



**HAL**  
open science

# Molecular conformations and game theory

Amélie Héliou

► **To cite this version:**

Amélie Héliou. Molecular conformations and game theory. Bioinformatics [q-bio.QM]. Université Paris Saclay (COMUE), 2017. English. NNT : 2017SACLX033 . tel-01633869

**HAL Id: tel-01633869**

**<https://pastel.hal.science/tel-01633869v1>**

Submitted on 13 Nov 2017

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

NNT : 2017SACLX033

THÈSE DE DOCTORAT  
DE L'UNIVERSITÉ PARIS-SACLAY  
PRÉPARÉE À L'ÉCOLE POLYTECHNIQUE

Ecole doctorale n°580

STIC

Spécialité de doctorat : Informatique

par

**MME. AMÉLIE HÉLIOU**

Conformations moléculaires et théorie des jeux

Thèse présentée et soutenue à Amphithéâtre Sophie Germain, bâtiment Alan Turing,  
Campus de l'école Polytechnique, le 31 août 2017.

Composition du Jury :

Mme.	CHRISTINE FROIDEVAUX	Professeure Université Paris-Sud	(Présidente)
M.	JUAN CORTÉS	Directeur de recherche LAAS-CNRS	(Rapporteur)
M.	RIDA LARAKI	Directeur de recherche Université Paris-Dauphine	(Rapporteur)
Mme	MATHILDE CARPENTIER	Maître de conférences Université Pierre et Marie Curie	(Examinatrice)
M.	YANNICK VIOSSAT	Maître de conférences Université Paris-Dauphine	(Examineur)
M.	RAPHAËL GUEROIS	Chercheur CEA CEA	(Examineur)
Mme.	JOHANNE COHEN	Directrice de recherche Université Paris-Sud	(Directrice de thèse)



*À Mémé et Abuelito, qui m'ont tant donné et resteront dans  
mon coeur.*

*"El que tropieza y no cae, adelanta terreno."*

Refrán Español

*"Don't dream of winning, train for it."*

Mohamed Farah



## Remerciements

Tout d'abord, je tiens à remercier ma directrice de thèse, Johanne Cohen, qui m'a encadrée durant ces trois années. Je remercie tous les membres du jury, en particuliers, mes rapporteurs, Juan Cortés et Rida Laraki, pour avoir attentivement relu et commenté ce manuscrit.

Je suis très reconnaissante envers Henry van den Bedem qui a accepté de m'encadrer sur le projet de mouvement de l'ARN. Il m'a ainsi permis de travailler avec son équipe (Dominik et Rasmus) principalement par visio-conférences mais aussi en m'invitant plusieurs fois pendant quelques mois à Stanford. Ces voyages ont été très enrichissants tant sur le plan scientifique que humain. J'en profite aussi pour remercier mon équipe AMIBio et le centre France-Stanford qui ont financé ces voyages.

Je remercie sincèrement Panayotis Mertikopoulos. Il a eu la patience de m'expliquer et de me faire comprendre des notions clés qui sont à la base des travaux en théorie des jeux présentés dans ce manuscrit. Je ne l'ai malheureusement rencontré qu'en début de troisième année, les sujets sur lesquels nous travaillons me plaisent beaucoup, je lui suis reconnaissante de m'avoir acceptée en Post-doc.

Je remercie mes co-bureaux successifs (Pauline, Nicolas, Bibeck, Frédéric, Jorge, Eliott et Laurent) pour avoir partagé mes journées. Je remercie également Afaf, Juraj, Julien, Élise et tous les membres de mon équipe et de l'équipe Grace pour les déjeuners, les discussions et gouters qui ont égayé mes journées. En particulier, Yann, qui a su prendre le temps de m'écouter et de me conseiller et Mireille qui a soutenu tous mes projets.

Je remercie tout particulièrement Steve qui m'a guidée dans mes débuts difficiles.

Je dois beaucoup aussi à Patrick, Kazia, et les coureurs d'Orsay, mes entraîneurs d'athlétisme et amis qui n'ont jamais cessé de m'encourager et de croire en moi. Une piqure de rappel régulière pour garder en tête que le travail est important mais qu'il y a beaucoup d'autres choses pour s'épanouir.

Je remercie mes amis qui m'ont soutenue et donnée confiance dans les moments difficiles. En particulier, je remercie Chouchou (Thomas), mon colocataire, pour avoir si bien géré la maison et Kiki (Romain) pour les repas décontractés.

Enfin, je ne remercierai jamais assez ma famille pour leur soutien inconditionnel. En particulier, je remercie ma mère, qui a toujours été là pour moi et m'a soutenue dans mes choix même s'il s'agissait de partir sur un autre continent pendant trois mois. Pour finir, je dois tant à ma soeur, Alice, mon éternelle complice, toujours prête à m'épauler et m'encourager.



# Résumé substantiel

Les protéines et acides ribonucléiques sont les principaux acteurs de nombreux processus cellulaires. Comprendre leurs fonctions, structures et interactions est un challenge important. Les méthodes expérimentales fournissent des informations sur la structure et la dynamique des molécules. Cependant les méthodes expérimentales sont limitées par le nombre de molécules qu'elles peuvent observer et les moyens qu'elles requièrent. Les travaux présentés dans ce document ont pour objectif de fournir de nouveaux outils pour l'analyse et la prédiction de la structure de molécules telles que les protéines et les ARNs (*Acide Ribonucléiques*).

Les protéines tout comme les ARNs sont des chaînes d'éléments : 20 acides aminés pour les protéines et 4 nucléotides pour les ARNs. Ces éléments interagissent entre eux pour former des liaisons fortes qui aboutissent à la structure secondaire. C'est-à-dire la formation d'hélices alpha et de feuillets beta chez la protéine et d'hélices et de boucles chez l'ARN. Enfin des interactions plus faibles, dont certaines avec le solvant, agencent les structures secondaires dans l'espace. Bien que non systématique, le lien entre la séquence et la structure des molécules est bien connu. Cependant alors que la prédiction de structure secondaire donne de bons résultats, il est encore très difficile de prédire la structure en trois dimensions depuis la séquence.

Dans un premier temps, nous nous sommes intéressés à l'extension du programme KGS (*Kino-Geometric Sampling*) aux calculs des transitions d'un ARN entre deux de ses conformations. Pour cela nous considérons que la structure secondaire est connue et stable. Puis nous déterminons les degrés de liberté de la molécule qui respectent les contraintes de conservation de sa structure secondaire. Notre procédure est basée sur la cinétique inverse pour trouver une transition entre deux conformations d'un ARN. Nous obtenons des résultats comparables à l'état de l'art, ce qui montre que notre sélection des degrés de liberté est pertinente. De plus, nous utilisons des données partielles, ce qui permet d'utiliser différents types de résultats expérimentaux.

Dans un second temps, nous abordons le problème du repliement protéique par une approche innovante de théorie des jeux. Nous représentons une protéine par un jeu où les joueurs sont les acides aminés et les stratégies sont les angles dièdres. La prédiction de structure peut alors être vue comme la recherche d'un équilibre dans un jeu de potentiel multi-joueur où la fonction de potentiel correspond à la qualité du repliement.

Pour cela, nous effectuons d'abord une analyse de la convergence de certains algorithmes. Nous nous intéressons particulièrement aux algorithmes de non-regret pour des jeux itératifs, non-coopératifs, avec un nombre fini de joueur et de stratégies par joueur. De tels algorithmes garantissent à chaque joueur un regret (différence de ce qu'il a gagné avec ce qu'il aurait pu gagner en conservant la meilleure des stratégies) sous-linéaire. Ces algorithmes convergent en moyenne vers la classe des CCE (*coarse correlated equilibria*) qui contient tous les équilibres de Nash (aucun joueur n'a intérêt à changer seul) mais aussi des équilibres peu rationnels (des stratégies dites *dominées* sont utilisées). Nous étudions en particulier un algorithme de



poids exponentiels appelé *Hedge*, et nous montrons qu'il garantit l'élimination des stratégies dominées et la convergence locale vers un équilibre de Nash ou la convergence globale vers cet équilibre s'il est unique. Puis, en limitant notre analyse aux jeux de potentiel, nous montrons qu'une classe plus large d'algorithmes, les algorithmes de régularisation, convergent vers un équilibre de Nash presque sûrement. Les résultats de convergence sont conservés lorsque les joueurs n'ont accès qu'à une information imparfaite et/ou bruitée. Nous avons ensuite appliqué cet algorithme au repliement protéique et obtenu des résultats prometteurs.

Nos travaux appellent à de nouveaux développements. Il serait particulièrement intéressant d'utiliser les positions relatives des atomes (distances entre eux) pour calculer la transition entre deux conformations plutôt que leur position absolue. Cette amélioration va nous permettre dans un futur proche de développer un outil d'analyse des données *DEER*. En ce qui concerne le repliement protéique nous devons améliorer la fonction d'utilité pour qu'elle reflète mieux le repliement de la protéine. Nous nous confrontons aussi au problème de non-optimalité de certains équilibres de Nash : un équilibre de Nash peut être éloigné de l'optimum.

# Contents

<b>Remerciements</b>	<b>v</b>
<b>Résumé substantiel</b>	<b>vi</b>
<b>List of Figures</b>	<b>xiii</b>
<b>List of Tables</b>	<b>xv</b>
<b>Foreword</b>	<b>1</b>
<b>1 Some background in structural biology</b>	<b>3</b>
1.1 Proteins and RNAs presentation . . . . .	3
1.1.1 Sequences . . . . .	4
1.1.1.1 Protein sequence . . . . .	4
1.1.1.2 RNA sequence . . . . .	5
1.1.2 Secondary structures . . . . .	6
1.1.2.1 Protein secondary structures . . . . .	7
1.1.2.2 RNA secondary structures . . . . .	8
1.1.3 Three dimensional structures . . . . .	9
1.1.4 Interactions . . . . .	10
1.2 Structures : determination, prediction and sampling . . . . .	11
1.2.1 Experimental techniques . . . . .	12
1.2.1.1 X-RAY . . . . .	12
1.2.1.2 NMR . . . . .	12
1.2.1.3 cryo-EM . . . . .	13
1.2.1.4 SAXS . . . . .	13
1.2.1.5 DEER . . . . .	14
1.2.2 Structure Prediction . . . . .	14
1.2.2.1 Coarse-grained approaches . . . . .	15
1.2.2.2 Template-based methods . . . . .	15
1.2.2.3 Template-free methods . . . . .	16
1.2.3 Motion modeling . . . . .	17
1.2.3.1 Torsional and cartesian coordinates . . . . .	17
1.2.3.2 Molecular dynamics . . . . .	18
1.2.3.3 Normal Mode Analysis . . . . .	20
1.2.4 Sampling and morphing strategies . . . . .	21
1.2.4.1 Monte-Carlo . . . . .	21
1.2.4.2 Tree exploration . . . . .	22
1.2.4.3 Other approaches . . . . .	23

<b>2</b>	<b>An introduction to Game theory</b>	<b>25</b>
2.1	Games presentation . . . . .	25
2.1.1	Extensive form . . . . .	26
2.1.2	Normal form . . . . .	27
2.2	Dominance and Equilibria . . . . .	28
2.2.1	Dominated strategy . . . . .	28
2.2.2	Nash Equilibria . . . . .	29
2.2.3	Correlated Equilibria . . . . .	30
2.2.4	Coarse Correlated Equilibria . . . . .	31
2.3	Iterative games and regret minimization . . . . .	31
2.3.1	Iterative games . . . . .	31
2.3.2	Regret minimization . . . . .	32
2.3.2.1	Internal regret . . . . .	32
2.3.2.2	External regret . . . . .	32
2.3.2.3	Expected Regret . . . . .	32
2.3.2.4	Pseudo Regret . . . . .	33
2.4	Games under study . . . . .	33
2.4.1	Bandit and semi-bandit game . . . . .	33
2.4.2	Exponential learning and no-regret . . . . .	33
<b>3</b>	<b>RNA conformation explorations with a robotics approach</b>	<b>35</b>
3.1	Molecule representation . . . . .	36
3.1.1	Kinematic graph . . . . .	36
3.1.2	Secondary structure conservation . . . . .	37
3.2	Clash-free directed motion . . . . .	40
3.2.1	Dynamic Clash-avoiding Constraints . . . . .	40
3.2.2	Directed motion . . . . .	41
3.2.2.1	Least square motion . . . . .	41
3.2.2.2	Comparison with nullspace projection . . . . .	42
3.2.3	Procedure overview and dataset . . . . .	47
3.2.3.1	Procedure overview . . . . .	47
3.2.3.2	Dataset . . . . .	48
3.3	Results . . . . .	48
3.3.1	Benchmark results . . . . .	48
3.3.1.1	Comparison to iMOD . . . . .	48
3.3.1.2	Performances using every fifth C5 atom . . . . .	50
3.3.1.3	Maintaining secondary structure between substates . . . . .	51
3.3.1.4	Distance to the heavy atom goal positions . . . . .	52
3.3.1.5	Study of 3D helical arrangements . . . . .	54
3.3.2	Conformational exchange of riboswitch <i>glnA</i> . . . . .	55
3.3.3	Run time . . . . .	56
3.4	Perspectives and conclusions . . . . .	57
3.4.1	Relative distances . . . . .	58
3.4.1.1	Relative distances approach . . . . .	58
3.4.1.2	Randomization . . . . .	58
3.4.1.3	Preliminary results . . . . .	58
3.4.1.4	Projected applications . . . . .	58
3.4.2	Conclusions . . . . .	59

