)$.

The first comparison (Figs. S4.14, S4.15, and S4.16, left columns) shows areas in purple with relative higher data concentration as small clusters whereas higher reconstruction concentration is much more dispersed. This dispersion however is not uniform as the reconstruction method only retains angles compatible with the loop closure.

The second comparison yields two complementary insights (Figs. S4.14, S4.15, and S4.16, right columns). The first one relates to steric constraints found before and after the tripeptide. The first (resp. third) difference plot indeed exhibits a vertical (resp. horizontal) stripe, showing that ϕ_1 is more constrained than ψ_1 (resp. ψ_3 more constrained than ϕ_3). In fact, picking arbitrary values for ϕ in the first amino acids or ψ in the third one would make loop closure impossible. The second one owes to the concentration exhibited by the X-like shape in the middle of the difference plot. This strong correlation (logarithmic scale used) indicates a strong coupling between the ϕ and ψ angles.

4.3.6 Biophysical analysis based on the potential energy of solutions

Rationale. As a separate assessment of the quality of reconstructions returned by `TLCdouble`, we compute the potential energy (denoted V) of the tripeptide backbone including heavy atoms (*i.e.* the carbonyl oxygens and C_β) involved in the specification of the regions occupied in Ramachandran diagrams (Fig. 2.7). This analysis imposes two constraints. First, we discard tripeptides containing PRO. Second, we assign a type to each tripeptide, out of 2^3 possibilities corresponding to the presence or absence of a GLY at each position. This type is used in particular to find the nearest neighbor of a reconstruction amongst all tripeptides of the same type in $\mathcal{A}_\mathcal{D}$, which we denote $nn_Y^{Class}(x)$. (See Rmk 4.1. In Eq. (4.7), the set Y is filtered to retain those tripeptides whose type matches that of x.) Practically, we present plots for the most abundant class, corresponding to tripeptides with a C_β at each position.

Using the `AMBER` ff14sb force field, three potential energy terms are taken into account :

- The first corresponds to the contribution of dihedral angles. Each such angle contributes $\sum_n (k(1 + \cos(n\phi - \phi_0)))$ with n the periodicity of the term, k the energy constant, ϕ_0 a phase shift angle, and ϕ is the torsion angle formed by the four bonded particles.
- The second term is the electrostatic interaction between non bonded particles. Each non bonded pair contributes $\frac{q_i q_j}{4\pi\epsilon d}$, where ϵ is the dielectric constant, q_i, q_j are the charges of the two particles, and d is their distance.
- The last term is the van der Waals interaction term. Each non bonded pair contributes $\epsilon(\frac{\theta^-}{d^{12}} + \frac{\theta^+}{d^6})$ where ϵ is a constant, θ^-, θ^+ are the repulsive and attractive Lennard-Jones terms, and d is the distance between particles.

In any case, only contributions impacted by the changes made by the TLC algorithm are taken into account. For the dihedral angles, this implies that proper dihedrals around the peptide bonds are not taken into account. For the non bonded interactions only pairs whose relative distance changes contribute.

To assess the potential energy of a reconstruction in $\mathcal{A}_{\overline{\mathcal{D}}}$, we compare this potential energy to a reference point. This can either be the nearest neighbor of each point ($nn_{\mathcal{A}_\mathcal{D}}(x)$, Eq. 4.7) or the data DataTripeptide(x) used to generate it (Eq. 4.2). This yields the following two relative changes for the potential energy $V_*(\cdot)$ with $* \in \{dihedral, elec., vdW\}$:

$$\Delta_r V_*(x) = \frac{V_*(x) - V_*(nn^{Class}_{\mathcal{A}_\mathcal{D}}(x))}{V_*(nn^{Class}_{\mathcal{A}_\mathcal{D}}(x))}, \forall x \in \mathcal{A}_{\overline{\mathcal{D}}}. \tag{4.8}$$

or

$$\Delta_r V_*(x) = \frac{V_*(x) - V_*(\text{DataTripeptide}(x))}{V_*(\text{DataTripeptide}(x))}, \forall x \in \mathcal{A}_{\overline{\mathcal{D}}}. \tag{4.9}$$

Using the whole database, we perform a scatter plot in the plane $(d_p^{(\mathcal{A}_\mathcal{D})}(\cdot), \Delta_r V_*(\cdot))$, and represent the resulting 3D histogram using a heatmap.

Results. The potential energy is a measure of the *strain* of reconstructions. The analysis of the three potential energies and the two comparison setups yields several interesting facts (Fig. 4.5):

- **Magnitude of angular changes.** We note that using the nearest neighbor of a reconstruction significantly reduces the L1 distance (Fig. 4.5: from $[100, 750]$ to $[0, 250]$ degrees), an indication on how a reconstruction from $\mathcal{A}_{\overline{\mathcal{D}}}$ differs from its data tripeptide in terms of dihedral angles.

- **Magnitude of potential energy changes ΔV_* in kcal/mol – Table S4.2.** The absolute difference ΔV_* has a different scale for the three potential energy terms used: $V_{dihedral}$ yields the smallest changes, then $V_{elec.}$ and finally V_{vdW}. The low energetic impact of changes to dihedral angles is what makes it a priority target to modify structures in protein molecules and why TLC is such an interesting approach. The changes in $V_{elec.}$ in the backbone of proteins are more sensitive to modifications done by TLC as the energy linearly depends on the inverse of the distance d between non bonded atoms. In the same spirit, with a larger exponent (d^{12}), the changes in V_{vdW} are the largest ones. It should be noted that $V_{vdW}(\text{DataTripeptide}(x))$ has a larger value than the difference $\Delta V_{vdW}(x)$.

- **Magnitude of relative changes – Table S4.3).** Out of the three potential energies, V_{vdW} displays a significant difference in terms of relative changes for the two reference tripeptide definitions: from $[-0.1, 0.5]$ (Fig. 4.5(C)) to $[0.05, 0.25]$ (Fig. 4.5(F)). Even though a reconstruction resembles less its ancestor than its nearest neighbor in terms of angular coordinates, the spread of relative changes is smaller for ancestors.

- **Centering and symmetry of relative changes.** Relative changes for $V_{dihedral}$ exhibit a relative symmetry about $\Delta_r V_{dihedral} = 0$ (Fig. 4.5(A,D)), which is expected due to the periodic form of this potential energy. A relative symmetry is also observed for V_{vdW}, about $\Delta_r V_{vdW} \sim 0.17$ and $\Delta_r V_{vdW} \sim 0.16$ respectively (Figs. 4.5(C,F)). This negative value shows that data tend to have a smaller V_{vdW}, yet reconstructions occasionally yield more favorable interactions. Finally, $V_{elec.}$ only displays negative values for relative changes (Figs. 4.5(B,E)), stressing the rather tight optimization of this potential energy in native structures.

- **Distance $d_1 = 0$ does not imply $\Delta V_{dihedral} = 0$.** It also appears that $d_p \to 0$ implies $\Delta_r V_{dihedral} \to 0$ (Fig. 4.5(A)). This can be explained by considering that if there is no difference in free dihedral angles then the energy term obtained corresponds to that of its reference. This is not true however for $\Delta_r V_{vdW}$ and $\Delta_r V_{elec.}$. When using $nn^{Class}_{\mathcal{A}_\mathcal{D}}(x)$ as reference these are impacted by the differences in the other internal coordinates, differences that impact interatomic backbone distances.

4.4 Discussion and outlook

Tripeptide loop closure (TLC) is a classical strategy to generate conformations of tripeptides, e.g. to reconstruct missing segments in structural data, or to implement move sets in simulation methods. Specifically, a TLC problem solves for six dihedral angles, keeping the remaining internal coordinates (bond lengths, valence angles) constant. Solutions are determined by the real roots of a degree 16 polynomial, which makes it very convenient to generate discrete conformations, but which raises questions regarding the biophysical relevance of solutions. The focus of this work is precisely to provide a detailed assessment of reconstructions, using tripeptides from the protein data bank as a reference.

From the computational standpoint, we show that multiprecision is required for the existence and the accuracy of reconstructions. From the geometric standpoint, it appears that the number of solutions depends on the endpoint to endpoint distance of the gap to be filled. Also, despite the fact that a mere five atoms are moving, RMSD up to ~ 6Å are observed, showing that TLC yields a significant conformational diversity.

From the statistical standpoint, we present a detailed comparison of angular distribution in the Ramachandran spaces of data and reconstructions, for each of the three positions in the tripeptide. The specific distribution for the second tripeptide in reconstructions is remarkable. This distribution features a central empty region –the *void* pattern, and is more uniform than classical Ramachandran distributions. Such differences actually owe to the different nature of these two distributions. On the one hand, classical Ramachandran distributions encode propensities observed in protein structures, typically extracted from the protein data bank. Such structures are biased toward (meta-)stables states, and one expects transient regions to be under-represented. On the other hand, Ramachandran distributions associated to reconstructions inherently encode the propensities of angles in the TLC reconstructions, which, as we have seen, endow the central atoms of the tripeptide with enhanced move capabilities.
The results thus show that, while reconstructions are themselves conditioned to the input PDB data, their bias towards (meta-)stable structures is less pronounced. Application-wise, the void pattern provides strong hints on how to interpolate between two tripeptides geometries. Given two conformations encoded by two points in the 6D space, one may indeed attempt to connect them while staying away from the void region in a manner akin to path planning in robotics. This strategy, which remains to be explored, poses two difficulties. First, the Ramachandran diagrams of the three a.a. in a tripeptide are coupled, so that coming up with a statistical model to sample tripeptides requires a 6D analysis. Second, whole backbone segments are more complex than mere tripeptides, so that the connexion must be made between these two classes of structural objects. Nevertheless, such a strategy may be particularly applicable to conformational sampling in less-structured systems such as intrinsically disordered proteins (IDPs), which would accompany recent awareness of the need for force field modifications. Finally, from the biophysical standpoint, we show that the potential energies associated with dihedral angles, electrostatic and van der Waals interactions incur changes of increasing magnitude, in this order. Non bonded distances are not considered in TLC and get impacted more significantly, the importance of changes depending on the weighting of the interatomic distance (via the distance exponent). Fully assessing the relevance of these solutions requires further work however. While local steric clashes may arise from the tripeptide geometry provided by solutions of TLC, such clashes may be palliated by performing a local repacking, or by minimizing the overall potential energy, as classically done in methods such as basin-hopping.

Overall, our work furthers the understanding of tripeptide geometries and their link to reconstructions yielded by the tripeptide loop closure. From the software standpoint, we anticipate that our robust open source implementation, available in the Structural Bioinformatics Library, will ease the use of TLC in various structural modeling projects in general, and the generation of conformation of flexible loops in particular.

4.5 Artwork

4.6 SI: Methods

4.6.1 Material: loops and tripeptides from the PDB

- Table S4.1

Figure 4.1 Tripeptide: atoms and degrees of freedom used for loop closure. **(A)** Classical tripeptide loop closure(TLC): the six dihedral angles represented correspond to the degrees of freedom used to solve the problem. N_1, $C_{\alpha;1}$, $C_{\alpha;3}$ and C_3 are constraints and do not move during loop closure. In between $C_{\alpha;1}$ and $C_{\alpha;3}$ 6 bond length, 7 bond angles and two ω dihedral angles are fixed. The algorithm has these 15 parameters and the anchor positions as constraints. **(B)** In tripeptide loop closure with gaps(TLCG), the dihedral degrees of freedom τ_i may be separated from each other by gaps.

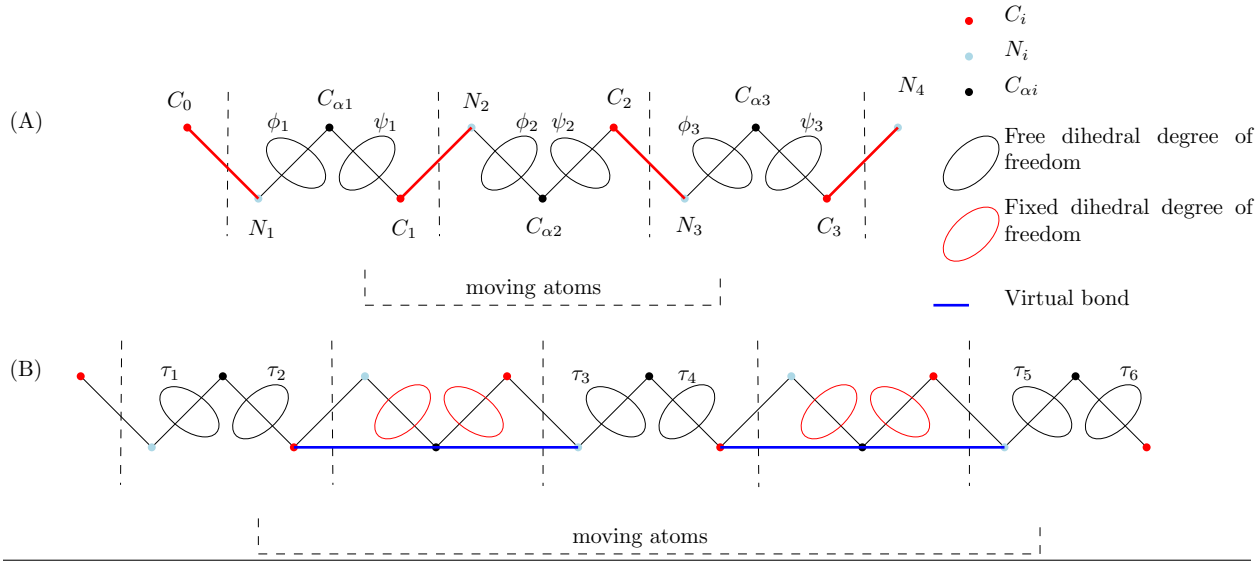


Table 4.1 Amino acid composition of tripeptides. **(A)** Percentage of tripeptides containing the indicated amino acid at least once. **(B)** Percentage of tripeptides containing an amino acid at least twice.

Amino Acid	Percentage of Tripeptides	Amino Acid	Percentage of Tripeptides
CYS	3.77	CYS	0.06
ASP	21.62	ASP	1.55
SER	19.34	SER	1.55
GLN	10.12	GLN	0.36
LYS	16.99	LYS	1.05
ILE	10.29	ILE	0.20
PRO	24.76	PRO	1.59
THR	16.42	THR	1.01
PHE	9.53	PHE	0.29
ASN	15.22	ASN	0.85
GLY	29.53	GLY	3.07
HIS	7.70	HIS	0.25
LEU	18.87	LEU	1.11
ARG	13.69	ARG	0.60
TRP	3.785	TRP	0.04
ALA	17.73	ALA	1.30
VAL	12.53	VAL	0.40
GLU	17.03	GLU	1.17
TYR	9.00	TYR	0.30
MET	4.27	MET	0.06
	(A)		(B)

Figure 4.2 Minimum RMSD between the reconstruction geometrically most similar (RMSD in Å) to the associated data tripeptide. (A) `TLCCoutsias` **(B)** `TLCdouble` **(C)** `TLCdouble`[-x2] – twice precision in mantissa **(D)** `TLCdouble`[-x4] – quadrice precision in mantissa. The logarithmic scale is defined between the smallest bin with a value greater than zero, and the maximum. The minimum of this scale is placed slightly above the intersection of the two axes, and empty bins are not represented.

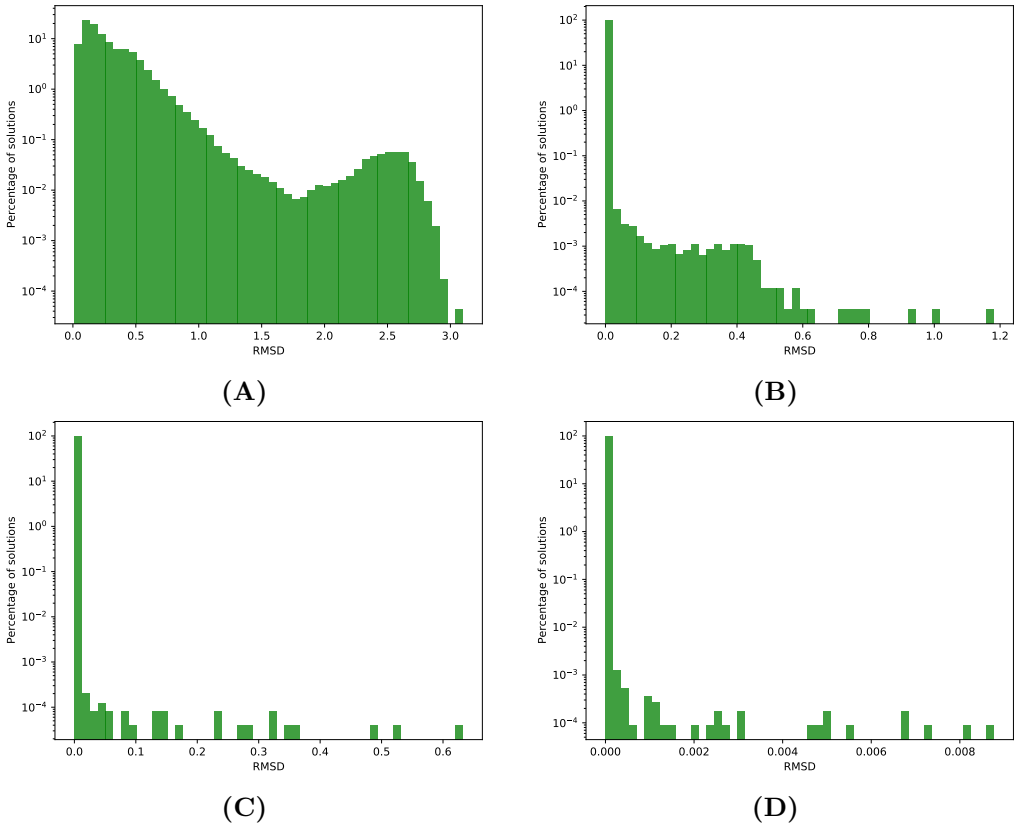
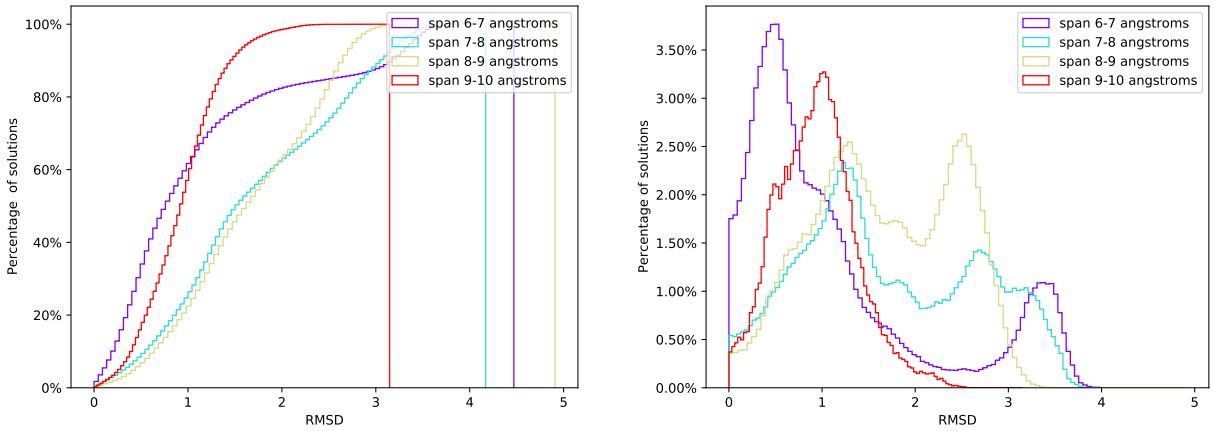


Figure 4.3 Solutions yielded by `TLCdouble[-x2]`: maximum RMSD between each set of reconstructions and the original data. Upon solving $TLC(l)$ for a tripeptide l, the solution most dissimilar to l in the RMSD sense is sought in the solutions set $Sol(l) = \{r_1, \ldots, r_k\}$. **(Left)** Cumulative histogram of this maximum RMSD. **(Right)** Regular histogram of the same.



left **right**

4.6.2 The TLC geometric model

- Fig. S4.6
- Fig. S4.7

- Fig. S4.8
- Fig. S4.9
- Fig. S4.10

4.6.3 Statistical analysis

Ramachandran distributions and their difference. For a given a.a. found at position $i = 1, 2, 3$ in a tripeptide, we consider the Ramachandran distribution in spaces $\mathcal{R}^{aa}_{\mathcal{D},i}$ and $\mathcal{R}^{aa}_{\overline{\mathcal{D}},i}$ respectively. Furthermore, we define the difference between these distributions in the two dimensional space defined by the signed differences $\Delta\phi_i$ and $\Delta\psi_i$, as follows:

$$\begin{cases} \phi_i, \overline{\phi}_i & , \psi_i, \overline{\psi}_i \text{ all } \in (-180, 180) \\ \Delta\phi_i & = \phi_i - \overline{\phi}_i \text{ adding -360 or +360 to keep } \Delta\phi_i \in (-180, 180) \\ \Delta\psi_i & = \psi_i - \overline{\psi}_i \text{ adding -360 or +360 to keep } \Delta\phi_i \in (-180, 180) \end{cases} \tag{4.10}$$

4.6.4 Biophysical analysis

Pairs of atoms contributing to the non bonded terms in eq. 4.8 and 4.9: All atoms pairs containing at least one impacted embedding of a heavy atom.

- All pairs containing C1

- All pairs containing O1
- All pairs containing N2
- All pairs containing CA2
- All pairs containing CB2
- All pairs containing C2
- All pairs containing O2
- All pairs containing N3

Any dihedral containing at least one of the atoms above is considered as contributing to the potential energy term relative to the impacted dihedral angles.

Table 4.2 Table of ΔV_* in kcal/mol.

Reference	Energy term	Figure	1rst quantile	median	third quantile
	$V_{dihedral}$	Fig. 4.5(A)	-0.206	0.068	0.575
	$V_{elec.}$	Fig. 4.5(B)	76.55	89.9847	102.24
$nn^{Class}_{\mathcal{A}_\mathcal{D}}(x)$	V_{vdW}	Fig. 4.5(C)	17661	25683	34116
	$V_{dihedral}$	Fig. 4.5(D)	-0.24	0.14	0.71
	$V_{elec.}$	Fig. 4.5(E)	81.07	92.89	104.88
x^{-1}	V_{vdW}	Fig. 4.5(F)	22615	24704	26874

Table 4.3 Table of $\Delta_r V_*$ ratios.

Reference	Energy term	Figure	1rst quantile	median	third quantile
	$V_{dihedral}$	Fig. 4.5(A)	-0.054	0.018	0.161
	$V_{elec.}$	Fig. 4.5(B)	2.739	4.016	5.953
$nn^{Class}_{\mathcal{A}_\mathcal{D}}(x)$	V_{vdW}	Fig. 4.5(C)	0.111	0.166	0.228
	$V_{dihedral}$	Fig. 4.5(D)	-0.065	0.041	0.202
	$V_{elec.}$	Fig. 4.5(E)	3.576	4.945	7.293
x^{-1}	V_{vdW}	Fig. 4.5(F)	0.145	0.159	0.174

Figure 4.4 Ramachandran distributions for ASP, GLY, and PRO. (Left column) Distributions for domains $\mathcal{R}_{\mathcal{D},2}$ **(Right column)** Distributions for domains $\mathcal{R}_{\overline{\mathcal{D}},2}$, with the superimposed Ramachandran template.

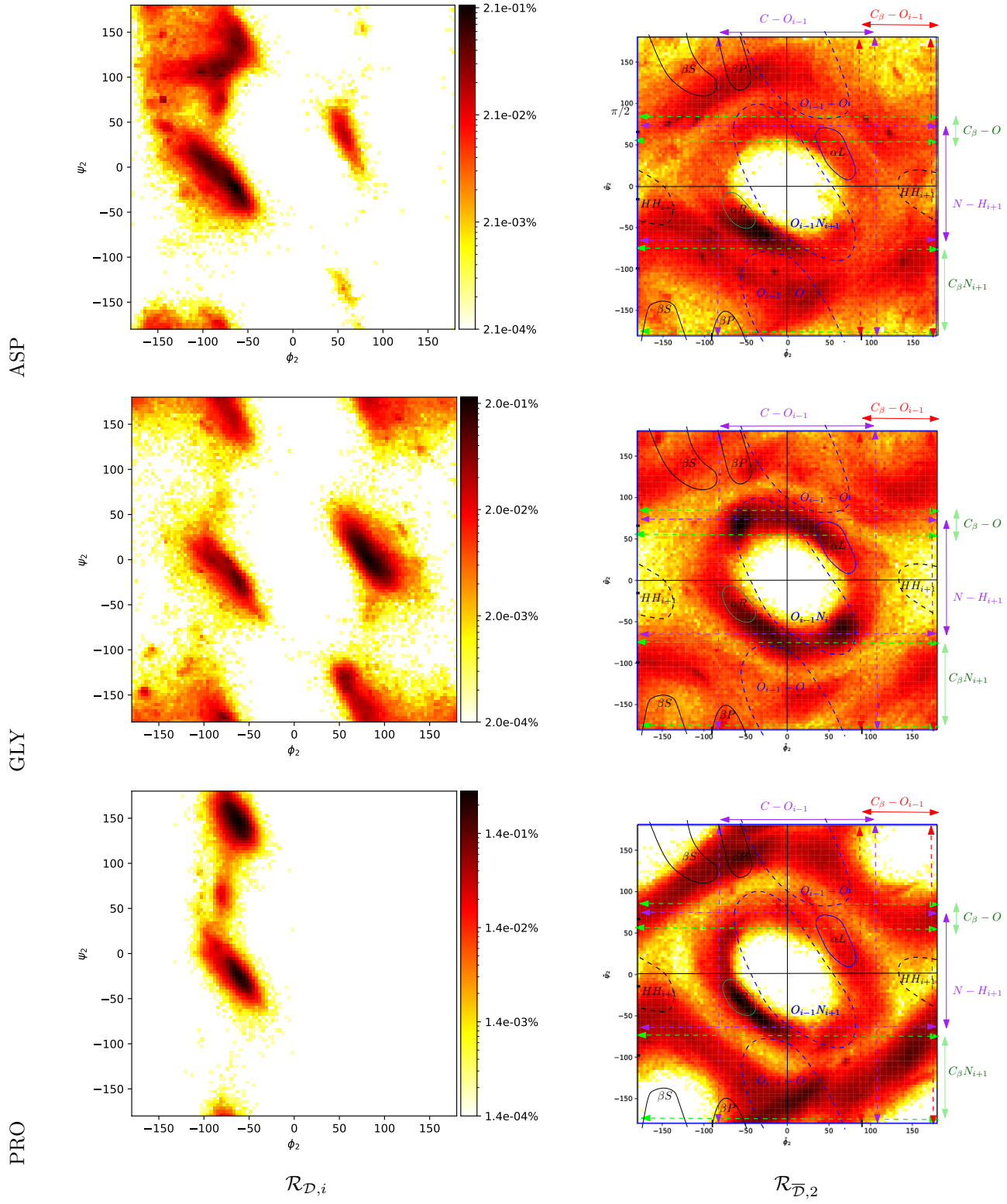


Figure 4.5 Relative changes of the potential energy: reconstructions in $\mathcal{A}_{\overline{\mathcal{D}}}$ versus a reference tripeptide, for all tripeptides of class ASP (i.e., without GLY and featuring a C_β at each position). Calculations involve all backbone heavy atoms, including the carbonyl oxygen and the $C\beta$. The y-coordinate is the sum of angular distances to the match used (L1 norm, Eq. 4.4). The color depends logarithmically on the percentage of all solutions in a bin. **(Top row)** Reference tripeptide for a reconstruction $x \in \mathcal{A}_{\overline{\mathcal{D}}}$ is the nearest neighbor of the same class $nn^{Class}_{\mathcal{A}_\mathcal{D}}(x)$. **(Bottom row)** Reference tripeptide DataTripeptide(x) for a reconstruction $x \in \mathcal{A}_{\overline{\mathcal{D}}}$ **(First column)** Potential energy of dihedral angles. **(Second column)**Electrostatic term, involving all pairs of atoms whose relative distance changes. **(Third column)** van der Waals term, involving all pairs of atoms whose relative distance changes.

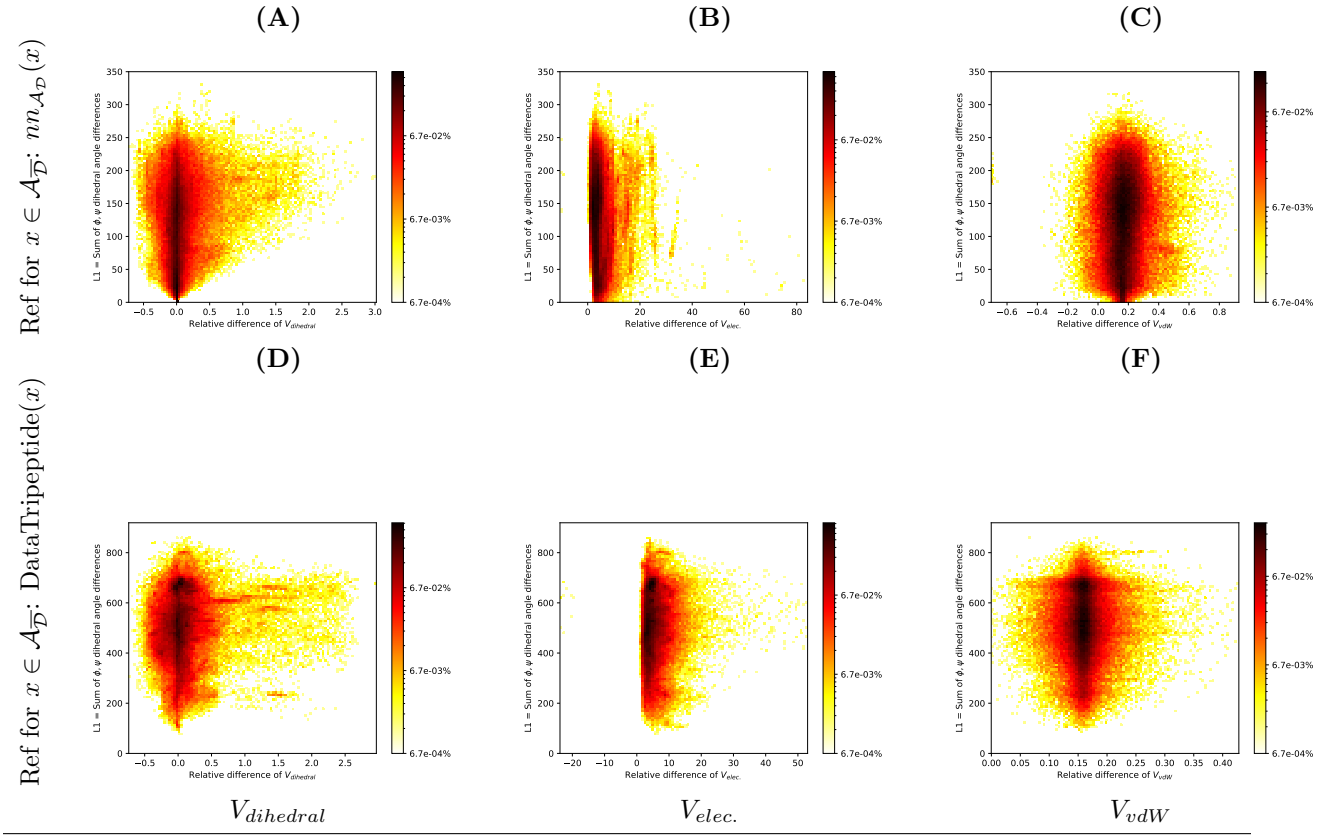


Figure 4.6 TLC: example reconstructions.

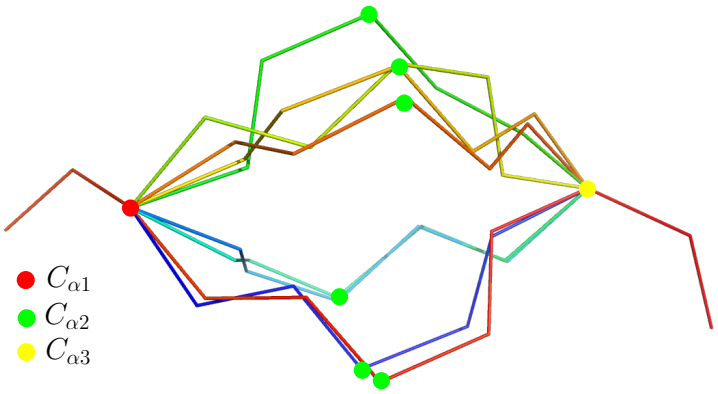


Figure 4.7 TLCG: example reconstructions sandwiching a beta sheet. PDBID 1vfb, chain C. The three amino acid defining the tripeptide are: $C_{\alpha;1}$ (resid: 41 GLN), green $C_{\alpha;2}$ (resid: 42 ALA), yellow $C_{\alpha;3}$ (resid: 54 GLY). A total of six reconstructions were obtained with `TLCdouble`[-x2]. Four are displayed for the sake of clarity. The blue one represents the original geometry.