<b>4</b>	<b>Finding alternate conformations of ligands</b>	<b>61</b>
4.1	Introduction . . . . .	61
4.1.1	Fitting the density for proteins . . . . .	61
4.1.2	Alternate conformation for ligands . . . . .	62
4.2	Method presentation . . . . .	62
4.2.1	Ligand sampling . . . . .	62
4.2.2	Fitting the density . . . . .	64
4.3	Preliminary results and Discussions . . . . .	65
4.3.1	Results . . . . .	65
4.3.2	Discussion . . . . .	65
<b>5</b>	<b>Multi-player exponential learning with full information</b>	<b>67</b>
5.1	Preliminaries . . . . .	68
5.1.1	Notations . . . . .	68
5.1.2	HEDGE algorithm . . . . .	70
5.2	Convergence in generic games . . . . .	71
5.2.1	Elimination of dominated strategies . . . . .	71
5.2.1.1	Technical Lemma . . . . .	71
5.2.1.2	Results . . . . .	73
5.2.1.3	Discussions . . . . .	74
5.2.2	Local convergence towards strict equilibrium . . . . .	74
5.2.2.1	Logit map properties . . . . .	74
5.2.2.2	Result . . . . .	75
5.2.2.3	Discussions . . . . .	76
5.2.3	Global convergence towards an unique strict equilibrium . . . . .	76
5.2.3.1	Kullback-Leiber divergence properties . . . . .	77
5.2.3.2	Results . . . . .	78
5.2.3.3	Discussion . . . . .	80
5.3	Global convergence in generic potential games . . . . .	80
5.3.1	Asymptotic Pseudotrajectory . . . . .	81
5.3.2	Global convergence in the continuous dynamics . . . . .	82
5.3.3	Global convergence in the potential game . . . . .	83
5.3.4	Convergence Rate . . . . .	84
5.4	Conclusions . . . . .	86
<b>6</b>	<b>Extension to regularized learning and partial information</b>	<b>87</b>
6.1	Dual-averaging regularized learning . . . . .	87
6.1.1	Preliminaries . . . . .	87
6.1.1.1	Penalty functions . . . . .	87
6.1.1.2	Algorithm . . . . .	89
6.1.2	Convergence in potential games . . . . .	89
6.1.2.1	Asymptotic Pseudotrajectory . . . . .	90
6.1.2.2	Global convergence with the continuous dynamics . . . . .	91
6.1.3	Global convergence in the potential game . . . . .	93
6.1.4	Convergence time . . . . .	93
6.2	Bandit setting . . . . .	94
6.2.1	$\epsilon$ -dynamic and $\epsilon$ -equilibrium . . . . .	94
6.2.2	Global convergence . . . . .	96
6.2.2.1	Asymptotic pseudo trajectory . . . . .	96
6.2.2.2	Global convergence with the continuous dynamics . . . . .	97
6.2.3	Global convergence in the potential game . . . . .	99

6.2.4	Convergence time . . . . .	99
6.3	Conclusions . . . . .	100
<b>7</b>	<b>From game theory to protein folding</b>	<b>101</b>
7.1	Game setting . . . . .	101
7.1.1	Dataset . . . . .	102
7.1.1.1	Secondary structure . . . . .	102
7.1.1.2	Type of amino acids . . . . .	102
7.1.2	Players . . . . .	102
7.1.3	Strategies . . . . .	103
7.1.3.1	Strategies in the "all AAs" setting . . . . .	104
7.1.3.2	Strategies in the "tripeptide" setting . . . . .	104
7.1.4	Utility functions . . . . .	104
7.1.4.1	Root Mean Squared Distance . . . . .	105
7.1.4.2	BC-score . . . . .	105
7.1.4.3	Distance based . . . . .	106
7.2	Algorithm and implementation . . . . .	107
7.2.1	Algorithm . . . . .	107
7.2.2	Implementation . . . . .	107
7.3	Results and discussions . . . . .	107
7.3.1	Results . . . . .	108
7.3.2	Discussion . . . . .	109
<b>8</b>	<b>Conclusions and perspectives</b>	<b>111</b>
	<b>Bibliography</b>	<b>115</b>
<b>A</b>	<b>Convex conjugates</b>	<b>125</b>
A.1	<i>Gibbs entropy</i> . . . . .	125
A.1.1	Convex conjugate . . . . .	125
A.1.2	Strong convexity . . . . .	126
A.2	General convex conjugates . . . . .	126
<b>B</b>	<b>Properties from [113]</b>	<b>127</b>

# List of Figures

1.1	Schematic representation of an eukaryotic cell. . . . .	4
1.2	Structure of a nucleotide monophosphate with an adenine base . . . . .	5
1.3	Two main protein's secondary structures. . . . .	7
1.4	Three torsional backbone angles. . . . .	8
1.5	Ramachandran plot [10]. . . . .	8
1.6	RNA secondary structures. . . . .	9
1.7	A protein and an RNA structure (3CT6 and 17RA). . . . .	10
1.8	Two complexes protein/RNA and protein/ligand . . . . .	11
1.9	X-ray electron density and fitted AAs . . . . .	12
1.10	NMR bundle . . . . .	13
1.11	Glutamate receptor and its intensity curve . . . . .	14
1.12	Different levels of coarse-graining . . . . .	15
1.13	Comparative modeling by homology [25] . . . . .	16
1.14	Fragment-based method [25] . . . . .	17
1.15	Illustration of bond length and angles in bonded energy . . . . .	18
1.16	RRT extending step [64] . . . . .	22
2.1	Extensive form's trees . . . . .	26
2.2	Normal form's matrices . . . . .	27
2.3	Dominated strategies illustration . . . . .	28
2.4	Matching pennies . . . . .	29
2.5	Traffic light . . . . .	30
3.1	Kinematic graph and tree. . . . .	37
3.2	An hydrogen bond and its constraints. . . . .	38
3.3	Difference between rotations and translations. . . . .	39
3.4	Dynamic Clash Constraints. . . . .	40
3.5	Direction to the goal positions. . . . .	41
3.6	Comparison between least square and gradient methods . . . . .	42
3.7	Flowchart . . . . .	47
3.8	Comparison with iMOD . . . . .	50
3.9	Clash free two stages procedure . . . . .	51
3.10	Transition of 2N3Q . . . . .	52
3.11	Helical arrangements . . . . .	54
3.12	Application to a riboswitch . . . . .	56
3.13	Run time analysis . . . . .	57
3.14	Opening motion of alpha domain . . . . .	59
4.1	Ligand workflow . . . . .	63
4.2	Sampling of degree of freedom . . . . .	64
4.3	Alternative ligane 2xu3 . . . . .	65
5.1	Flow example . . . . .	85

6.1	$\epsilon$ -dynamics . . . . .	95
7.1	Player definition . . . . .	103
7.2	Ramachandran plots . . . . .	104
7.3	Distribution of distances . . . . .	106
7.4	Best result in all AAs . . . . .	108
7.5	Best result in tripeptide . . . . .	109

# List of Tables

1.1	The 4 RNA nucleobases . . . . .	5
1.2	The 20 AAs and their category . . . . .	6
3.1	Dataset . . . . .	49
3.2	Base pairs classification . . . . .	53





# Foreword

This document presents research work on bioinformatics. It addresses subjects that can seem heterogenous, however, the global motivation under all projects is the same: to develop theoretically proved methods for addressing difficult molecular structures challenges. We strongly believe that bioinformatics, and in particular, the understanding of molecular structures, can benefit from new approaches providing additional guarantees.

In Chapter 1, we introduce the biological problems we addressed: how to model the flexibility of a molecule and how to find its three dimensional conformations. We first present the two kinds of molecules we focus on: proteins and ribonucleic acids. Then, we explain the challenges that represent the understanding of their structures. We also present the most used methods to address these problems and their limitations.

In Chapter 2, we introduce game theory, we present the games and algorithms which properties we study in Chapter 5 and 6. This chapter starts by a presentation of games and their equilibria. Then, we introduce the concept of regret minimization, and exponential-weights algorithms that we study in Chapter 5 and 6.

In Chapter 3, we present a kinematics-based procedure to morph a ribonucleic acid molecule between conformational substates. This method is based on a graph approach and it introduces a new application to morphing with sparse information. It opens the door to other developments and applications on experimental data.

The Chapter 4 gives an overview of an ongoing project for finding alternative conformations of ligands in electron density maps. As well as the morphing procedure, we develop here a tool that aims to be used by experimentalists, helping them to exploit their experimental data.

The Chapters 5 and 6 present new convergence results of a class a no-regret algorithm. No-regret algorithms are extensively studied for their time average property of no-regret. Here we show that under light assumptions they also converge asymptotically. In Chapter 5, we present local convergence on generic games as well as global convergence results on potential games with a famous instance of no-regret algorithm called HEDGE. In Chapter 6 we extended our results to more realistic settings and a boarder class of algorithms.

In Chapter 7, we present a protein folding method based on our results in games' convergence. We present how a protein can be understood as a multiplayer iterative game, and we apply HEDGE algorithm to it. We also tested a few utility functions and obtained encouraging results.



## Chapter 1

# Some background in structural biology

### Contents

---

<b>1.1</b>	<b>Proteins and RNAs presentation . . . . .</b>	<b>3</b>
1.1.1	Sequences . . . . .	4
1.1.1.1	Protein sequence . . . . .	4
1.1.1.2	RNA sequence . . . . .	5
1.1.2	Secondary structures . . . . .	6
1.1.2.1	Protein secondary structures . . . . .	7
1.1.2.2	RNA secondary structures . . . . .	8
1.1.3	Three dimensional structures . . . . .	9
1.1.4	Interactions . . . . .	10
<b>1.2</b>	<b>Structures : determination, prediction and sampling . . . . .</b>	<b>11</b>
1.2.1	Experimental techniques . . . . .	12
1.2.1.1	X-RAY . . . . .	12
1.2.1.2	NMR . . . . .	12
1.2.1.3	cryo-EM . . . . .	13
1.2.1.4	SAXS . . . . .	13
1.2.1.5	DEER . . . . .	14
1.2.2	Structure Prediction . . . . .	14
1.2.2.1	Coarse-grained approaches . . . . .	15
1.2.2.2	Template-based methods . . . . .	15
1.2.2.3	Template-free methods . . . . .	16
1.2.3	Motion modeling . . . . .	17
1.2.3.1	Torsional and cartesian coordinates . . . . .	17
1.2.3.2	Molecular dynamics . . . . .	18
1.2.3.3	Normal Mode Analysis . . . . .	20
1.2.4	Sampling and morphing strategies . . . . .	21
1.2.4.1	Monte-Carlo . . . . .	21
1.2.4.2	Tree exploration . . . . .	22
1.2.4.3	Other approaches . . . . .	23

---

## 1.1 Proteins and RNAs presentation

Every single second, thousands of molecular processes and interactions occur in our cells. They are essential for our heart to beat, our brain to think, our cells to live, etc. Proteins as well as *Ribonucleic Acids* (RNAs) are the workhorses of this permanent activity.

Most of the genetic information of a cell is encoded in its *Deoxyribonucleic Acid* (DNA). In eucaryote organisms like Humans, DNA is contained in the nucleus (Figure 1.1). RNAs are often introduced as messengers between DNA in the nucleus

and the cell cytoplasm to create proteins. During the transcription, part of a DNA molecule is copied into RNA. RNA molecules undergo some maturation events and go through nuclear pores to reach the cytoplasm. Once in the cytoplasm some RNAs called RNAs messenger are translated into proteins by ribosomes, this is the translation step. The translation is based on the RNA sequence and requires the intervention of other RNAs and proteins.

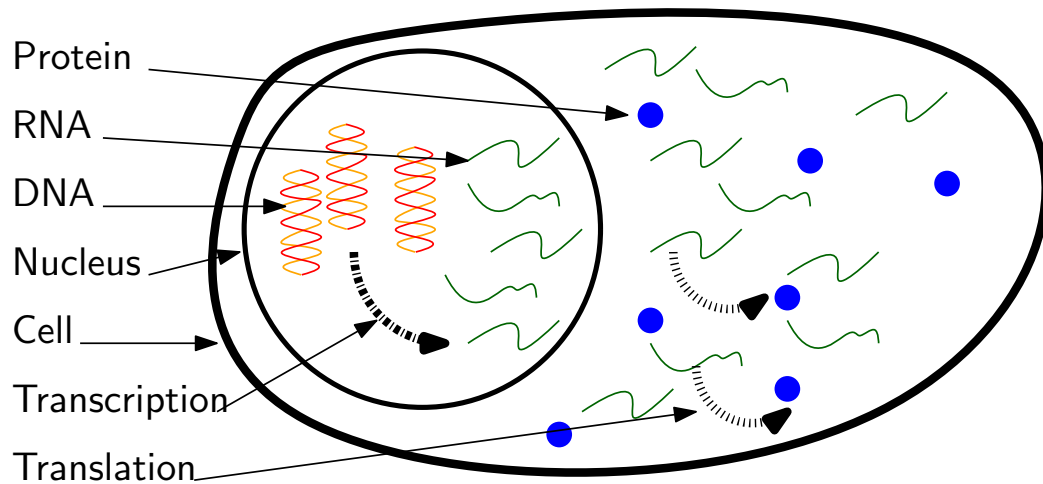


FIGURE 1.1: Schematic representation of an eukaryotic cell, and a few macromolecules.

Non-coding RNAs gather all RNAs that are not translated into protein. However they are essential and play major role in many cellular mechanisms along with proteins [1].

In this section, we present the basis of proteins and RNAs, from their sequence to their structures.

### 1.1.1 Sequences

Proteins as well as RNAs are chains (one or an assembly of multiple chains for some proteins) of residues. Each of those residues are often represented by a one letter code. They can be seen as a string although proteins and RNAs have a different alphabet. We will first describe proteins sequence and then RNAs sequence.

#### 1.1.1.1 Protein sequence

The protein alphabet has 20 letters corresponding to the 20 *Amino Acids* (AAs) (table 1.2). AAs are different from one another, but they all have the same central carbon atom,  $C_{\alpha}$ , an amino group ( $NH_2$ ) and a carboxyl group ( $COOH$ ) linked to the  $C_{\alpha}$ . This common base is called backbone.

AAs are linked by peptide bonds between their amino group ( $NH_2$ ) and their carboxyl group ( $COOH$ ). The formation of the peptide bond also forms a water molecule ( $H_2O$ ) while the amino group and the carboxyl group become respectively  $NH$  and  $C=O$ . However, the rest of the AA, called the side chain differs between AAs. The side chain attached to the  $C_{\alpha}$  is different in the 20 AAs. AAs can be gathered in groups according to the physico-chemical characteristics of their side chain. A lot of different classifications are possible relying on different criteria and dividing

the AAs in different number of groups [2]. In our study, described in Chapter 7, we chose to use a three groups classification : hydrophobic, polar and charged.

This classification is useful as one can consider that AAs of the same group have the similar properties.

### 1.1.1.2 RNA sequence

A RNA is a sequence of nucleotides. A nucleotide is formed by a phosphore group, a five member sugar ring and a base (Figure 1.2). Phosphore group and sugar ring are the same for all nucleotides. However 4 different bases exist for the RNA (table 1.1).

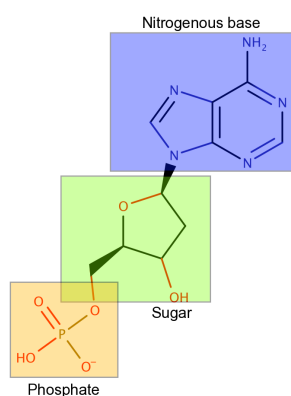


FIGURE 1.2: Structure of a nucleotide monophosphate with an adenine base.

Category	Structure	Name	Structure	Name
Purines		Adenine A		Guanine G
		Uracil U		Cytosine C

TABLE 1.1: The 4 RNA nucleobases.

Adenine and guanine are purines, they have two merged rings. Uracil and cytosine are pyrimidines, they only have one ring. Nucleotides can interact and form pairs. The main base pairs, the more stable, are the *Watson-Crick* (WC) base pairs [3] adenine (A) with uracil (U) and cytosine (C) with guanine (G). They are usually complemented with the less stable Wobble (G-U) base pairs.

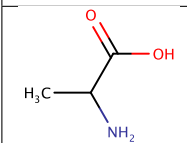
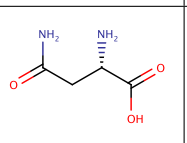
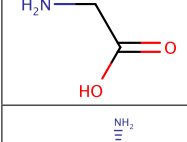
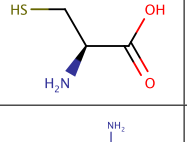
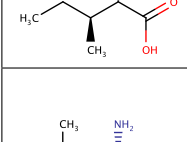
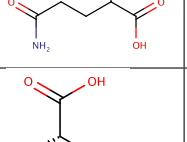
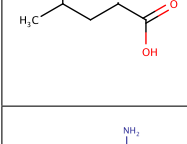
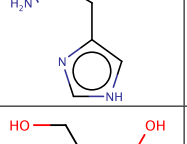
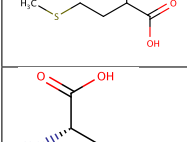
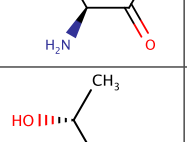
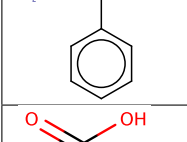
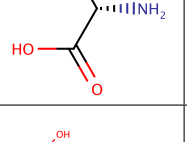
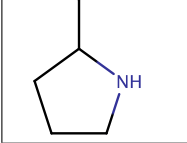
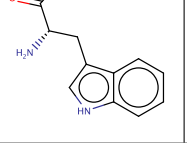
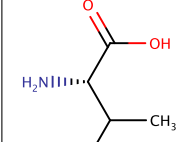
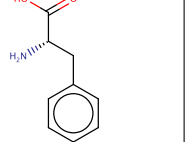
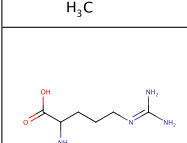
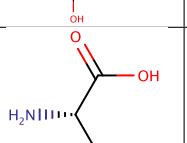
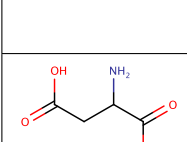
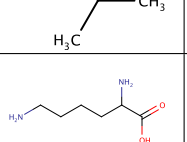
Structure	Name	Category	Structure	Name	Category
	Alanine Ala / A	hydrophobic		Asparagine Asn / N	polar
	Glycine Gly / G	hydrophobic		Cysteine Cys / C	polar
	Isoleucine Ile / I	hydrophobic		Glutamine Gln / E	polar
	Leucine Leu / L	hydrophobic		Histidine His / H	polar
	Methionine Met / M	hydrophobic		Serine Ser / S	polar
	Phenylalanine Phe / F	hydrophobic		Threonine Thr / T	polar
	Proline Pro / P	hydrophobic		Tryptophane Trp / W	polar
	Valine Val / V	hydrophobic		Tyrosine Tyr / Y	polar
	Arginine Arg / R	charged		Glutamic acid Glu / E	charged
	Aspartic acid Asp / D	charged		Lysine Lys / K	charged

TABLE 1.2: The 20 AAs and their category. The glycine side chain is only an hydrogen atom, not represented here.

### 1.1.2 Secondary structures

Proteins as well as RNAs have an important secondary structure resulting on strong interactions between AAs or nucleotides. The secondary structure is very important

in the three dimensional structure determination.

We start by studying the protein's secondary structure.

### 1.1.2.1 Protein secondary structures

Many secondary structures can be distinguished [4] but in the rest of the manuscript we will only focus on the two main structures [5] (Figure 1.3):

- the  $\alpha$ -helix;
- the  $\beta$ -sheet.

Regions not structured into helices or sheets are called loops.

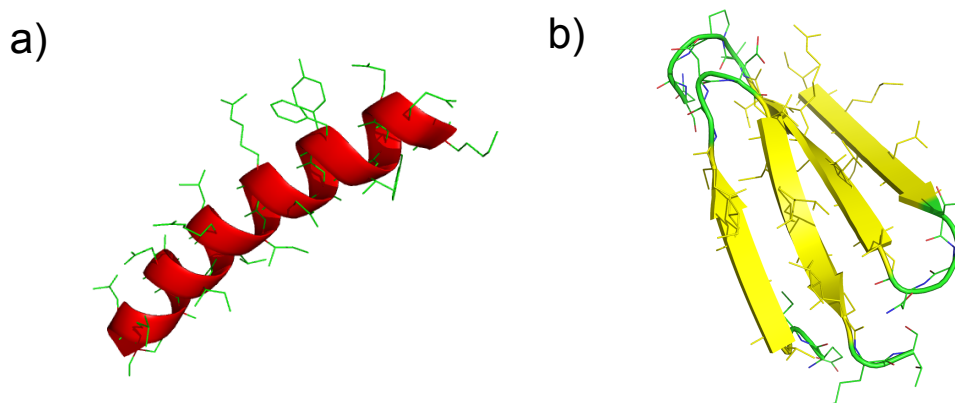


FIGURE 1.3: Two main protein's secondary structures.(a) The alpha helix. (b) The beta sheet.

Three torsional angles:  $\omega$ ,  $\phi$  and  $\psi$  (1.4), determine the backbone configuration. The peptide bond region formed by the  $C\alpha$ , C and O of an AA and the NH, and  $C\alpha$  of the next AA is planar: all atoms lay in the same plane [6].

Because of planarity of peptide bonds, the angle  $\omega$  is either  $0^\circ$  (*cis*) or  $180^\circ$  (*trans*). The *trans* configuration is present in 98% of the cases [8], we therefore consider that all peptides are in a *trans* configuration.

Due to steric constraints, some ranges of the angles  $\phi$  and  $\psi$  are unreachable. These two angles differ between secondary structures. The Ramachandran plot [9] represents the accessible region for the  $(\phi, \psi)$  couples. It points out that different secondary structures reach different areas of the plot (Figure 1.5).

The secondary structure of a protein can be inferred from its three dimensional structure. Softwares like *Dictionary of Protein Secondary Structure* (DSSP) [11] can determine the secondary structure based on atoms position. It uses an eight groups classification that can be easily gathered into helices, sheets and loops. It is also possible although less accurate to predict the secondary structure from the sequence. Well-known softwares, like PSIPRED [12], use alignment against sequences with known secondary structures, relying on the hypothesis that structures are more conserved than sequences, so two similar sequences are likely to present the same structure. Other methods rely on learning algorithms like *Hidden Markov Model* (HMM) [13] trained on a database to predict new sequences.



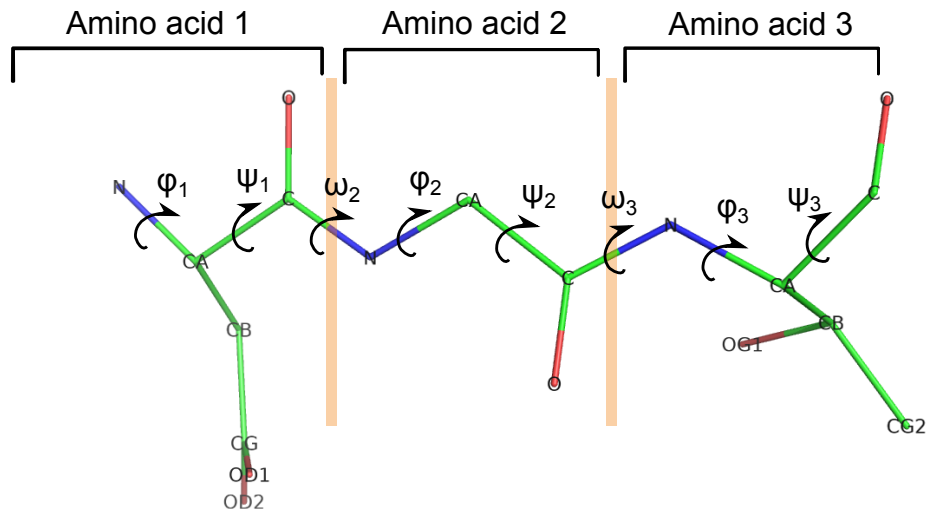


FIGURE 1.4: Three torsional backbone angles of amino acids. Figure done with PyMol [7].

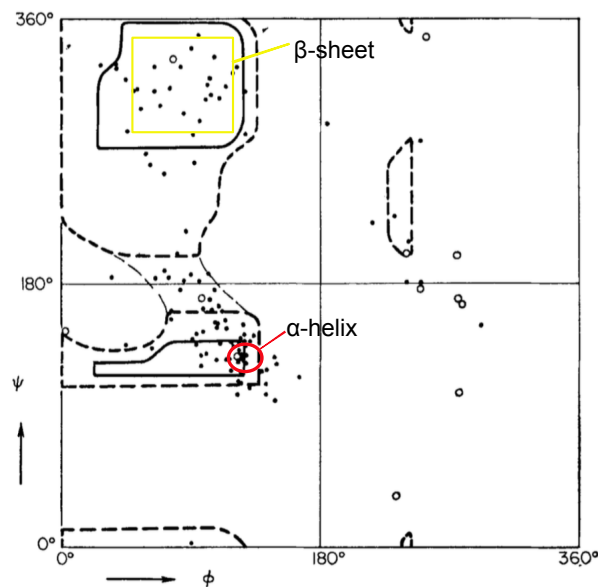


FIGURE 1.5: Ramachandran plot from [10]. Allowed configurations are outlined by a bold line and partially allowed conformations, that are less stable, by a light line.  $\alpha$ -helix and  $\beta$ -sheet regions are respectively delimited in red and yellow. Note that the original plot does not represent the  $\alpha$ -helix and  $\beta$ -sheet regions but clouds of points with different symbols for different secondary structures.