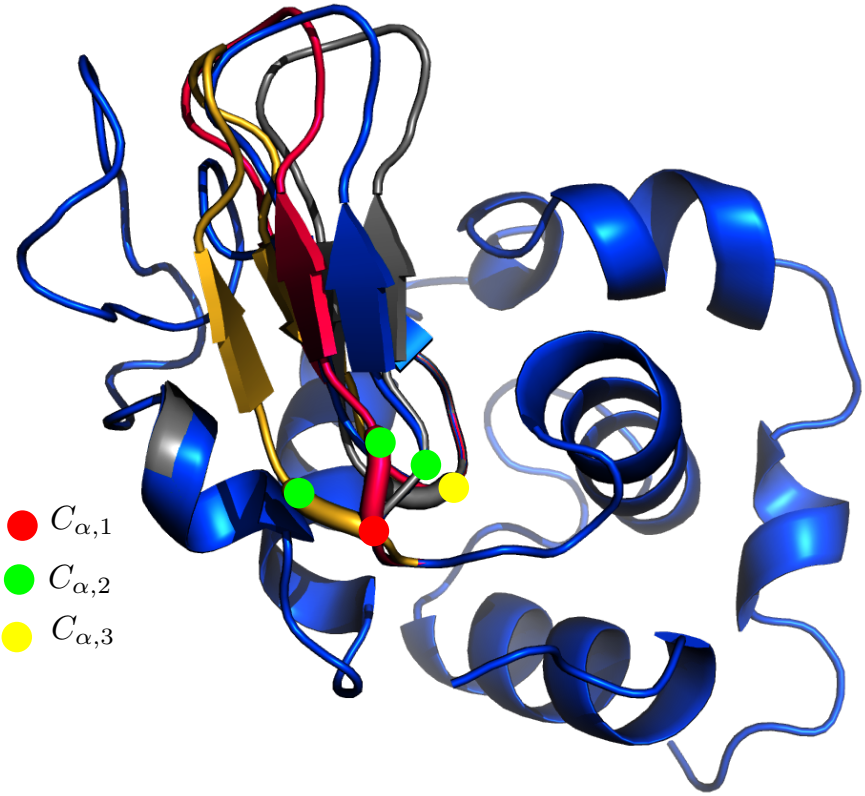
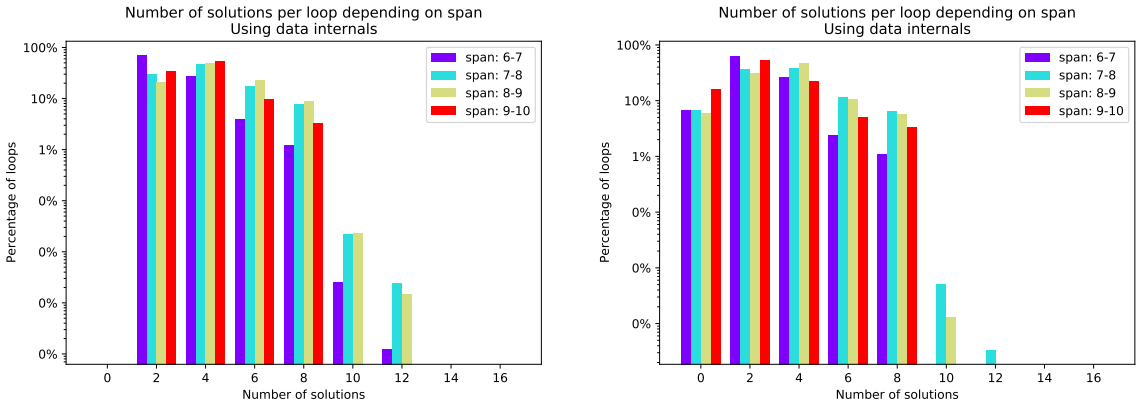


Figure 4.8 Number of solutions for all TLC problems in our database $\mathcal{D}$. (Left) Fixed internals (bond lengths, valence angles) from the data **(Right)** Canonical values for these internal coordinates, from [CSJD04].



Data internals | Canonical values for fixed internals

Figure 4.9 Distribution of displacement for the five moving atoms. Solving a TLC results in five moving atoms (Fig. 4.1). For all displaced atoms in the loop closure generated solutions this is the distribution of the displacement in Angstroms when compared to the original data used to formulate the loop closure.

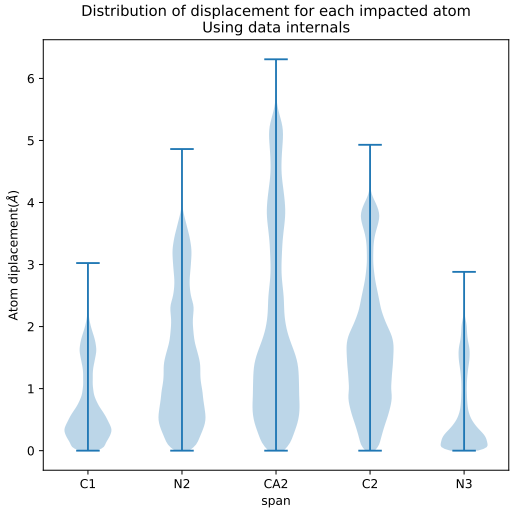
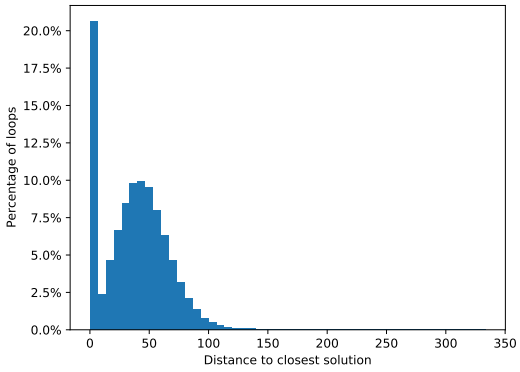
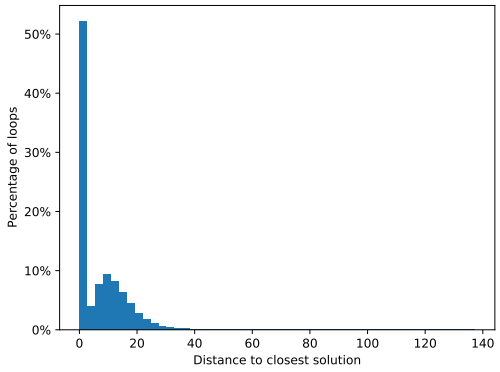


Figure 4.10 Distances to nearest neighbors, see Eq. 4.7, in degrees.



$d_p^{(\mathcal{A}_\mathcal{D})}(x), x \in \mathcal{A}_{\overline{\mathcal{D}}}$



$d_p^{(\mathcal{A}_{\overline{\mathcal{D}}})}(x), x \in \mathcal{A}_\mathcal{D}$

Figure 4.11 Amino acid: ASP. **(Left column)** Distributions in Ramachandran domains $\mathcal{R}_{\mathcal{D},i}, i = 1, 2, 3$ **(Middle column)** Distributions in Ramachandran domains $\mathcal{R}_{\overline{\mathcal{D}},i}, i = 1, 2, 3$

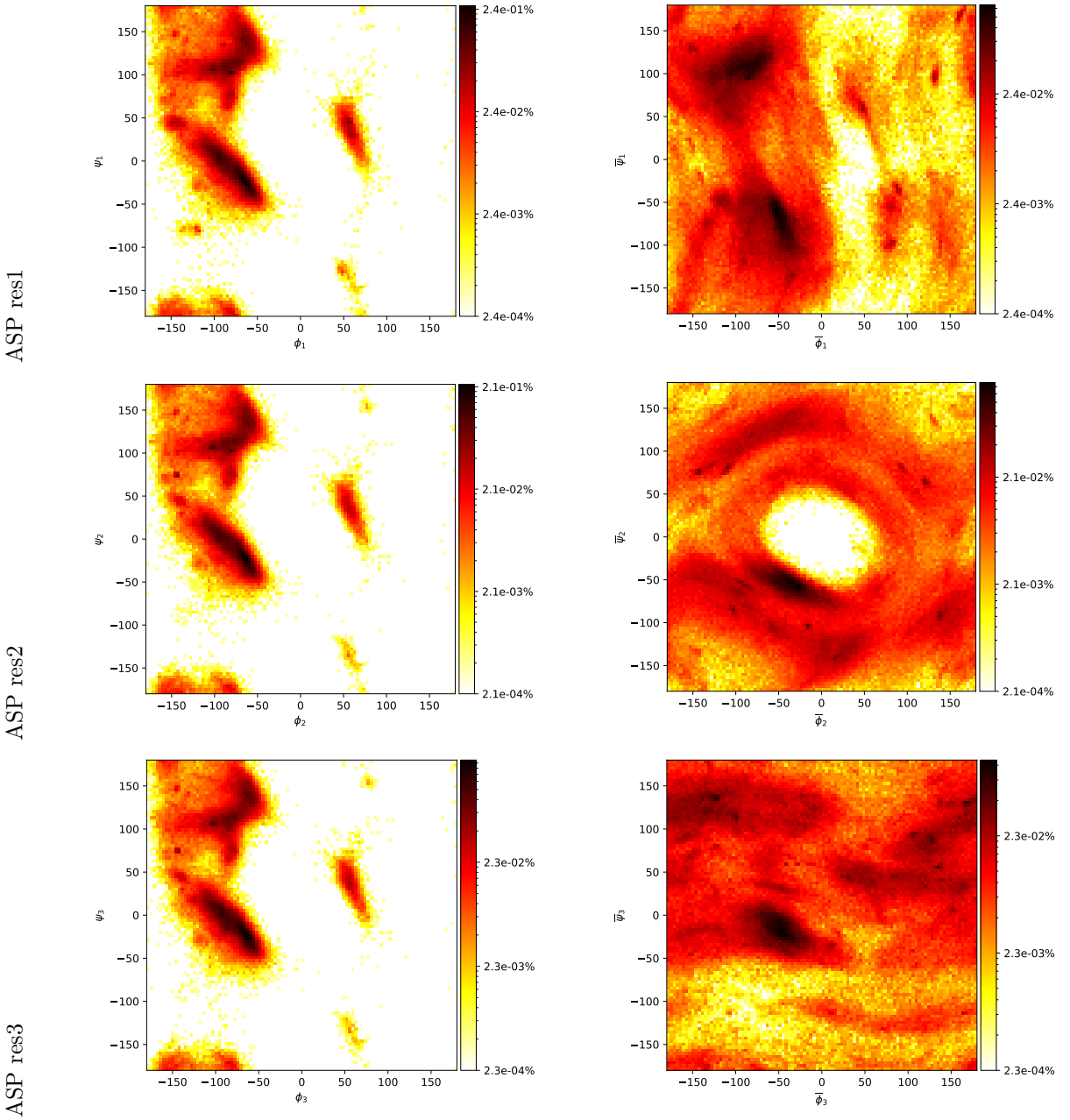


Figure 4.12 Amino acid: GLY. (Left column) Distributions in Ramachandran domains $\mathcal{R}_{\mathcal{D},i}, i = 1, 2, 3$ **(Middle column)** Distributions in Ramachandran domains $\mathcal{R}_{\overline{\mathcal{D}},i}, i = 1, 2, 3$

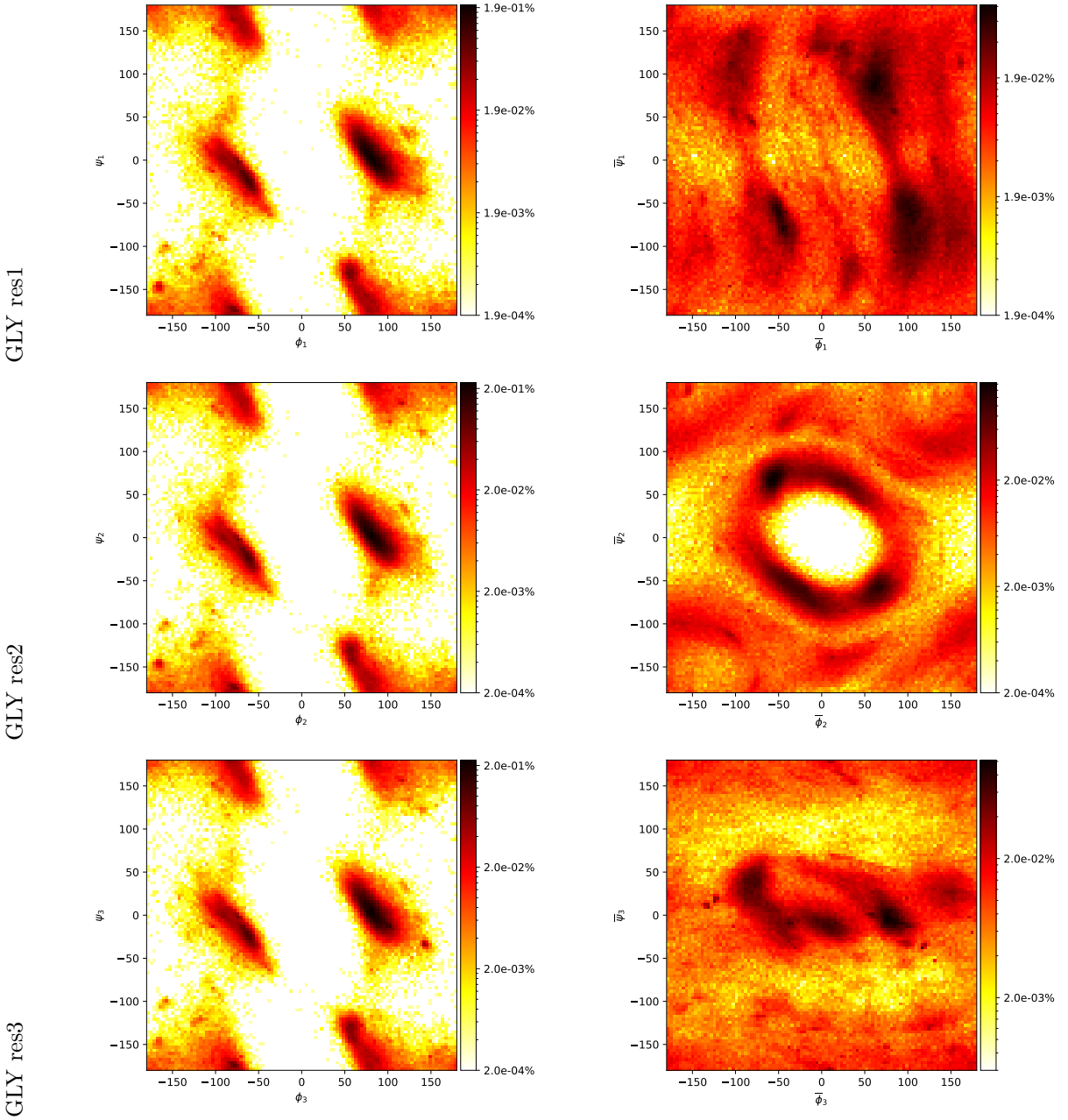


Figure 4.13 Amino acid: PRO. **(Left column)** Distributions in Ramachandran domains $\mathcal{R}_{\mathcal{D},i}, i = 1, 2, 3$ **(Middle column)** Distributions in Ramachandran domains $\mathcal{R}_{\overline{\mathcal{D}},i}, i = 1, 2, 3$

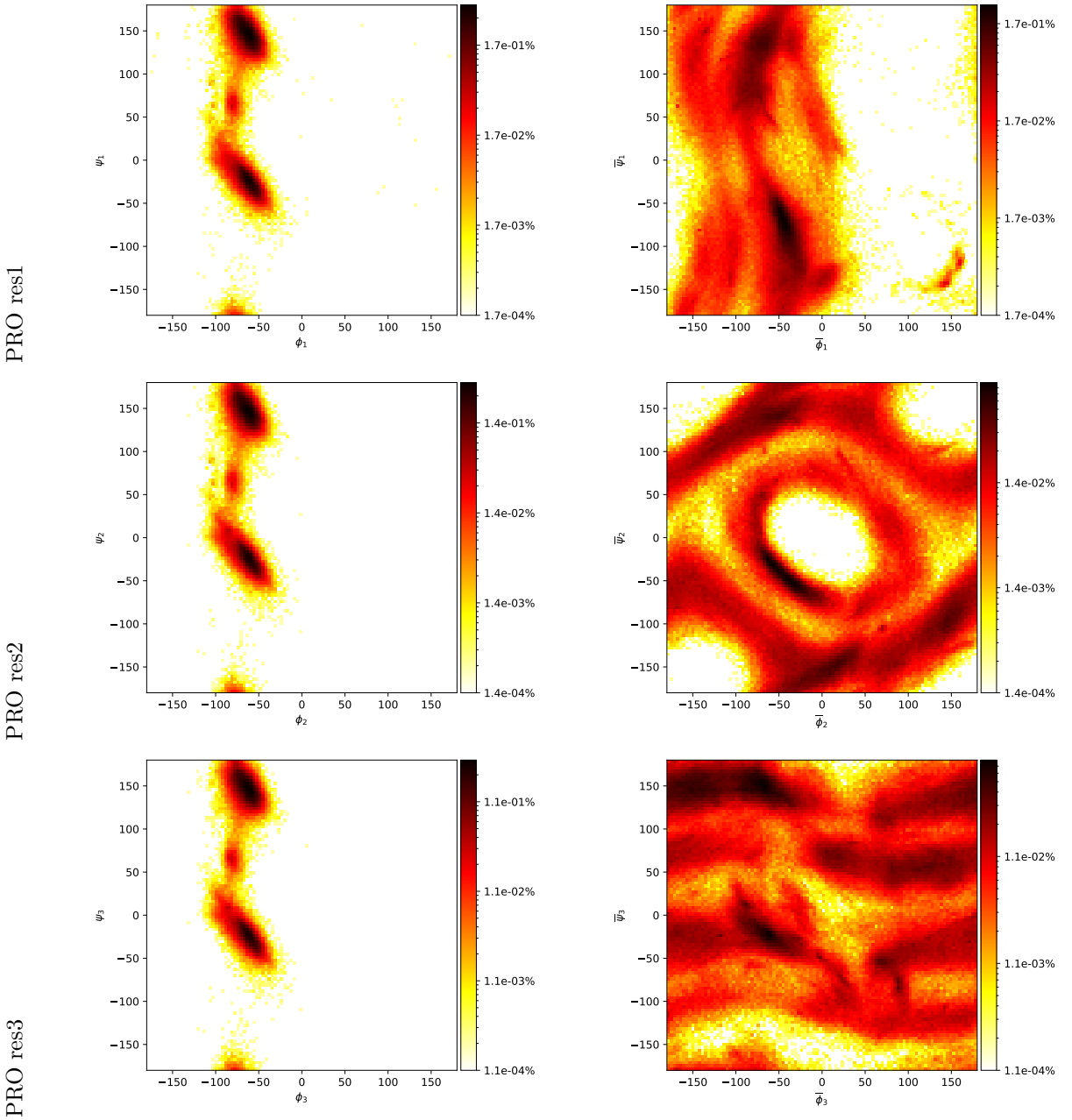


Figure 4.14 Data versus reconstructions: amino acid ASP. (Left column: density difference map) Difference in bin population between the two figures on the same line in (Fig. S4.11). Two color maps are used for the sake of clarity: the (blue to purple) (resp. (yellow to black)) map is used when data (resp. reconstructions) have a higher relative population. **(Right column: difference correlation map)** Oriented angular distance between $\mathcal{R}_{\mathcal{D},i}, i = 1, 2, 3$ and each of their corresponding reconstructions $\mathcal{R}_{\overline{\mathcal{D}},i}, i = 1, 2, 3$ (Fig. S4.11).

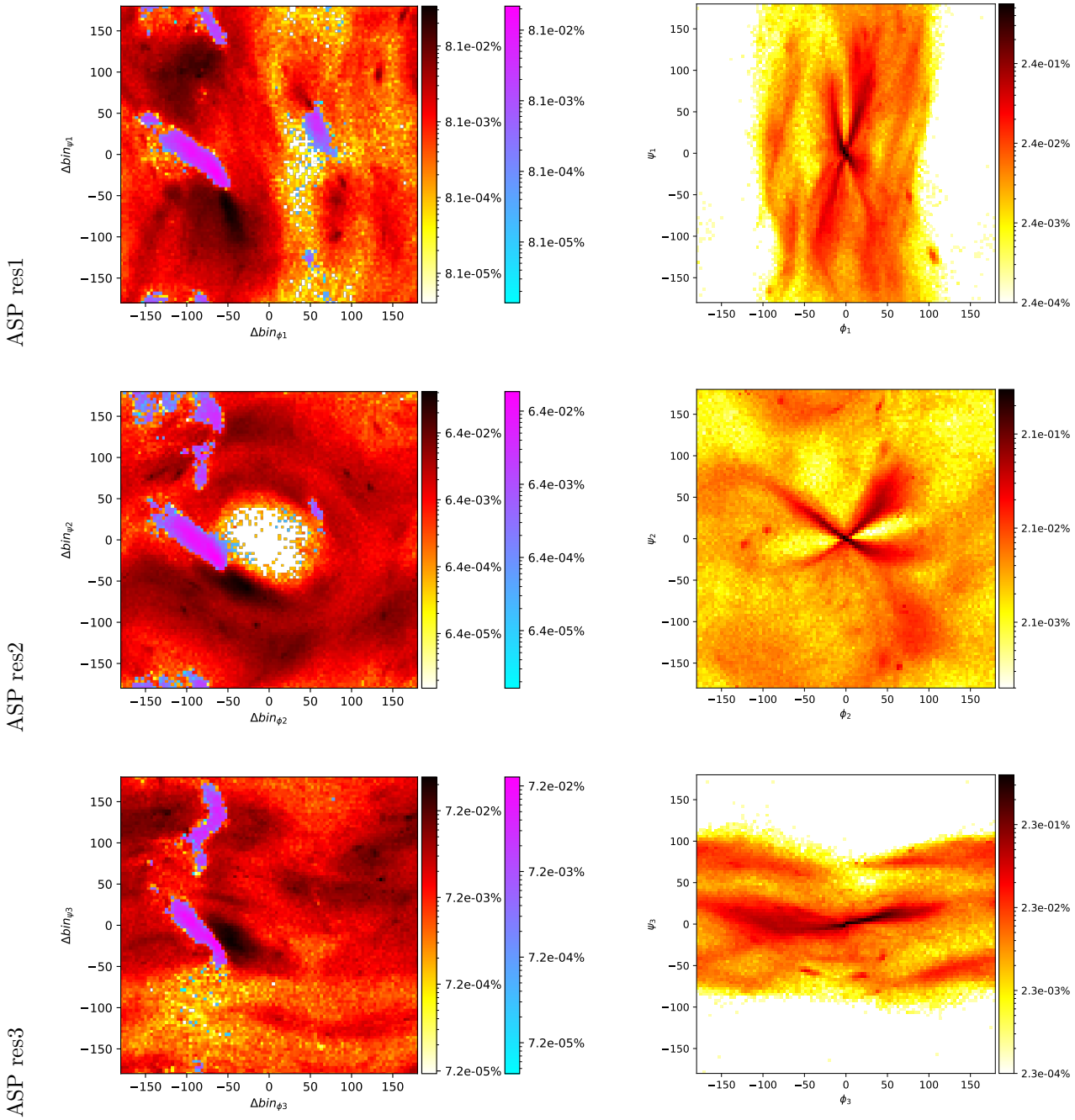


Figure 4.15 Data versus reconstructions: amino acid GLY. **(Left column: density difference map)** Difference in bin population between the two figures on the same line in (Fig. S4.12). Two color maps are used for the sake of clarity: the (blue to purple) (resp. (yellow to black)) map is used when data (resp. reconstructions) have a higher relative population. **(Right column: difference correlation map)** Oriented angular distance between $\mathcal{R}_{\mathcal{D},i}, i = 1, 2, 3$ and each of their corresponding reconstructions $\mathcal{R}_{\overline{\mathcal{D}},i}, i = 1, 2, 3$ (Fig. S4.12).

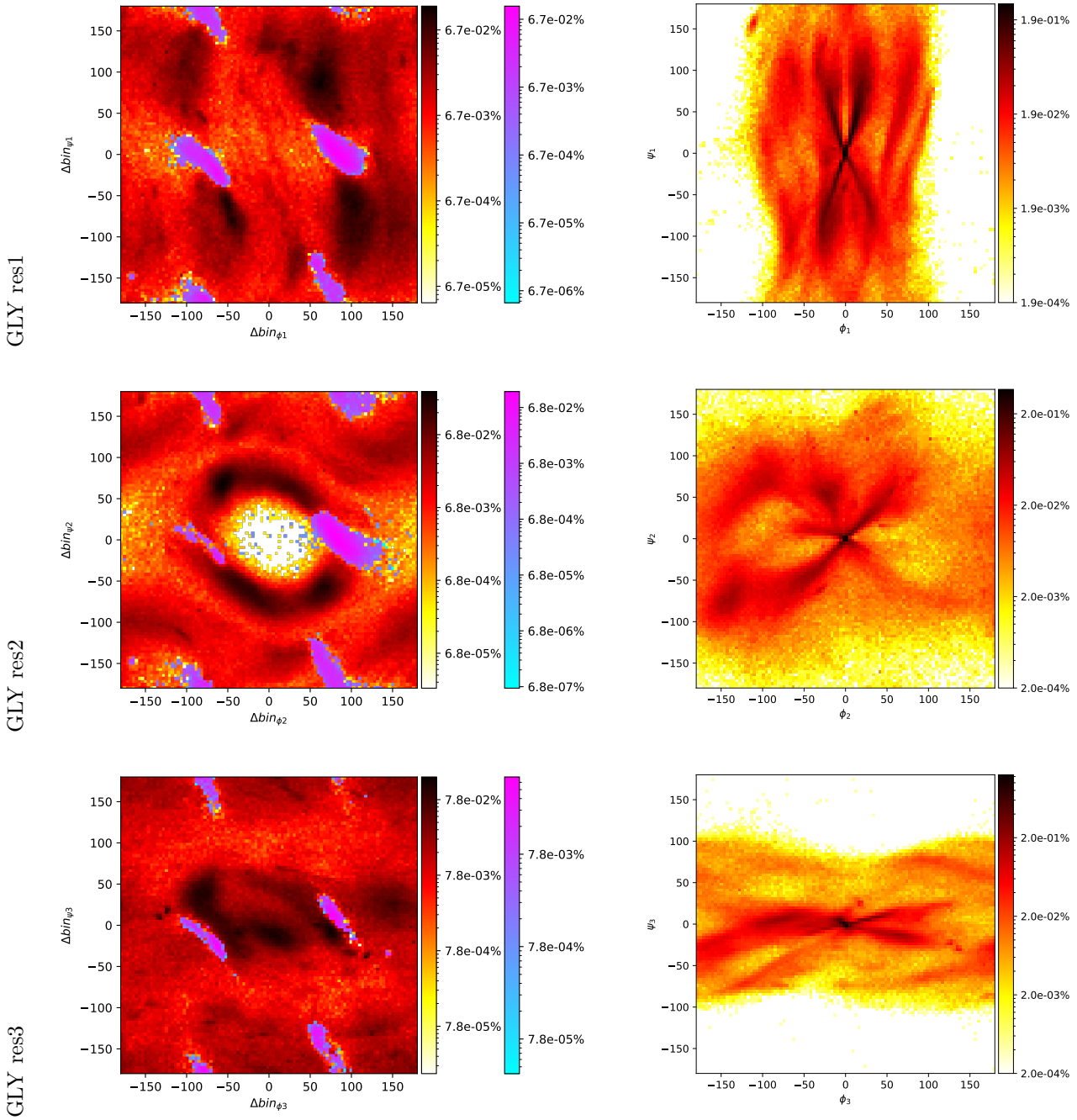
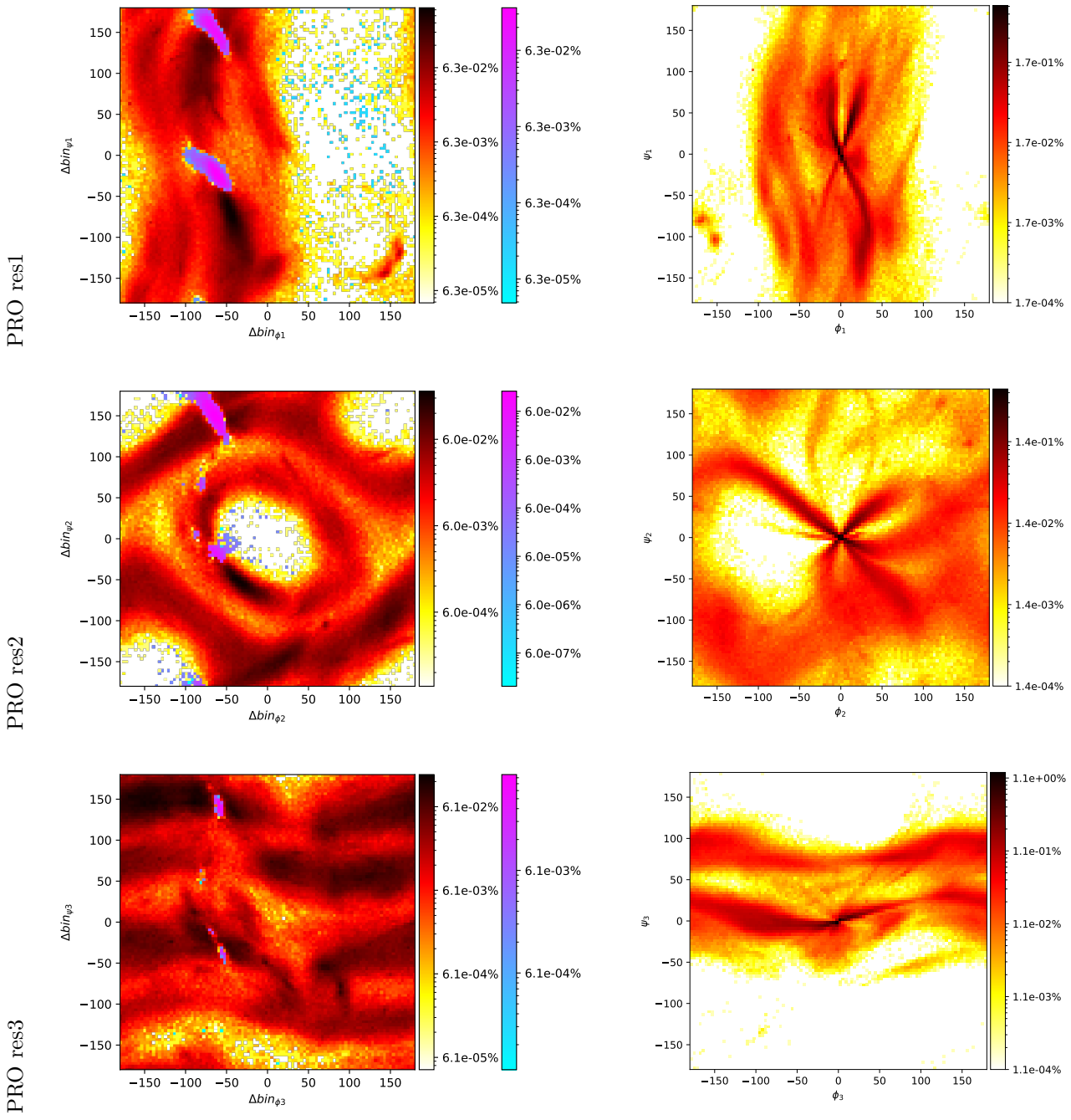


Figure 4.16 Data versus reconstructions: amino acid PRO. **(Left column: density difference map)** Difference in bin population between the two figures on the same line in (Fig. S4.13). Two color maps are used for the sake of clarity: the (blue to purple) (resp. (yellow to black)) map is used when data (resp. reconstructions) have a higher relative population. **(Right column: difference correlation map)** Oriented angular distance between $\mathcal{R}_{\mathcal{D},i}, i = 1, 2, 3$ and each of their corresponding reconstructions $\mathcal{R}_{\overline{\mathcal{D}},i}, i = 1, 2, 3$ (Fig. S4.13).



Chapter 5

Enhanced conformational exploration of protein loops using a global parameterization of the backbone geometry

5.1 Introduction

In this chapter we put forward a global continuous parameterization of the conformational space of a given loop. This parameterization is based on considering the loop as a series of tripeptides seperated by peptide bodies (the four atoms $C_\alpha - C - N - C_\alpha$). This chapter is organized as follows:

- Sec. 5.2 provides a high-level description of the method
- Sec. 5.3 introduces (mandatory) background material
- Sec. 5.4 details the algorithms
- Sec. 5.5 present experiments
- Finally, Sec. 5.6 discusses future work

Nb: Section S5.7.1 contains a compendium of the main notations used throughout the paper.

5.2 Algorithm overview

5.2.1 Geometric model and ingredients

We consider a loop L consisting of $M = 3 \times m$ amino acids, including one or two a.a. on the boundary of the loop if necessary to obtain a multiple of three. We work in the rigid geometry model [EH91], in which bond lengths, valence angles, and peptide bond dihedral angle are fixed. In this model, the internal geometry of each tripeptide is defined by 12 angles [CSJD04], whence an overall angular configuration space $\mathcal{A}$ of dimension $12m$ for the m tripeptides. (As we shall see later, this model can be relaxed, see Rmk. 5.6.)