### 1.1.2.2 RNA secondary structures

The secondary structure of the RNA relies on base interactions. Base pair interactions stabilize the molecule structure because a consequent energy (up to 6kcal/mol [14]) is needed to break them. The strongest bases interactions are Watson-Crick (A-U and C-G), but others like Wobble [15] interactions between guanine and uracil also exist and play an important role. Stability of base pair interactions also depends on

their orientation. Consecutive base pairs form double-stranded helices that are very important for RNA structure stability (Figure 1.6). A pseudoknot is an helix with one strand forming an helix elsewhere. Secondary structure can be represented by a sequence of parenthesis, a "(" for the first base of a pair, ")" for the last base of a pair and "." for unpaired bases. Pseudoknots result in intersecting parenthesis (Figure 1.6).

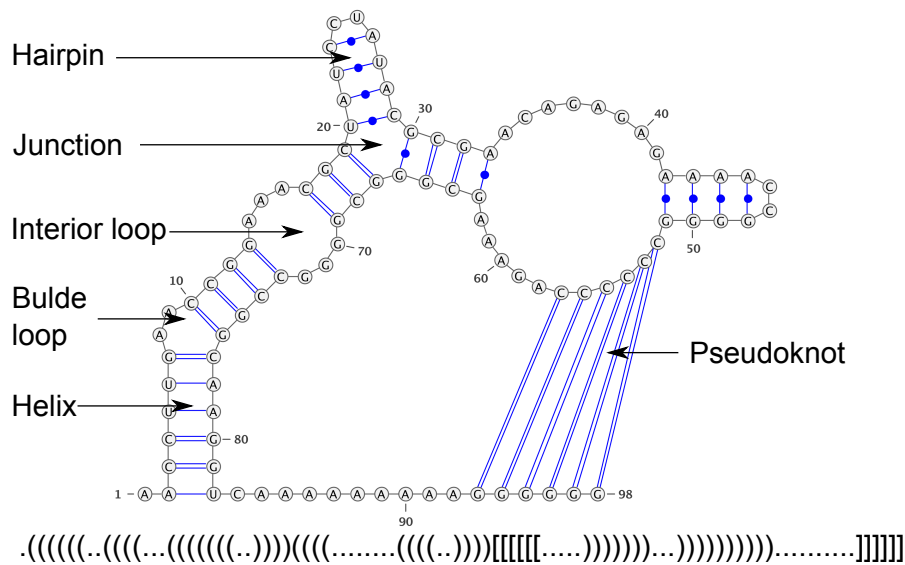


FIGURE 1.6: RNA secondary structures with helices and different kind of junctions. Waston-Crick base pairs are represented by one or two blue lines, others base pairs are represented by a line with a circle. Made with VARNA [16]

The RNA secondary structure can be determined from the three dimensional structure by software like RNAView [17] which finds base pairs interaction and classifies them according to the Leontis classification [18]. Dynamic programming techniques allow to find the optimal pair matching to maximize the number of interactions (without pseudoknot) since Nussinov in 1978 [19]. Most of those algorithms do not consider pseudoknots as the problem was shown to be NP-hard [20]. The *stacking* energy resulting of the interaction of two base pairs in top of each other, was taken into account by Zuker [21]. Many other dynamic programming methods have been released [22], [23] and some can take into account some pseudoknots [24]. A RNA sequence can adopt different secondary structures, new methods tend to output multiple possible secondary structures with a probability score.

### 1.1.3 Three dimensional structures

Often, the three dimensional structure of molecules, called also conformation, is essential for their function. However, in some cases, the protein has no main structure. These proteins are called intrinsically disordered proteins and represent only a 30% of the proteins [25]. The rest of the proteins adopt a few main conformations, they fluctuate around those with high frequency and from time to time they can transition from one to an other. The three dimensional structure of proteins and RNAs results from the arrangement of the secondary structure into space. The possibilities

of arrangements are infinite, but a few main classes of folds have been exhibited for RNAs [26]–[28] as well as for proteins [4], [29].

Proteins known structures can be hierarchically classified according to their folds. Various databases exist, three of the main databases are CATH [30], SCOP [31] and Pfam [32]. CATH is a four level hierarchical database: Class, Architecture, Topology and Homologous superfamily, it contains a total of 1373 different topologies. SCOP has a slightly different hierarchy: Class, Fold, Superfamily, Family, Domain and Species. Pfam has also its own hierarchy: Family, Domain, Repeat, Motifs, Coiled-Coil and Disordered. The main difference of Pfam is that it associates a HMM to each family.

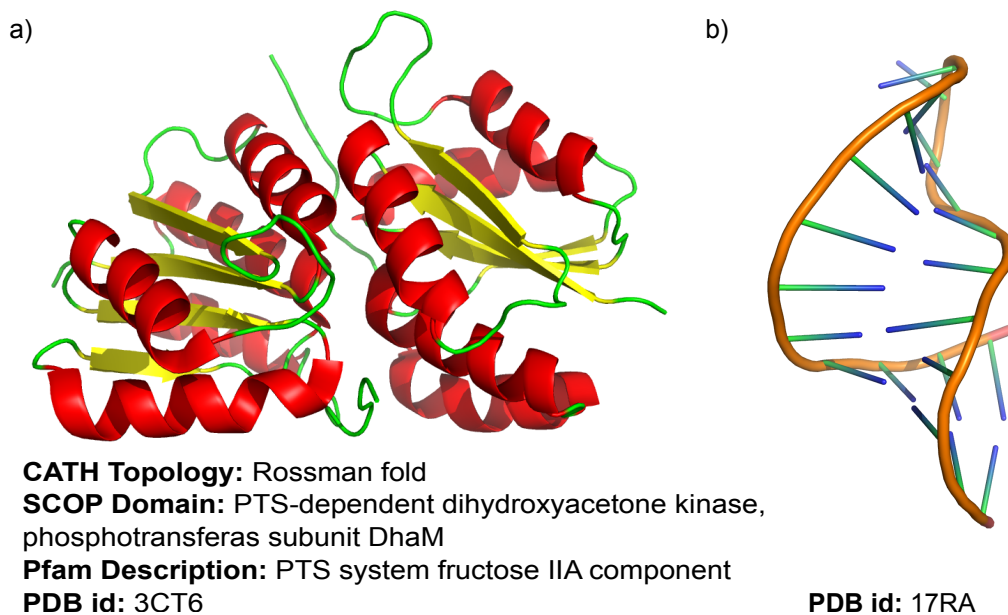


FIGURE 1.7: a) A protein structure 3CT6 and its classifications. b) A RNA structure 17RA.

The *Protein Data Bank* (PDB) [33] gathers more than 84,000 experimentally solved protein and RNA structures. However some of the structures are redundant or very similar. A PDB entry often links the structure to its CATH, SCOP and Pfam classifications (Figure 1.7a). The PDB also contains RNA structures (Figure 1.7b) but with less cross-informations. Most experimental techniques for determining structures (see Section 1.2.1) give only one result. However it is well-known that proteins as well as RNAs are flexible and can adopt different structures.

#### 1.1.4 Interactions

Molecules interact with each other. Two peptide chains can interact to form the proper protein, if the interaction is not supposed to be broken we call it the *quaternary structure*, as the protein does not exist without it. On the contrary, some interactions are only temporary. The formation of complexes and the interaction of molecules is a key step in many cellular processes like the transcription. It is therefore essential to understand the interacting process between molecules. There are many kinds of complexes with different types of molecules. Proteins are often *receptors* and bind either a smaller protein or peptide (only a few AAs), a RNA or a ligand which can be a small molecule like a nucleotide or an ion (Figure 1.8). Interface and binding pocket refer to the same thing: the nucleotides involved in the interaction.

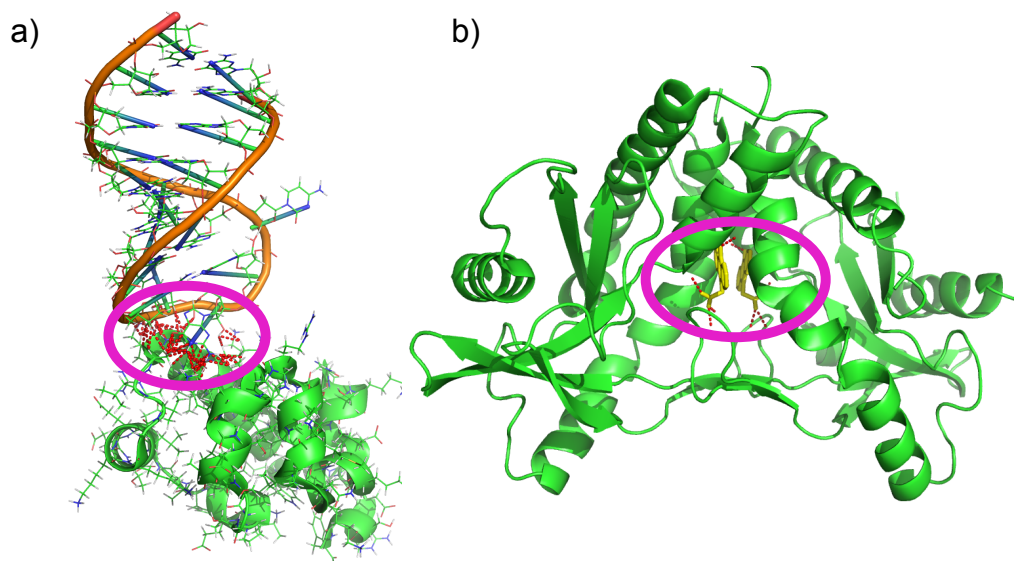


FIGURE 1.8: a) A protein/RNA complex 2B6G, the interface is encircled in purple and inter-molecule hydrogen bonds appear in red dashed lines. b) A protein/ligand complex, the binding pocket is encircled in purple, with hydrogen bonds in red dashed lines.

A RNA can also bind ligands and peptides. The predictions of molecules interactions is a two-fold strategy. First we need to know the structure of each partner as their structures can change while interacting. Second we also need to find the interaction site. This second step can be done either by an extensive search (RosettaDOCK) [34], or using experimental information (HADDOCK, High Ambiguity Driven protein-protein DOCKing) [35], or by recognizing AAs that are often in interface areas [36]. Reviews [37] present many other methods to predict molecular interactions. *Critical Assessment of PRediction of Interactions* (CAPRI) is a worldwide competition to assess the quality of interaction predictions [38].

## 1.2 Structures : determination, prediction and sampling

Determining the structures of molecules is of tremendous importance when one wants to understand how they participate to cellular processes. Elaboration of drugs often relies on the structure of the drug and its efficiency depends on its interaction with others molecules. Experimental methods described in the first subsection can determine the structure with different accuracy. However these methods are time consuming and cannot be applied to any proteins or RNAs. Therefore computational methods that predict the structure of molecules are of main interest. As for the interaction prediction, there is a worldwide competition, called *Critical Assessment of Techniques for Protein Structure Prediction* (CASP), to assess the quality of structure prediction and prediction accuracy [39]. This is extremely important as it allows to test prediction method on new structures. Although the same approach can often be applied to proteins as well as RNAs, most methods can only consider proteins some others address only RNAs and only a few can handle both. In this section we first described different experimental methods, then we will focus on computational methods.

## 1.2.1 Experimental techniques

There are several experimental methods for solving a molecule structure. Here we present three main ones: X-ray crystallography (89.5% of the PDB structures), *Nuclear Magnetic Resonance* (NMR) (9.1% of the PDB structures) and *cryo-Electronic Microscopy* (EM) (1.1% of the PDB structures). We also present *Small Angles X-Ray Scattering* (SAXS) and *Double Electron-Electron Resonance* (DEER) experiments which do not solve the molecular structure but give precious insight on the molecule shape.