Our algorithm uses a strategy similar to Hit-and-Run (HAR) [BBR$^+$87] to sample a region $\mathcal{V} \subset \mathcal{A}$ (Fig. 5.1). The region $\mathcal{V}$ defines necessary conditions for the m TLC problems to admit solutions, and intersections with the hyper-surfaces bounding it are used to generate configurations of the whole loop. Individual solutions to the m TLC problems are then obtained in a region $\mathcal{S} \subset \mathcal{V}$. The Cartesian product of solutions for the m tripeptides defines the new conformations of the loop L.

We now introduce the ingredients in turn.

Geometric model. The four atoms making up the peptide bond $(C_{\alpha;1}, C_1, N_2, C_{\alpha;2})$ form a rigid body termed the *peptide body* (Fig. S5.9). For the sake of exposure, we call the two segments $C_{\alpha;1} - C_1$ and $N_2 - C_{\alpha;2}$ the *legs* of the tripeptide, and the tripeptide minus its legs the *tripeptide core*. We model the loop as a sequence of peptide bodies P_k connecting tripeptides cores $T_k^{'}$ (Fig. 5.2):

$$L = P_0 \; T_1^{'} \; P_1 \; \ldots \; P_{k-1} \; T_k^{'} \; P_k \; \ldots \; P_{m-1} T_m^{'} P_m. \tag{5.1}$$

(Nb: strictly speaking, P_0 and P_m contain each two atoms of the loop L.) The main idea to generate conformations of L is to sample the positions of peptide bodies independently using rigid motions, and then, to solve individual TLC problems. To describe this strategy more precisely, the following ingredients are needed.

Tripeptide loop closure. Tripeptide Loop Closure is a method computing all possible valid geometries of a tripeptide, under two types of constraints. First, the first and last two atoms of the tripeptide, *i.e.* its legs, are fixed. Second, all internal coordinates are fixed, except the six (ϕ, ψ) dihedral angles of the three C_α carbons.

TLC admits at most 16 solutions corresponding to the real roots of a degree 16 polynomial. These solutions have been shown to be geometrically diverse (atoms are moving up to 5Å), and low potential energy [ORC22]. Solving TLC can be done using three rigid bodies associated with the three edges of the triangle involving the three C_α carbons. The rotations of these rigid bodies are described by three angles τ_1, τ_2, τ_3, two of which can be eliminated to yield the degree 16 polynomial. The coefficients of this polynomial depends on $3 \times 4 = 12$ angles describing the internal geometry of the tripeptide [CSJD04]. This 12 dimensional space is denoted $\mathcal{A}_k$ for the tripeptide T_k. Taking the Cartesian product of the individual angular spaces of the m tripeptides yields a $12m$ dimensional space denoted $\mathcal{A}$.

Necessary conditions for TLC to admit solutions. In the angular space $\mathcal{A}_k$, we have recently exhibited a region $\mathcal{V}_k$ defining necessary conditions for TLC to admit solutions [OC22a]. For a given tripeptide, this region is defined from 24 implicit equations involving the 12 variables parameterizing TLC. The corresponding space for all tripeptides is denoted $\mathcal{V}$. This space contains the solution space $\mathcal{S} \subset \mathcal{V}$, such that each tripeptide admits solutions.

Identifying active constraints with Hit-and-Run. To sample $\mathcal{S}$, we use techniques and ideas coming from geometric optimization. To introduce them, recall that a linear program consists in finding the minimum of a linear functional under linear constraints defining a (high dimensional) polytope. The hyperplanes contributing to the definition of the polytope are termed active, and the remaining ones redundant. To identify the latter, the Hit-and-Run (HAR) algorithm was invented long ago [BBR$^+$87]. In a nutshell, given a starting point inside the polytope, HAR iteratively proceeds as follows: shoot a random ray inside the polytope and identify the nearest hyperplane intersected; generate a point onto the segment defined by the starting and the intersection point; then iterate. Since then, this algorithm has been modified to generate points following a Gaussian distribution, a key step in the computation of the volume of polytopes [CV16]. Other random walks serving similar purposes are billiard walk and Hamiltonian Monte Carlo [LV18, CPC22], as well as walks based on piecewise deterministic processes [CCF22]. Such methods play a key role in our algorithm too.

5.2.2 Algorithm: wrapping up

Similarly to HAR, our algorithm consists of consecutive steps. Each step generates a conformation L' of the loop L by moving the peptide bodies. Given the internal coordinates of L', we solve TLC for each individual tripeptide, and take the Cartesian product of these solutions–if any.

To see how the conformation L' is generated, let $SE(3)$ be the special Euclidean group representing rigid motions (translation+rotation) in 3D. The $m-1$ peptide bodies being rigid bodies, we move them in 3D space using rigid motions parameterized over the motion space $\mathcal{M} = (SE(3))^{m-1}$. We consider a (random) ray in $\mathcal{M}$, whose parameter t is called the *time*. Every point on this ray defines a rigid motion applied to each peptide body. Since the tripeptide legs are moving due to this motion, the 12 angular coordinates of each tripeptide become time dependent. We use the image of the ray in the angle space $\mathcal{A}$ to find intersections with the hyper-surfaces defining the aforementioned necessary conditions (Fig. 5.1). In a manner similar to HAR, these intersections are used to generate a point in the solution space $\mathcal{S}$. Each such point encodes an internal geometry for each tripeptide, so that TLC can be solved for each individual tripeptide. As said above, the solutions to the individual TLC problems are then combined. The ability to generate efficiently points in $\mathcal{S}$ depends on the stringency of necessary conditions defining $\mathcal{V}$, that is to say on the volume of the region $\mathcal{S}\backslash\mathcal{V}$.

We now detail these ingredients.

5.3 Background and notations for peptides and TLC

5.3.1 Peptides and tripeptides

Peptides, peptide bonds, tripeptides, and protein loops. Atoms within a tripeptide are denoted as $C_{\alpha;3k-2}, C_{\alpha;3k-1}, C_{\alpha;3k}$, and likewise for the C and N atoms (Fig. 5.2). Note that with these notations, one has $A_{4k-3} = N_{3k-2}$, $A_{4k-2} = C_{\alpha;3k-2}$, $A_{4k-1} = C_{\alpha;3k}$ and $A_{4k} = C_{3k}$.

As noticed above, the two segments $N_{3k-2}C_{\alpha;3k-2}$ and $C_{\alpha;3k}C_{3k}$ form *legs* of the tripeptide, while the tripeptide minus its legs form the *tripeptide core* $T_k^{'}$. Note that for two consecutive tripeptides, the second leg of T_k and the first one of T_{k+1} form the peptide bond. Note also that in the decomposition of Eq. 5.1, $P_0 = A_1A_2$ and $P_m = A_{4m-1}A_{4m}$ play a special role: these two fixed segments are called *anchors*.

5.3.2 Tripeptide loop closure (TLC) with fixed legs

TLC uses constraints on the tripeptide legs and internal coordinates (See Sec. 5.2). We may also recall that TLC induces a partition of the nine atoms in the tripeptide T_k into two classes. On the one hand, the first two and the last two atoms, *i.e.* the legs, are fixed. On the other hand, the remaining five middle atoms are moving. When considering all solutions of TLC on an exhaustive database of tripeptides extracted from the PDB, these atoms move up to 5Å[ORC22].

Solutions of TLC [CSJD04] rely on the following observations (Fig. 5.3(A,B) [1]):

- TLC involves three rigid bodies: the first two involve the five atoms in-between the first and third C_α carbons; the third one consists of the four atoms defining the legs of the tripeptide.
- The solution space of TLC can be modeled using rotation angles denoted $\{\sigma_{k,i}, \tau_{k,i}\}$ associated to the three rigid bodies. (Nb: the two angles associated with the $C_{\alpha;k,i}$ carbon are $\sigma_{k,i-1}$ and $\tau_{k,i}$.) Positions of the rigid bodies must respect the valence angles θ_i at the three C_α carbons. The rotation of a rigid body about its $C_\alpha - C_\alpha$ axis only impacts the valence angle constraints at its endpoints.
- Searching for solutions to the loop closure is akin to searching for rotation combinations of the angles $\{\sigma_{k,i}, \tau_{k,i}\}$ respectful of θ angles. $\sigma_{k,i-1}$ is the rotation angle of N_i atoms around their corresponding axis. $\tau_{k,i}$ is the rotation angle for C_i around its axis (Fig. 5.3(A,B)).

The geometry of the backbone can be used to define local frames at each C_α carbon ([CSJD04] and Fig. 5.3(B)), based on three vectors: $\hat{\mathbf{Z}}_{\mathbf{k,i}}$ – unit vector along two consecutive C_α carbons, $\hat{\mathbf{r}}^{\tau}_{\mathbf{k,i}}$ – to define the

[1] When talking of individual tripeptides i is used as an index with $i \in \{1, 2, 3\}$. These indices are counted mod 3, that is $i - 1 = i + 2$.

rotation of angle $\tau_{k,i}$, $\hat{\mathbf{r}}^{\sigma}_{\mathbf{k,i}}$ – to define the rotation of angle $\sigma_{k,i}$. Using these local frames, one defines the angles $\alpha_{k,i}, \xi_{k,i}, \eta_{k,i}$, with indices $i = 1, 2, 3$ – counted modulo three, for the tripeptide T_k (Fig. S5.10):

$$\begin{cases} \alpha_{k,i} = \angle \hat{\mathbf{Z}}_{\mathbf{k,i}} \hat{\mathbf{Z}}_{\mathbf{k,i+2}}; & \alpha_{k,i} \in [0, \pi) \\ \xi_{k,i} = \angle - \hat{\mathbf{Z}}_{\mathbf{k,i}} \hat{\mathbf{r}}^{\sigma}_{\mathbf{k,i}}; & \xi_{k,i} \in [0, \pi) \\ \eta_{k,i} = \angle \hat{\mathbf{Z}}_{\mathbf{k,i}} \hat{\mathbf{r}}^{\tau}_{\mathbf{k,i}}; & \eta_{k,i} \in [0, \pi) \\ \delta_{k,i} = \angle C_{k,i} C_{\alpha;k,i}, C_{\alpha;k,i} C_{\alpha;k,i+1}, C_{\alpha;k,i+1} N_{k,i+1} & \delta_{k,i} \in [0, 2\pi) \end{cases} \tag{5.2}$$

Definition. 5.1. *Let* $\mathbf{A}_{k,i} = \{\alpha_{k,i}, \eta_{k,i}, \xi_{k,i-1}, \delta_{k,i-1}\}$ *be the set of angles associated with* $C_{\alpha;i}$ *of the k-th tripeptide* T_k*. The* angular representation *of the tripeptide* T_k *is the 12-tuple* $\mathbf{A}_k = \{\mathbf{A}_{k,1}, \mathbf{A}_{k,2}, \mathbf{A}_{k,3}\}$.

The corresponding 12-dimensional space is denoted $\mathcal{A}_k$.

5.3.3 Tripeptide and necessary constraints for TLC

From now on, we assume that the peptide of interest is the k-th tripeptide in our loop, see Eq. (5.1).

In recent work [OC22a], we have introduced necessary conditions for TLC to admit solutions. For each of the three angles $\tau_{k,i}$, these so-called *depth 1 inter-angular constraints* are based on intervals to which $\tau_{k,i}$ must belong. These intervals, which are parameterized by the angular representation of the peptide, are denoted as follows:

$$\begin{cases} \mathcal{I}_{\tau_{k,i}} = \{I_{\tau_{k,i}}\} \text{ with } I_{\tau_{k,i}} = [I^{\min}_{\tau}(\mathbf{A}_{k,i}), I^{\max}_{\tau}(\mathbf{A}_{k,i})] \\ \mathcal{I}_{\tau_{k,i}|\delta} = \{I_{\tau_{k,i}|\delta}\} \text{ with } I_{\tau_{k,i}|\delta} = [I^{\min}_{\tau|\delta}(\mathbf{A}_{k,i+1}), I^{\max}_{\tau|\delta}(\mathbf{A}_{k,i+1})] \end{cases} \tag{5.3}$$

There are two intervals of each type, and their pairwise intersection results in four so-called *depth one validity intervals* or DOVI. As established in [OC22a], the bounds of these angles depend on the values

$$\arccos \frac{+\cos(\theta_i \pm \xi_{i-1}) + \cos\eta_i \cos\alpha_i}{\sin\eta_i \sin\alpha_i}. \tag{5.4}$$

For a given tripeptide, we may consider the mapping from its angular representation in the angle space $\mathcal{A}_k$ to the validity intervals:

$$\text{DOVI}_{\tau_{k,i}}(\cdot) : \mathcal{A}_k \mapsto (\mathcal{I}_{\tau_{k,i}} \cap \mathcal{I}_{\tau_{k,i}|\delta})^4. \tag{5.5}$$

That is, upon fixing the angular representation of the tripeptide (Def. 5.1), we obtain up to four validity intervals, or the empty set if the four intersections are empty. As reported in the companion paper [OC22a], our necessary conditions are rather tight.

Remark 5.1. *The function* $DOVI_{\tau_{k,i}}$ *is obtained using the interval* $I_{\tau_{k,i}}$ *whose definition requires the angles* $\alpha_{k,i}, \eta_{k,i}, \xi_{k,i-1}$ *for* $I_{\tau_{k,i}}$*, and the interval* $I_{\tau_{k,i}|\delta}$ *whose definition requires the angles* $\alpha_{k,i+1}, \eta_{k,i+1}, \xi_{k,i}, \delta_{k,i}$*. The number of parameters is thus seven. For the sake of conciseness, we use the supersets* $\mathbf{A}_{k,i}$ *and* $\mathbf{A}_{k,i+1}$*. See [OC22a] for details.*

5.4 Algorithm: details

5.4.1 Tripeptides with moving legs

Moving peptides bodies. When considering the decomposition of Eq. (5.1), the $m-1$ peptide bodies move independently. The motion of one peptide body is parameterized by the special Euclidean group $SE(3)$, which combines one translation and one rotation. To be more specific, let S^2 be the sphere of directions in $\mathbb{R}^3$, and A a positive real number. The motion space $\mathcal{R}$ for one peptide body is defined via the motion space

$$\mathcal{R} : (S^2 \times [0, A)) \times (S^2 \times [0, 2\pi)) \subset SE(3). \tag{5.6}$$

The term $S^2 \times [0, A)$ codes the translation defined by a unit vector and a real number in $[0, A)$, while the term $S^2 \times [0, 2\pi)$ codes the rotation defined by an angle about a direction given by a unit vector on S^2. Therefore, specifying a random rigid motion for each peptide body requires $2(m-1)$ unit vectors. We pool these vectors into a $6(m-1)$-dimensional vector denoted V in the sequel. The value of A defines a trade-off between the relative magnitude of the translation and rotation. We use the default value $A = 2\pi$, as we hardly noticed any incidence for this parameter (data not shown). Summarizing, the overall motion space for peptide bodies is the $6(m-1)$ dimensional space:

$$\mathcal{M} = \mathcal{R}^{m-1}. \tag{5.7}$$

Using a 1-parameter family in the motion space. We restrict motions in $\mathcal{M}$ to a a 1-parameter family, performing the following linear interpolation defined by vector V:

$$\text{Ray}(V) = \{\gamma(t) = Id + tV, \text{ with } \gamma(0) = Id\}. \tag{5.8}$$

The restriction of this one parameter family to each peptide body defines a rigid transformation

$$\gamma_k : [0, 1] \mapsto SE(3), \gamma_k(0) = Id, \tag{5.9}$$

such that the position of the k-th peptide body $P_k(t)$ at time t satisfies

$$P_k(t) = \gamma_k(t) P_k(0). \tag{5.10}$$

The full equations for this motion are provided in the supplementary section 5.7.4.

5.4.2 Validity domain and overall configuration space $\mathcal{A}$

We now wish to use the depth one validity constraints for the m peptides, whose legs are moving as just explained. To this end, we concatenate the angular representations of the m tripeptides (Def. 5.1), and define:

Definition. 5.2. *(Angular conformational space $\mathcal{A}$) The* angular conformational space *of the loop L is the $12m$ dimensional space defined by the product of the m angular space of the individual tripeptides:*

$$\mathcal{A} \stackrel{Def}{=} \prod_{k=1}^{m} \mathcal{A}_k. \tag{5.11}$$

Fixing the positions of the peptide bodies in Eq. (5.1) yields the angular representations of the m tripeptides. We therefore define a mapping from the motion space into the global angular space:

$$f_{\mathcal{M} \to \mathcal{A}} : \mathcal{M} \mapsto \mathcal{A} \tag{5.12}$$

Having discussed the depth one validity interval for one tripeptide– see Eq. (5.5), we can finally aggregate such conditions:

Definition. 5.3. *(Angular validity domain $\mathcal{V}$.) The* angular *validity domain $\mathcal{V}_k$ of the angle $\tau_{k,i}$ of the k-th tripeptide is the subset of $\mathcal{A}_k$ such that $DOVI_{\tau_{k,i}}(\cdot) \neq \emptyset$.*

The angular *validity domain of the loop L is the subset $\mathcal{V} \subset \mathcal{A}$ such that*

$$\forall k = 1, \ldots, m, \forall i = 1, \ldots, 3, \forall a \in \mathcal{V} : DOVI_{\tau_{k,i}}(a) \neq \emptyset.$$

Note that there are $3m$ individual angular validity domains since each tripeptide has 3 angles τ.

Points in $\mathcal{V}$ satisfy necessary conditions. However, for a point $p \in \mathcal{V}$, one or several tripeptide may not admit any valid geometry. We therefore define:

Definition. 5.4. *(Solution space $\mathcal{S}$) The* solution space $\mathcal{S} \subset \mathcal{V}$ *of the loop L is the subspace of $\mathcal{A}$ such that TLC admits at least one solution for each tripeptide. A point in $\mathcal{S}$ (resp. $\mathcal{V}\backslash\mathcal{S}$) is termed* fertile *(resp. sterile).*

Let s_k the number of solutions yielded by TLC for a point $p \in \mathcal{S}$. The Cartesian product of these sets yields a total number of embeddings, *i.e.* conformations, equal to $\Pi_{k=1,\ldots,m} s_k$.

Remark 5.2. *Note that the degrees of freedom are defined for rigid bodies in-between tripeptides while the constraints are defined within the tripeptides (Fig. 5.2).*

5.4.3 Kinetic validity intervals

We now wish to use our 1-parameter family of motions to explore the solutions space $\mathcal{S}$ via an exploration of the valid space $\mathcal{V}$.

The tripeptide legs move according to the motion imposed to the peptide bodies (Eq. 5.10). It is therefore possible to define a time dependent (aka kinetic) version of the angles $\mathbf{A}_{k,i}$:

$$\mathbf{A}_{k,i}(t) = (f^{(\alpha)}_{(k,i)}(t), f^{(\xi)}_{(k,i)}(t), f^{(\eta)}_{(k,i)}(t), f^{(\delta)}_{(k,i)}(t)), \tag{5.13}$$

with

$$\begin{cases} f^{(\alpha)}_{(k,i)}(t) & : \text{ function computing the angle } \alpha_{k,i} \text{ at time } t \\ f^{(\xi)}_{(k,i)}(t) & : \text{ function computing the angle } \xi_{k,i} \text{ at time } t \\ f^{(\eta)}_{(k,i)}(t) & : \text{ function computing the angle } \eta_{k,i} \text{ at time } t \\ f^{(\delta)}_{(k,i)}(t) & : \text{ function computing the angle } \delta_{k,i} \text{ at time } t \end{cases} \tag{5.14}$$

Once plugged into the intervals of Eq. (5.3), these functions make it possible to define a kinetic version of the four static validity intervals:

Definition. 5.5. *(Kinetic validity intervals) The* kinetic validity intervals *for a given angle $\tau_{k,i}$ of a tripeptide T_k are the validity intervals obtained for the time varying angles $\mathbf{A}_{k,i}(t)$:*

$$\begin{cases} I_{\tau_{k,i}}(t) = [I^{min}_{\tau}(\mathbf{A}_{k,i}(t)), I^{max}_{\tau}(\mathbf{A}_{k,i}(t))] \\ I_{\tau_{k,i}|\delta}(t) = [I^{min}_{\tau|\delta}(\mathbf{A}_{k,i+1}(t)), I^{max}_{\tau|\delta}(\mathbf{A}_{k,i+1}(t))] \end{cases} \tag{5.15}$$

Remark 5.3. *The time dependent angles are computed as follows (Fig. S5.10):*

- *The fixed internal coordinates within each tripeptide are sufficient to determine the value of $\eta_{k,1}, \xi_{k,1}, \eta_{k,2}$ and $\xi_{k,2}$. (Note that these are defined in the rigid bodies associated with $C_{\alpha;3k-2}C_{\alpha;3k-1}$ or $C_{\alpha;3k-1}C_{\alpha;3k}$.)*
- *The position of the* legs *are sufficient to define $\eta_{k,3}$ and $\xi_{k,3}$.*
- *The leg positions together with the fixed internal coordinates are sufficient to compute all three $\alpha_{k,i}, i \in \{1,2,3\}$ angles as these angles are defined by the C_α triangle.*

Remark 5.4. *The motions of consecutive rigid bodies is constrained by the triangle inequality between the three consecutive C_α atoms (Fig. 5.3). Indeed, these atoms must satisfy the following triangle inequality:*

$$\|C_{\alpha;3k-2}C_{\alpha;3k}\| \leq \|C_{\alpha;3k-2}C_{\alpha;3k-1}\| + \|C_{\alpha;3k-1}C_{\alpha;3k}\|\,. \tag{5.16}$$

Note that following the rigidity of peptide bodies, the two right hand side distances are fixed.

5.4.4 Sampling: one step

Sampling $\mathcal{V}$ with Hit-and-Run. We sample the validity domain $\mathcal{V}$ using the Hit-and-Run algorithm (Fig. 5.1 and [BBR+87]). For a ray $\mathrm{Ray}(V)$ in the motion space (Eq. 5.8), consider the restriction of this ray to the valid space $\mathcal{V}$, that is

$$\mathrm{Ray}_{\mathcal{V}}(V) = \{\gamma(t) \in \mathrm{Ray}(V) \mid f_{\mathcal{M}\to\mathcal{A}}(\gamma(t)) \in \mathcal{V}\}. \tag{5.17}$$

The Hit-and-Run algorithm consists of iteratively sampling a new point on $\mathrm{Ray}_{\mathcal{V}}(V)$, so that the restriction of the ray to the valid space $\mathcal{V}$ must be computed.

To see how, consider two kinetic intervals $I_{\tau_{k,i}}(t) \in \mathcal{I}_{\tau_{k,i}}$ and $I_{\tau_{k,i}|\delta}(t) \in \mathcal{I}_{\tau_{k,i}|\delta}$ as specified in Eq. (5.15). For these intervals, consider the limit conditions (Fig. 5.4):

$$\begin{cases} I_{\tau}^{\max}(\mathbf{A}_{k,i}(t)) = I_{\tau|\delta}^{\min}(\mathbf{A}_{k,i+1}(t)), \\ \text{ or } I_{\tau}^{\min}(\mathbf{A}_{k,i}(t)) = I_{\tau|\delta}^{\max}(\mathbf{A}_{k,i+1}(t)) \end{cases} \tag{5.18}$$

For a given $\tau_{k,i}$ angle, there are 8 such conditions, namely two (Eqs. (5.18)) for each of the depth one validity interval. And since there are three $\tau_{k,i}$ angles per tripeptide, we obtain 24 conditions.

With these ingredients, our algorithm operates as follows:

- Generate a random ray $\mathrm{Ray}(V)$ in the motion space $\mathcal{M}$.
- (`get_tau_tmax`, Algorithm 2 and Sec. S5.7.4) For a given $\tau_{k,i}$ angle, find out the largest interval $[0, t_{\max}]$ such that the $\mathrm{DOVI}_{\tau_{k,i}}$ is different from the $\emptyset$ on this interval (Nb: an upper bound on $t_{\max}$ is obtained from the triangle inequality applied to the C_α carbons, see remark 5.4.)
- (`LS_one_step`, Algorithm 3) Take the intersection of all such intervals for the $3m$ angles, generate a t value on the resulting interval, and apply the corresponding motions to the tripeptide legs. This yields a candidate conformation $L_{\text{cand.}} \in \mathcal{V}$ of the loop L.
- (`Loop_sampler`, Algorithm 4). Perform `LS_one_step` until $L_{\text{cand.}} \in \mathcal{S}$. Once obtained, start again from $L_{\text{cand.}}$ and iterate.

Remark 5.5. *In the* real random access memory model *(real RAM), which assumes exact calculations with real numbers, Algorithm* `LS_one_step` *is exact. In practice, our implementation uses multiprecision numbers and root finding routines provided by Maple [MGH+03]. Due to the cost of such operations, algorithm 3 can be further optimized, see algorithm S6.*

Leaving the realm of multiprecision, an approximate version has also been developed to strike a compromise between exactness and performances, see `LS_one_step_approx` *(Algorithm 5). This variant performs a regular sampling of the ray, from which t_{max} is estimated.* `LS_one_step_approx` *is the version used in the experiments thereafter.*

5.4.5 Sampling: combining several steps

We use the building block `Loop_sampler` to define two algorithms. In our Experiments, the loops assessed are those generated by these two algorithms, without any relaxation/energy minimization or post-processing.

Unmixed loop sampler. Combining steps of `Loop_sampler` yields algorithms $\mathbf{ULS}_{One|All;N_{ES}}^{N_V;N_{OR}}[p_0]$, whose parameters are as follows:

1. p_0: the starting point/conformation in space $\mathcal{S}$.
2. One—All: a point in the solution space $\mathcal{S}$ generates a total of $N_m = \Pi_{k=1,\ldots,m} s_k$ loop conformations, with s_k the number of TLC solutions for the tripeptide T_k. The flag One—All states whether we choose one embedding at random, or keep them all.

3. N_{ES}: for a given HAR trajectory, the number of embedding steps performed.

4. N_V: number of HAR trajectories started at p_0, each defined by a random vector defining a ray in the motion space $\mathcal{M}$.

5. N_{OR}: the output rate in the form $1/n$, with n the number of HAR steps performed along a HAR trajectory, before an *embedding step* is performed–as dictated by the flag One—All. An output rate of one means that all embeddings steps are exploited.

For example, $\mathbf{ULS}^{5;1/4}_{One;1000}$ uses five HAR trajectories with an output rate of 1/4, and 1000 embedding steps, each retaining a single embedding. Thus, the number of loop conformations returned is exactly 1250. On the other hand, $\mathbf{ULS}^{1;1}_{All;1000}$ uses a single HAR trajectory of 1000 steps with an output rate of one, retaining all solutions at each step. The number of loop conformations generated is at least 2000, and at most 1000×16^m.

Mixed loop sampler. In the previous version of the algorithm, peptide bodies remain rigid during the whole simulation. To alleviate this constraint, we also provide the following two-step variant of the algorithm, denoted $\mathbb{MLS}^{N_V;N_{OR}}_{One|All;N_{ES}}[p_0]$. In short, every other HAR step, the loop is shorted by three residues (two a.a. on one end, one on the other), and a HAR step is performed for this reduced model. One solution is then picked at random, and the updated positions of the peptide bodies used for the next HAR step.

Remark 5.6. *We have recalled above the two types of constraints used by TLC: the legs' positions and internal coordinates. Practically, we use standard values for internal coordinates [CSJD04, ORC22]. These internal coordinates can be changed and sampled in the course of the algorithm, an option not used in our experiments.*

In using these standard coordinates, we assume that all tripeptides of the loop have angular parameters $\mathbf{A}_k \in \mathcal{S}$.

5.5 Experiments

5.5.1 Material and methods

Implementation. Our implementation is sketched in Sec. S5.7.3. Consider a loop together with a valid starting point p_0 – see below. First, the $12(m-1)$ Cartesian coordinates of the peptide bodies are extracted, together with the 12 Cartesian coordinates of the two loop anchors (4 points in total). Then, the steps are iteratively performed as described above for the unmixed and mixed versions of the loop sampler.

We compare our samplers against the state-of-the-art method `MoMA-LS` [BMV+19] discussed in Introduction. We note however that the comparison is not perfectly fair since `MoMA-LS` also samples three ω angles in the loop before using tripeptide loop closure. Importantly, we noted that the ω angle preceding the first tripeptide of the loop is also sampled (Fig. S5.15). This degree of freedom induces a rotation of all atoms in the loop, including $C_{\alpha;1}$ which is fixed in our algorithm.

Loops tested. Several loop datasets have been assembled, see e.g. [JPR+04, ZZLF11, MSD18, BMV+19]. Note that a loop refers to a set of structures with the same sequence and anchor positions which can be superimposed via a rigid motion. Most of these loops comprise between 12 and 15 amino acids. In the sequel, we focus on three such loops.

• PTPN9-MEG2. The first one is a 12 a.a. long loop found in the in human protein tyrosine phosphatase PTPN9-MEG2 [QZC+02, ZLT+12], between residue 466 and 477. For this case, four conformations (aka landmarks) have been crystallized: L_0 : 4GE2.pdb/chain A, L_1:2PA5.pdb/chain A, L_2: 4GE6.pdb/chain B, L_3:4ICZ.pdb/chain A. Interestingly, three of these loops form a cluster (lRMSD < 0.1, Table S5.2), while

L_3 is significantly different (lRMSD > 1.5). We choose L_0 as a starting point, since it is furthest away from L_3.

• CCP-W191G. The second loop is a 15 a.a. long loop found in cytochrome C peroxidase (CCP), a water-soluble heme-containing enzyme reducing hydrogen peroxide (H_2O_2) to water. CCP contains three cavities which are hydrophobic (cavity: L99A), slightly polar (cavity: L99A/M102Q), and anionic (cavity: W191G), the latter binding almost exclusively small monocations. Out of the several crystal structures reported [GSB$^+$08], one of them features N-methyl-1-phenylmethanamine – N-Methylbenzylamine for short in the W191G cavity. This binding is of interest, as the aforementioned 15 a.a. long loop flips out by nearly 12Å, opening the cavity to the bulk solvent for the entry/exit of the ligand [GSB$^+$08].

• CDR-H3-HIV. To illustrate the ability of our method to handle long loops as a whole, we process a 30 a.a. long complementarity-determining region (CDR H3) loop, one of the longest CDR observed in human antibodies [PMW$^+$10]. Broadly neutralizing antibodies against the human immunodeficiency virus type of 1 (HIV-1) exhibit two typical features, namely an extensive affinity maturation (accomplished over long periods of time), and an exceptionally long heavy chain CDR.

5.5.2 Conformational diversity

To assess the conformational diversity of a set of conformations generated, we plot the root mean square fluctuations (RMSF) of the $3m$ heavy atoms $\{N, C_\alpha, C\}$ of the loop backbone, in the form of boxplots. (Recall that the RMSF of a given atom is the stdev of distances between its positions and their center of mass.)

Loop PTPN9-MEG2. We first analyze the RMSF values observed for the loop PTPN9-MEG2 (Fig. 5.6). A general observation is the bell shape traced by the RMSF median marks, which is expected since the middle of the loop incurs less steric constraints than its endpoints. To compare the methods, the RMSF plots for `MoMA-LS` converge rapidly. A median of $\sim 2-3$Å in the middle of the loop is obtained, with numerous extreme/outlier configurations. Our algorithm needs more steps to stabilize, reaching a stable distribution for 500 conformations. Overall, our methods generate RMSF fluctuations larger than those from `MoMA-LS`, with $\mathbf{ULS}^{1;1}_{One;\cdot}$ and $\mathbb{MLS}^{1;1}_{One;\cdot}$ yielding median RMSF values $\sim 5-6$Å and ~ 8Å respectively near the center of the loop.

Our plots also shed light on the various ingredients of our method. A marked difference is observed between $\mathbf{ULS}^{N_V;N_{OR}}_{\text{One—All};N_{ES}}$ and $\mathbb{MLS}^{N_V;N_{OR}}_{\text{One—All};N_{ES}}$. The RMSF plots of the former contain plateaus of length 4 corresponding to the atoms found in rigid peptide bodies. Those of the latter do not, a consequence of the shift shift along the backbone inherent to the removal of three amino acids.

Otherwise, an important point is the stability of our method with respect to the parameter One—All and to the number of vectors N_V. Beyond 500 conformations, little variation is actually observed (Fig. 5.6 versus Fig. S5.11 and Fig. S5.12).

CCP-W191G. The patterns for this slightly longer loop are similar to those observed for the previous one, so that we focus solely on the most striking point. Interestingly, despite the lack of sampling of the ω angle, our algorithms reach a max RMSF circa 7.5Å, while `MoMA-LS` culminates at about 3.7Å(Fig. 5.7 and Fig. S5.13.

The ability to generate such diverse ensembles is clearly an advantage over more classical methods such as Molecular Mechanics/Generalized Born Surface Area (MM/GBSA) which fail from sampling conformations as diverse as 12Å[GSB$^+$08].

CDR-H3-HIV. Loops beyond 15 a.a. are usually considered to be beyond reach [MSD18, BCC21]. To illustrate the capabilities of our method, we process a 30 a.a. long loop CDR-H3-HIV (Fig. 5.8), one of the longest CDR observed in human antibodies [PMW$^+$10]. The CDR3 resembles an axe, with a handle and a head (Fig. 5.8(A)). This CDR represents alone 42% of the surface area exposed by the CDRs [PMW$^+$10].

Remarkably, compared to the two loops just discussed, `MoMA-LS` exhibits a much larger diversity (Fig. S5.14). Naturally, the longer the loop, the larger the benefits of also sampling the ω angle preceding the loop. The RMSF plots for our algorithm show a flattened bell shape curve $\mathbb{MLS}^{10;1/2}_{One;500}$, $\mathbb{MLS}^{10;1/2}_{One;5000}$,$\mathbb{MLS}^{10;1}_{One;500}$ and $\mathbb{MLS}^{10;1}_{One;5000}$, with a maximum RMSF near 12Å.

It has been speculated that the head of this CDR3 can substantially deform, possibly to maneuver into a recessed epitope [PMW$^+$10]. Our simulations mitigates this intuition (Fig. 5.8(B,C,D)). On the one hand, while the *middle* of the head does deform substantially, in particular in the vertical direction, the front and the back appear quite rigid. On the other hand, the stem of the axe exhibits a substantial lateral flexibility. Naturally, these preliminary observations call for further structural analysis in the presence of the antigens.

5.5.3 Exploration of the conformational landscape

To assess the ability of the algorithm to explore a complex conformational landscape, we focus on loops for which several conformations have been obtained experimentally. Consider a set $\{L_j\}, j = 1, \ldots, J$ of J loop conformations, called *landmarks*. To assess the amount of conformational space explored, we generate conformations, and check the min and max lRMSD distances of these conformations to all landmarks.

Loop PTPN9-MEG2. These distances are of special interest in the context of the 2-cluster structure of the four conformations of PTPN9-MEG2 (Table S5.2).

Starting from L_0, we first study the ability to move away from the cluster $L_0/L_1/L_2$ (maxlRMSD values for columns L_1 and L_2, Table 5.1). For a fixed number of conformations (50/500/5000), the lRMSD observed for our algorithms are significantly larger than those obtained with the loops from `MoMA-LS`. Consistent with the analysis of RMSF, the variant $\mathbf{ULS}^{N_V;N_{OR}}_{\text{One—All};N_{ES}}$ outperforms all contenders.

Also starting from L_0, we next investigate the speed at which we approach the significantly different conformation L_3 (minlRMSD values for column L_3, Table 5.1). The values reported by our methods are slightly worse than those from `MoMA-LS` (Table 5.1): best `MoMA-LS`: 0.99Å; best $\mathbf{ULS}^{N_V;N_{OR}}_{\text{One—All};N_{ES}}$: 1.46Å; best $\mathbb{MLS}^{N_V;N_{OR}}_{\text{One—All};N_{ES}}$ 1.40Å. However, as noticed above, `MoMA-LS` also samples the ω angle preceding the loop. Inspecting ω values, one obtains: $\omega(L_0) : -177°$; $\omega(L_3) : -165°$; ω(best from `MoMA-LS`): $-167°$. It is therefore the sampling of this dihedral angle which favors `MoMA-LS`.

5.5.4 Failure rate and running time

Algorithm 3 fails as soon as one TLC does not admit any solution. This failure probability depends on the number of tripeptides, and naturally depends on the discrepancy between the two spaces $\mathcal{S}_k$ and $\mathcal{V}_k$, that is on the volume of the region $\mathcal{V}_k \backslash \mathcal{S}_k$. In turn, this failure naturally impacts the running time of algorithm `Loop_sampler`.

Calculations were run on a desktop DELL Precision 7920 Tower (Intel Xeon Silver 4214 CPU at 2.20GHz, 64 Go of RAM), under Linux Fedora core 32. Each HAR is processed on a single CPU core. For PTPN9-MEG2, there there is on average 0.69 failure per success when tested on $\mathbf{ULS}^{1;1}_{One;1000}[L_0]$ and 2.92 with $\mathbb{MLS}^{1;1}_{One;1000}[L_0]$.

The average time taken for one step by $\mathbf{ULS}^{1;1}_{One;1000}[L_0]$ is 0.04 seconds, and 0.17 for $\mathbb{MLS}^{1;1}_{One;1000}[L_0]$. The latter algorithm involves more operations than the former, and as just noticed, also incurs a higher failure rate. Whence the increased running time.

For the long loop CDR-H3-HIV, the average failure per success becomes 1.18 for $\mathbf{ULS}^{1;1}_{One;1000}$ and 6.09 for $\mathbb{MLS}^{1;1}_{One;1000}$. The average time per step in $\mathbf{ULS}^{1;1}_{One;1000}$ becomes 0.21 seconds, and 0.98 for $\mathbb{MLS}^{1;1}_{One;1000}[L_0]$.

Remark 5.7. *Parameter One—All has no impact on failure rate since all solutions are computed in any case.*

5.6 Outlook

Method. Loops sampling methods raise difficult mathematical problems due to the high dimensionality of the parameter space, and the non linear interaction between the degrees of freedom (dof). Current state-of-the art methods belong to two main classes. The first one consists of methods relying on kinematic loop closure; such methods first perturb selected dof (the prerotation step), and proceed with loop closure (the postrotation step). However, a first difficulty is to balance the amplitude of changes incurred by pre and post dof, to avoid steric clashes during the loop closure step. Another difficulty lies in the non linear nature of the solution space. For systems involving n dof, such methods typically results in a solution space which is a $n-6$ dimensional manifold. Sampling this manifold is usually done via back-projection upon walking the tangent space, which is numerically challenging and imposes rather local changes. A second class of methods of utmost importance exploit structures from the Protein Data Bank, and possibly resort to loop closure too. However, such methods face a combinatorial explosion when the loop length increases. As a matter of fact, modeling as a whole loops beyond 15 amino acids is still considered out of reach.

Our work introduces a new paradigm for this problem, based on a global geometric parameterization of the loop relying on a decomposition into tripeptides. The method lies in the lineage of the Hit-and-Run algorithm, invented long ago to identify redundant constraints in a linear program. Since then, HAR and related techniques have proven essential to sample high dimensional distributions in bounded and unbounded domains, yielding effective polynomial time algorithms of low complexity to compute the volume of polytopes in hundreds of dimensions [LV18, CV16, CPC22, CCF22]. The connexion between these algorithms and loop sampling is non trivial, as using HAR to generate loop conformations involves two new ingredients. The first one is a description of the loop sampling problem in a fully dimensional conformational space, as it is the absence of codimension which removes the constraint to follow a curved manifold. We achieve such a description using the intrinsic description of tripeptides. The second one is the design of necessary conditions for the individual tripeptide problems to admit solutions. These conditions can then be used in a manner akin to the hyperplanes of the polytope, to explore the region of interest and generate novel conformations.

Our results improve on those produced by a recent state-of-the-art method. On classical loop examples (12 to 15 a.a.), we show that our solutions enjoy wider RMSF fluctuations. We also show that our method copes easily with a 30 a.a. long loop as a whole, a loop length usually considered beyond reach. Last but not least, it should be stressed that our method is parameter free, as the generation process does not depend on any statistical or biophysical model.

Future work. Computational Structural Biology recently underwent a very significant progress with the advent of deep learning methods for structure prediction [JEP+21, BDA+21]. However, such methods generally face difficulties for unstructured and/or highly flexible regions [RP21]. Also, they do not yield insights on the intrinsic complexity of the problem. In this context, our work opens new perspectives in structural modeling. In terms of structure, we anticipate several straightforward applications. The ability of our sampler to generate very diverse ensembles of conformations should prove key to investigate systems with highly flexible regions, including enzymes, membrane transporters, CDRs, and also intrinsically disordered proteins. The realm of thermodynamics appears more challenging. As discussed in Introduction, methods in the lineage of `Conrot` come with correction factors which, once incorporated into Metropolis-Hastings and Monte Carlo sampling, ensure that the correct distribution (typically canonical) is sampled. Our work primarily focuses on the geometric rather than thermodynamic setting. In fact, current sampling methods of choice are multiphase / adaptive sampling methods, including meta-dynamics, Wang-Landau, etc [LSR10, BMS15]. A question of critical importance in future work will be to ensure that our exploration methods are suitable to sample NVE and/or NVT ensembles. The connexion with polytope volume calculations is a strong hint that this may indeed be the case, and that sampling micro-canonical ensembles may be possible. If so, our paradigm may eventually yield a definitive step for structural and thermodynamic predictions. Meanwhile, our method can still be used in the context of global optimization and energy landscapes, which decouples structure, thermodynamics, and dynamics Upon discovering (deep) local minima, one can sample their basins [Wal03] using classical MC methods.

Algorithm 2 `get_tau_tmax`. For a given angle $\tau_{k,i}$, find the largest value of $t_{\max}$ of t such that $\mathrm{DOVI}_{\tau_{k,i}}(p(t)) \neq \emptyset$ on the segment $[0, t^{\Delta}_{max}]$.

for $I_{\tau_{k,i}}(t) \in \mathcal{I}_{\tau_{k,i}}(t)$ **do**
  for $I_{\tau_{k,i}|\delta}(t) \in \mathcal{I}_{\tau_{k,i}|\delta}(t)$ **do**
    $S = S \cup$ numerical solutions for Eqs. 5.18 $t \in [0, t^{\Delta}_{max}]$
Sort S by ascending order
Let t_l be the l-th element of S
$l = 1$
$u_l := \frac{t_l + t_{l+1}}{2}$
// Stop when no validity interval can be defined for $\tau_{k,i}$
while $\mathrm{DOVI}_{\tau_{k,i}}(f_{\mathcal{M}\to\mathcal{A}}\gamma(u_l)) \neq \emptyset$ **do**
  $t_{max} = t_k$
  $l = l + 1$
return $\{t_{max}\}$

Algorithm 3 `LS_one_step`. Given a starting point $p_0 \in \mathcal{S}$ and a random direction V in the motion space $\mathcal{M}$, the algorithm finds the nearest intersection p_{near} of the image of the ray $\mathrm{Ray}(V)$ (by the map $f_{\mathcal{M}\to\mathcal{A}}$) with a surface constraint, and generates a random value on the segment $[0, t_{\max}]$. Then, applies the corresponding motion to peptide bodies of the loop L.

Input: $p_0 \in \mathcal{S}$: starting point in the fertile space
Input: V: direction in motion space
Output: a point $p_{out} \in \mathcal{V}$
Var $t^{\Delta}_{\max}$: initialized using the smallest value of $t > 0$ breaking triangular inequality in a given tripeptide
V: Random direction (Eq. 5.8)
$S = \{t^{\Delta}_{\max}\}$
for k $\in \{1, ..., m\}$ **do**
  for i $\in \{1, 2, 3\}$ **do**
    $S = S \cup \mathrm{get\_tau\_tmax}(\tau_{k,i})$
// Get the smallest value – most stringent condition
$t_{\max} = \min S$
// Output the next sample
$t_s \leftarrow \mathrm{Uniform}(0, t_{\max})$
Apply the rigid transforms defined by t_s to the $m - 1$ peptide bodies
return Loop L with moved peptide bodies

Algorithm 4 `Loop_sampler`. Given a starting point $p_0 \in \mathcal{S}$, algorithm `Loop_sampler` iterates `LS_one_step` until $L_{\text{cand.}}$ yields solution(s) for all tripeptides in the loop. This process is then repeated iteratively from $L_{\text{cand.}}$.

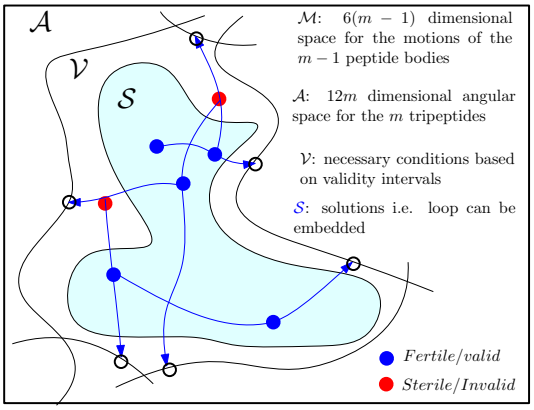
Input: $p_0 \in \mathcal{V}$
$p_{tmp} = p_0$
$Sample = \emptyset$
while not done **do**
  $is\_in\_S = false$
  while not is_in_S **do**
    Generate random direction V
    $L_{\text{cand.}} \leftarrow \mathrm{LS\_one\_step}(p_{tmp}, V)$
    Solve individual TLC for the m peptide bodies
    if all m tripeptide have at least one solution **then**
      $is\_in\_S = true$
  Combine the individual solutions obtained for the individual tripeptides

Algorithm 5 `LS_one_step_approx`. Given a starting point $p_0 \in \mathcal{S}$, a random direction V in the motion space $\mathcal{M}$, and a number of iteration X, the algorithm uniformly samples between 0 and $t_{\max}$, and finds the largest value u_l such that $\text{DOVI}_{\tau_{k,i}}(f_{\mathcal{M}\to\mathcal{A}}(\gamma(u_l))) \neq \emptyset$. It then iterates between the step were it stopped and the one before it until $\text{DOVI}_{\tau_{k,i}}(f_{\mathcal{M}\to\mathcal{A}}\gamma(u_l)) \neq \emptyset$. If $X \to \infty$ the $t_{\max}$ obtained using this algorithm corresponds to the one obtained using `LS_one_step`.

Input: $p_0 \in \mathcal{S}$: starting point in the fertile space
Input: V: direction in motion space
Input: X: max number of iteration to obtain approximate solution
Output: a point $p_{out} \in \mathcal{V}$
Var $t^{\Delta}_{\max}$: initialized using the smallest value of $t > 0$ breaking triangular inequality in a given tripeptide
V: Random direction (Eq. 5.8)
$u_l := 0$
$x = 1$
// Identify the first iteration failing the condition
while $\text{DOVI}_{\tau_{k,i}}(f_{\mathcal{M}\to\mathcal{A}}\gamma(u_l)) \neq \emptyset$ **do**
    $u_l = (x/X)t_{\max}$
    $x = x + 1$
// Slice the failing interval into X bits and iterate
$t_{min} = (x-1)/X t_{\max}$
$x = 1$
while $\text{DOVI}_{\tau_{k,i}}(f_{\mathcal{M}\to\mathcal{A}}\gamma(u_l)) \neq \emptyset$ **do**
    $u_l = t_{min} + \frac{t_{\max}(x)}{X^2}$
    $x = x + 1$
$t_{\max} = t_{min} + \frac{t_{\max}(x-1)}{X^2}$
// Output the next sample
$t_s \leftarrow \text{Uniform}(0, t_{\max})$
Apply the rigid transforms defined by t_s to the $m-1$ peptide bodies
return Loop L with moved peptide bodies

5.7 Artwork

Figure 5.1 Sampling a loop involving m tripeptides: algorithm overview. Spaces used: $\mathcal{A}$: a $12m$ dimensional angular space coding the internal geometry of all tripeptides; $\mathcal{V} \subset \mathcal{A}$: a region characterized by necessary conditions for the m individual TLC problems to admit solutions; $\mathcal{S} \subset \mathcal{V}$ corresponds to individual geometries of the tripeptides such that TLC admits solutions for each tripeptide. The Hit-and-Run algorithm is used to find intersection (empty bullets) between 1D trajectories (blue curves) in the angular space of the tripeptides, and hyper-surfaces bounding the regions defining necessary conditions for the m individual TLC problems to admit solutions. One point is then generated on the curve segment joining the staring point and the intersection point. This point is fertile if all TLC problems admit solutions, and sterile otherwise. The number of conformations obtained is the product of the individual numbers for the m tripeptides.



The supporting information is organized as follows:

- Section S5.7.1: notations,
- Section S5.7.2: algorithm,
- Section S5.7.3: implementation,
- Section S5.7.4: sampling,
- Section S5.7.5: material,
- Section S5.7.6: results.

Figure 5.2 Loop decomposition into tripeptides and peptide bodies, and associated geometric model. (A) Each ellipsis and its two legs correspond to one tripeptides. In red, the peptide bond between the consecutive tripeptides T_k and T_{k+1}. The peptide body encompasses the peptide bond, as well as one atom to the left and the right. **(B)** Indexing of atoms within the k-th tripeptide. **(C)** Geometry of the peptide bond linking tripeptides T_k and T_{k+1} with constrained bond lengths, valence angles, and torsion angle – in red. These four atoms form the rigid body P_k.

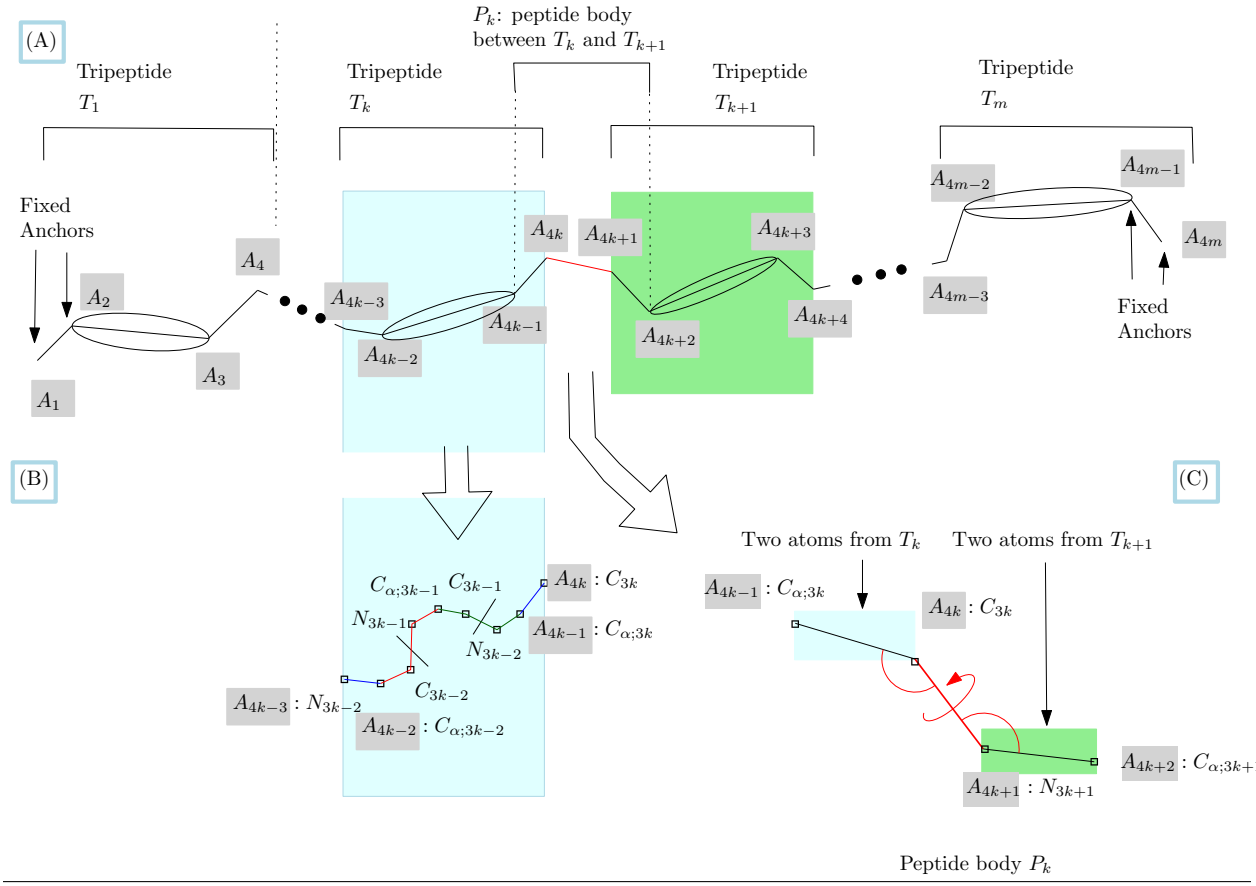


Figure 5.3 Geometric model used for an individual tripeptide. **(A)** Tripeptide with moving legs. Given internal coordinates and two rigid bodies around a tripeptide the C_α triangle can be defined together with $\{\alpha_i, \eta_i, \xi_i\}$ angles. **(B)** $\mathcal{J}^{(1)}_{\sigma_i}$ and $\mathcal{J}^{(1)}_{\tau_i}$. **(C)** Illustration of the relationship between rigid body positions, $\{\alpha_i, \eta_i, \xi_i\}$ angles and the *depth one inter-angular constraint.*

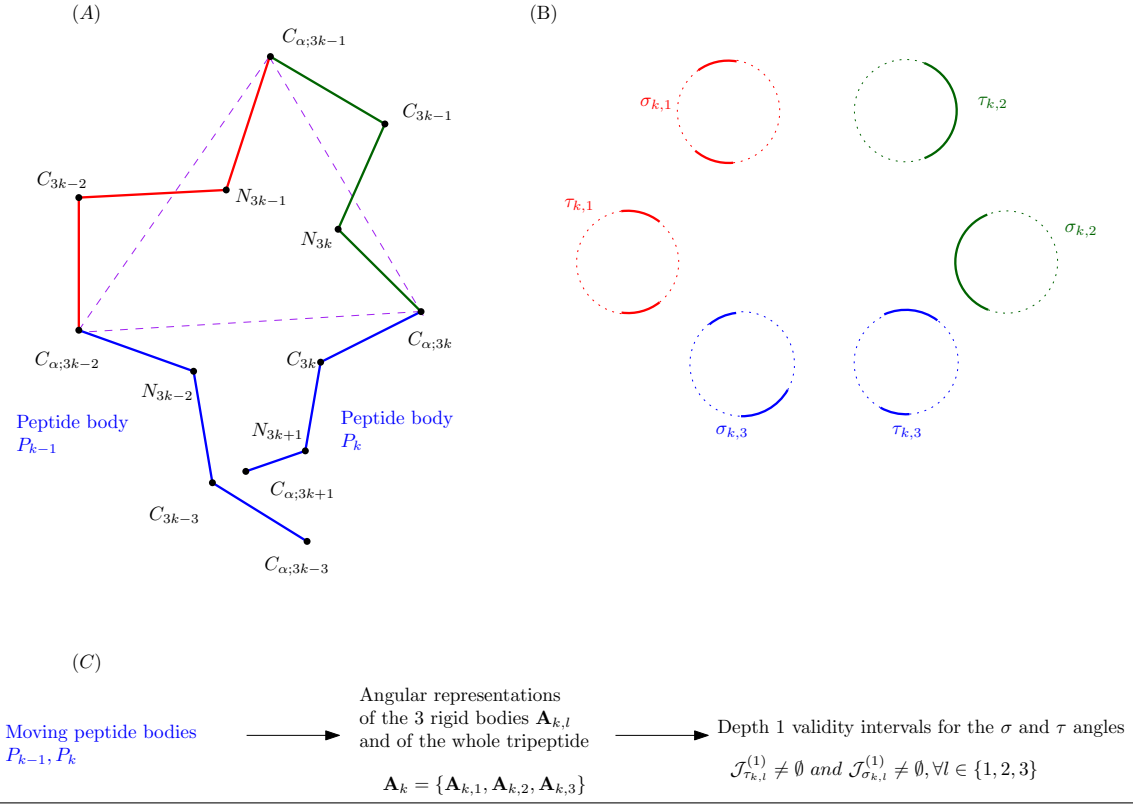


Figure 5.4 Kinetic validity intervals. We focus on a given interval pair $I_{\tau_{k,i}} \in \mathcal{I}_{\tau_{k,i}}$ and $I_{\tau_{k,i}|\delta} \in \mathcal{I}_{\tau_{k,i}|\delta}$ for the angle $\tau_{k,i}$ from tripeptide T_k. The legs of T_k are moving with P_{k-1} and P_k. These movements impact the positions of the interval endpoints via the angles $\mathbf{A}_{k,i}(t)$ and $\mathbf{A}_{k,i+1}(t)$. **(A)** The interiors of the two intervals intersect. **(B)** The intervals intersect on their boundary–a limit case. The arrow indicate the derivative of the endpoints of intervals with respect to time.

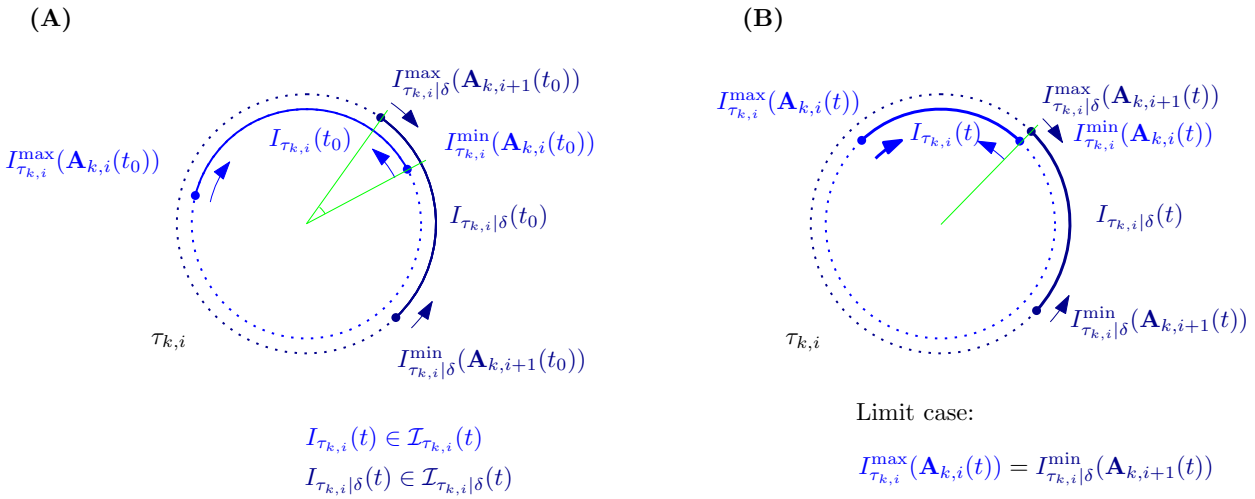


Figure 5.5 Interpolation in the space of rigid motions $\mathcal{R}$ and associated transformations applied to rigid bodies. The figure features two peptide bodies P_k and P_{k+1} in the loop segment $T_k^{'}\ P_kT_{k+1}^{'}\ P_{k+1}T_{k+2}^{'}$. The initial positions of the bodies are denoted $P_k(0)$ and $P_{k+1}(0)$ respectively; these bodies must satisfy a distance constraint materialized by the green line segment – $length < S$. Each rigid body undergoes a translation (unit vectors $\mathbf{T}_k^{(t)}$ and $\mathbf{T}_{k+1}^{(t)}$ respectively) composed with a rotation (unit vectors $\mathbf{V}_k^{(r)}$ and $\mathbf{V}_{k+1}^{(r)}$ respectively). The positions corresponding to time t are denoted $P_k(t)$ and $P_{k+1}(t)$ respectively. The distance between the last C_α of $P_k(t)$ and the first C_α of $P_{k+1}(t)$ is constrained by the triangular inequality (SI Sec. 5.7.4). This constraint is represented by the maximum length S on the figure.

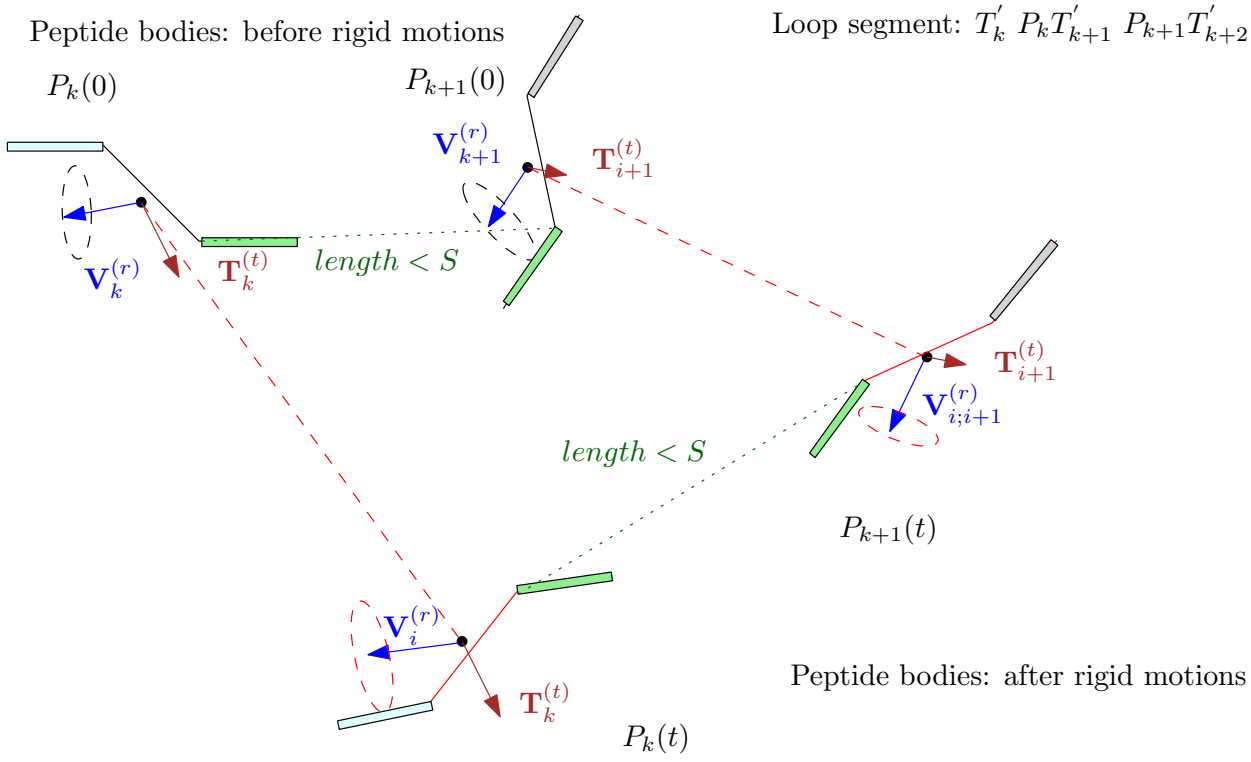


Figure 5.6 Loop PTPN9-MEG2: Backbone RMSF for the 12 amino acid long loop PTPN9-MEG2. Simulations started from the conformation/landmark L_o – see text. Each tick on the x-axis corresponds to a heavy atom of the loop – 36 in this case. For `MoMA-LS`, note that only one atom is fixed on the left hand side of the loop, since the ω angle preceding the loop is also sampled.