### 1.2.1.1 X-RAY

X-ray crystallography is the most used method. It allows to obtain a precise structure and its resolution. The molecule is purified and then crystallized [40], this step

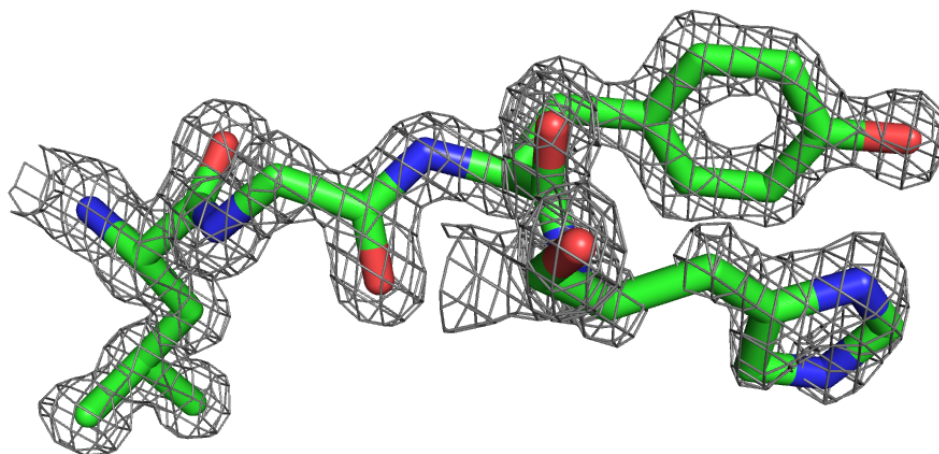


FIGURE 1.9: Electron density map and a few residues of a X-ray determined protein, 3Q6Y.

is crucial and can be impossible or very difficult for some molecules in particular RNAs. The crystal is then subjected to an synchrotron X-ray beam. The protein scatters the X-rays and the pattern is analyzed to calculate an electron density map (Figure 1.9). Amino acids are fitted into the density to give a structure. In Chapter 4 we present a method to improve the fitting of ligands in their binding pocket. Although this method gives high precision results, it is believed that cristal structure slightly differ from the structure of *in vivo* molecules.

### 1.2.1.2 NMR

When using NMR spectroscopy the purified molecule is placed in a static magnetic field [41]. With this magnetic field, the space becomes anisotropic and nuclei spins can be observed. NMR spectroscopy does not requires cristallization but it requires a large sample of molecules as well as enrichment in rare isotopes to observe nuclei spins. NMR spectroscopy gives precious information that allows to retrieve the conformation using suitable software. NMR resolved structures are generally a bundle of different structures (Figure 1.10). The width of the bundle can be interpreted as

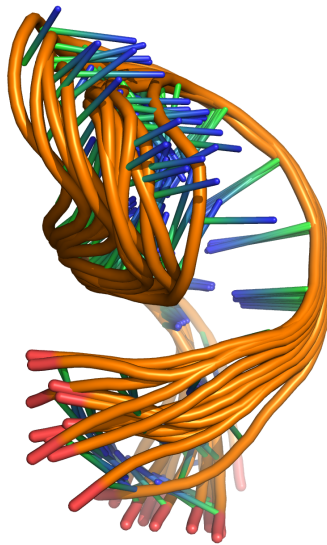


FIGURE 1.10: NMR bundle of the RNA 17RA.

a local dynamic as well as the accuracy of the prediction [42]. In Chapter 3 we use NMR bundle to test our morphing procedure.

### 1.2.1.3 cryo-EM

Cryo-EM uses a small frozen sample and an electron beam. The sample is positioned along the electron beam. And a detector analyzes the electrons that did not collide with the sample. Tremendous improvement of the technique occurred in the last decades. Among others, the detector can now capture movies and reconstruct precise structures [43]. The final output of this method is really similar to X-ray crystallography (Figure 1.9). Cryo-EM is mainly used on proteins difficult to crystallize and macromolecular complexes. The resolution is still a bit low compared to X-ray crystallography but hopefully this gap will be filled in a few years [44].

### 1.2.1.4 SAXS

SAXS is an experimental method that provides low-resolution structural information about macromolecules in solution. Although the provided resolution is very low compared to X-ray crystallography, NMR spectroscopy or cryo-EM, the method is very interesting because it uses in solution molecules and can analyse them in various conditions, for example with and without a ligand. It is also often used for RNAs as these are more complicated to crystallize [45]. The protocol is similar to the X-ray crystallography except that the sample is in solution instead of a crystal. This implies that the signal is isotropic instead of anisotropic. Therefore the output is a intensity single curve (Figure 1.11). From the curve, one can infer the radius of giration of the molecule that is about its diameter. Usually the structure is not inferred form the intensity curve. However it is very easy to compute a curve from a known structure. The comparison of experimental and theoretical curves allows to choose which structures among a set of structures were present in the solution [46].

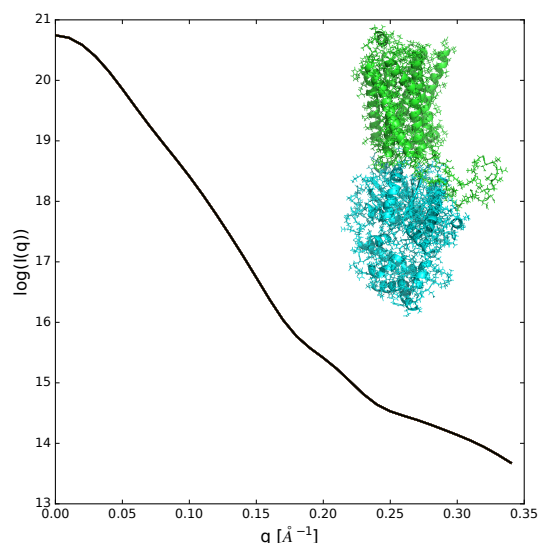


FIGURE 1.11: A glutamate receptor and its intensity curves.

### 1.2.1.5 DEER

DEER is a spectroscopic approach similar to NMR that measures spin resonance. Instead of observing nuclei spins, it observes probes or labels spins. Probes are molecules added at precise points of the molecule. DEER measurements is a distribution of distances between probes. This can be very useful to identify large motion like the opening of a domain. One drawback is that probes can be long and flexible. The distance between two probes does not exactly correspond to the distance between the two anchor points. We use DEER data in an extension of the method presented in Chapter 3.

Now that we presented experimental methods for structure determination, we will focus on computational methods.

## 1.2.2 Structure Prediction

Only around 11% of proteins with a known sequence have an experimentally solved structure. The PDB database [33] contains 37,843 protein structures with less than 90% of sequence identity and the UniRef [47] database contains 332,971 protein sequences with less than 90% of identity<sup>1</sup>.

For RNAs it is even worse with only 1,226 experimentally solved structures in the PDB database with less than 90% of sequence identity.

In addition, it is commonly believed that structure can be inferred from sequence [48]. Therefore, developing structure prediction methods is extremely important for confirming the assumption that it is possible and for complementing experimental methods.

The CASP assessment evaluates the quality of protein structure prediction methods every 2 years and identifies fields of progress and fields that need to be developed further [39]. There is no such assessment for RNAs although the CAPRI assessment for interaction prediction usually includes RNAs.

<sup>1</sup>On the 5th of April 2017

One main issue of molecule structure prediction is the dimension of the problem. A protein contains many degrees of freedom, taking all of them into account at the same time is extremely challenging. A broadly used way to overcome this problem is to use a multi-step and *coarse-grained* approach.

Most methods are developed for proteins but the same approaches could often be applied to RNAs. Here we present the principles of some methods without considering if they apply to RNAs or proteins.

### 1.2.2.1 Coarse-grained approaches

A coarse-grained approach consists of gathering multiple atoms into one single object, thus drastically reducing the dimension of the problem. For a more detailed review of different coarse-grained methods the reader can look into Noid's survey [49].

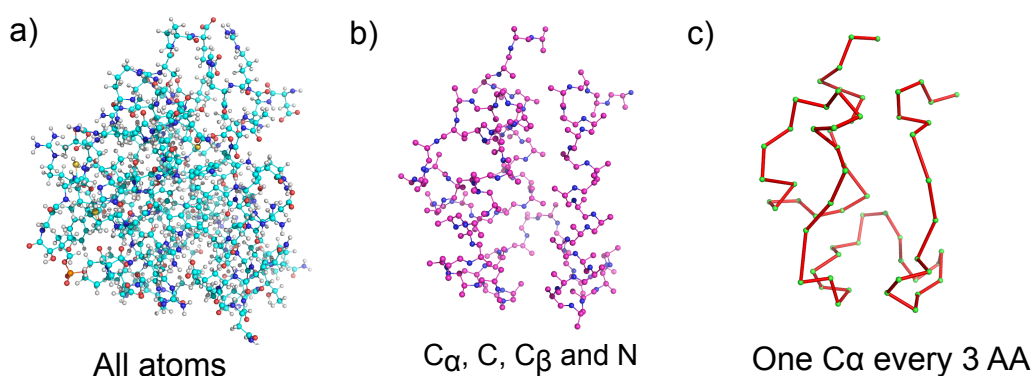


FIGURE 1.12: Different levels of coarse-graining for protein (example with PDB id 2N6Y). a) Full atom representation. b) Only four atoms are represented per AA (only three for the glycine that does not have  $C_{\beta}$ ). This representation is very convenient as the backbone and its dihedral angles are present, the side chain is only represented by its first atom  $C_{\beta}$ . c) Only one  $C_{\alpha}$  atom per tripeptides (three AAs).

The finer is the coarse-grained representation the closer to reality is the model but the higher is the dimension of accessible conformations. Figure 1.12 illustrates different levels of representation from all atoms to one point for few residues. A usual way to take the best of coarse-grained approaches is to use different levels of coarse-graining in different steps: the scale becoming finer as the computation advances [50]. However retrieving an all atoms representation from a coarse-grained structure is challenging [51].

### 1.2.2.2 Template-based methods

Structure prediction methods can be divided in two main categories: the template-based methods and the template-free methods. A template-based method as defined in CASP needs *template* structures to build the prediction. The definition of a template structure is complicated [52] but it mostly lays on sequence identity. A *target*, i.e., a sequence with unknown structure, is classified into the template-free category if there are no homologous proteins with a known structure.



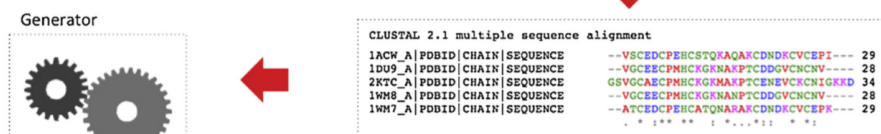
Dorn et al. [25] did a detailed review of different methods for structure prediction. Analysis of the performance of the best methods can also be found in proceedings of CASP meeting every two years, [53] is the most recent one. Here we give a short overview of the main methods.

Most template based methods have an approach as described in Figure 1.13. The sequence is aligned against sequences from a database to find one or more similar sequences with known structures. The structures of similar sequences are used to model the structure of the target sequence.

#### 1. Searching the PDB for Homologous Proteins



#### 2. Sequence Alignment



#### 3. Build Conformational Models

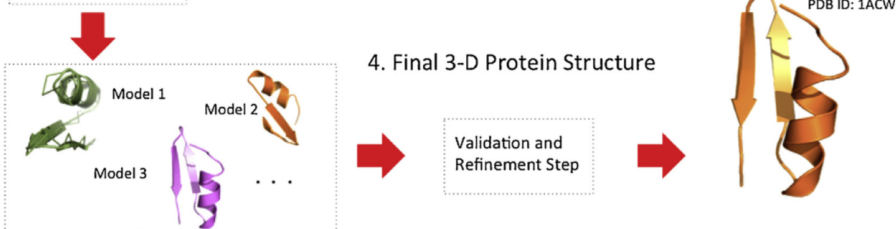


FIGURE 1.13: Schematic representation of a typical process of comparative modeling by homology. Initially, template proteins are identified. Then the sequence of the target protein is aligned against the sequence of the protein-templates, and then a model is built and validated, obtaining in the end, the 3-D structure of the target protein. If necessary, the final structure may undergo a refinement process.

Taken from [25].

Although very accurate, template-based methods have a main drawback: they cannot predict a new fold [39]. Some methods include a refinement steps to refine the template based model on regions not well covered. This can be done with methods close to the molecular dynamics (see Section 1.2.3.2) or template-free methods.

#### 1.2.2.3 Template-free methods

Template-free methods are used for sequences with no template structure available in the PDB database. The most successful methods until recently were fragments methods (Figure 1.14). This approach comes from the observation that protein structure can be divided in common structural motifs [54]. Homologous fragments are assembled according to criteria that depend on the methods and usually involve local and non-local information. This process can be done multiple times before reaching a satisfying structure.