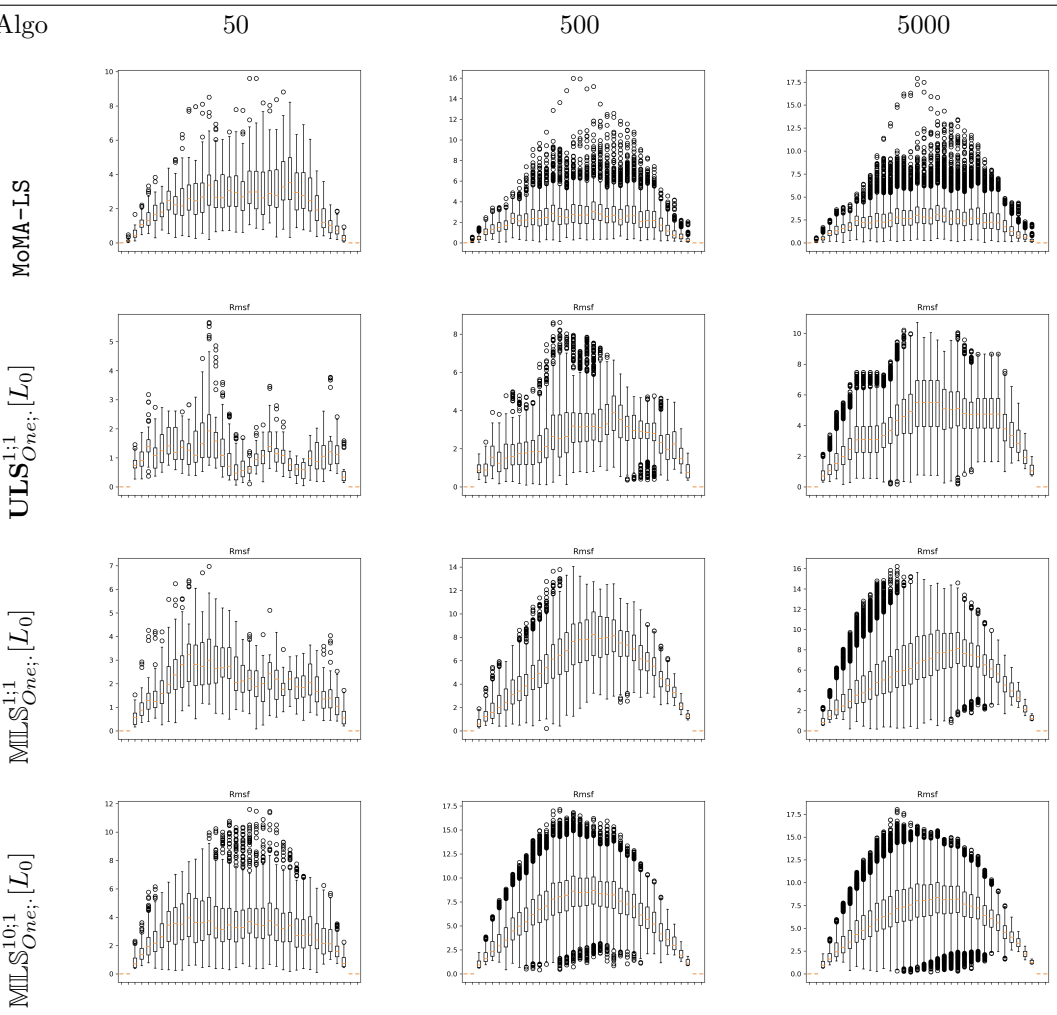
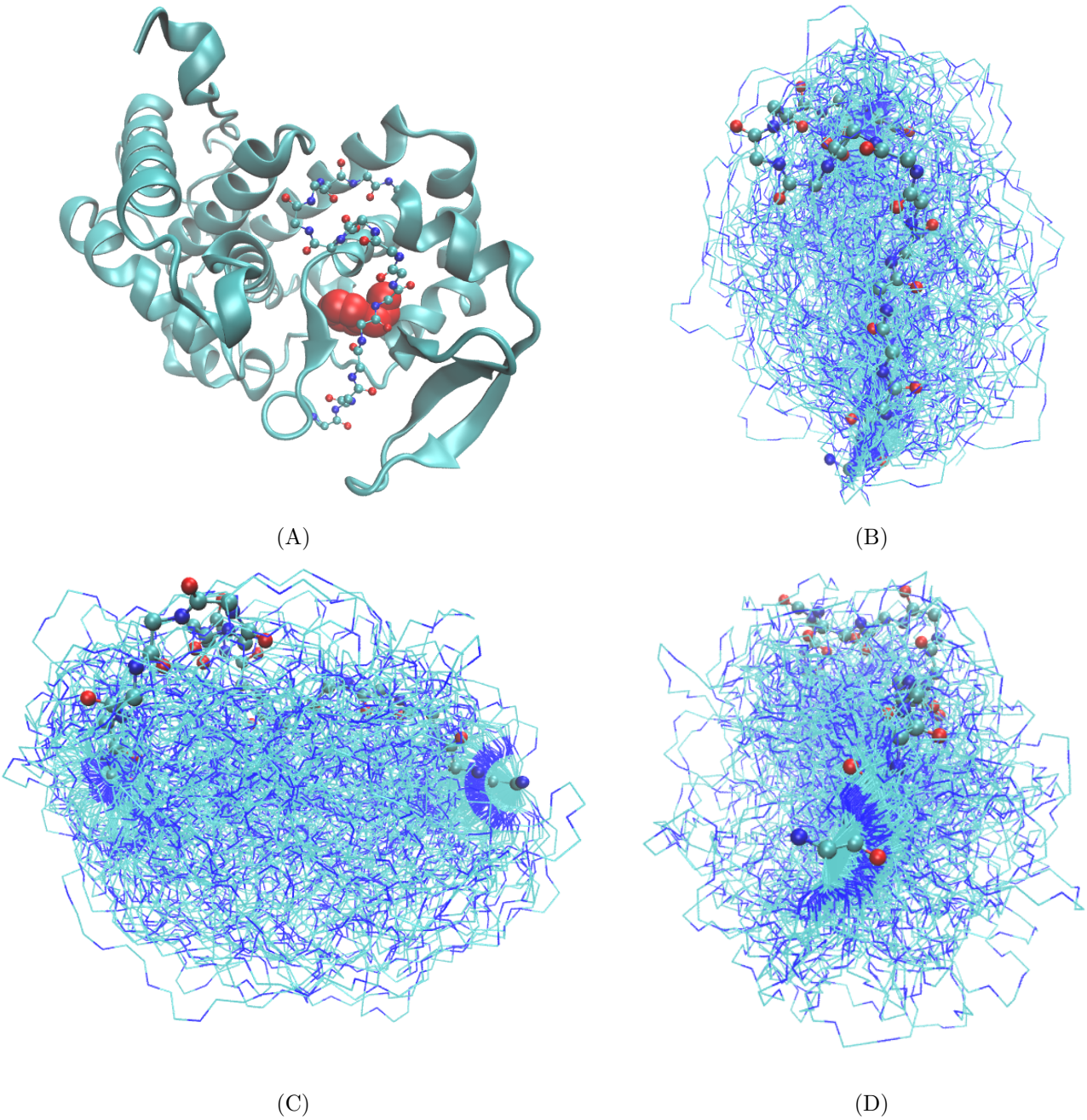


Table 5.1 Loop PTPN9-MEG2: exploration to reach landmark conformations. Four conformations of loop PTPN9-MEG2 form two clusters: L_0, L_1, L_2 and L_3. For `MoMA-LS`, we compute min and max lRMSD distances to these landmarks. For $\mathbf{ULS}^{N_V;N_{OR}}_{\text{One—All};N_{ES}}$ and $\mathbb{MLS}^{N_V;N_{OR}}_{\text{One—All};N_{ES}}$, starting from L_0, we investigate the ability to get away from the cluster (maxlRMSD values) and to approach conformation L_3 (minlRMSD values).

	L_1 min/maxlRMSD	L_2 min/maxlRMSD	L_3 min/maxlRMSD
`MoMA-LS`, 50	1.00/3.81	1.03/3.80	1.38/4.11
`MoMA-LS`, 500	0.78/4.29	0.77/4.30	1.11/4.70
`MoMA-LS`, 5000	0.74/4.92	0.73/4.94	0.99/4.97
$\mathbf{ULS}^{1;1}_{One;50}[L_0]$	0.41/2.21	0.42/2.23	1.43/2.60
$\mathbf{ULS}^{10;1}_{One;50}[L_0]$	0.39/3.09	0.38/3.09	1.39/3.50
$\mathbf{ULS}^{1;1/4}_{One;50}[L_0]$	0.59/3.04	0.59/3.05	1.34/3.45
$\mathbf{ULS}^{10;1/4}_{One;50}[L_0]$	0.46/3.53	0.47/3.56	1.34/3.73
$\mathbb{MLS}^{1;1}_{One;50}[L_0]$	0.46/3.99	0.47/4.01	1.59/4.56
$\mathbb{MLS}^{10;1}_{One;50}[L_0]$	0.43/4.02	0.43/4.03	1.53/4.75
$\mathbb{MLS}^{1;1/4}_{One;50}[L_0]$	1.80/5.05	1.81/5.07	2.20/5.38
$\mathbb{MLS}^{10;1/4}_{One;50}[L_0]$	1.35/5.45	1.36/5.47	1.81/5.61
$\mathbf{ULS}^{1;1}_{One;500}[L_0]$	0.46/3.77	0.46/3.79	1.36/4.19
$\mathbf{ULS}^{10;1}_{One;500}[L_0]$	0.38/4.88	0.37/4.89	1.36/4.97
$\mathbf{ULS}^{1;1/4}_{One;500}[L_0]$	0.63/5.25	0.64/5.28	1.45/5.54
$\mathbf{ULS}^{10;1/4}_{One;500}[L_0]$	0.59/5.17	0.59/5.21	1.45/5.55
$\mathbb{MLS}^{1;1}_{One;500}[L_0]$	0.61/5.47	0.61/5.48	1.60/6.12
$\mathbb{MLS}^{10;1}_{One;500}[L_0]$	0.52/5.86	0.53/5.87	1.52/6.46
$\mathbb{MLS}^{1;1/4}_{One;500}[L_0]$	1.69/5.66	1.71/5.69	1.93/6.05
$\mathbb{MLS}^{10;1/4}_{One;500}[L_0]$	1.45/5.75	1.43/5.77	1.68/6.30
$\mathbf{ULS}^{1;1}_{One;5000}[L_0]$	0.48/5.26	0.49/5.29	1.42/5.51
$\mathbf{ULS}^{10;1}_{One;5000}[L_0]$	0.43/5.36	0.43/5.40	1.40/5.74
$\mathbf{ULS}^{1;1/4}_{One;5000}[L_0]$	0.56/5.19	0.56/5.22	1.45/5.58
$\mathbf{ULS}^{10;1/4}_{One;5000}[L_0]$	0.46/5.42	0.47/5.46	1.46/5.80
$\mathbb{MLS}^{1;1}_{One;5000}[L_0]$	0.71/5.83	0.72/5.86	1.56/6.22
$\mathbb{MLS}^{10;1}_{One;5000}[L_0]$	0.57/5.96	0.57/5.99	1.52/6.48
$\mathbb{MLS}^{1;1/4}_{One;5000}[L_0]$	1.82/5.88	1.83/5.89	1.66/6.32
$\mathbb{MLS}^{10;1/4}_{One;5000}[L_0]$	1.45/6.06	1.44/6.10	1.46/6.59

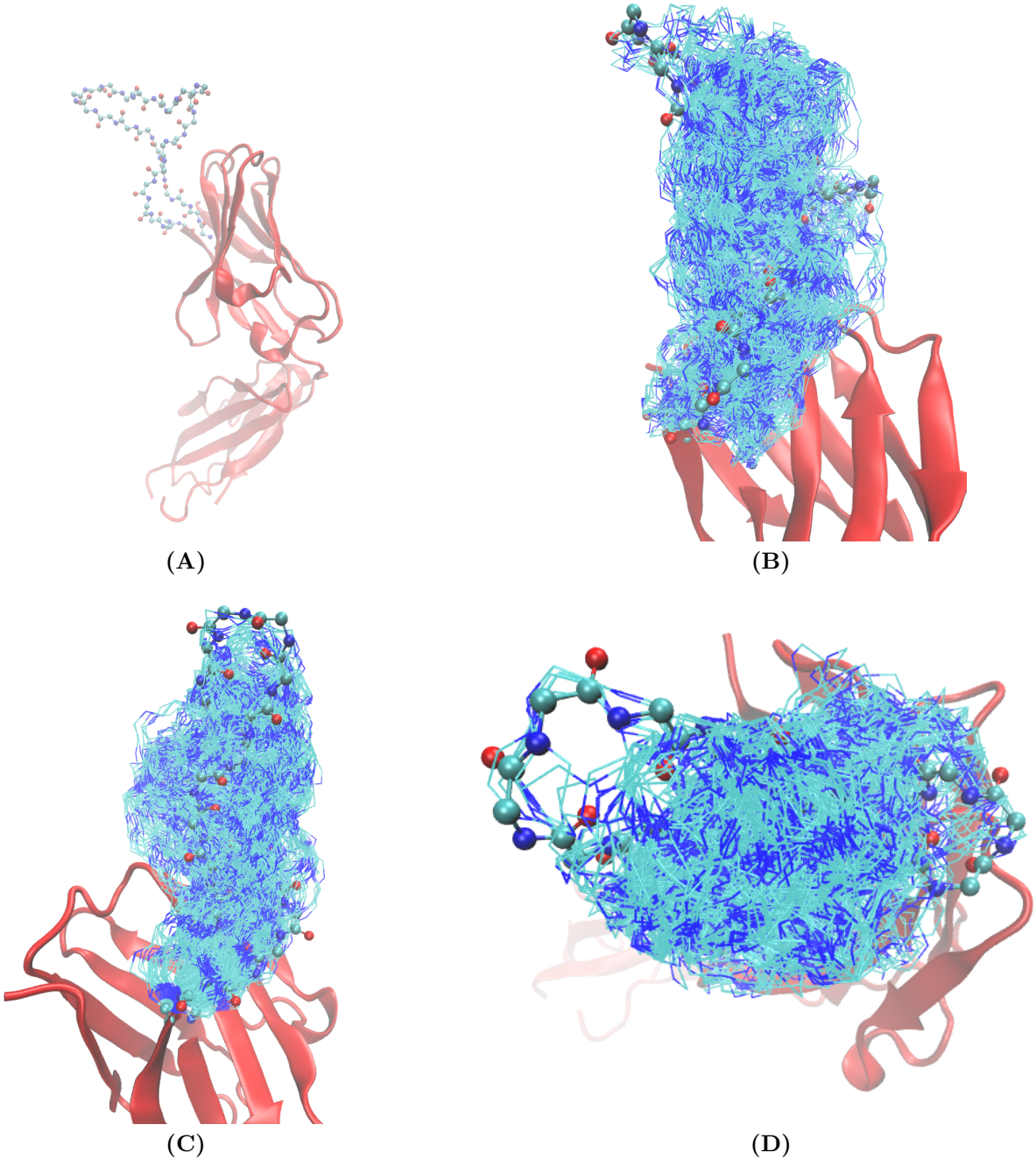
Figure 5.7 CCP-W191G. Loop studied specification: pdbid: 2rbt, chain X, residues 186-200. Conformations generated by algorithm $\mathbb{MLS}^{1;1}_{One;250}$. **(A)** Overview of the protein: cartoon mode: protein; CPK mode: loop; VDW representation: ligand N-Methylbenzylamine. **(B,C,D)** Top, side, front view of the loop conformations. Protein omitted for the sake of clarity.



(A) (B)

(C) (D)

Figure 5.8 Complementarity-determining region (CDR-H3) raised against HIV-1: sampling a 30 amino acid long loop. PG16 is an antibody with neutralization effect on HIV-1 [PMW$^+$10]. Loop specification: pdbid: 3mme; chain A; residues: 93-100, 100A-100T, 101, 102. Conformations generated by algorithm $\mathbb{MLS}^{1;1}_{One;250}$. **(A)** Variable domain (red) and the 30 a.a. long CDR3. **(B,C,D)** Side/front/top view of 250 conformations .



(A) **(B)** **(C)** **(D)**

5.7.1 Notations: cheatsheet

Tripeptides and the whole loop L.

- Legs of a tripeptide T_k: the first two and last two atoms, i.e. left leg $(N_1, C_{\alpha;1})$, right leg $(C_{\alpha;3}, C_3)$. The tripeptide core $T_k^{'}$ is the tripeptide minus the legs.
- Peptide body P_k: the rigid body defined by the right leg and the left leg of two consecutive tripeptides.
- Loop anchors are the first two and last two atoms in the loop.
- Loop: a sequence of m tripeptide:

$$L = P_0\ T_1^{'}\ P_1\ \ldots\ P_{k-1}\ T_k^{'}\ P_k\ \ldots\ P_{m-1} T_m^{'} P_m.$$

  Decomposes into left anchor + sequence of (tripeptide core+peptide body) + right anchor.

Angular representations.

- Tripeptide T_k, four tuple of angles around the $C_{\alpha;i}$: $\mathbf{A}_{k,i} = \{\alpha_{k,i}, \eta_{k,i}, \xi_{k,i-1}, \delta_{k,i-1}\}$ with $i \in \{1,2,3\}$ – counted modulo three.
- Tripeptide angular representation aggregating three four tuples: $\mathbf{A}_k = \{\mathbf{A}_{k,1}, \mathbf{A}_{k,2}, \mathbf{A}_{k,3}\}$.
- Angular conformational space of a tripeptide: 12-dimensional space $\mathcal{A}_k$
- Angular conformational space of the loop L: $12m$-dimensional space $\mathcal{A} = \prod_{k=1}^{m} \mathcal{A}_k$.
- Functions returning the 4 angles α, ξ, η and δ as a function of the legs of a tripeptide: $f^{(\alpha)}_{(k,i)}, f^{(\xi)}_{(k,i)}, f^{(\eta)}_{(k,i)}, f^{(\delta)}_{(k,i)}$
- Validity intervals for the angle $\tau_{k,i}$:

$$\begin{cases} \text{Initial validity interval:} I_{\tau_{k,i}} = [I^{\min}_{\tau}(\mathbf{A}_{k,i}), I^{\max}_{\tau}(\mathbf{A}_{k,i})] & \text{Sets: } \mathcal{I}_{\tau_{k,i}} = \cup I_{\tau_{k,i}} \\ \text{Rotated validity interval:} I_{\tau_{k,i}|\delta} = [I^{\min}_{\tau|\delta}(\mathbf{A}_{k,i+1}), I^{\max}_{\tau|\delta}(\mathbf{A}_{k,i+1})] & \text{Sets: } \mathcal{I}_{\tau_{k,i}|\delta} = \cup I_{\tau_{k,i}|\delta} \end{cases}$$

- Mapping from the 12 angles $\mathbf{A}_{k,i}$ into the set of validity intervals:

$$\text{DOVI}_{\tau_{k,i}}(\cdot) : \mathcal{A}_k \mapsto (\mathcal{I}_{\tau_{k,i}} \cap \mathcal{I}_{\tau_{k,i}|\delta})^4.$$

- Angular validity domain of angle $\tau_{k,i}$ for the tripeptide T_k: the subset of $\mathcal{A}_k$ such that $\text{DOVI}_{\tau_{k,i}}(\cdot) \neq \emptyset$.
- Depth j validity intervals for $\tau_{k,i}$: $\mathcal{J}^{(j)}_{\tau_{k,i}}$.

Motions.

- The $6(m-1)$ dimensional space of rigid motions for the $m-1$ peptide bodies: $\mathcal{M}$
- Kinetic (dept one) validity intervals:

$$\begin{cases} I_{\tau_{k,i}}(t) = [I^{\min}_{\tau}(\mathbf{A}_{k,i}(t)), I^{\max}_{\tau}(\mathbf{A}_{k,i}(t))] \\ I_{\tau_{k,i}|\delta}(t) = [I^{\min}_{\tau|\delta}(\mathbf{A}_{k,i+1}(t)), I^{\max}_{\tau|\delta}(\mathbf{A}_{k,i+1}(t))] \end{cases}$$

Spaces and validity domains.

- Angular conformational space $\mathcal{A}$

$$\mathcal{A} \overset{Def}{=} \prod_{k=1}^{m} \mathcal{A}_k.$$

- The *angular* validity domain $\mathcal{V}$ of L:

$$\mathcal{V} \subset \mathcal{A} \text{ such that } \forall k, \forall i, \forall a \in \mathcal{V} : \text{DOVI}_{\tau_{k,i}}(a) \neq \emptyset.$$

- The Hit-and-Run algorithm consists of iteratively sampling a new point on $\text{Ray}_{\mathcal{V}}(p_0)V$.

- Solution space $\mathcal{S}$

$$\mathcal{S} \subset \mathcal{V} \text{ such that TLC admits at least one solution for each tripeptide } T_k.$$

5.7.2 Algorithm

Figure 5.9 The peptide body: a rigid body associated with a peptide bond. Internal coordinates marked in red are fixed. The fixed values of the coordinates $\omega, \nu_{i+1}, d_{i+2}$ are such that the position of $C_{\alpha;2}$ is uniquely determined given positions for the previous three. Note in particular that the distance between $C_{\alpha;1}$ and $C_{\alpha;2}$ is fixed.

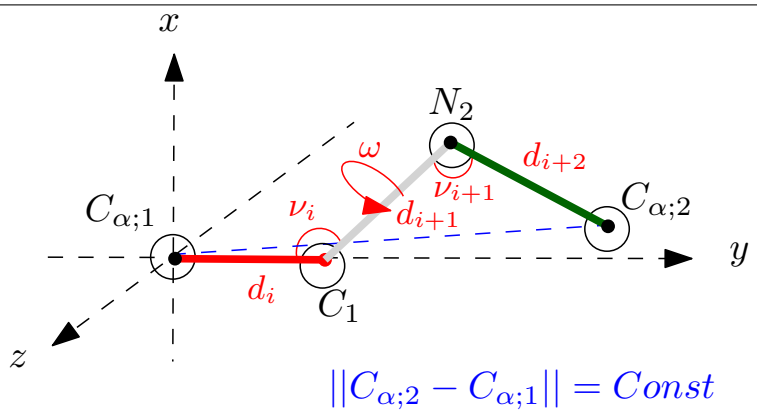


Figure 5.10 Local frames and associated varibales. Adapted from [CSJD04].

- Orthonormal local frames:

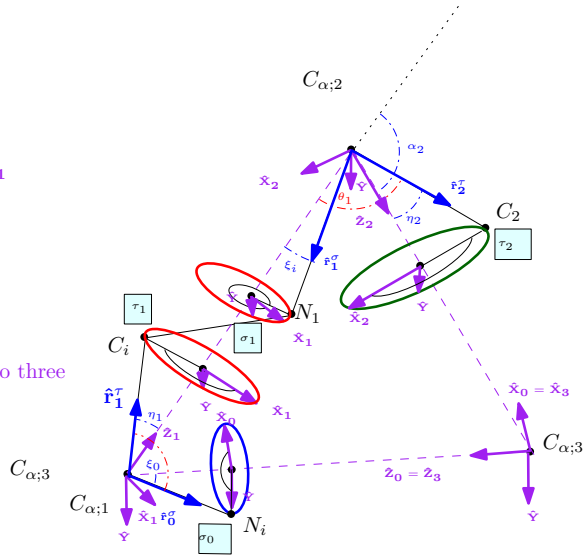
  Nb: $\hat{\mathbf{Z}}_\mathbf{i}$ = Unit vector along $\mathbf{C}_{\alpha;\mathbf{i}}\mathbf{C}_{\alpha;\mathbf{i+1}}$

  $\hat{\mathbf{Y}}_\mathbf{i} \equiv \hat{\mathbf{Z}}_{\mathbf{i-1}} \times \hat{\mathbf{Z}}_\mathbf{i}$ Nb: $\hat{\mathbf{Y}}_\mathbf{i} = \hat{\mathbf{Y}}$

  $\hat{\mathbf{X}}_\mathbf{i} = \hat{\mathbf{Y}}_\mathbf{i} \times \hat{\mathbf{Z}}_\mathbf{i} = (\hat{\mathbf{Z}}_\mathbf{i} \cdot \hat{\mathbf{Z}}_{\mathbf{i+2}})\hat{\mathbf{Z}}_\mathbf{i} - \hat{\mathbf{Z}}_{\mathbf{i+2}}$

  Indices $i \in \{1, 2, 3\}$ are counted modulo three

- Vectors to model the rotations of a toms at $C_{\alpha;i}$:
  - $\hat{\mathbf{r}}^\sigma_{\mathbf{i-1}}$: rotation of C_i
  - $\hat{\mathbf{r}}^\tau_\mathbf{i}$: rotation of N_i



Algorithm 6 `LS_one_step`**: optimized version**. In this optimized version of Algorithm 3, the upper bound $t_{\max}$ is updated incrementally for all $\tau_{k,i}$ angles, which makes it possible to seek individual roots (for a given $\tau_{k,i}$ angle) on a shorter interval.

Input: $p_{in} \in \mathcal{S}$: starting point in the fertile space
Input: V: direction in motion space
Output: a point $p_{out} = \mathcal{V}$

Var $t_{\max}$: initialized using the smallest value of $t > 0$ breaking triangular inequality in a given tripeptide

V: Random direction (Eq. 5.8)
for k $\in \{1, ..., m\}$ **do**
  for i $\in \{1, 2, 3\}$ **do**
    // Angle $\tau_{k,i}$: process the (at most) 24 equations
    $S = \{t_{\max}\}$
    // Process all interval pairs
    for $I_{\tau_{k,i}}(t) \in \mathcal{I}_{\tau_{k,i}}(t)$ **do**
      for $I_{\tau_{k,i}|\delta}(t) \in \mathcal{I}_{\tau_{k,i}|\delta}(t)$ **do**
        $S_{tmp} \leftarrow$ numerical solutions for Eqs. 5.18 $t \in [0, t_{\max}]$
        $S = S \cup S_{tmp}$
    Sort S by ascending order
    Let t_l be the l-th element of S
    $u_l := \frac{t_l + t_{l+1}}{2}$
    $l = 1$
    // Stop when no validity interval can be defined for $\tau_{k,i}$
    while $\text{DOVI}_{\tau_{k,i}}(\tau_{k,i}(u_l)) \neq \emptyset$ **do**
      $t_{\max} = t_k$
      $l = l + 1$
// Output the next sample
$t_s \leftarrow \text{Uniform}(0, t_{\max})$
Apply the rigid transforms defined by t_s to the $m - 1$ peptide bodies

5.7.3 Implementation

- Loop_sampler. The sampler generates the necessary random directions and applies the rigid transformation to each P_k at each step.

- LS_tripep_validity_domain. The individual tripeptide validity domain class contain methods mapping P_{k-1} and P_k to $\mathcal{A}$ as well as computing $t_{\max}$ (Algo. 2)

- LS_bb_embeder. The backbone embedder: performs TLC on all tripeptides using standard internal coordinates and double precision.

5.7.4 Sampling rigid body positions along interpolation paths

In this section, we provide the details about Eq. 5.10, which we repeat for the sake of exposure:

$$P_k(t) = \gamma_k(t) P_k.$$

Rigid body representation

We have noticed that the two anchor points on each side of a peptide bond (four atoms in total), form a rigid body (Fig. 5.2(C)). This rigid body enjoys three translational and three rotational degrees of freedom (dof).

Note that using homogeneous coordinates, the matrix 4×4 matrix giving the coordinates of this rigid body reads as

$$P_k = \begin{pmatrix} A_{4i-1} & A_{4i} & A_{4i+1} & A_{4i+2} \\ 1 & 1 & 1 & 1 \end{pmatrix} \tag{5.19}$$

In the sequel, we will consider a parameterized such matrix, denoted $P_k(t)$, with t a real number.

Translation

- $\mathbf{U}_i^{(t)}, i = 1, \ldots, m-1$: unit vectors drawn uniformly at random on the sphere of directions on S^2. Used to define the directions of translations.
- $C_i^{(t)}, i = 1, \ldots, m-1$: $m-1$ uniformly random variables in $(0, 2\pi)$, to define the norm of translation vectors.
- Translation vector: $\mathbf{T}_i^{(t)} = C_i^{(t)} * \mathbf{U}_i^{(t)}, i = 1, \ldots, m-1$

Using homogeneous coordinates, the corresponding transformation reads as follows:

$$\tilde{\mathbf{T}}_\mathbf{i}(t) = \begin{pmatrix} 1 & 0 & 0 & t\mathbf{T}_{i;x}^{(t)} \\ 0 & 1 & 0 & t\mathbf{T}_{i;y}^{(t)} \\ 0 & 0 & 1 & t\mathbf{T}_{i;z}^{(t)} \\ 0 & 0 & 0 & 1 \end{pmatrix} = \begin{pmatrix} I & t\mathbf{T}_i^{(t)} \\ 0 & 1 \end{pmatrix} \tag{5.20}$$

Rotation

- $\mathbf{U}_i^{(r)}, i = 1, \ldots, m-1$: units vectors drawn uniformly at random on the sphere of directions on S^2. Used to define the rotation axis.
- $C_i^{(r)}, i = 1, \ldots, m-1$: $m-1$ uniformly random variables in $(0, 2\pi)$, to define the rotation angles.
- Rotation: of angles $C_i^{(r)}$ around the direction $\mathbf{U}_i^{(r)}$.

In homogeneous coordinates:

For $\tilde{\mathbf{R}}_\mathbf{i}(t)$ consider $\theta = tC_i^{(r)}$ and $R(\theta, \mathbf{U}_i^{(r)})$ the rotation matrix corresponding to a rotation of θ around axis $\mathbf{U}_i^{(r)}$:

$$\tilde{\mathbf{R}}_\mathbf{i}(t) = \begin{pmatrix} cos\theta + \mathbf{U}_{i;x}^{(r)2}(1-cos\theta) & \mathbf{U}_{i;x}^{(r)}\mathbf{U}_{i;y}^{(r)}(1-cos\theta) - \mathbf{U}_{i;z}^{(r)}sin(\theta) & \mathbf{U}_{i;x}^{(r)}\mathbf{U}_{i;z}^{(r)}(1-cos\theta) - \mathbf{U}_{i;y}^{(r)}sin(\theta) & 0 \\ \mathbf{U}_{i;y}^{(r)}\mathbf{U}_{i;x}^{(r)}(1-cos\theta) + \mathbf{U}_{i;z}^{(r)}sin(\theta) & cos\theta + \mathbf{U}_{i;y}^{(r)2}(1-cos\theta) & \mathbf{U}_{i;y}^{(r)}\mathbf{U}_{i;z}^{(r)}(1-cos\theta) - \mathbf{U}_{i;x}^{(r)}sin(\theta) & 0 \\ \mathbf{U}_{i;z}^{(r)}\mathbf{U}_{i;x}^{(r)}(1-cos\theta) + \mathbf{U}_{i;y}^{(r)}sin(\theta) & \mathbf{U}_{i;z}^{(r)}\mathbf{U}_{i;y}^{(r)}(1-cos\theta) - \mathbf{U}_{i;x}^{(r)}sin\theta & cos\theta + \mathbf{U}_{i;z}^{(r)2}(1-cos\theta) & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$$\tilde{\mathbf{R}}_\mathbf{i}(t) = \begin{pmatrix} (R(\theta, \mathbf{U}_i^{(r)})) & 0 \\ 0 & 1 \end{pmatrix} \tag{5.21}$$

The three types of constraints are equivalent as they are applied on the distance between two points. In all cases the two points are each part of a rigid body and in each cases each point will be subjected to a different translation and rotation. The kinematic function can then be expressed using the rigid body transformations.

Complete transformation

- $P_k(0)$ is the homogeneous 3 dimension coordinates of a rigid body without translation and rotation.
- $P_k(t)$ is the homogeneous 3 dimension coordinates of a rigid body with translated and rotated using $\tilde{\mathbf{T}}_\mathbf{i}(t)$ and $\tilde{\mathbf{R}}_\mathbf{i}(t)$.
- T_i^0 is the translation of the center of mass of $P_k(0)$ to the origin and T_i^{-1} the opposite.

Still using homogeneous coordinates we obtain:

$$\gamma_k(t) = \tilde{\mathbf{T}}_\mathbf{i}(t) T_i^{-1} \tilde{\mathbf{R}}_\mathbf{i}(t) T_i^0 \tag{5.22}$$

The complete transformation applied to P_k becomes:

$$P_k(t) = \gamma_k(t) P_k \tag{5.23}$$

$\gamma_k(t)$ can be applied to any individual atom in P_k.

Numerical root finding and tmax

When numerically searching for a solution to $Eq5.18$ an initial search interval is needed:

Given leg positions for a given tripeptide the three C_α carbons within satisfy a triangle inequality (S5.9). Using the proper indices, this constraint reads as

$$\|C_{\alpha;3i} - C_{\alpha;3i-2}\| < L_{C_{\alpha;3i-2}C_{\alpha;3i-1}} + L_{C_{\alpha;3i-1}C_{\alpha;3i}} \tag{5.24}$$

As noticed above, the distances With $L_{C_{\alpha;3i-2}C_{\alpha;3i-1}}$ and $L_{C_{\alpha;3i-1}C_{\alpha;3i}}$ are fixed. If this is not satisfied we have a forbidden sample as it belongs to $\mathcal{A}\backslash\mathcal{V}$.

- The triangular inequality is used to find an upper bound for numerical root finding.
- **Remark 5.8.** *So long as the translation vectors are not the same there will always be a points where the distance is greater than a given value (Fig. 5.5).*
- Let c_i and c_{i+1} be the centers of the rotation circles for both atoms on which the constraint applies. These correspond to the orthogonal projections on their respective rotational axes.
- With $S = L_{C_{\alpha;3i-2}C_{\alpha;3i-1}} + L_{C_{\alpha;3i-1}C_{\alpha;3i}}$.
- With r_i and r_{i+1} the respective radii of said circles, the triangular inequality will necessarilly be invalid:

$$\left\| \tilde{\mathbf{T}}_{\mathbf{i+1}}(t) c_{i+1}(t) - \tilde{\mathbf{T}}_\mathbf{i}(t) c_i(t) \right\| = S + r1 + r2 \tag{5.25}$$

- This corresponds to a univariate second degree polynomial with one positive and one negative root. The upper limit of our initial constraint with both rotation and translation is the positive root.

In the loop the smallest of such values among all tripeptides is selected as an initial upper bound for t_max.

Table 5.2 Least RMSD matrix between landmark pairs for the loop PTPN9-MEG2. The first three conformations form a cluster.

	L_0	L_1	L_2	L_3
L_0		0.099	0.072	1.574
L_1			0.087	1.550
L_2				1.559

5.7.5 Material

loops used

`MoMA-LS` parameters

Here we summarize the parameters used in `MoMA-LS` for our experiments:

- The ratio of van der Waals radii used for collision detection is 0.5. The minimum value is used as we do not implement collision detection;
- Residue-dependent pseudo-atoms at the C_β positions are not used for collision detection for the same reason;
- Side chains are omitted as they are not considered in our algorithms as of now;
- One solution is kept for inverse kinematics as we mostly compare to the version using one solution for inverse kinematics in our algorithm;
- The number of sampled states is 50 500, or 5000.

Remark 5.9. *A general post-processing strategy in loop generation consists of checking the absence of steric clash between the $N, C_\alpha, C, O, C_\beta$ atoms. Denoting R_i and R_j the van der Waals radii of two atoms i and j, the usual criterion consists of checking that $d_{ij} > (R_i + R_j) \geq d_{min}$, usually taken in the range $0.5 - 0.7$, see [MCK09, BMV+19]. The conformations may also energy minimized, a step which is mandatory when dealing with all atom models.*

5.7.6 Results

Figure 5.11 Loop PTPN9-MEG2: tests with algorithm $\mathbf{ULS}^{N_V;N_{OR}}_{All;N_{ES}}[L_0]$. Compare against Fig. 5.6 to see the incidence of option All.

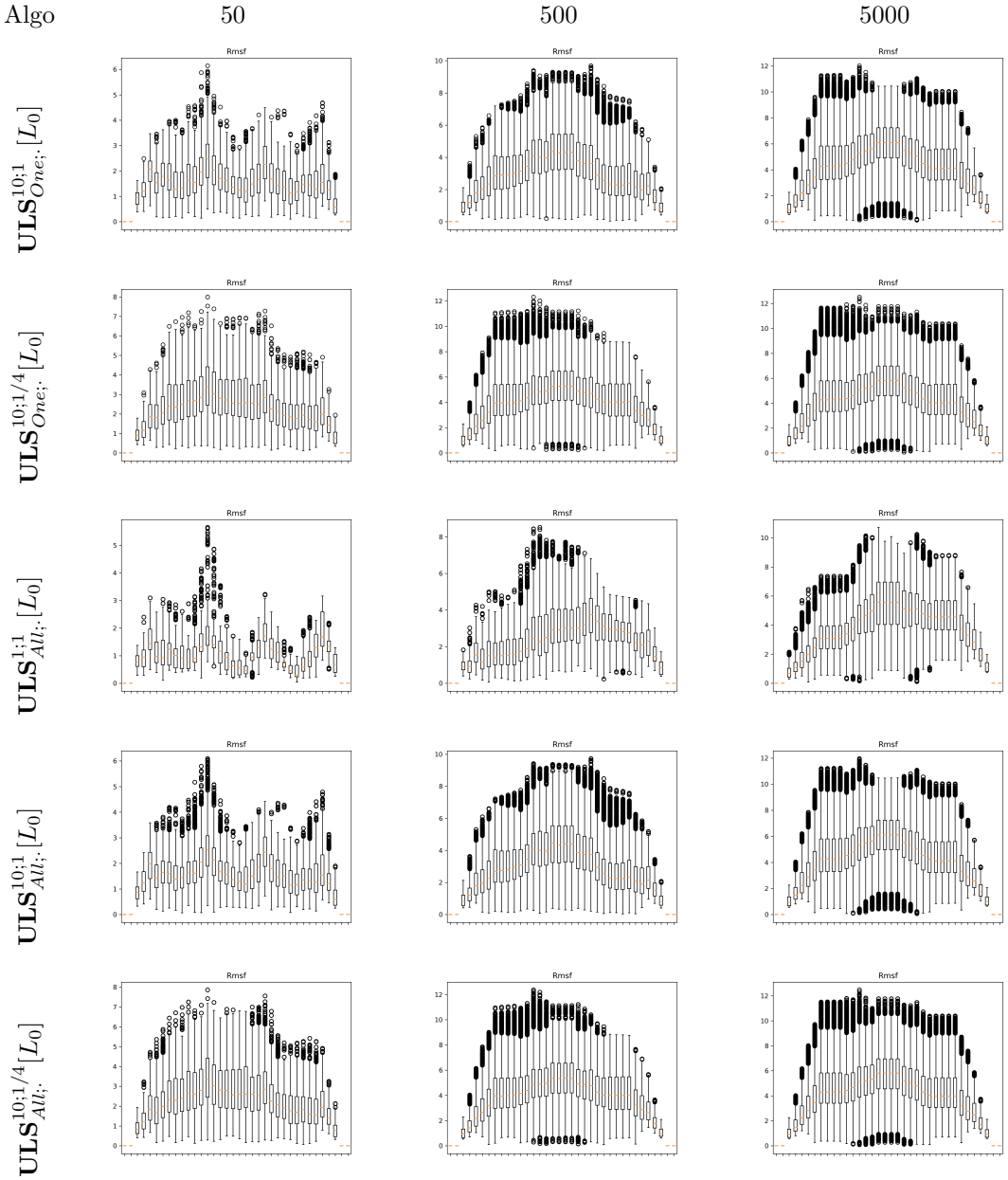


Figure 5.12 Loop PTPN9-MEG2: tests with algorithm $\mathbb{MLS}^{N_V;N_{OR}}_{All;N_{ES}}[L_0]$. Compare against Fig. 5.6 to see the incidence of option All.

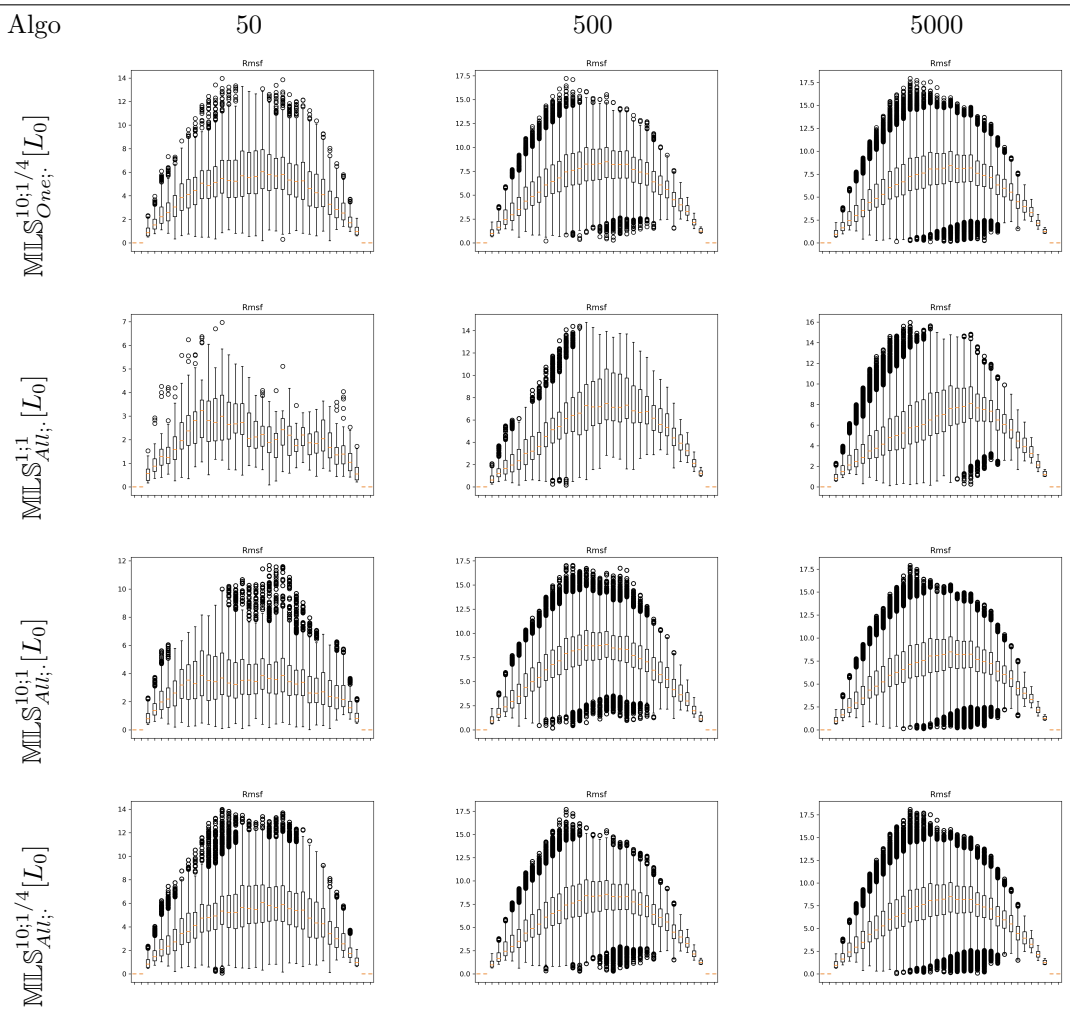


Figure 5.13 Loop CCP-W191G: tests with algorithm $\mathbb{MLS}^{N_V;N_{OR}}_{one;N_{ES}}$ and MoMA-LS.

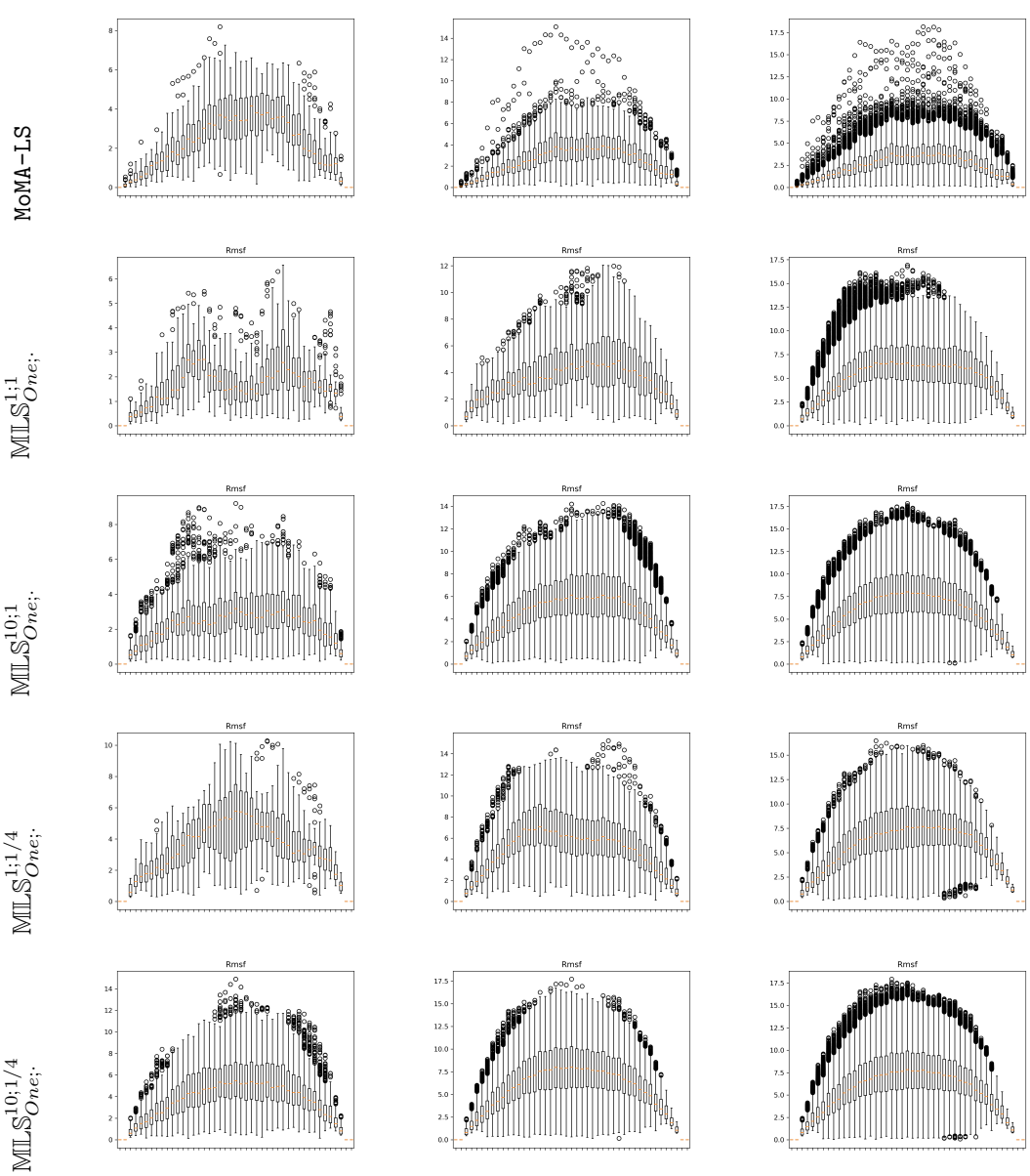


Figure 5.14 Loop CDR-H3-HIV: tests with algorithm $\mathbb{MLS}^{N_V;N_{OR}}_{one;N_{ES}}$ and MoMA-LS.

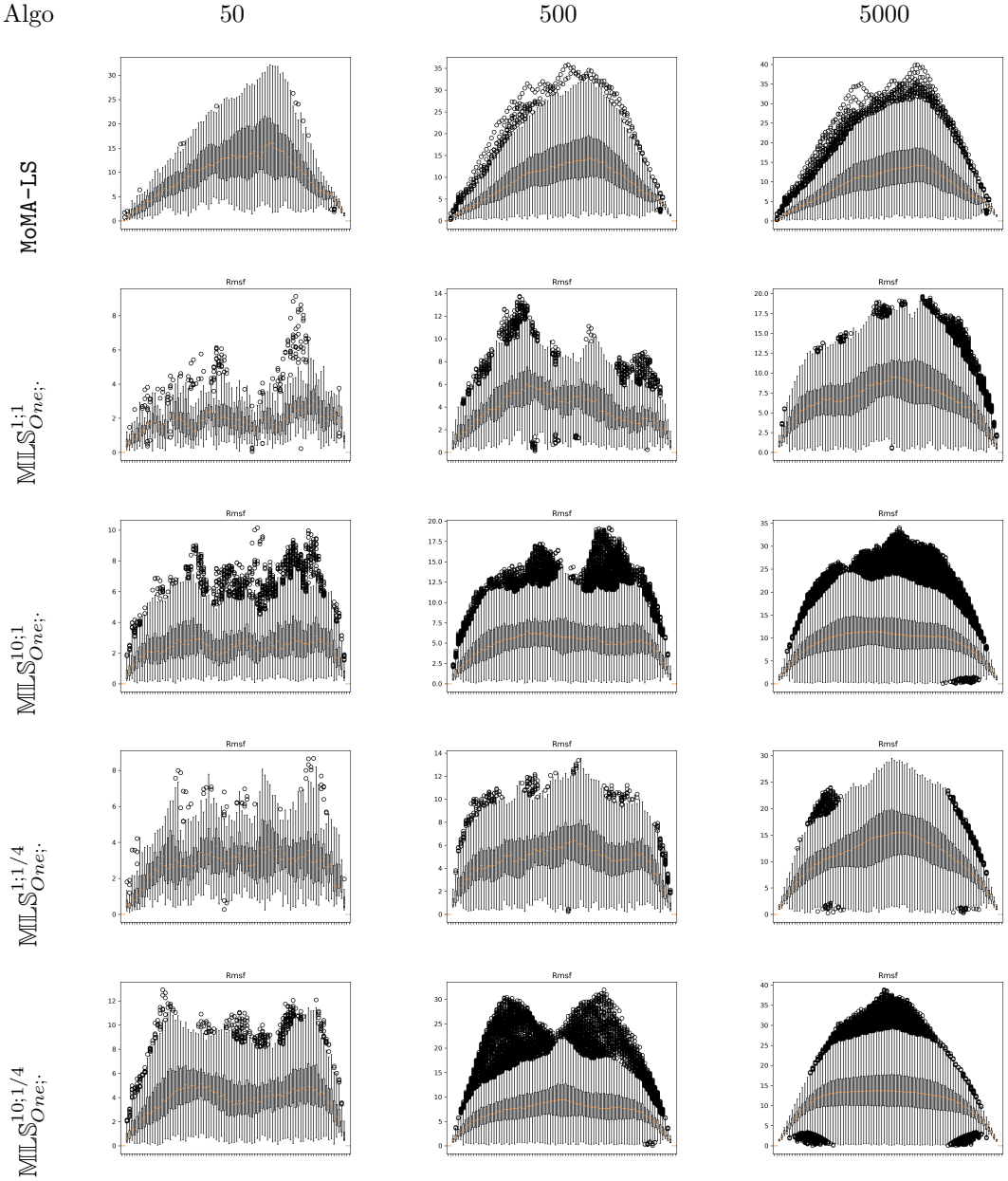
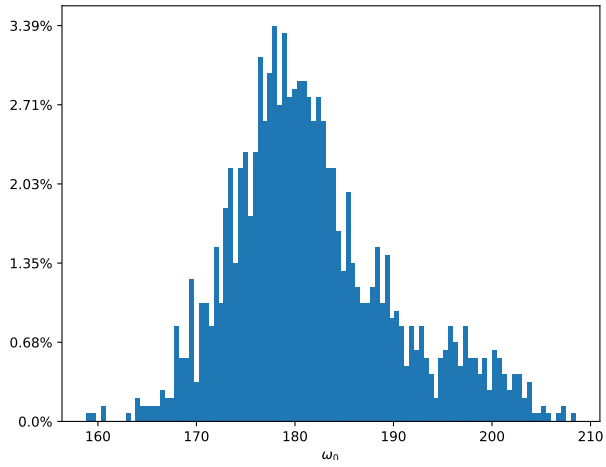


Figure 5.15 ω_0 angle values impacting $C_{\alpha;1}$ position in `MoMA-LS`. This histogram is made from the sample of 5000 conformations obtained using `MoMA-LS` and L_0 of loop PTPN9-MEG2. The ω_0 angle is the torsion angle around the peptide bond preceding the loop.



Chapter 6

Fréchet mean and p-mean on the unit circle

6.1 Introduction

Fréchet mean and generalizations.

Transferring the notion of the center of mass, the point minimizing the sum of square distances in a point set, to the unit circle S^1 is of particular interest since this space encodes angles, where the motion observed in molecular systems is concentrated in terms of internal coordinates. Generalizations of the center of mass in general metric spaces are referred to as Fréchet mean [Fré48].

In the following, we focus on p-means defined on the unit circle S^1, for $p > 1$. As the point set is critical (The case $p = 1$ requires trivial adaptations.)

Consider n angles $\Theta_0 = \{\theta_i\}_{i=1,\dots,n}$. Practically, since real data are known with finite precision, we treat angles as rational numbers. Consider the embedding of an angle onto the unit circle, that is $X(\theta) = (\cos\theta, \sin\theta)^\mathsf{T}$. The geodesic distance between two points $X(\theta)$ and $X(\theta_i)$ on S^1, denoted $d(\cdot,\cdot)$, satisfies

$$d(X(\theta), X(\theta_i)) = \min(\mid \theta - \theta_i \mid, 2\pi - \mid \theta - \theta_i \mid) = 2\arcsin\frac{\|X(\theta) - X(\theta_i)\|}{2}. \tag{6.1}$$

Consider a set of positive weights $\{w_i\}_{i=1,\dots,n}$. For an integer $p \geq 1$, consider the function involving the weighted distances to all points, i.e.

$$F_p(\theta) = \sum_{i=1,\dots,n} w_i f_i(\theta), \text{ with } f_i(\theta) = d^p(X(\theta), X(\theta_i)). \tag{6.2}$$

We denote its minimum

$$\theta^* = \arg\min_{\theta\in[0,2\pi)} F_p(\theta). \tag{6.3}$$

For units weights and $p = 2$, the value obtained is the Fréchet mean. In that case, the candidate minimizers (local minima of Eq. 6.2) form the vertices of a regular polygon [HsH15]. The previous expression can also be seen as a distance to a point mass probability distribution on S^1. For a general probability distribution on S^1, necessary and sufficient conditions for the existence of a Fréchet mean have been worked out [Cha13]. In the same paper, the authors propose a quadratic algorithm–regardless of numerical issues–to compute the Fréchet mean for the particular case of a point mass probability distribution. In a more general setting, a stochastic algorithm finding p-means wrt a general measure on the circle has also been proposed [AM16].

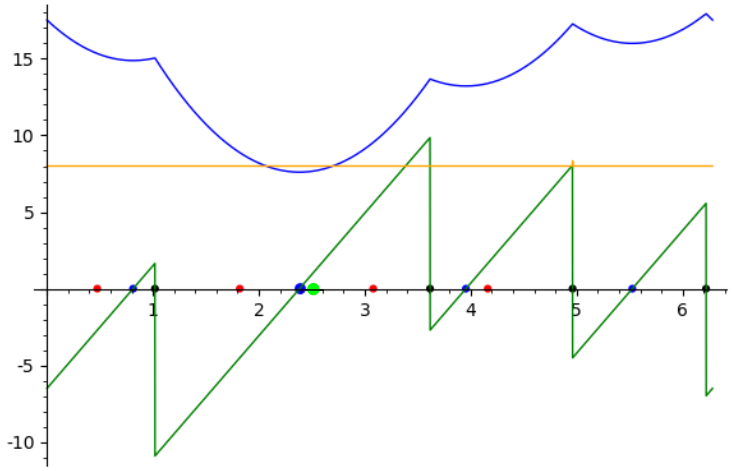
Remark 6.1. *In the subsequent sections, the weights in Eq. 6.2 are omitted – rational weights do not change our analysis. Our implementation, however, does use them.*

Robustness and numerical issues. From a mathematical standpoint, computing the p-mean is a non-convex optimization problem, and one may assume that calculations are carried out in the standard real RAM computer model, which assumes that exact operations on real numbers are available at constant time per operation [PS85]. From a practical standpoint though, numbers in real computers are represented with finite precision [MBdD+18]. The ensuing rounding errors are such that algorithms written in the real RAM model may loop, crash, or terminate with an erroneous answer, even for the simplest 2D geometric calculations [KMP+08].

Robust geometric algorithms, which deliver what they are designed for, can be developed using the Exact Geometric Computation (EGC) paradigm [YD95], which is central in the Computational Geometry Algorithms Library (CGAL) [cga]. The EGC relies on so-called *exact predicates* and *constructions*. A predicate is a function whose output belongs to a finite set, while a construction exhibits a new geometric object from the input data. For example, the predicate `Sign`(x) returns the sign $\{negative, null, positive\}$ of the arithmetic expression x. As we shall see, designing robust predicates for p-means on S^1 is connected to transcendental number theory since expressions involving π are dealt with. In particular, one needs to evaluate the sign of such expressions, which raises decidability issues [CCK+06].

Combinatorial complexity issues. The computation of the p-means also raises a combinatorial complexity issue. Function F_p being a sum over n terms, k function evaluations yield a complexity $O(kn)$, which is quadratic if there is a linear number of local minima. Therefore, the fact that using candidate minimizers form a regular polygon [HsH15] does not directly yield a linear time algorithm even if the angles are sorted. As we shall see, the piecewise maintenance of the expression of the function does so, though. For the sake of conciseness, combinatorial complexity is plainly referred to as complexity in the sequel.

Figure 6.1 Fréchet mean of four points on S^1 (Functions) blue: function F_2; green: derivative F_2'; orange: second derivative F_2'' **(Points)** red bullets: data points; black bullets: antipodal points; blue bullets: local minima of the function; large blue bullet: Fréchet mean θ^*; green bullet: circular mean Eq. 6.14.



6.1.1 Contributions

This paper makes three contributions regarding p-means of a finite point set. First, we show that the function F_p is determined by a very simple combinatorial structure, namely a partition of S^1 into circle arcs. Second, we give an explicit expression for F_p, deduce that the problem is decidable, and present an algorithm computing p-means. Third, we present an effective and robust implementation, based on multi-precision interval arithmetic.

6.2 p-mean of a finite point set on S^1: characterization

6.2.1 Notations

In the following, angles are in $[0, 2\pi)$. We first define:

Definition. 6.1. *For each angle $\theta_i \in [0, \pi)$, we define $\theta_i^+ = \theta_i + \pi$. The set of all such angles is denoted $\Theta^+ = \{\theta_i^+\}$. For each angle $\theta_i \in [\pi, 2\pi)$, we define $\theta_i^- = \theta_i - \pi$. The set of all such angles is denoted $\Theta^- = \{\theta_i^-\}$. The* antipodal set *of Θ_0 is the set of angles $\Theta^\pm = \Theta^+ \cup \Theta^-$.*

Altogether, these angles yield the larger set

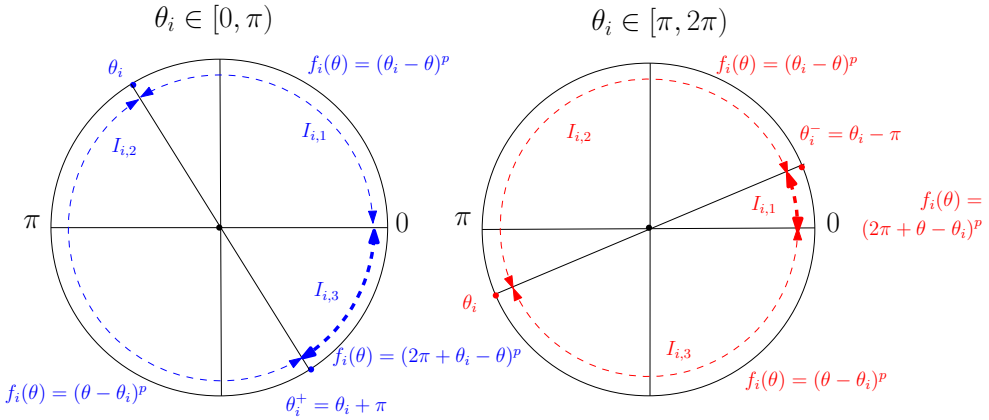
$$\Theta = \Theta_0 \cup \Theta^\pm. \tag{6.4}$$

The $2n$ angles in Θ are generically denoted α_i or α_j. Note however that when referring to an angle in the continuous interval $[0, 2\pi)$, θ is used.

To each angle θ_i, we associate three so-called *elementary intervals* (Fig. 6.2):

- $\theta_i \in [0, \pi) : I_{i,1} = (0, \theta_i), I_{i,2} = (\theta_i, \theta_i^+), I_{i,3} = (\theta_i^+, 2\pi)$.
- $\theta_i \in [\pi, 2\pi) : I_{i,1} = (0, \theta_i^-), I_{i,2} = (\theta_i^-, \theta_i), I_{i,3} = (\theta_i, 2\pi)$.

Figure 6.2 The partition of S^1 into circle arcs, and the piecewise functions defining F_p. The three elementary intervals defined by angles in $[0, \pi)$ and $[\pi, 2\pi)$ respectively. Bold circle arcs indicate that f_i has a transcendental expression i.e. involves π.



6.2.2 Partition of S^1

We also consider the partition of $[0, 2\pi)$ induced by the intersection of the $3n$ intervals $\{I_{i,1}, I_{i,2}, I_{i,3}\}$ (Fig. 6.2). More specifically, we choose one interval (out of three) for each function f_i, and intersect them all:

Definition. 6.2. *The elementary intervals $I_{i,j}$ define a partition of S^1 based on the following intervals:*

$$\mathcal{I} = \{ \bigcap_{i=1,\dots,n} (I_{i,1} \vee I_{i,2} \vee I_{i,3}) \text{ with } \bigcap_{i=1,\dots,n} I_{i,\cdot} \neq \emptyset \}. \tag{6.5}$$

In the following, open intervals from $\mathcal{I}$ are denoted (α_j, α_{j+1}).

Remark 6.2. *From the previous definition, it appears that the intervals in $\mathcal{I}$ may be ascribed to nine types since the left endpoint is an angle θ_i or an antipodal angle θ_i^+ or θ_i^-, and likewise for the right endpoint.*

6.2.3 Piecewise expression for F_p

We use the previous intervals to describe the piecewise structure of F_p. We define the following piecewise functions (Fig 6.2):

$$\theta_i \in [0,\pi) : f_i(\theta) = \begin{cases} (\theta_i - \theta)^p, \text{ for } \theta \in I_{i,1}, \\ (\theta - \theta_i)^p, \text{ for } \theta \in I_{i,2}, \\ (2\pi + \theta_i - \theta)^p, \text{ for } \theta \in I_{i,3}. \end{cases} \tag{6.6}$$

$$\theta_i \in [\pi,2\pi) : f_i(\theta) = \begin{cases} (2\pi + \theta - \theta_i)^p, \text{ for } \theta \in I_{i,1}, \\ (\theta_i - \theta)^p, \text{ for } \theta \in I_{i,2}, \\ (\theta - \theta_i)^p, \text{ for } \theta \in I_{i,3}. \end{cases} \tag{6.7}$$

The previous equations give the piecewise expression of $F_p(\theta)$ (Eq. 6.2), from which one derives the following, which characterizes the derivative at points in $\alpha_j \in \Theta$:

$$\Delta f'_{i|\theta} = \lim_{\theta \searrow \alpha_j} f'_i(\theta) - \lim_{\theta \nearrow \alpha_j} f'_i(\theta) \tag{6.8}$$

Remark 6.3. *Let θ_{max} be the antipodal value of the largest $\theta_i \in \Theta_0$ larger than π, and θ_{min} the antipode of the smallest $\theta_i \in \Theta_0$ smaller than π. The function F_p is transcendental in $[0,\theta_{max})$ and $(\theta_{min}, 2\pi]$ – its expression involves π. Also, the function F_p is algebraic on $(\theta_{max}, \theta_{min})$. See Fig. 6.2.*

Using Eq. 6.8, the following is immediate:

Lemma. 6.1. *For $p > 1$, the function f_i and its derivatives satisfy:*

- *The function f_i is continuous on S^1.*
- *The derivative f'_i is continuous on S^1 except at the antipodal value of θ_i, where $\Delta f'_{i|antipode(\theta_i)} = -2p\ \pi^{p-1}$.*
- *The second order derivative f''_i is non negative on S^1.*

The previous lemma tells us that $F_p^{'}$ incurs drops at antipodal points, and then keeps increasing again on the interval starting at that point. Finding local minima of F_p therefore requires finding those intervals from $\mathcal{I}$ where $F_p^{'}$ vanishes, which happens at most once:

Lemma. 6.2. *For $p > 1$, the function F_p has at most one local min. on each interval in $\mathcal{I}$.*

6.3 Algorithm

The observations above are not sufficient to obtain an efficient algorithm: since there are $2n$ intervals and since the function has linear complexity on each of them, a linear number of function evaluations has quadratic complexity. We get around this difficulty by maintaining the expression of the function at angles in Θ.

6.3.1 Analytical expressions and nullity of $F_p^{'}$

The function F_p and its derivative. We first derive a compact, analytical expression of F_p and $F_p^{'}$. Following Eqs. 6.6 and 6.7, the expressions of $f_i(\theta)$ and $f'_i(\theta)$ can be written as

$$f'_i(\theta) = k_i \times (a_i + \varepsilon_i \theta)^{p-1}, \text{ with } k_i \in \{-p, p\}, a_i \in \{-\theta_i, 2\pi - \theta_i, \theta_i, 2\pi + \theta_i\}, \varepsilon_i \in \{-1, +1\}. \tag{6.9}$$

On open intervals (α_j, α_{j+1}), the function reads as the following polynomial

$$F_p(\theta) = \sum_{i=1}^{n} (a_i + \varepsilon_i \theta)^p = \sum_{j=0}^{p} b_j \theta^j, \text{ with } b_j = \sum_{i=1}^{n} \binom{p}{j} a_i^{p-j} \varepsilon_i^j. \tag{6.10}$$

Similarly, the derivative $F_p^{'}(\theta)$ reads as a degree $p-1$ polynomial:

$$F_p^{'}(\theta) = \sum_{i=1}^{n} k_i(a_i + \varepsilon_i\theta)^{p-1} = \sum_{j=0}^{p-1} c_j\theta^j, \text{ with } c_j = \sum_{i=1}^{n} k_i \binom{p-1}{j} a_i^{p-1-j}\varepsilon_i^j. \tag{6.11}$$

In the following, we assume that the coefficients of F_p and $F_p^{'}$ are stored in two vectors B and C of size $p+1$ and p respectively, so that evaluating the function or its derivative at a given θ has cost $O(p)$.

Nullity of $F_p^{'}$: algebraic versus transcendental expressions. The previous equations call for two important comments. First, from the combinatorial complexity standpoint, if the coefficients of the polynomials are known, evaluating F_p and $F_p^{'}$ has cost $O(p)$. Second, from the numerical standpoint, locating local minima of F_p requires finding intervals from $\mathcal{I}$ on which $F_p^{'}$ vanishes. Identifying such intervals is key to the robustness of our algorithm. Practically, since an interval is defined by two consecutive values in the set Θ, we need to check that the sign of $F_p^{'}$ differs at these endpoints. The cornerstone is therefore to decide the sign of $F_p^{'}$ at angles in Θ (input angles or their antipodes), and the following is a simple consequence of Lindemann's theorem on the transcendence of π:

Lemma. 6.3. *If the angular values $\theta_i \in \Theta_0$ are rational numbers, checking whether $F_p^{'}(\alpha_i) \neq 0$ for any $\alpha_i \in \Theta$ is decidable. Moreover, when $F_p^{'}$ has a transcendental expression and α_i is rational, $F_p^{'} \neq 0$.*

Proof. We first consider the case $\alpha_i \in \Theta_0$, and distinguish the two types of intervals – see Remark 6.3. First, consider an interval where F_p has an algebraic expression. We face a purely algebraic problem, and deciding whether $F_p^{'}(\alpha_i) \neq 0$ can be done using classical bounds, e.g. Mahler bounds [LPY05, YYD+10]. Second, consider an interval where F_p has a transcendental expression. Then, $F_p^{'}(\alpha_i)$ can be rewritten as a polynomial of degree $p-1$ in π. Lindemann's theorem on the transcendence of π implies that $F_p^{'}(\alpha_i) \neq 0$.

Consider now the case where $\alpha_i \in \Theta^{\pm}$, that is $\alpha_i = \alpha_j \pm \pi$. Each individual term $f_i{}'(\alpha_i)$ also has the form $(c_i\pi + q_i)^{p-1}$, with $c_i \in \mathbb{N}$ and $q_i \in \mathbb{Q}$, so that the latter case also applies. □

6.3.2 Algorithm

Upon creating and sorting the set Θ, which has complexity $O(n \log n)$, the algorithm involves four steps for each interval in $\mathcal{I}$.

Identify the intervals where $F_p^{'}$ vanishes. By lemmas 6.1 and 6.2, there is at most one local minimum per interval, which requires checking the signs of $F_p^{'}$ to the right and left bounds of an interval (α_j, α_{j+1}). Using the functional forms encoded in vector C, computing these derivatives has the same complexity as the previous step. However, this step calls for two important comments:

- For $\alpha_i \in \Theta$, checking whether $F_p^{'}(\alpha_i) \neq 0$ is decidable – Lemma 6.3. However, the arithmetic nature of the number α_i must be taken into account, as rational numbers (input angles) and transcendental numbers (antipodal points) must be dealt with using different arithmetic techniques. See below.
- Not all intervals (α_j, α_{j+1}) can provide a root. Indeed, once $F_p^{'}(\alpha_i) > 0$, since the individual second order derivatives are positive, $F_p^{'}$ cannot vanish until one crosses one $\alpha_j \in \Theta^{\pm}$. As we shall see, this observation is easily accommodated in Algorithm 7.