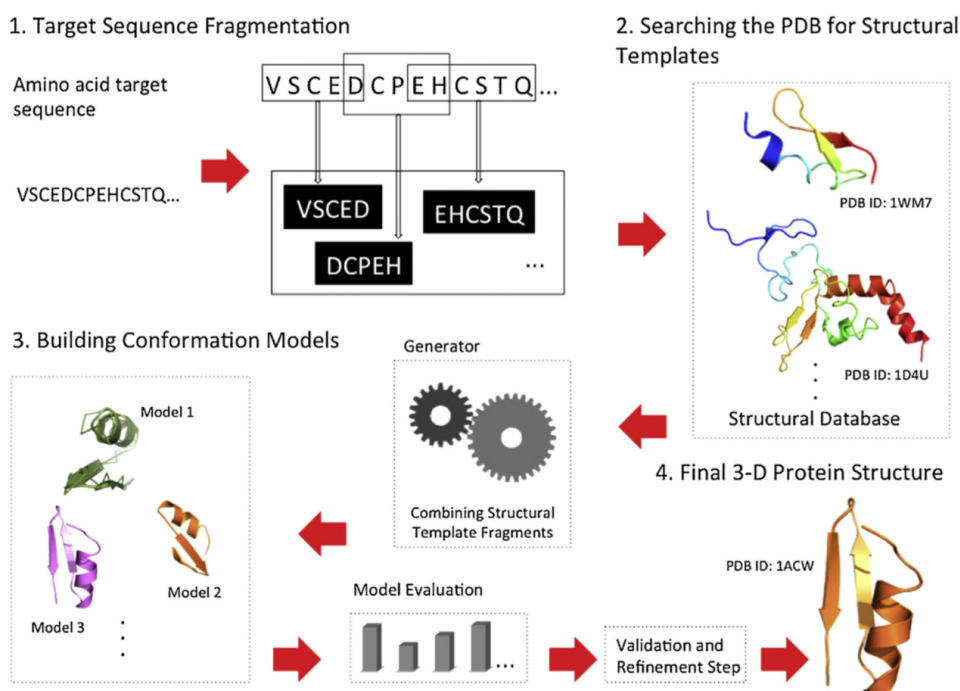


FIGURE 1.14: General schematic representation of a fragment-based method for the 3D protein structure prediction problem: a target sequence is fragmented, models are obtained from the PDB, the fragments are classified, the conformation is constructed and when appropriate, conformation is refined. Taken from [25].

In Chapter 7 we present a template-free approach using small fragments (tripeptides) and integrating database statistics for a scoring function. This method is based on a similar approach for RNAs [55].

Template-free as well as template-based methods tend to incorporate additional information into their model. For example it is common to use predictions of secondary structure to bias the prediction process. Coevolutionary informations based on sequence evolution are used to predict contacts [56] and are also more and more used and lead to the efficient methods. The best prediction in CASP11 free modeling was found by Baker's group incorporating contacts prediction in the score function.

### 1.2.3 Motion modeling

Motion modeling is a key issue of conformation sampling as well as motion planning. Here we give an overview of the main approaches: atomic perturbations with molecular dynamics, and extraction of the main components with normal mode analysis.

#### 1.2.3.1 Torsional and cartesian coordinates

Most of the methods can use torsional or cartesian coordinates. Cartesian coordinates represent the position of an atom by its 3 dimension vector position.

Torsional coordinates are internal coordinates that correspond to torsional angles around bonds. Torsional moves approaches in molecular motions only consider torsions, i.e., rotations around bonds, as degrees of freedom. This implies to fix the value of all bond lengths as well as planar angles, which is generally true.

This approach can be used as a first step of a coarse-grained method using only backbone torsion [57]. This is very natural in protein as the two main degrees of freedom of an AA's backbone are its  $\psi$  and  $\phi$  torsion angles (see Section 1.1.2). It is also relevant for RNA even if the backbone has more than two main degrees of freedom.

In Chapter 3, we present a method that uses torsional moves coordinated such that the secondary structure of the molecule is preserved. The method for protein structure prediction presented in Chapter 7 also uses torsional moves.

### 1.2.3.2 Molecular dynamics

Molecular dynamics is a particular method as it aims at generating a trajectory as close as possible to a "real" molecular motion. To do so it considers force fields that apply to any atoms of the molecules. The movement of each atom is usually determined by numerically solving Newton's equation of motion.

$$m_i \cdot a_i = F_i \quad (1.1)$$

Where  $m_i$  is the mass of atom  $i$ ,  $a_i$  its acceleration and  $F_i$  the forces that applied to it.

**Force field** Multiple force fields exist, the main are *Assisted Model Building and Energy Refinement* (AMBER) [58] and *Chemistry at HARvard Macromolecular Mechanics* (CHARMM) [59]. Conservative forces are obtained through the derivation of potential energy. The potential energy of an atom can usually be divided into bonded and non-bonded terms. Bonded energy comes from the deformation of a bond between two atoms (Figure 1.15).

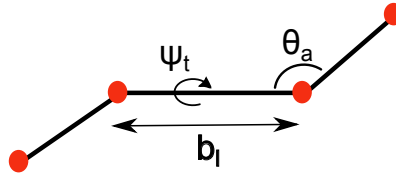


FIGURE 1.15: Illustration of bond length and angles in bonded energy

$$U_{bonded} = \sum_{b \in bonds} \frac{1}{2} k_b (b_l - b_0)^2 + \sum_{a \in angles} \frac{1}{2} k_a (\theta_a - \theta_0)^2 + \sum_{t \in torsions} \frac{1}{2} k_t (1 + \cos(n_t \psi_t + \psi_0)) \quad (1.2)$$

The first term corresponds to the harmonic potential with a constant  $k_b$  and a reference length  $b_0$ ,  $b_l$  is the current length of the bond. The second term is the harmonic angle potential with a constant  $k_a$  and a reference angle  $\theta_0$ ,  $\theta_a$  is the current angle. The last term is the cosine form of the dihedral potential with a constant  $k_t$ , a reference angle  $\psi_0$  and a multiplicity  $n_t$  which depend on the bond,  $\psi_t$  is the current angle.

The non-bonded energy corresponds to van der Walls interaction (first term) and the electrostatic interaction (second term).

$$U_{non-bonded} = \sum_i \sum_{j>i} \epsilon_{ij} \left( \left( \frac{r_{0ij}}{r_{ij}} \right)^{12} - 2 \left( \frac{r_{0ij}}{r_{ij}} \right)^6 \right) + \sum_i \sum_{j>i} \frac{q_i q_j}{4\pi\epsilon_0 r_{ij}} \quad (1.3)$$

Where  $r_{0ij}$  is the distance at equilibrium,  $\epsilon$  is the depth of the energy,  $\epsilon_0$  is the dielectric constant,  $q_i$  is the charge of atom  $i$  and  $r_{ij}$  the distance between atom  $i$  and  $j$ .

The solvent can either be implicit and considered as an additional term in the force field or explicit and represented by atoms. Equation 1.1 is a differential equation (acceleration is the second derivative of the position) and can be solved with various initialization conditions.

**Initialization** Initial positions are chosen depending on the simulation goal. The initial distribution of velocities is usually drawn from the Maxwell-Boltzmann distribution

$$P(v_{i,\alpha}) = \sqrt{\frac{m}{2\pi k_b T}} \exp\left(-\frac{mv_{i,\alpha}^2}{2k_b T}\right) \quad (1.4)$$

where  $v_{i,\alpha}$  is the  $\alpha = (x, y, z)$  component of the velocity of atom  $i$ ,  $m$  is its mass,  $T$  the temperature and  $k_b$  the Boltzmann constant.

**Time scale** Equation 1.1 is numerically solved, i.e., the solution is iteratively evaluated with a small time step. This time step is usually on the order of the femtosecond ( $10^{-15}$  s) [60]. Timescale of motion varies from femtoseconds for bond vibrations to millisecond in the case of large domain motions [61].

Equation 1.1 is a microscopic equation that is solved for each atom. However we want to estimate macroscopic properties of the system for example its energy. Then it is important to introduce the concept of an *ensemble* as "a collection of systems described by the same set of microscopic interactions and sharing a common set of macroscopic properties" [62].

**Temperature** The instantaneous temperature of a molecular system is related to its average (over the ensemble) kinetic energy by the equipartition theorem.

$$\frac{1}{2} \sum_{i=1}^N m_i \langle v_i^2 \rangle = \frac{3}{2} N k_b T \quad (1.5)$$

Where  $\langle v_i^2 \rangle$  is the root-mean-square speed of atom  $i$ ,  $m_i$  its mass,  $N$  is the number of atoms,  $k_b$  the Boltzmann constant and  $T$  the temperature. Therefore the instantaneous temperature can be chosen by scaling the velocities.

There exist different kinds of ensembles depending on which macroscopic properties are preserved.

**Microcanonical ensemble** The microcanonical ensemble (NVE) is obtained when the number of particle (N), the volume (V) and the energy (E) are fixed. It is obtained by integrating equation 1.1.

The total energy is fixed so the sum of potential energy and kinetic energy is fixed. The ensemble can be used to find various conformations of the same energy level. The temperature is not fixed.

**Canonical ensemble** The canonical ensemble (NVT) is obtained when the number of particle (N), the volume (V) and the temperature (T) are fixed. This ensemble is often used as a first step to minimize the energy. Obtaining a constant temperature involves coupling the system with a heat bath. This can be done by different ways, all resulting in considering nonconservative forces (forces that do not derived from a potential) added in equation 1.1.

**Isothermal-isobaric ensemble** The isothermal-isobaric ensemble (NPT) is obtained when the number of particle (N), the pression (P) and the temperature (T) are fixed.

Molecular dynamics is very accurate for small time scale motions and limited for long time scale error as numerically solving the motion equations implies a cumulative error. In addition, in order to apply a molecular dynamics method one needs to have a good understanding of the system he wants to analyze in order to choose the right solvent, ensemble and other parameters.

### 1.2.3.3 Normal Mode Analysis

Physical objects such as molecules have degrees of freedom. A normal mode is a combinaison of degrees of freedom. From an equilibrium position an object can oscillate along normal modes with a given frequency. An object has as many normal modes as degrees of freedom. "Normal" means that they are independent, an oscillation along a normal mode does not induce oscillations on others. A general move can be described as a superposition of moves along normal modes.

To find normal modes and their frequencies, the potential energy of a system in a state  $S$  near an equilibrium state  $S^0$  is developed in Taylor series. Let consider a system composed of  $n$  atoms  $a_1, \dots, a_n$  each one having three degrees of freedom corresponding to the directions  $x, y$  and  $z$ .

$$\begin{aligned} V(a_1, \dots, a_n)|_S &= V(a_1, \dots, a_n)|_{S^0} + \sum_{i,\alpha} \frac{\partial V}{\partial a_{i\alpha}}|_{S^0} (a_{i\alpha} - a_{i\alpha}^0) \\ &+ \frac{1}{2} \sum_{i,\alpha} \sum_{j,\beta} \frac{\partial^2 V}{\partial a_{i\alpha} \partial a_{j\beta}}|_{S^0} (a_{i\alpha} - a_{i\alpha}^0)(a_{j\beta} - a_{j\beta}^0) \\ &+ O(\|a - a^0\|^3) \end{aligned} \quad (1.6)$$

where  $\alpha$  and  $\beta$  can be  $x, y$  or  $z$  and  $a_{i\alpha}$  is the coordinate of atom  $i$  along the direction  $\alpha$  and  $a_{i\alpha}^0$  corresponds to the equilibrium state  $S^0$ .

As  $S^0$  is an equilibrium, the second term is null. We can remove the first term because the potential energy is defined up to an additive constant. So neglecting the third order terms we obtain.

$$V(a_1, \dots, a_n)|_S = \frac{1}{2} \sum_{i,\alpha} \sum_{j,\beta} \frac{\partial^2 V}{\partial a_{i\alpha} \partial a_{j\beta}}|_{S^0} (a_{i\alpha} - a_{i\alpha}^0)(a_{j\beta} - a_{j\beta}^0) \quad (1.7)$$

The study of the Lagrangian equation of motion done in [63] shows that the normal modes are the eigenvectors of the dynamical matrix  $D$  and their frequencies are the inverse of the associated eigenvalues. Where  $D$  is a  $3n$  by  $3n$  matrix corresponding to the Hessian matrix of the potential weighted by the mass of atoms such that

$$D_{i\alpha,j\beta} = \frac{\partial^2 V}{\partial a_{i\alpha} \partial a_{j\beta}} \Big|_{S^0} / \sqrt{m_i m_j}$$

Low frequencies modes correspond to collective conformational changes whereas high frequencies modes correspond to localized displacements.

The usual way of doing a *Normal Mode Analysis* (NMA) on molecules is to use an *Elastic Network Model* (ENM) [63]. Any couple of atoms separated by a distance less than a specified distance cutoff are considered to be linked by an elastic. Therefore the potential energy is:

$$V = \sum_{i,j} \frac{1}{2} k_{i,j} (d_{i,j} - d_{i,j}^0)^2 \quad (1.8)$$

where  $d_{i,j}$  is the distance between atoms  $i$  and  $j$ ,  $k_{i,j}$  is the constant associated to the spring that models their interaction and  $d_{i,j}^0$  is their distance at equilibrium.

Low frequencies modes are then used for sampling the conformational spaces or finding a trajectory toward a different conformation.

## 1.2.4 Sampling and morphing strategies

Once that moves are defined, different strategies can be used to choose moves and sample the conformational space or move toward an other conformation. First we present the Monte-Carlo method to sample according to a defined distribution. This method can be combined with other methods like *Rapidly exploring Random Trees* (RRTs), presented after.