In the following, we denote $\mathrm{SD}(p-1)$ the cost of deciding the sign (negative, zero, positive) of $F_p^{'}(\theta)$, for $\theta \in \Theta$.

Compute the unique root of $F_p^{'}$. Since $F_p^{'}$ is piecewise polynomial, finding its real root has constant time complexity for $p \leq 5$. Otherwise, a numerical method can be used [KRS16]. In the following, we denote $\mathrm{RF}(p-1)$ the cost of isolating the real root of a degree $p-1$ polynomial.

Evaluate F_p at a local minimum. Once the angle θ_m corresponding to a local minimum has been computed, we evaluate $F_p(\theta_m)$ using Eq. 6.10. This evaluation has $O(p)$ complexity since the coefficients of the polynomial are known.

Maintain the polynomials F_p and $F_p^{'}$. Following Eqs. 6.10 and 6.11, the function and its derivative only change when crossing an angle from Θ. At such an angle, updating the vectors B and C has complexity $O(p)$. Overall, this step therefore has complexity $O(np)$.

We summarize with the following output-sensitive complexity:

Theorem. 6.1. *Algorithm 7 computes the p-mean with $O(n \log n + np + nSD(p-1) + kRF(p-1) + kp)$ complexity, with k the number of local minima of F_p.*

6.3.3 Generic implementation

In the following, we present an implementation of our algorithm based on predicates, i.e. functions deciding branching points.

Pseudo-code, predicates and constructions Our algorithm (Algo. 7) takes as input a list of angular values (in degrees or radians) and the value of p. Following Remark 6.1, an optional file containing the weights may be passed. If $p > 5$, we take for granted an algorithm computing the root of $F_p^{'}$ on an interval. As a default, we resort to a bisection method which divides the interval into two, checks which side contains the unique root of $F_p^{'}$, and iterates until the width of the interval is less than some user specified value τ (supporting information (SI) Algo. 9). The interval returned is called the *root isolation interval*. Our algorithm was implemented in generic C++ in the Structural Bioinformatics Library [CD17], as a template class whose main parameter is a geometric kernel providing the required predicates and constructions. We now discuss these–see Sec. 6.3.4 for their robust implementation.

Predicates. The algorithm involves two predicates:

- `Sign`$(F_p^{'}(\theta))$. Predicate used to determine the sign of the $F_p^{'}(\theta)$ with $\theta \in [0, 2\pi)$ (SI Algo. 9).
- `Interval_too_wide`(θ_l, θ_r). Predicate used to determine whether the root isolation interval has width less than τ (SI Algo. 9). It is true if $\theta_r - \theta_l > \tau$, and false otherwise.

Constructions.

- **Updating representations.**. Updating the coefficients in B and C is necessary at each $\alpha_i \in \Theta$: for $F_p(\theta)$ (resp. $F_p^{'}(\theta)$), we subtract the contribution of $f_i(\theta)$ (resp. $f_i^{'}(\theta)$) before α_i, and add that of $f_i(\theta)$ (resp. $f_i^{'}(\theta)$) after α_i.
- **Find_root**. To computing the root of $F_p^{'}$ on an interval (α_j, α_{j+1}), we resort to a bisection method $p > 3$ (SI Algo. 9), with radical based formulae otherwise.

Remark 6.4. *A kernel based on floating point number types, the* `double` *type in our case, is easily assembled, see* `SBL::GT::Inexact_predicates_kernel_for_frechet_mean` *in SI Sec. 6.3.5. As noticed earlier, it comes with no guarantee. In particular, the algorithm may terminate with an erroneous result if selected predicates are falsely evaluated.*

6.3.4 Robust implementation based on exact predicates

Number types for lazy evaluations. Following the Exact Geometric Computation exact predicates are gathered in a *kernel*. We circumvent rounding errors using interval number types which are certified to contain the exact value of interest. That is, an expression x is represented by the interval $[\underline{x}, \overline{x}] \ni x$. The bounds of these intervals may have a fixed precision, which corresponds to the CGAL::Interval_nt number type [cga]. Or the bounds may be multiprecision, e.g. `Gmpfr` from `Mpfr` [FHL+07], which corresponds to the CGAL::Gmpfi type [cga]. We now explain how these types are used to code exact predicates.

The `Sign` predicate. We distinguish the algebraic and transcendental cases, performing multiprecision calculations only if needed (Fig. 6.3).

Algorithm 7 p-mean calculation: generic algorithm for $p > 1$ in the real RAM model

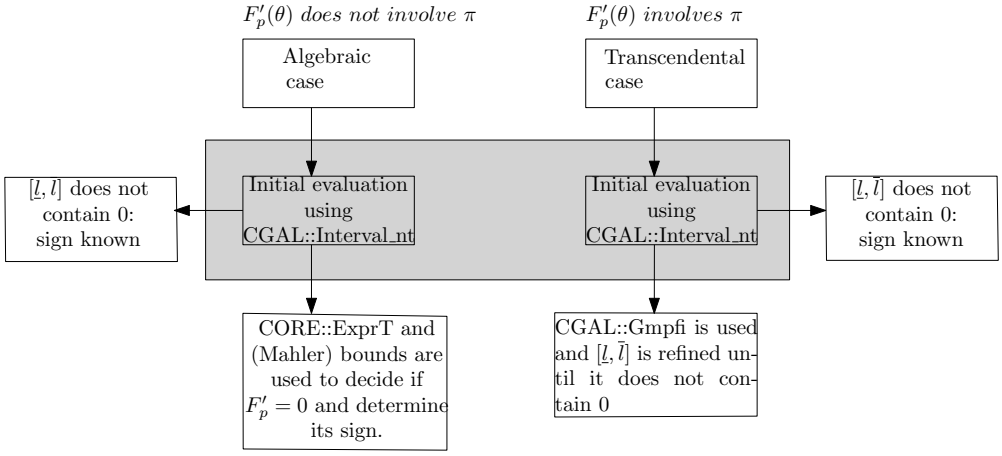
Θ: $vector[1, 2n]$ containing all the angles
B: $vector[1, p+1]$ to store the coefficients of the polynomial $F_p(\theta)$ Eq. 6.10
C: $vector[1, p]$ to store the coefficients of the polynomial $F_p^{'}(\theta)$ Eq. 6.11
θ^* // Angle corresponding to the global minimum of F_p
Root_remains = true // flag indicating whether a root must be sought on (α_j, α_{j+1})

// Initialization
Compute $\Theta^{\pm}$ and form sorted Θ
α_0: first angle in Θ
Store the coefficients of F_p into the vector B for the interval $(0, \alpha_0)$
Store the coefficients of $F_p^{'}$ into vector C for the interval $(0, \alpha_0)$
Compute $l \leftarrow F_p^{'}(\theta)$ for $\theta \rightarrow 0^+$ using Eq. 6.11 and vector C
Update_root(`Sign`(l))//Updates Root_remains see SI Algo. 8
if `Sign`(l) is null **then**
  Compute $F_p(0)$ using vector B and Eq. 6.10, and possibly update θ^*.

// For each angle, handle {interval ending, coefficients in B and C, interval starting}
for all α_i **in** Θ **do**
  if Root_remains **then**
    Compute $r \leftarrow F_p^{'}(\theta)$ for $\theta \rightarrow \alpha_i^-$ using Eq. 6.11 and vector C
    Update_root(`Sign`(r))//Updates Root_remains see Algo. SI 8
    if `Sign`(r) is positive **then**
      $\theta_c \leftarrow$ **Find_root**(α_{i-1}, α_i)
      Compute $F_p(\theta_c)$ using vector B and Eq. 6.10, and possibly update θ^*.
    else if `Sign`(r) is null **then**
      Compute $F_p(\alpha_i)$ using vector B and Eq. 6.10, and possibly update θ^*.
  Update the coefficients of F_p stored in vector B upon crossing α_i
  Update the coefficients of $F_p^{'}$ stored in vector C upon crossing α_i
  if $\alpha_i \in \Theta^{\pm}$ **then**
    Compute $l \leftarrow F_p^{'}(\theta)$ for $\theta \rightarrow \alpha_i^+$ using Eq. 6.11 and vector C
    Update_root(`Sign`(l))//Updates Root_remains see SI Algo. 8
    if `Sign`(l) is null **then**
      Compute $F_p(\alpha_i)$ using vector B and Eq. 6.10, and possibly update θ^*.

// Process the interval ending at 2π
Compute $r \leftarrow F_p^{'}(\theta)$ for $\theta \rightarrow 2\pi^-$ using Eq. 6.11 and vector C
if Root_remains **then**
  if `Sign`(r) is positive **then**
    $\theta_c \leftarrow$ **Find_root**$(\theta_{2n}, 2\pi)$
    Compute $F_p(\theta_c)$ using vector B and Eq. 6.10, and possibly update θ^*
  else if `Sign`(r) is null **then**
    Compute $F_p(2\pi)$ using vector B and Eq. 6.10, and possibly update θ^*.

Figure 6.3 Number types used in the `Sign` predicate. Note that CGAL::Interval_nt is used in the algebraic and transcendental cases, while the remaining number types are only used if required.



•**Transcendental case: multiprecision interval arithmetic.** When F_p is transcendental and α_i rational, $F_p^{'}(\alpha_i)$ is positive or negative (lemma 6.3). Another case where $F_p^{'}(\alpha_i) \neq 0$ is when $\alpha_i \in \Theta^{\pm}$. In our implementation this situation is faced in two cases. First, in the main algorithm (Algo. 7), `Sign`(l) or `Sign`(r): l and r are transcendental if $\alpha_i \in \Theta^{\pm}$. Second, in the root finding algorithm(SI Algo. 9), `Sign`$(F_p^{'}(c))$: c is transcendental if α_{i-1} or $\alpha_i \in \Theta^{\pm}$. In both cases, we proceed in a lazy way: first, we try to conclude using CGAL::Interval_nt; if this interval contains zero, we switch to CGAL::Gmpfi (Fig. 6.3), refine the interval bounds, and conclude. Refining the interval consists of iteratively doubling the number of bits used to describe all numbers–including π, until a conclusion can be reached.

•**Algebraic case: zero separation bounds.** When F_p has a rational expression and α_i is rational, `Sign`$(F_p^{'}(\alpha_i))$ may be zero (SI Fig. 6.7). In this case, an input angle may also correspond to a local minimum of F_p. To decide whether $F_p^{'}(\alpha_i) = 0$, we resort to zero separation bounds and multiprecision interval arithmetic.

Let us consider $F_p^{'}(\alpha_i)$ as an arithmetic expression E, using a number of authorized operations($\pm, \times, /$ in our case). A separation bound is a function *sep* such that the value ξ of expression E is lower bounded by $sep(E)$ in the following manner:

$$If\ \xi \neq 0 \text{ then } \text{sep}(E) \leq |\xi| \tag{6.12}$$

Considering $\tilde{\xi}$ an approximation of ξ and Δ an upper bounded error $|\tilde{\xi} - \xi|$.

$$If\ |\tilde{\xi}| + \Delta < \text{sep}(E) \text{ then } \xi = 0. \tag{6.13}$$

Practically, we proceed in a lazy way, in two steps (Fig. 6.3). First, using CGAL::Interval_nt with double precision, we check whether we can conclude on $F_p^{'}(\alpha_i) \neq 0$. If not–the interval contains zero, we use CORE::ExprT[KLPY99] to determine the zero separation bound and decide if $F_p^{'}(\alpha_i) = 0$. If not, we finally determine the sign.

Predicate `Interval_too_wide`(θ_l, θ_r). Returns true when $\underline{\theta_r} - \overline{\theta_l} > \tau$, false if $\overline{\theta_r} - \underline{\theta_l} \leq \tau$. Similarly to the sign predicate, we distinguish the transcendental and algebraic cases to check whether $\theta_l - \theta_r - \tau = 0$. Supposing τ and Θ_0 are rational $\theta_l - \theta_r - \tau$ is transcendental if the initial α_{i-1} or $\alpha_i \in \Theta^{\pm}$. If transcendental the interval is refined in the same way as the transcendental case of the `Sign` predicate. Otherwise the expression is algebraic and the precision is raised until an exact computation can be performed.

6.3.5 Software availability

The source code is available in the package *Frechet mean for* S^1 of the Structural Bioinformatics Library (SBL), a library proposing state-of-the art methods in computational structural biology [CD17], see `https://sbl.inria.fr/doc/Frechet_mean_S1-user-manual.html` and `https://sbl.inria.fr/`.

For end-users, the package provides executables corresponding to the robust and non-robust implementations. Given a list of angles and the value of p, the program returns sorted list of pairs (angular value of local minimum, function value) by increasing value of F_p. A Jupyter notebook `Frechet_mean_S1.ipynb` using `SAGE` (`https://www.sagemath.org/`) is also provided.

For developers, The C++ code of our algorithm is provided in the class
`SBL::GT::Frechet_mean_S1`, which is templated by the kernel. Two kernels are provided, namely (i) Non-robust kernel: `SBL::GT::Inexact_predicates_kernel_for_frechet_mean`. A plain floating point(double) number type is used, and (ii) Robust kernel:
`SBL::GT::Lazy_exact_predicates_kernel _for_frechet_mean`. See Sec. 6.3.4.

6.4 Experiments

6.4.1 Overview

Our experiments target three aspects, namely (i) robustness, (ii), comparison of the Fréchet mean against the classical circular mean, and (iii) computational complexity. Practically, three sets of angles are used. (Dataset 1) Randomly generated angles. (Dataset 2) So-called dihedral angles χ_i in proteins, defined by 4 consecutive atoms on the side chains of amino acids. (Recall that a protein is a polymer of amino acids, and that the 20 natural a.a. differ by their so-called side chains. See Fig. 6.6 for an example.) These angles are known to be dependent, and correlations between them are key to reduce the dimensionality of the conformation space of proteins [TWS$^+$10]. Using the Protein Data Bank, we retained 27093 PDB files with a resolution of 3 angstroms or better. For all polypeptide chains in these files, we computed all dihedral angles of all standard (20) amino-acids. This results in 240 classes of dihedral angles, containing from 50,227 to 439,793 observations. (Dataset 3) Also protein dihedral angles, but from a so-called *rotamer* library [SDJ11]. Rotamers (rotational isomers) are preferred conformations adopted by side chains, used to characterize protein conformations.

Note that in all cases, angles being given with finite precision (they are derived from experimentally determined atomic coordinates), they are treated as rational numbers.

6.4.2 Robustness

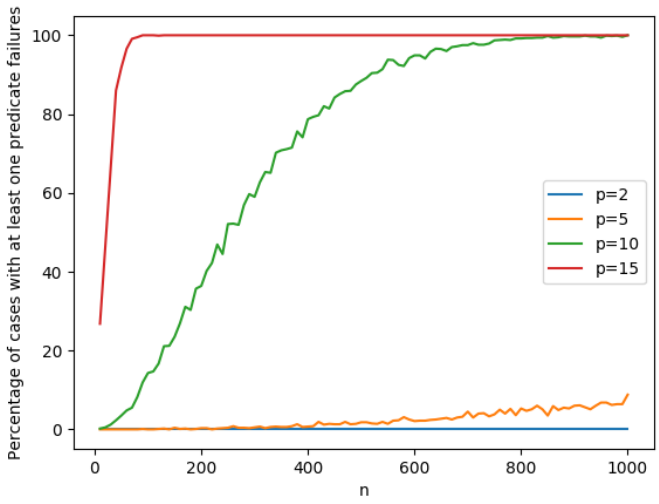
Using our robust interval-based implementation, we count the fraction of cases for which at least one predicate triggers refinement during an execution. We use sets of $n \in [10, 1000]$ angles generated uniformly at random in $[0, 2\pi)$, and perform 1000 repeats for each value of n (SI Fig. 6.4). For large values of p, whenever $n > 1000$, all executions require interval refinement. Even for $p = 2$ and $n = 10^5$, refinement is triggered in 1.3% of the cases. In all the cases where refinement was triggered, doubling the precision was sufficient to solve the predicate.

6.4.3 Fréchet mean

Fréchet mean versus circular mean. A classical way to estimate the circular mean of a set of angles is the *resultant* or *circular mean*, defined as follows [MJ09]:

$$\overline{\theta} = \text{atan2}(\sum_i \sin\theta_i/n, \sum_i \cos\theta_i/n). \tag{6.14}$$

Figure 6.4 Fraction of program runs for which at least one predicate execution triggers refinement, as a function of n and p. The number of repeats for each value of n is 1000.



The circular mean does not minimize F_p, but minimizes instead [JS01, Section 1.3]:

$$\overline{\theta} = \arg\min \sum_{i=1,\ldots,n} d(\theta_i, \theta), \text{ with } d(\alpha, \beta) = 1 - \cos(\alpha - \beta). \tag{6.15}$$

Given a set of angles, we compare the variance of these angles with respect to the Fréchet mean θ^* and the circular mean $\overline{\theta}$, respectively. Two datasets were used for such experiments: first, randomly generated sets of $n = 30$ angles uniformly at random in $[0, 2\pi)$, with 1000 repeats; second, the aforementioned dihedral angles in protein structures.

For both types of data, the variance obtained for $\overline{\theta}$ is significantly larger than that obtained for θ^*, typically up to 25% (Fig. 6.5). This shows the interest of using θ^* in data analysis in general, and to center angles prior to principal components analysis in particular.

6.4.4 Computation time and complexity

The complexity of Algorithm 7 (Theorem. 6.1) has three main components: the sorting step, the updates of vectors B and C, and the numerics. We wish in particular to determine whether the $n \log n$ sorting term dominates.

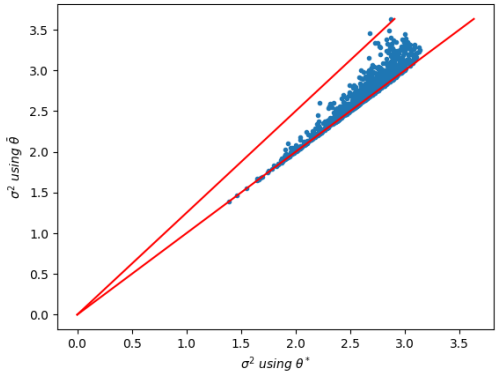
For $p \in \{2, 5, 10, 15\}$, we use sets of $n \in [10^3, 10^5]$ angles generated uniformly at random in $[0, 2\pi)$, and perform 5 repeats for each value of n. For $p = 2$, the number of angles is pushed up to $n = 10^7$, with the same number of repeats. In any case, a linear complexity is practically observed (SI Fig. 6.8) showing that for the values of n used, the constants associated with the linear time update of the data structures and the numerics take over the $n \log n$ term of the sorting step.

6.4.5 Application to clustering on the flat torus

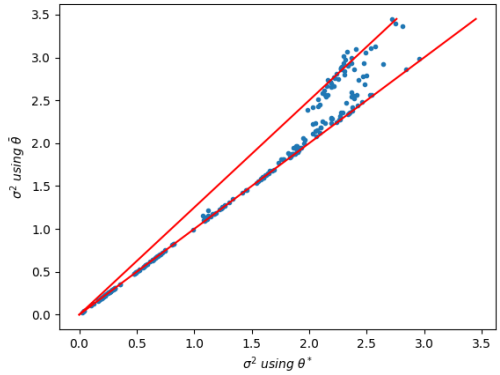
Rotamers characterize the geometry of protein side chains (Sec. 6.4.1). State of the art rotameric libraries treat the dihedral angles independently [SDJ11]. For the a.a. lysine (LYS), (Fig. 6.6(Inset)), four angles and 3 canonical values for each yield $3^4 = 81$ rotamers.

We undertake the problem of clustering side chains conformations using k-means++ [AV07]. While k-means is a classical clustering method, the problem solved is non convex and inferring the *right* number of

Figure 6.5 Variance of angles with respect to the Fréchet mean θ^* and the circular average $\overline{\theta}$. (Left) Comparison using a simulated set with $n = 30$ angles at random in $[0, 2\pi)$, with 1000 repeats. **(Right)** Comparison for the 243 classes dihedral angles in protein structures–see text. **(Both panels)** In red $y = x$ and $y = 5/4x$.



Simulated data



Torsion angles from protein structural data

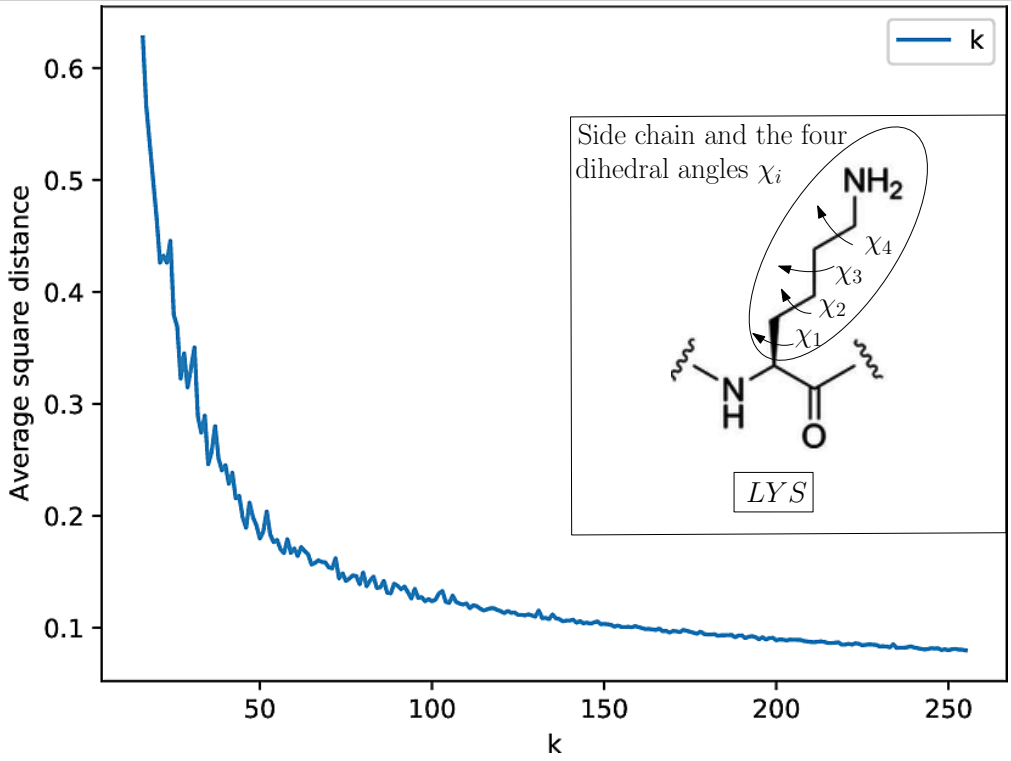
clusters is always problematic [CMTW19]. One way to mitigate this difficulty consists of tracking an elbow in the plot of the k-means functional [Ng12]. Using the lysine (LYS) a.a. as example, we work directly on the 4D flat torus $(S^1)^4$, and center the data within a cluster using our Fréchet algorithm. Varying the value of k shows a sharp decline of the k-means++ criterion circa $k = 40$, and then a gradual straightening of the average squared distance (Fig. 6.6). Working directly on the flat torus therefore makes it possible to capture correlations between individual dihedral angles. The application to a significant reduction (factor of two or so) of rotamers will be reported elsewhere.

6.5 Outlook

The Fréchet mean and the p-mean are of central importance as zero dimensional statistical summaries of data which do not live in Euclidean spaces. For the particular case of S^1, this paper develops the first robust algorithm computing the p-mean. Our algorithm is effective for large number of angular values and large values of p as well, yet, robustness requires predicates and constructions using interval multiprecision arithmetic. For the particular case of the Fréchet mean ($p = 2$), we show that the circular mean should not be used for a substitute to the circular center of mass, as it results in a significantly larger variance.

We foresee two main developments. Application-wise, our results on protein side chain conformations hint at a significant reduction (factor of two or so) of rotamers, which should prove instrumental to foster the diversity of conformational explorations. Also, our centering procedure will help generalizing principal components analysis (PCA) on the flat torus. In theoretical realm, our strategy may be used both to study the intrinsic difficulty of computing p-means (in terms of lower bounds), and to design effective algorithms. Indeed, as evidenced by the S^1 case, the combinatorial structure defined by the cut-loci of the points determines all key properties. A first case would be that of p-means on the unit sphere, for which there exist efficient algorithms to maintain arrangements of circles.

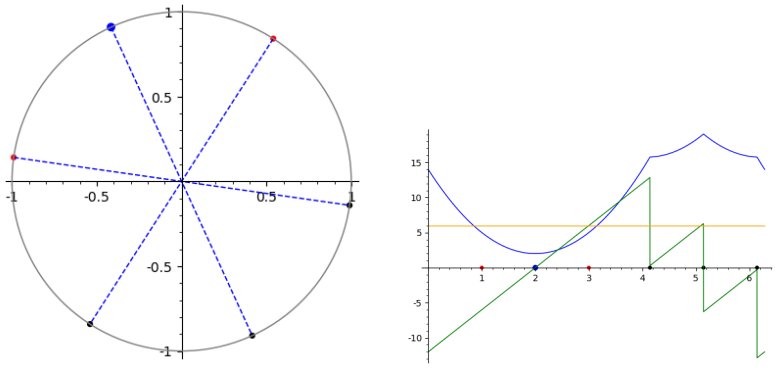
Figure 6.6 k-means++ using Fréchet mean as center performed on 4-dimensional flat torus coding the conformational space of the side chain of the Lysine amino acid. x-axis: number of clusters k. y-axis: average squared distance to the closest cluster center.



6.6 Supporting information

6.6.1 Algorithm

Figure 6.7 An interval where F_p has an algebraic expression and $F_p^{'}(\theta) = 0$. Illustration of $F_p, F_p^{'}, F_p^{''}$ for $p = 2$ and three angles $\Theta_0 = \{\theta_1 = 1, \theta_2 = 2, \theta_3 = 3\}$. Color conventions as in Fig. 6.1. In this case, $F_2'(\theta_2) = 0$, which must be numerically ascertained to ensure the correctness of the algorithm.



$n = 3, p = 2$ $\quad$ $n = 3, p = 2$

Algorithm 8 Update_root($Sign$): Updates the Root_remains buffer in main algorithm(Algo. 7)

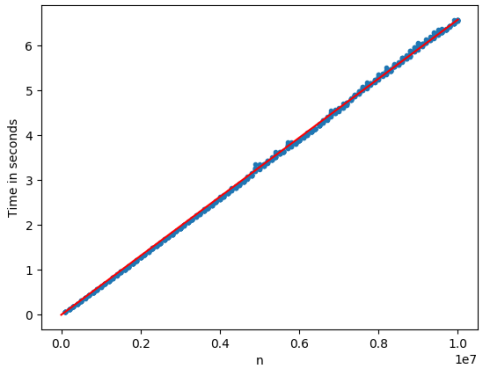
$Sign \in$ {positive,negative,null} // Sign of the derivative used to update the presence of roots on (α_j, α_{j+1})
Root_remains $\leftarrow$ true // flag indicating whether a root must be sought on (α_j, α_{j+1})
if $Sign$ is negative **then**
    Root_remains $\leftarrow$ true
else if $Sign$ is positive **then**
    Root_remains $\leftarrow$ false
else if $Sign$ is null **then**
    Root_remains $\leftarrow$ false

Algorithm 9 Find_root(α_{i-1}, α_i): generic algorithm for $p > 5$

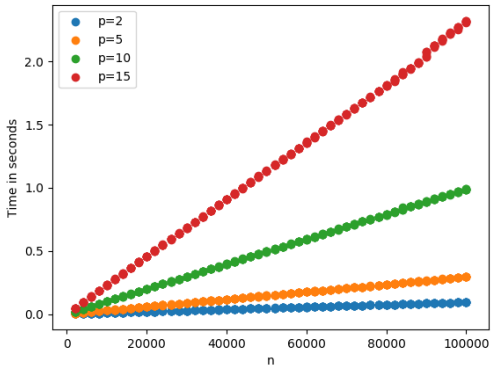
α_{i-1}, α_i: the left and right endpoints of the initial interval
τ: Threshold to stop binary search if interval is small enough
c: Center of interval g
$\theta_l \leftarrow \alpha_{i-1}, \theta_r \leftarrow \alpha_i$ // Interval being bisected
while `Interval_too_wide`(θ_l, θ_r) **do**
    $c \leftarrow \theta_l + (\theta_r - \theta_l)/2$
    $S \leftarrow$ `Sign`$(F_p^{'}(c))$
    if S is positive **then**
        $\theta_r \leftarrow c$
    else if S is negative **then**
        $\theta_l \leftarrow c$
    else if S is null **then**
        $\theta_r \leftarrow c$
        $\theta_l \leftarrow c$
$\theta_c \leftarrow \theta_l + (\theta_r - \theta_l)/2$

6.6.2 Results

Figure 6.8 Fréchet mean: computation time depending as a function of n and p. The samples of size n are generated at random angles at random in $[0, 2\pi)$. **(Left)** The red line joins $0, 0$ to the average time of the largest point sets($nmax = 10e^7$). **(Right)** Each color corresponds to a value of $p \in \{2, 5, 10, 15\}$.



$p = 2$, $nmax = 10e^7$



$p \in \{2, 5, 10, 15\}$, $nmax = 10e^5$

Chapter 7

Outlook

Current contributions. Despite the advent of deep learning methods [JEP+21, BDA+21], the sampling of unstructured or highly flexible regions is still an open problem as the data based nature of such methods biases them toward available structural data, made mainly of structured regions.

This work aims to answer this problem through a method avoiding data based solutions.

With this purpose the goal was to design a local move set that would take a conformation and return a modified one in order to iteratively explore conformation space in a Monte Carlo like algorithm.

Considering the sensitivity to variations of internal coordinates in force fields used in computational structural biology, it is necessary when designing such a move set to aim at impacting primarily the *softer* dihedral angles. As surveyed in chapter 2, many models and algorithms have been developed on the subject.

Considering the bias towards (meta-)stables states found in Protein data bank structures, and the potential under representation of transient regions, the importance of using (Tripeptide Loop Closure) TLC like methods to decrease the said bias becomes apparent.

This motivated for us the use of TLC as a building block to sample loop conformations. In chapter 3 we produced improved necessary conditions for TLC to yield solutions depending on the *legs'* positions and the internal coordinate constraints.

Having analyzed the output of TLC, in chapter 4, and showing its potential to yield a significant conformational diversity, the next step was to implement it in a more general protein loop conformational exploration method.

In order to do so the necessity for the use of multiprecision during the analysis of TLC output was considered, and a robust open source implementation was developed. It is available in the Structural Bioinformatics Library, and, together with the necessary conditions for TLC to have solutions, it is used in the generation of conformation of protein loops in chapter 5.

This yielded an efficient method generating diverse conformations tested on loops of up to 30 amino acids.

Our sampler can already be applied to generate very diverse ensembles of conformations to investigate systems with highly flexible regions.

These conformations thus generated however do not yet consider the side chains of each residue in the loops. This motivated the final contribution presented in this thesis, the first building block to produce a clustering of side chain angular configuration space was produced in the form of an efficient and robust algorithm to compute the Fréchet mean.

Future work and potential impact Despite the tightness of the necessary conditions presented in chapter 3, we have left the problem of improving the tightness of our constraints (notably using their iterated versions) as a partially open problem.

The convergence of this iterative algorithm and the design of an efficient algorithm could constitute a future contribution.

If so it would be immediately translated into more efficient versions of our chapter 5 loop-sampling algorithms as it would bring the constraint validity space $\mathcal{V}$ closer to the solution space $\mathcal{S}$.

A second improvement could be applied to the two step version of our algorithm $\mathbb{MLS}^{N_V;N_{OR}}_{\text{One—All};N_{ES}}$, it could be easily improved to be compatible with any loop size and not only loops dividable in tripeptides.

On top of this improvement, variations in internal coordinates could be considered and sampled simultaneously as the current degrees of freedom, starting with ω angle variations.

If this improved version is combined with a side chain sampling method this would yield a potentially and efficient application generating complete conformation and benefiting from the already diverse backbone conformation output.

Once this method is produced it could potentially be used to obtain reliable approximations for various thermodynamic/kinetic quantities.

To accomplish this it would be necessary to ensure that exploration methods produced this way are suitable to sample NVE and/or NVT ensembles.

The connexion with polytope volume calculations is a strong hint that this may indeed be the case, and that sampling micro-canonical ensembles may be possible.

This could lead further down the line to a response to one of the main endeavors of computational structural biology, making predictions of observables either structural, thermodynamic, or kinetic.

In any case such a method should at least be a useful tool to further the understanding of the dynamics of biomolecules.

Bibliography

[AJ91] F. Allen and O. Johnson. Automated conformational analysis from crystallographic data. 4. statistical descriptors for a distribution of torsion angles. *Acta Crystallographica Section B: Structural Science*, 47(1):62–67, 1991.

[AlQ19] Mohammed AlQuraishi. Parallelized natural extension reference frame: parallelized conversion from internal to cartesian coordinates. *Journal of computational chemistry*, 40(7):885–892, 2019.

[AM06] S.A. Adcock and A.J. McCammon. Molecular dynamics: survey of methods for simulating the activity of proteins. *Chemical reviews*, 106(5):1589–1615, 2006.

[AM14] M. Arnaudon and L. Miclo. Means in complete manifolds: uniqueness and approximation. *ESAIM: Probability and Statistics*, 18:185–206, 2014.

[AM16] M. Arnaudon and L. Miclo. A stochastic algorithm finding p-means on the circle. *Bernoulli*, 22(4):2237–2300, 2016.

[ASR88] Milton Abramowitz, Irene A Stegun, and Robert H Romer. Handbook of mathematical functions with formulas, graphs, and mathematical tables, 1988.

[AV07] D. Arthur and S. Vassilvitskii. k-means++: The advantages of careful seeding. In *ACM-SODA*, page 1035. Society for Industrial and Applied Mathematics, 2007.

[BBEJ+12] Sandro Bottaro, Wouter Boomsma, Kristoffer E. Johansson, Christian Andreetta, Thomas Hamelryck, and Jesper Ferkinghoff-Borg. Subtle Monte Carlo updates in dense molecular systems. *Journal of chemical theory and computation*, 8(2):695–702, 2012.

[BBR+87] H. Berbee, C. Boender, A. Ran, C. Scheffer, R. Smith, and J. Telgen. Hit-and-run algorithms for the identification of nonredundant linear inequalities. *Mathematical Programming*, 37(2):184–207, 1987.

[BC13] Peter Bürgisser and Felipe Cucker. *Condition: The geometry of numerical algorithms*, volume 349. Springer Science & Business Media, 2013.

[BCC21] Amélie Barozet, Pablo Chacón, and Juan Cortés. Current approaches to flexible loop modeling. *Current Research in Structural Biology*, 3:187–191, 2021.

[BD15] G. Biau and L. Devroye. *Lectures on the nearest neighbor method*. Springer, 2015.

[BDA+21] 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.

[Bet05] Marcos R Betancourt. Efficient monte carlo trial moves for polypeptide simulations. *The Journal of chemical physics*, 123(17):174905, 2005.

[BFH+13] W. Boomsma, J. Frellsen, T. Harder, S. Bottaro, K. Johansson, P. Tian, K. Stovgaard, C. Andreetta, S. Olsson, and J.B. Valentin. PHAISTOS: A framework for markov chain Monte Carlo simulation and inference of protein structure. *Journal of computational chemistry*, 34(19):1697–1705, 2013.

[Bin74] Christopher Bingham. An antipodally symmetric distribution on the sphere. *The Annals of Statistics*, pages 1201–1225, 1974.

[BK85] Robert E Bruccoleri and Martin Karplus. Chain closure with bond angle variations. *Macromolecules*, 18(12):2767–2773, 1985.

[BKD96] J. Baker, A. Kessi, and B. Delley. The generation and use of delocalized internal coordinates in geometry optimization. *The Journal of chemical physics*, 105(1):192–212, 1996.

[BKP88] C.L. Brooks, M. Karplus, and B. Montgomery Pettitt. *Advances in Chemical Physics, Proteins: A Theoretical Perspective of Dynamics, Structure, and Thermodynamics*. Wiley, 1988.

[BLvdB15] D. Budday, S. Leyendecker, and H. van den Bedem. Geometric analysis characterizes molecular rigidity in generic and non-generic protein configurations. *Journal of the Mechanics and Physics of Solids*, 83:36–47, 2015.

[BMRW01] O.M. Becker, A. D. Mackerell, B. Roux, and M. Watanabe. *Computational Biochemistry and Biophysics*. M. Dekker, 2001.

[BMS15] Rafael C Bernardi, Marcelo CR Melo, and Klaus Schulten. Enhanced sampling techniques in molecular dynamics simulations of biological systems. *Biochimica et Biophysica Acta (BBA)-General Subjects*, 1850(5):872–877, 2015.

[BMV+19] A. Barozet, K. Molloy, M. Vaisset, T. Simeon, and J. Cortés. A reinforcement-learning-based approach to enhance exhaustive protein loop sampling. *Bioinformatics*, 36(4):1099–1106, 2019.

[BT12] Carl Ivar Branden and John Tooze. *Introduction to protein structure*. Garland Science, 2012.

[BZS+12] Robert B Best, Xiao Zhu, Jihyun Shim, Pedro EM Lopes, Jeetain Mittal, Michael Feig, and Alexander D MacKerell Jr. Optimization of the additive CHARMM all-atom protein force field targeting improved sampling of the backbone ϕ, ψ and side-chain $\chi 1$ and $\chi 2$ dihedral angles. *Journal of chemical theory and computation*, 8(9):3257–3273, 2012.

[CCF22] A. Chevallier, F. Cazals, and P. Fearnhead. Efficient computation of the the volume of a polytope in high-dimensions using piecewise deterministic markov processes. In *AISTATS*, 2022.

[CCK+06] E-C. Chang, S.W. Choi, D.Y. Kwon, H. Park, and C. Yap. Shortest path amidst disc obstacles is computable. *International Journal of Computational Geometry & Applications*, 16(05n06):567–590, 2006.

[CD17] F. Cazals and T. Dreyfus. The Structural Bioinformatics Library: modeling in biomolecular science and beyond. *Bioinformatics*, 7(33):1–8, 2017.

[CDJ03] Adrian A Canutescu and Roland L Dunbrack Jr. Cyclic coordinate descent: A robotics algorithm for protein loop closure. *Protein science*, 12(5):963–972, 2003.

[CDO21] F. Cazals, B. Delmas, and T. O'Donnell. Fréchet mean and p-mean on the unit circle: decidability, algorithm, and applications to clustering on the flat torus. In D. Coudert and E. Natale, editors, *Symposium on Experimental Algorithms*, Sophia Antipolis, 2021. Lipics.

[cga] CGAL, Computational Geometry Algorithms Library. http://www.cgal.org.

[CGOS13] F. Chazal, L. Guibas, S. Oudot, and P. Skraba. Persistence-based clustering in riemannian manifolds. *J. ACM*, 60(6):1–38, 2013.

[Cha13] B. Charlier. Necessary and sufficient condition for the existence of a Fréchet mean on the circle. *ESAIM: Probability and Statistics*, 17:635–649, 2013.

[CLW+16] E. Coutsias, K. Lexa, M. Wester, S. Pollock, and M. Jacobson. Exhaustive conformational sampling of complex fused ring macrocycles using inverse kinematics. *Journal of chemical theory and computation*, 12(9):4674–4687, 2016.

[CMTW19] F. Cazals, D. Mazauric, R. Tetley, and R. Watrigant. Comparing two clusterings using matchings between clusters of clusters. *ACM J. of Experimental Algorithms*, 24(1):1–42, 2019.

[CPC22] A. Chevallier, S. Pion, and F. Cazals. Improved polytope volume calculations based on Hamiltonian Monte Carlo with boundary reflections and sweet arithmetics. *J. of Computational Geometry*, NA(NA), 2022.

[Cra89] Peter C Craig. An introduction to anadromous fishes in the alaskan arctic. *Biological Papers of the University of Alaska*, 24:27–54, 1989.

[CS04] Juan Cortés and Thierry Siméon. Sampling-based motion planning under kinematic loop-closure constraints. In *Algorithmic Foundations of Robotics VI*, pages 75–90. Springer, 2004.

[CSJD04] Evangelos A Coutsias, Chaok Seok, Matthew P Jacobson, and Ken A Dill. A kinematic view of loop closure. *Journal of computational chemistry*, 25(4):510–528, 2004.

[CSRST04] J. Cortés, T. Siméon, M. Remaud-Siméon, and V. Tran. Geometric algorithms for the conformational analysis of long protein loops. *Journal of computational chemistry*, 25(7):956–967, 2004.

[CSWD06] E. Coutsias, C. Seok, M. Wester, and K. Dill. Resultants and loop closure. *International Journal of Quantum Chemistry*, 106(1):176–189, 2006.

[CV16] B. Cousins and S. Vempala. A practical volume algorithm. *Mathematical Programming Computation*, 8(2):133–160, 2016.

[DABV+18] L. Denarie, I. Al-Bluwi, M. Vaisset, T. Siméon, and J. Cortés. Segmenting proteins into tripeptides to enhance conformational sampling with monte carlo methods. *Molecules*, 23(2):373, 2018.

[DAIRR06] Ian W Davis, W Bryan Arendall III, David C Richardson, and Jane S Richardson. The backrub motion: how protein backbone shrugs when a sidechain dances. *Structure*, 14(2):265–274, 2006.

[DB10] K. Dill and S. Bromberg. *Molecular driving forces: statistical thermodynamics in biology, chemistry, physics, and nanoscience*. Garland Science, 2010.

[DBT93] LR Dodd, TD Boone, and DN Theodorou. A concerted rotation algorithm for atomistic monte carlo simulation of polymer melts and glasses. *Molecular Physics*, 78(4):961–996, 1993.

[DCC15] Kristina Djinovic-Carugo and Oliviero Carugo. Missing strings of residues in protein crystal structures. *Intrinsically disordered proteins*, 3(1):e1095697, 2015.

[DDMHL92] Johan Desmet, Marc De Maeyer, Bart Hazes, and Ignace Lasters. The dead-end elimination theorem and its use in protein side-chain positioning. *Nature*, 356(6369):539–542, 1992.

[DGL96] L. Devroye, L. Györfi, and G. Lugosi. *A probabilistic theory of pattern recognition*, volume 31. Springer Verlag, 1996.

[DH55] Jacques Denavit and Richard S Hartenberg. A kinematic notation for lower-pair mechanisms based on matrices. 1955.

[DSUS08] A Keith Dunker, Israel Silman, Vladimir N Uversky, and Joel L Sussman. Function and structure of inherently disordered proteins. *Current opinion in structural biology*, 18(6):756–764, 2008.

[DW] L. Dicks and D.J. Wales. The use of sequence-dependent rotamer information in global optimisation of proteins. *Preprint*.

[EH91] Richard A Engh and Robert Huber. Accurate bond and angle parameters for x-ray protein structure refinement. *Acta Crystallographica Section A: Foundations of Crystallography*, 47(4):392–400, 1991.

[EH10] H. Edelsbrunner and J. Harer. *Computational topology: an introduction*. American Mathematical Soc., 2010.

[EM99] Ioannis Z Emiris and Bernard Mourrain. Computer algebra methods for studying and computing molecular conformations. *Algorithmica*, 25(2):372–402, 1999.

[FD+00] András Fiser, Richard Kinh Gian Do, et al. Modeling of loops in protein structures. *Protein science*, 9(9):1753–1773, 2000.

[Fer99] A. Fersht. *Structure and Mechanism in Protein Science: A Guide to Enzyme Catalysis and Protein Folding*. Freeman, 1999.

[FHL+07] L. Fousse, G. Hanrot, V. Lefèvre, P. Pélissier, and P. Zimmermann. MPFR: A multiple-precision binary floating-point library with correct rounding. *ACM Transactions on Mathematical Software (TOMS)*, 33(2):13, 2007.

[Fie99] M. Field. *A practical introduction to the simulation of molecular systems*. Cambridge University Press, 1999.

[Fré48] M. Fréchet. Les éléments aléatoires de nature quelconque dans un espace distancié. In *Annales de l'institut Henri Poincaré*, volume 10, pages 215–310, 1948.

[FS02] D. Frenkel and B. Smit. *Understanding molecular simulation*. Academic Press, 2002.

[Gha97] Zoubin Ghahramani. Learning dynamic bayesian networks. In *International School on Neural Networks, Initiated by IIASS and EMFCSC*, pages 168–197. Springer, 1997.

[GK73] K. Grove and H. Karcher. How to conjugatec 1-close group actions. *Mathematische Zeitschrift*, 132(1):11–20, 1973.

[GMW+03] Jeffrey J Gray, Stewart Moughon, Chu Wang, Ora Schueler-Furman, Brian Kuhlman, Carol A Rohl, and David Baker. Protein–protein docking with simultaneous optimization of rigid-body displacement and side-chain conformations. *Journal of molecular biology*, 331(1):281–299, 2003.

[GRD12] Pablo Gainza, Kyle E Roberts, and Bruce R Donald. Protein design using continuous rotamers. *PLoS Comput Biol*, 8(1):e1002335, 2012.

[GS70] Nobuhiro Go and Harold A Scheraga. Ring closure and local conformational deformations of chain molecules. *Macromolecules*, 3(2):178–187, 1970.

[GS78] N. Gō and H. Scheraga. Calculation of the conformation of cyclo-hexaglycyl. 2. application of a monte-carlo method. *Macromolecules*, 11(3):552–559, 1978.

[GSB+08] Alan P Graves, Devleena M Shivakumar, Sarah E Boyce, Matthew P Jacobson, David A Case, and Brian K Shoichet. Rescoring docking hit lists for model cavity sites: predictions and experimental testing. *Journal of molecular biology*, 377(3):914–934, 2008.

[HB05] Bosco K Ho and Robert Brasseur. The ramachandran plots of glycine and pre-proline. *BMC Struct Biol*, 5:14, Aug 2005.

[HBFdB17] A. Héliou, D. Budday, R. Fonseca, and H. Van den Bedem. Fast, clash-free rna conformational morphing using molecular junctions. *Bioinformatics*, 33(14):2114–2122, 2017.

[HBP+10] Tim Harder, Wouter Boomsma, Martin Paluszewski, Jes Frellsen, Kristoffer E Johansson, and Thomas Hamelryck. Beyond rotamers: a generative, probabilistic model of side chains in proteins. *BMC bioinformatics*, 11(1):306, 2010.

[HGD15] Mark A Hallen, Pablo Gainza, and Bruce R Donald. Compact representation of continuous energy surfaces for more efficient protein design. *Journal of chemical theory and computation*, 11(5):2292–2306, 2015.

[HHH78] Kenneth Henderson Hunt, Kenneth Henderson Hunt, and Kenneth H Hunt. *Kinematic geometry of mechanisms*, volume 7. Oxford University Press, USA, 1978.

[HJD17] Mark A Hallen, Jonathan D Jou, and Bruce R Donald. LUTE (local unpruned tuple expansion): Accurate continuously flexible protein design with general energy functions and rigid rotamer-like efficiency. *Journal of Computational Biology*, 24(6):536–546, 2017.

[HKD13] Mark A Hallen, Daniel A Keedy, and Bruce R Donald. Dead-end elimination with perturbations (deeper): A provable protein design algorithm with continuous sidechain and backbone flexibility. *Proteins: Structure, Function, and Bioinformatics*, 81(1):18–39, 2013.

[HL95] Jenn-Kang Hwang and Wen-Fa Liao. Side-chain prediction by neural networks and simulated annealing optimization. *Protein Engineering, Design and Selection*, 8(4):363–370, 1995.

[HLRR16] Bradley J Hintze, Steven M Lewis, Jane S Richardson, and David C Richardson. Molprobity's ultimate rotamer-library distributions for model validation. *Proteins: Structure, Function, and Bioinformatics*, 84(9):1177–1189, 2016.

[HMFB12] Thomas Hamelryck, Kanti Mardia, and Jesper Ferkinghoff-Borg. *Bayesian methods in structural bioinformatics*. Springer, 2012.

[HMK18] Daniel Hilger, Matthieu Masureel, and Brian K Kobilka. Structure and dynamics of gpcr signaling complexes. *Nature structural & molecular biology*, 25(1):4–12, 2018.

[HNR68] Peter E Hart, Nils J Nilsson, and Bertram Raphael. A formal basis for the heuristic determination of minimum cost paths. *IEEE transactions on Systems Science and Cybernetics*, 4(2):100–107, 1968.

[HsH15] T. Hotz and s. Huckemann. Intrinsic means on the circle: Uniqueness, locus and asymptotics. *Annals of the Institute of Statistical Mathematics*, 67(1):177–193, 2015.

[HTB03] Bosco K Ho, Annick Thomas, and Robert Brasseur. Revisiting the Ramachandran plot: hard-sphere repulsion, electrostatics, and h-bonding in the alpha-helix. *Protein Sci*, 12(11):2508–22, Nov 2003.

[JEP+21] 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.

[JPR+04] Matthew P Jacobson, David L Pincus, Chaya S Rapp, Tyler JF Day, Barry Honig, David E Shaw, and Richard A Friesner. A hierarchical approach to all-atom protein loop prediction. *Proteins: Structure, Function, and Bioinformatics*, 55(2):351–367, 2004.

[JS01] S.R. Jammalamadaka and A. SenGupta. *Topics in Circular Statistics*. World Scientific, 2001.

[JT86] T Alwyn Jones and Soren Thirup. Using known substructures in protein model building and crystallography. *The EMBO journal*, 5(4):819–822, 1986.

[JWLM78] Joel Janin, Shoshanna Wodak, Michael Levitt, and Bernard Maigret. Conformation of amino acid side-chains in proteins. *Journal of molecular biology*, 125(3):357–386, 1978.

[KD94] Patrice Koehl and Marc Delarue. Application of a self-consistent mean field theory to predict protein side-chains conformation and estimate their conformational entropy. *Journal of molecular biology*, 239(2):249–275, 1994.

[KGJH14] Gerhard Kurz, Igor Gilitschenski, Simon Julier, and Uwe D Hanebeck. Recursive bingham filter for directional estimation involving 180 degree symmetry. *Journal of Advances in Information Fusion*, 9(2):90–105, 2014.

[KGLK05] R. Kolodny, L. Guibas, M. Levitt, and P. Koehl. Inverse kinematics in biology: The protein loop closure problem. *The International Journal of Robotics Research*, 24(2-3):151–163, 2005.

[KKW12] John Kuriyan, Boyana Konforti, and David Wemmer. *The molecules of life: Physical and chemical principles*. Garland Science, 2012.

[KLPY99] V. Karamcheti, C. Li, I. Pechtchanski, and C. Yap. A core library for robust numeric and geometric computation. In *Proceedings of the fifteenth annual symposium on Computational geometry*, pages 351–359. ACM, 1999.

[KMP+08] L. Kettner, K. Mehlhorn, S. Pion, S. Schirra, and C. Yap. Classroom examples of robustness problems in geometric computations. *Computational Geometry*, 40(1):61–78, 2008.

[KRS16] A. Kobel, F. Rouillier, and M. Sagraloff. Computing real roots of real polynomials... and now for real! In *Proceedings of the ACM on International Symposium on Symbolic and Algebraic Computation*, pages 303–310. ACM, 2016.

[KS83] W. Kabsch and C. Sander. Dictionary of protein secondary structure: pattern recognition of hydrogen-bonded and geometrical features. *Biopolymers: Original Research on Biomolecules*, 22(12):2577–2637, 1983.

[KS18] Alfred Kume and Tomonari Sei. On the exact maximum likelihood inference of fisher–bingham distributions using an adjusted holonomic gradient method. *Statistics and Computing*, 28(4):835–847, 2018.

[KSDJ09] Georgii G Krivov, Maxim V Shapovalov, and Roland L Dunbrack Jr. Improved prediction of protein side-chain conformations with SCWRL4. *Proteins: Structure, Function, and Bioinformatics*, 77(4):778–795, 2009.

[Lat12] Jean-Claude Latombe. *Robot motion planning*, volume 124. Springer Science & Business Media, 2012.

[LaV06] S. LaValle. *Planning algorithms*. Cambridge university press, 2006.

[Lem20] J.A. Lemkul. Pairwise-additive and polarizable atomistic force fields for molecular dynamics simulations of proteins. *Prog Mol Biol Transl Sci*, 170:1–71, 2020.

[LL98] Andrew R Leach and Andrew P Lemon. Exploring the conformational space of protein side chains using dead-end elimination and the a* algorithm. *Proteins: Structure, Function, and Bioinformatics*, 33(2):227–239, 1998.

[LPY05] C. Li, S. Pion, and C. Yap. Recent progress in exact geometric computation. *The Journal of Logic and Algebraic Programming*, 64(1):85–111, 2005.

[LRH03] Richard A Lee, Moe Razaz, and Steven Hayward. The DynDom database of protein domain motions. *Bioinformatics*, 19(10):1290–1291, 2003.

[LS87] Z. Li and H.A. Scheraga. Monte Carlo-minimization approach to the multiple-minima problem in protein folding. *PNAS*, 84(19):6611–6615, 1987.

[LS91] Christopher Lee and S Subbiah. Prediction of protein side-chain conformation by packing optimization. *Journal of molecular biology*, 217(2):373–388, 1991.

[LSR10] T. Lelièvre, G. Stoltz, and M. Rousset. *Free energy computations: A mathematical perspective*. World Scientific, 2010.

[LV18] Y.T. Lee and S. Vempala. Convergence rate of Riemannian Hamiltonian Monte Carlo and faster polytope volume computation. In *STOC*, pages 1115–1121. ACM, 2018.

[LZZ14] Shide Liang, Chi Zhang, and Yaoqi Zhou. LEAP: Highly accurate prediction of protein loop conformations by integrating coarse-grained sampling and optimized energy scores with all-atom refinement of backbone and side chains. *Journal of computational chemistry*, 35(4):335–341, 2014.

[MAR10a] M Merced Malabanan, Tina L Amyes, and John P Richard. A role for flexible loops in enzyme catalysis. *Current opinion in structural biology*, 20(6):702–710, 2010.

[Mar10b] Kanti V Mardia. Bayesian analysis for bivariate von mises distributions. *Journal of Applied Statistics*, 37(3):515–528, 2010.

[MBdD+18] J.-M. Muller, N. Brunie, F. de Dinechin, C. Jeannerod, M. Joldes, V. Lefèvre, G. Melquiond, N. Revol, and S. Torres. Handbook of floating-point arithmetic. 2018.

[MC94] Dinesh Manocha and John F Canny. Efficient inverse kinematics for general 6r manipulators. *IEEE transactions on robotics and automation*, 10(5):648–657, 1994.

[MCK09] D.J. Mandell, E.A. Coutsias, and T. Kortemme. Sub-angstrom accuracy in protein loop reconstruction by robotics-inspired conformational sampling. *Nature Methods*, 6:551 – 552, 2009.

[MGH+03] Michael B Monagan, Keith O Geddes, K Michael Heal, George Labahn, SM Vorkoetter, James McCarron, and Paul DeMarco. Maple 9: Advanced programming guide. 2003.

[MJ09] K. Mardia and P. Jupp. *Directional statistics*, volume 494. John Wiley & Sons, 2009.

[MLL08] R James Milgram, Guanfeng Liu, and Jean-Claude Latombe. On the structure of the inverse kinematics map of a fragment of protein backbone. *Journal of computational chemistry*, 29(1):50–68, 2008.

[MNK+17] Claire Marks, Jaroslaw Nowak, Stefan Klostermann, Guy Georges, James Dunbar, Jiye Shi, Sebastian Kelm, and Charlotte M Deane. Sphinx: merging knowledge-based and ab initio approaches to improve protein loop prediction. *Bioinformatics*, 33(9):1346–1353, 2017.

[MSD18] C. Marks, J. Shi, and C. Deane. Predicting loop conformational ensembles. *Bioinformatics*, 34(6):949–956, 2018.

[MT93] M. MacArthur and J. Thornton. Conformational analysis of protein structures derived from nmr data. *Proteins: Structure, Function, and Bioinformatics*, 17(3):232–251, 1993.

[Mur12] Kevin P Murphy. *Machine learning: a probabilistic perspective*. MIT press, 2012.

[MWS+14] Kenji Mochizuki, Chris S Whittleston, Sandeep Somani, Halim Kusumaatmaja, and David J Wales. A conformational factorisation approach for estimating the binding free energies of macromolecules. *Physical Chemistry Chemical Physics*, 16(7):2842–2853, 2014.

[Ng12] A. Ng. Clustering with the k-means algorithm. *Machine Learning*, 2012.

[NOS05] Kimberly Noonan, David O'Brien, and Jack Snoeyink. Probik: Protein backbone motion by inverse kinematics. *The International Journal of Robotics Research*, 24(11):971–982, 2005.

[OC22a] T. O'Donnell and F. Cazals. Geometric constraints within tripeptides and the existence of tripeptide reconstructions. *Submitted*, 2022.

[OC22b] T. O'Donnell and F. Cazals. Protein loops sampling based on a global parameterization of the backbone conformational space. *Submitted*, 2022.

[ORC22] T. O'Donnell, C.H. Robert, and F. Cazals. Tripeptide loop closure: a detailed study of reconstructions based on Ramachandran distributions. *Proteins: structure, function, and bioinformatics*, 90(3):858–868, 2022.

[OSY+12] V. Ozenne, R. Schneider, M. Yao, J-R. Huang, L. Salmon, M. Zweckstetter, M. Jensen, and M. Blackledge. Mapping the potential energy landscape of intrinsically disordered proteins at amino acid resolution. *Journal of the American Chemical Society*, 134(36):15138–15148, 2012.

[PC94] D. Parsons and J. Canny. Geometric problems in molecular biology and robotics. In *ISMB*, pages 322–330, 1994.

[Pen18] X. Pennec. Barycentric subspace analysis on manifolds. *The Annals of Statistics*, 46(6A):2711–2746, 2018.

[PFPB79] Peter Pulay, Geza Fogarasi, Frank Pang, and James E Boggs. Systematic ab initio gradient calculation of molecular geometries, force constants, and dipole moment derivatives. *Journal of the American Chemical Society*, 101(10):2550–2560, 1979.

[PH10] Martin Paluszewski and Thomas Hamelryck. Mocapy++–a toolkit for inference and learning in dynamic bayesian networks. *BMC bioinformatics*, 11(1):1–6, 2010.

[PHR+05] J. Parsons, J.B. Holmes, J.M. Rojas, J. Tsai, and C.E.M. Strauss. Practical conversion from torsion space to cartesian space for in silico protein synthesis. *Journal of computational chemistry*, 26(10):1063–1068, 2005.

[PMW+10] Marie Pancera, Jason S McLellan, Xueling Wu, Jiang Zhu, Anita Changela, Stephen D Schmidt, Yongping Yang, Tongqing Zhou, Sanjay Phogat, John R Mascola, et al. Crystal structure of PG16 and chimeric dissection with somatically related PG9: structure-function analysis of two quaternary-specific antibodies that effectively neutralize HIV-1. *Journal of virology*, 84(16):8098–8110, 2010.

[PRT+07] J. Porta, L. Ros, F. Thomas, F. Corcho, J. Cantó, and J. Pérez. Complete maps of molecular-loop conformational spaces. *Journal of computational chemistry*, 28(13):2170–2189, 2007.

[PS85] F. Preparata and M. Shamos. *Computational geometry: an introduction*. Springer Science & Business Media, 1985.

[PYB+12] Andreas Prlić, Andrew Yates, Spencer E Bliven, Peter W Rose, Julius Jacobsen, Peter V Troshin, Mark Chapman, Jianjiong Gao, Chuan Hock Koh, Sylvain Foisy, et al. Biojava: an open-source framework for bioinformatics in 2012. *Bioinformatics*, 28(20):2693–2695, 2012.

[QH09] Guoying Qi and Steven Hayward. Database of ligand-induced domain movements in enzymes. *BMC structural biology*, 9(1):1–9, 2009.

[QZC+02] Ying Qi, Runxiang Zhao, Hongxi Cao, Xingwei Sui, Sanford B Krantz, and Z Joe Zhao. Purification and characterization of protein tyrosine phosphatase ptp-meg2. *Journal of cellular biochemistry*, 86(1):79–89, 2002.

[Ram63] Gopalasamudram Narayana Ramachandran. Stereochemistry of polypeptide chain configurations. *J. Mol. Biol.*, 7:95–99, 1963.

[RP21] Kiersten M Ruff and Rohit V Pappu. Alphafold and implications for intrinsically disordered proteins. *Journal of Molecular Biology*, 433(20):167208, 2021.

[RR90] Madhusudan Raghavan and Bernard Roth. Kinematic analysis of the 6r manipulator of general geometry. In *International symposium on robotics research*, pages 314–320. Citeseer, 1990.

[Sax79] James B Saxe. Embeddability of weighted graphs in k-space is strongly NP-hard. In *Proc. of 17th Allerton Conference in Communications, Control and Computing, Monticello, IL*, pages 480–489, 1979.

[SBMVC21] M. Simsir, I. Broutin, I. Mus-Veteau, and F. Cazals. Studying dynamics without explicit dynamics: a structure-based study of the export mechanism by AcrB. *Proteins: structure, function, and bioinformatics*, 89:259–275, 2021.

[SDJ11] Maxim V Shapovalov and Roland L Dunbrack Jr. A smoothed backbone-dependent rotamer library for proteins derived from adaptive kernel density estimates and regressions. *Structure*, 19(6):844–858, 2011.

[SK08] Colin A Smith and Tanja Kortemme. Backrub-like backbone simulation recapitulates natural protein conformational variability and improves mutant side-chain prediction. *Journal of molecular biology*, 380(4):742–756, 2008.

[SK12] Amarda Shehu and Lydia E Kavraki. Modeling structures and motions of loops in protein molecules. *Entropy*, 14(2):252–290, 2012.

[SK13] Amelie Stein and Tanja Kortemme. Improvements to robotics-inspired conformational sampling in rosetta. *PloS one*, 8(5):e63090, 2013.

[SMLL+10] D. E. Shaw, P. Maragakis, K. Lindorff-Larsen, S. Piana, R. O. Dror, M. P. Eastwood, J. A. Bank, J. M. Jumper, J. K. Salmon, Y. Shan, and W. Wriggers. Atomic-level characterization of the structural dynamics of proteins. *Science*, 330(6002):341–346, 2010.

[STH08] D. Sheppard, R. Terrell, and G. Henkelman. Optimization methods for finding minimum energy paths. *The Journal of chemical physics*, 128(13):134106, 2008.

[STM+77] ROSEMARIE Swanson, BENES L Trus, NEIL Mandel, GRETCHEN Mandel, OLGA B Kallai, and RICHARD E Dickerson. Tuna cytochrome c at 2.0 a resolution. i. ferricytochrome structure analysis. *Journal of Biological Chemistry*, 252(2):759–775, 1977.

[SW+05] Andrew J Sommese, Charles W Wampler, et al. *The Numerical solution of systems of polynomials arising in engineering and science*. World Scientific, 2005.

[SXK+13] A. Schmidt, H. Xu, A. Khan, T. O'Donnell, S. Khurana, L. King, J. Manischewitz, H. Golding, P. Suphaphiphat, A. Carfi, E. Settembre, P. Dormitzer, T. Kepler, R. Zhang, A. Moody, B. Haynes, H-X. Liao, D. Shaw, and S. Harrison. Preconfiguration of the antigen-binding site during affinity maturation of a broadly neutralizing influenza virus antibody. *PNAS*, 110(1):264–269, 2013.

[TRVD16] Clare-Louise Towse, Steven J Rysavy, Ivan M Vulovic, and Valerie Daggett. New dynamic rotamer libraries: data-driven analysis of side-chain conformational propensities. *Structure*, 24(1):187–199, 2016.

[TWS+10] D. Ting, G. Wang, M. Shapovalov, R. Mitra, M.I. Jordan, and R. Dunbrack. Neighbor-dependent ramachandran probability distributions of amino acids developed from a hierarchical dirichlet process model. *PLoS Comput Biol*, 6(4):e1000763, 2010.

[UJ03] Jakob P Ulmschneider and William L Jorgensen. Monte Carlo backbone sampling for polypeptides with variable bond angles and dihedral angles using concerted rotations and a gaussian bias. *The Journal of chemical physics*, 118(9):4261–4271, 2003.

[Uve13] Vladimir N Uversky. Unusual biophysics of intrinsically disordered proteins. *Biochimica et Biophysica Acta (BBA)-Proteins and Proteomics*, 1834(5):932–951, 2013.

[Wal03] D. J. Wales. *Energy Landscapes*. Cambridge University Press, 2003.

[WMP14] Lee-Ping Wang, Todd J Martinez, and Vijay S Pande. Building force fields: an automatic, systematic, and reproducible approach. *The journal of physical chemistry letters*, 5(11):1885–1891, 2014.

[YD95] C. Yap and T. Dubé. The exact computation paradigm. In *Computing in Euclidean Geometry*, pages 452–492. World Scientific, 1995.

[YTH+02] Jinn-Moon Yang, Chi-Hung Tsai, Ming-Jing Hwang, Huai-Kuang Tsai, Jenn-Kang Hwang, and Cheng-Yan Kao. GEM: A gaussian evolutionary method for predicting protein side-chain conformations. *Protein Science*, 11(8):1897–1907, 2002.

[YYD+10] J. Yu, C. Yap, Z. Du, S. Pion, and Hervé H. Brönnimann. The design of core 2: A library for exact numeric computation in geometry and algebra. In *International Congress on Mathematical Software*, pages 121–141. Springer, 2010.

[ZLT+12] Sheng Zhang, Sijiu Liu, Rongya Tao, Dan Wei, Lan Chen, Weihua Shen, Zhi-Hong Yu, Lina Wang, David R Jones, Xiaocheng C Dong, et al. A highly selective and potent ptp-meg2 inhibitor with therapeutic potential for type 2 diabetes. *Journal of the American Chemical Society*, 134(43):18116–18124, 2012.

[ZRST15] Stefano Zamuner, Alex Rodriguez, Flavio Seno, and Antonio Trovato. An efficient algorithm to perform local concerted movements of a chain molecule. *PloS one*, 10(3):e0118342, 2015.

[Zuc10] D.M. Zuckerman. *Statistical Physics of Biomolecules: An Introduction*. CRC Press, 2010.

[ZZLF11] Suwen Zhao, Kai Zhu, Jianing Li, and Richard A Friesner. Progress in super long loop prediction. *Proteins: Structure, Function, and Bioinformatics*, 79(10):2920–2935, 2011.