### 1.2.4.1 Monte-Carlo

A Monte-Carlo method consists on estimating the average of a function on an ensemble by sampling the ensemble and then evaluating the function on the samples and computing the average. Monte-Carlo can use preferential sampling to sample according to a distribution function giving high probability to important value of the function so that the variance is reduced. Monte-carlo method is usually coupled with a Metropolis criterion to sample the probability law. The aim of a Monte-Carlo protein simulation is to sample conformations according to the Boltzmann distribution. The Boltzmann distribution is such that the probability to find a protein in a conformation  $c$  with an energy  $E_m$  at temperature  $T$  is :

$$\mathbb{P}(c) = \frac{\exp^{-\frac{E_m}{k_b T}}}{\sum_{c'} \exp^{-\frac{E_{c'}}{k_b T}}} \quad (1.9)$$

where  $k_b$  is the Boltzmann constant.

The simulation is usually done with the number of particles/conformations ( $N$ ), volume ( $V$ ) and temperature ( $T$ ) fixed and the energy  $E$  is the fluctuating variable.

The Monte-Carlo sampling algorithm is a three step algorithm:

1. Select a particle  $p$  (the initial if its the first iteration) and compute its energy  $E_p$
2. Randomly perturb  $p$  into  $p'$  and compute its energy  $E_{p'}$

3. Use the Metropolis criterion to accept or reject  $p'$ .

Therefore the Monte-Carlo sampling algorithm can be viewed as a Markov process because the generation of a new particle only depends on the current particle and the matrix acceptance from the Metropolis criterion.

The Metropolis criterion defined the probability to accept a new particle  $p'$ :

$$\begin{aligned} \mathbb{P}(p \text{ to } p') &= 1 \text{ if } E_{p'} \leq E_p \\ \mathbb{P}(p \text{ to } p') &= \exp^{-\frac{E_{p'} - E_p}{k_b T}} \text{ otherwise} \end{aligned} \quad (1.10)$$

This approach is limited by the size on the conformational space but it is often coupled with other techniques.

#### 1.2.4.2 Tree exploration

A common way to explore a conformational space is to construct a tree from the initial conformation. Various strategies have been developed to guarantee that the full space or a pre-defined portion of the space will be explore if the algorithm is ran long enough.

A widely used approach for that is RRTs [64]. It comes form the area of Robotics. The construction of a RRT initialized on an initial conformation  $x_{init}$  as steps illustrated in Figure 1.16.

1. Generate a random conformation  $x$ .
2. Find  $x_{near}$ , the closest conformation to  $x$ , already in the tree.
3. Perform a small move from  $x_{near}$  to  $x$  that gives  $x_{new}$ .

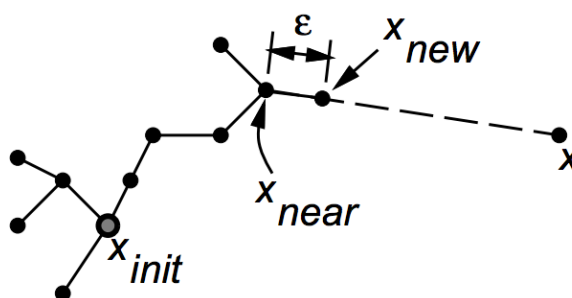


FIGURE 1.16: RRT extending step.  $x_{init}$  is the first conformation.  $x$  is a randomly generated conformation.  $x_{near}$  is the nearest existing conformation to  $x$ ,  $x_{new}$  is the generated conformation. Figure taken from [64]

All kind of moves can be used to generate the new conformation for example a combination of normal modes [65] or torsional angles [66]. Also the acceptance or not of  $x_{new}$  can be based on collision checking by rejecting a conformation if two atoms are too close [66] or using a metropolis criteria [67].

Tree exploration can be used for blindly sampling the conformational space as well as for finding a trajectory between two conformations. In the later, bias can be introduced in the choice of  $x_{near}$  to guide the exploration towards the goal conformation [68].

Extensions of the classic RRT exist that bring more guarantees such as asymptotically optimal trajectories [69].

### 1.2.4.3 Other approaches

In Chapter 3 we present a morphing method that does not construct a tree but just a path. A move is chosen among torsion combinations such that it reduces the distance to the goal conformation.

iMod [70] uses a NMA approach to generate a trajectory between two conformations. It biases the motion towards the goal conformation by accepting a move only if the *Root Mean Square Deviation* (RMSD) to the goal conformation is reduced. Otherwise it generates a new move by combining normal modes.

Finally some approaches use a hierarchical approach [71] adapting the number of degrees of freedom to their objective.





## Chapter 2

# An introduction to Game theory

### Contents

---

<b>2.1 Games presentation</b>	<b>25</b>
2.1.1 Extensive form	26
2.1.2 Normal form	27
<b>2.2 Dominance and Equilibria</b>	<b>28</b>
2.2.1 Dominated strategy	28
2.2.2 Nash Equilibria	29
2.2.3 Correlated Equilibria	30
2.2.4 Coarse Correlated Equilibria	31
<b>2.3 Iterative games and regret minimization</b>	<b>31</b>
2.3.1 Iterative games	31
2.3.2 Regret minimization	32
2.3.2.1 Internal regret	32
2.3.2.2 External regret	32
2.3.2.3 Expected Regret	32
2.3.2.4 Pseudo Regret	33
<b>2.4 Games under study</b>	<b>33</b>
2.4.1 Bandit and semi-bandit game	33
2.4.2 Exponential learning and no-regret	33

---

Game Theory appeared as a field in the middle of the 20th century with the book of John von Neumann and Oskar Morgenstern [72]. At first, game theory was mainly applied to politics and economy. More recently, it has also found important applications in biology and telecommunication networks, addressing problems like population dynamics and resources allocation (see [73] for examples).

Game theory addresses competitive problems that can be represented by players choosing a strategy among others. The strategy choice is dependent on other players actions.

In this chapter, we present the main concepts of game theory. The first section introduces important notions for game's definition and representations. The second section presents different games and equilibria. The third section introduces the concept of regret minimization. The last section presents a class of games that we will study later.

## 2.1 Games presentation

We focus on a certain class of games: the finite non-cooperative games.

*Finite* means that the number of players is finite and each players has a finite set of strategies.

*Non-cooperative* means that each player rationally acts to satisfy his own interest (players do not form coalitions).

Players tend to maximise the *utility* (also called *payoff*) they receive. Games can mainly be represented in two forms: the extensive form and the normal form.

### 2.1.1 Extensive form

Extensive form represents games as trees. The game starts at the root, at each node one of the players (depending on the node) makes a choice. Each edge leaving the node represents a possible strategy. The game ends at the leaves. The utility of each player is written at the leaves. This form is very convenient for representing games where players play sequentially. As an example we can consider the game illustrated in Figure 2.1. This game is a two player game - a store and a customer. The store plays first by choosing to put its articles on sale or to keep a normal price. The customer plays next by choosing to buy or not something. The utility of each players are written at the leaves.

Two different games can be studied on this example depending on the information available to the customer. In the perfect information setting, the customer can choose to act differently on sale or not. This situation is illustrated in the top tree of Figure 2.1, the customer knows if he is on the left (sale) or right (normal) node.

Without perfect information the customer does not know if he is on the left (sale) or the right (normal) node. That situation is represented in the bottom tree, the two nodes are gathered by the same line symbolizing that the customer can be in either one.

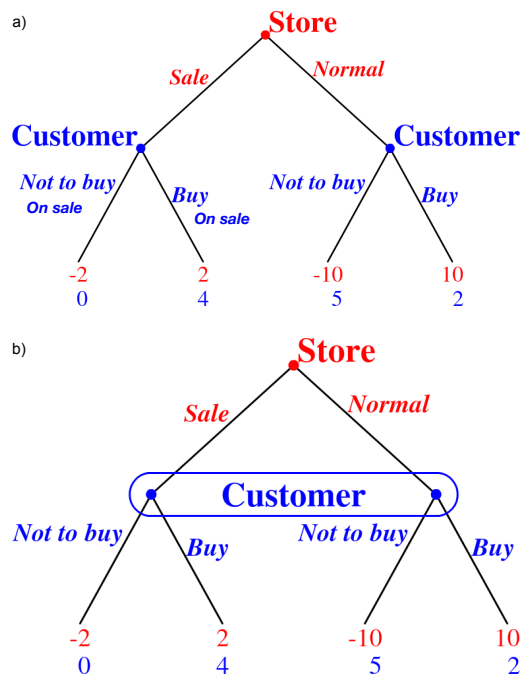


FIGURE 2.1: Tree representation of a game in extended form. On top (a), the customer has perfect information, at the bottom (b) he has imperfect information. The store has to choose between sale and normal price. a) In perfect information setting, the customer can choose differently for both choices of the store. b) In imperfect information setting, the customer does not know what the store chose. Payoffs are written at the leaves of the tree. The figures was made with the Game Theory Explorer website [74].

### 2.1.2 Normal form

Normal form is a natural way of representing games where players play simultaneously. The game can then be represented by a matrix, which dimensions are the number of strategies per players. The utilities of players are written in cells.

a)

		Customer			
		Not to buy on sale Not to buy	Not to buy on sale Buy	Buy on sale Not to buy	Buy on sale Buy
Store	Sale	0	0	4	4
	Normal	-2	-2	2	2
		5	2	5	2
		-10	10	-10	10

b)

		Customer	
		Not to buy	Buy
Store	Sale	0	4
	Normal	-2	2
		5	2
		-10	10

FIGURE 2.2: Matrix of the normal form of the same game. (a) Players have perfect information, the size of their strategies space grows exponentially with the number of choices they can do. (b) Players have imperfect information, the size of the matrix is smaller as the customer does not have to make a choice for each strategy of the store. However the number of distinct payoffs in both information setting is the same. They are just repeated perfect information.

Sequential games can also be represented in normal form. Figure 2.2 represents previous two player games in normal form. In this example each row corresponds to a strategy of the customer and each column to a strategy of the store. When the customer has imperfect information, the sequential game is equivalent to a one shot game, the order does not matter. However when players have a perfect information, the matrix size becomes huge: exponential on the number of choices players can do (Figure 2.2a). Indeed, to represent an extensive game into a matrix we need to take into account all the possible choices. In that case the customer can choose between four different strategies {(buy on sale, buy at normal price), (buy one sale, not to buy at normal price), (not to buy on sale, buy at normal price), (not to buy on sale, not to buy at normal price)}.

A normal form game can also be defined as a triplet  $(\mathcal{N}, \mathcal{S}, u)$ , representing the players, strategies and utility functions.  $\mathcal{N}$  denotes the set of  $N$  players indexed for 1 to  $N$ . Each player  $i$  has a finite set of strategies (called *pure strategies* or *pure actions*)  $\mathcal{S}_i = \{s_{i1}, s_{i2}, \dots, s_{iS_i}\}$  of cardinal  $S_i$ . The cartesian product of all  $\mathcal{S}_i$ ,  $\mathcal{S} = \prod_{i=1}^N \mathcal{S}_i$ , denotes all combinations of players' pure strategies. An element of  $\mathcal{S}$  corresponds to a pure strategy for each player. A player can also choose a non deterministic strategy called a *mixed strategy*.

**Definition 2.1.** A mixed strategy  $x_i$  of player  $i$  is a  $S_i$ -dimensional vector, corresponding to the probability of choosing each pure strategy. Letting  $x_{is}$  with  $s \in \mathcal{S}_i$  be the components of  $x_i$  we have  $\sum_s x_{is} = 1$  and  $0 \leq x_{is} \leq 1$ .

A mixed strategy of player  $i$  is  $x_i$ , a  $S_i$ -dimensional vector containing the probability for player  $i$  of playing each of its pure strategy. We denote  $\mathcal{X}_i = \Delta \mathcal{S}_i$  the continuous ensemble of mixed strategies of player  $i$  and  $\mathcal{X} = \prod_{i=1}^N \mathcal{X}_i$  the game's *strategy space*. An element  $x$  of  $\mathcal{X}$  contains a mixed strategy for each player, it can be written  $x = (x_1, \dots, x_N)$  where each  $x_i$  is a  $S_i$  dimensional vector. The

*support* of a mixed strategy is the pure strategies which probabilities are not null,  $\text{supp}(x_i) = \{s \in \mathcal{S}_i, x_{is} > 0\}$ . A pure strategy  $s_i$  is also a mixed strategy with only one non-zero coordinate, i.e. a singleton as support.

Each player  $i$  has an *utility function* (or *payoff function*)  $u_i : \mathcal{X} \rightarrow \mathbb{R}$ . We only consider *gain games*, players want to maximize their utility. A player  $i$  aims at maximizing its utility  $u_i(x) = u_i(x_1, \dots, x_N) = u_i(x_i, x_{-i})$  where  $x_{-i}$  is the action of all players except  $i$ .

## 2.2 Dominance and Equilibria

Now that we described what a game is and how games are represented, we will look into solution concepts. Algorithms are conceived in order to help players reaching a solution. We can define different solution concepts and so design different algorithms.

Most solutions describe a situation that rational players would choose. An intuitive way of characterizing the rationality of players is the elimination of *dominated strategy*.

### 2.2.1 Dominated strategy

A pure strategy is dominated if it is always suboptimal compared to another.

**Definition 2.2.** A pure strategy  $s_i$  of player  $i$  is dominated by  $s'_i$  (and written  $s_i \prec s'_i$ ) if

$$u_i(s_i; s_{-i}) \leq u_i(s'_i; s_{-i}) \quad \text{for all } s_{-i} \in \mathcal{S}_{-i} \equiv \prod_{j \neq i} \mathcal{S}_j, i \in \mathcal{N}. \quad (2.1)$$

When the inequality is strict the strategy  $s_i$  is *strictly dominated* by  $s'_i$ .

An interesting property for an algorithm is to guarantee that the probability of playing dominated strategies decreases to 0, thus guaranteeing that players will not play a suboptimal strategy.

(a) Prisoner's Dilemma payoff matrix:

		Prisoner 2	
		Confess	Silent
Prisoner 1	Confess	-3, -3	0, -5
	Silent	-5, -1	-1, -1

(b) Game matrix showing dominance elimination:

		P 2		
		A	B	C
P 1	A	2, 2	1, 1	-4, -4
	B	1, 1	0, 0	-1, -1
	C	-4, -4	-1, -1	-2, -2

FIGURE 2.3: (a) The *prisoner dilemma*: the strategy silent is strictly dominated for both players. The only strategy left is confess, although it leads to a smaller payoff than if both players remain silent. (b) The strategy C is strictly dominated by strategy B. Once the strategy C is removed, the strategy B is strictly dominated by the strategy A. After elimination of dominated strategies the only remaining is strategy A.

Figure 2.3 illustrates two games in normal form with strictly dominated strategies.

The *prisoner dilemma* is a two player games: two prisoners. They both have two possibilities either to confess or to remain silent. If they both remain silent, the police does not know which one is guilty and they both go to prison for one year. If one confesses and not the other, the one who confessed walks free in the other is incarcerated for five years. If they both confess they both go to prison for three years.

The silent strategy is strictly dominated by the confess strategy for both player. The only stable equilibrium is the confess/confess situation, although both players would do less prison in the silent/silent position. However if a player chooses to remain silent, the other has interest to confess so the situation is not stable.

The second game is borrowed from [75] and will be taken as example later, its stable equilibrium can be found by iteratively removing strictly dominated strategies. We can first see that strategy C is strictly dominated by strategy B. Once the strategy C is removed, the strategy B is strictly dominated by strategy A.

When only one strategy remains after iteratively removing strictly dominated strategies, the game is said *dominance-solvable*.

### 2.2.2 Nash Equilibria

A stronger and famous notion of rationality in a multi-player and non-cooperative game is a *Nash Equilibrium* (NE) [76]. In a NE, players have no interest in changing alone their strategy. Each player will not benefit from changing his strategy while the others keep their strategy.

**Definition 2.3.** A strategy  $x^*$  is a NE if and only if:

$$u_i(x_i^*; x_{-i}^*) \geq u_i(x_i; x_{-i}^*) \quad \forall i \in \mathcal{N}, \forall x_i \in \mathcal{X}_i.$$

		P2	
		Head	Tail
P1	Head	-1 1	1 -1
	Tail	1 -1	-1 1

FIGURE 2.4: The matching pennies games. Each player has a penny and can choose among two strategies - heads and tails. Player 1 wins if the two pennies match, player 2 wins if they do not.

When the inequality is strict the equilibrium is a strict NE. When the NE is a pure strategy it is a *pure Nash Equilibrium*.

In particular in a NE, strictly dominated strategies of the game have a probability 0 of being played. In the two examples in Figure 2.3 the remaining strategies are strict pure NE.

John Nash showed that any finite game admits at least a NE, however it does not have to be pure, it can be mixed. Figure 2.4 illustrates a game with no pure equilibrium. This game, the matching pennies, involves two players. Each player has a coin and can choose among two strategies - heads and tails. Player one wins 1 and player 2 wins -1 if the two pennies match. Player one wins -1 and player 2 wins 1 if the two pennies do not match. This game has no pure NE, however playing each strategy with probability 0.5 is a mixed NE.

The example of the Prisoner dilemma (Figure 2.3) shows that Nash Equilibria can be unsatisfactory by missing better payoffs because players cannot trust each other. To overcome this problem, we can study equilibria involving a coordinator.

### 2.2.3 Correlated Equilibria

*Correlated Equilibria* are a larger class of equilibria introduced by Robert Aumann [77]. Correlated equilibria involve a coordinator. The coordinator assigns a probability  $p(s)$  to each strategy in  $\mathcal{S}$ . A strategy  $s$  is then randomly drawn following the probability  $p$ . Each player is informed of the strategy the coordinator chose for him,  $s_i$ , but not of the others.  $p$  is a correlated equilibrium if no player can gain more by playing an other strategy than  $s_i$ .

		P2	
		Cross	Stop
P1	Cross	-10	0
	Stop	1	0

FIGURE 2.5: The traffic light games. Two drivers are at an intersection, each driver has two strategies: cross or stop. If they both cross, they crash and lose. If one stops and the other crosses the one that crosses wins.

**Definition 2.4.** The probability distribution  $p$  is a correlated equilibrium if for all players  $i \in \mathcal{N}$  and for all action  $s_i, s'_i \in \mathcal{S}_i$ ,

$$\sum_{s_{-i} \in \mathcal{S}_{-i}} u_i(s_i, s_{-i})p(s_i, s_{-i}) \geq \sum_{s_{-i} \in \mathcal{S}_{-i}} u_i(s'_i, s_{-i})p(s_i, s_{-i}) \quad (2.2)$$

The traffic light game (Figure 2.5) has three NE: two corresponding to crossing for only one car, one driver wins 1 the other wins nothing, one corresponding to

crossing with probability  $\frac{1}{11}$  and stopping with probability  $\frac{10}{11}$ , which leads to crashing with probability  $\frac{1}{121}$ , crossing alone with probability  $\frac{10}{121}$ , so a null expectation. This game has a correlated equilibrium with a strictly positive expectation equal for both players and avoiding the crashing situation. It consists in letting one car cross and stopping the other with probability  $\frac{1}{2}$  in a coordinated way: exactly what a traffic light does.

This class is broader than NE. Indeed given a NE,  $x^* = (x_1^*, \dots, x_N^*)$ , the probability distribution  $p(s) = \prod_{i=1}^N x_{is_i}^* \forall s \in \mathcal{S}$  such that  $s = (s_1, \dots, s_N)$  is a correlated equilibrium. A correlated equilibrium also gives 0 probability to strictly dominated strategies.

## 2.2.4 Coarse Correlated Equilibria

Correlated equilibria are included in a broader class, *Coarse Correlated Equilibrium* (CCE). A probability distribution  $p(s)$  chosen by a coordinator is a CCE if no player could gain more by always playing the same mixed strategy rather than playing the pure strategies the coordinator tells him to play.

**Definition 2.5.** The probability distribution  $p$  is a CCE if for all players  $i \in \mathcal{N}$  and for all action  $x \in \mathcal{X}_i$ ,

$$\sum_{s \in \mathcal{S}} u_i(s)p(s) \geq \sum_{s \in \mathcal{S}} u_i(x, s_{-i})p(s) \quad (2.3)$$

CCE includes correlated equilibria, but, contrary to them, they can attribute non-zero probability to strictly dominated strategies. For example in the panel b of Figure 2.3, playing (A,A), (B,B) and (C,C) with probability  $\frac{1}{3}$  is a CCE, although C is a strictly dominated strategy. Viossat and Zapechelnyuk [75] even showed that in some games a CCE can attribute positive probability only to dominated strategies.

## 2.3 Iterative games and regret minimization

### 2.3.1 Iterative games

Iterative games can involve one or multiple players. The same game is played a number of times (*time horizon*) that can be as large as wanted and even unknown.

Iterative games are very interesting when the utility is not fixed in a matrix or when players are not aware of the context, for example they do not know the dimension of the matrix. At each iteration players receive their utilities, and thus acquire a better knowledge of the game. Players need to find a trade-off between exploring all the strategies and exploiting the best strategy found so far.

When the utility is not fixed, it can depend on a probability law. It can also be chosen by an adversary that plays against the player. That case forbids deterministic algorithms because the adversary could predict each action and attribute a low utility to them.

The utility can also depend of an environnement, for example other players playing at the same game. We will focus latter on this setting. Each player plays as if it was alone, trying to minimize its regret. However, the utility received by a player depends on the choice of all players.



### 2.3.2 Regret minimization

Regret minimization is a concept in games where players play iteratively and want to optimize the sum of their payoffs. An algorithm is said to have a no-regret property if the regret is sublinear in the number of iterations, so the average over time of the regret is negative or converges to zero. There exist different definitions of regrets [78], we present four of them in the following.

#### 2.3.2.1 Internal regret

**Definition 2.6.** Given a player  $i$  with a strategy set  $\mathcal{S}_i$ , an utility function  $u_i$ , and a time horizon  $T$ , its internal regret is

$$R_i(T) = \max_{s \in \mathcal{S}_i} \max_{s' \in \mathcal{S}_i} \left( \sum_{t=1}^T [u_i(s', s_{-i}(t)) - u_i(s_i(t), s_{-i}(t))] \mathbb{1}_{s=s_i(t)} \right). \quad (2.4)$$

In words, the internal regret is the difference between the payoff of the player and the maximal payoff he could have earned by inverting strategies. That is to say, by playing  $s'$  instead of  $s$  each time the coordinator told him to play  $s$ . The internal regret is always positive.

#### 2.3.2.2 External regret

**Definition 2.7.** Given a player  $i$  with a strategy set  $\mathcal{S}_i$ , an utility function  $u_i$ , and a time horizon  $T$ , its external regret is

$$R_e(T) = \max_{s \in \mathcal{S}_i} \left( \sum_{t=1}^T u_i(s, s_{-i}(t)) - u_i(s_i(t), s_{-i}(t)) \right). \quad (2.5)$$

In words, the external regret is the difference between the gain of the player and the maximal gain he could earn by playing the same strategy at each step. The external regret can be negative, if the choices of the player are better than following any fixed strategy at each step.

It has been shown that if all players play according to a no-regret (external) algorithm, then their strategies converge to the *Hannan set* which is also the set of CCE [79], [80].

Taking into account that the payoff of an action can be non-deterministic and that players can also play in a non-deterministic way (mixed strategies), we can define two other types of regret based on expectation [81].

#### 2.3.2.3 Expected Regret

**Definition 2.8.** Given a player  $i$  with a strategy set  $\mathcal{S}_i$  and an utility function  $u_i$ , its expected regret is

$$R_{exp}(T) = \mathbb{E}_{s(t) \sim x(t)} \left[ \max_{s' \in \mathcal{S}_i} \left( \sum_{t=1}^T u_i(s', s_{-i}(t)) - u_i(s_i(t), s_{-i}(t)) \right) \right]. \quad (2.6)$$

The expected regret is the expectation of the external regret.

### 2.3.2.4 Pseudo Regret

**Definition 2.9.** Given a player  $i$  with a strategy set  $\mathcal{S}_i$  and an utility function  $u_i$ , its pseudo regret is

$$R_{pseudo}(T) = \max_{s' \in \mathcal{S}_i} \mathbb{E}_{s(t) \sim x(t)} \left[ \sum_{t=1}^T u_i(s', s_{-i}(t)) - u_i(s_i(t), s_{-i}(t)) \right]. \quad (2.7)$$

Given those definitions we have  $R_{pseudo}(T) \leq R_{exp}(T)$ . In the case of stochastic utility, deterministic algorithms were developed to minimize the pseudo regret [82].

## 2.4 Games under study

We will now present *bandit* and *semi-bandit games* that were extensively studied in terms of regret minimization.

### 2.4.1 Bandit and semi-bandit game

Bandit and semi-bandit games usually involve one player. The player chooses a strategy among his strategy set and receives his utility. In bandit game, the player only knows the utility of the chosen strategy. In contrary, in semi-bandit game, the player knows (after playing) the utility he would have received by playing any other strategy.

We will focus on the case when the utility depends on the environnement. More particularly we focus on multi-player games, so the utility function is a function of  $\mathcal{S} = \prod_{i=1}^N \mathcal{S}_i$  into  $\mathbb{R}$ . However players are not aware of other players actions. Therefore the utility they receive for a strategy can change over time as it also depends of the strategies played by others players.

### 2.4.2 Exponential learning and no-regret

We are now presenting an *Exponential Weights* (EW) algorithm [83] that we will study in Chapters 5 and 6. EW algorithms attribute a weight to each strategy. The probability of playing a pure strategy is proportional to the exponential of this strategy's weight. At each step the weights of strategies are increased or decreased according to the utility received. Here we present one example of EW algorithms in the bandit setting and its no-regret properties.

**Algorithm 2.1** HEDGE with time horizon  $T$

- 1 Initially set  $y_s(1), \forall s \in \mathcal{S}$  and the initial mixed strategy  $x(1) = \Lambda(y(1))$  where the logit map  $\Lambda$  is defined as

$$\Lambda(y) = \frac{1}{\sum_{s \in \mathcal{S}} \exp(y_s)} (\exp(y_s))_{s \in \mathcal{S}}. \quad (2.8)$$

- 2 **for** each round  $t$ :
- 3     The player draws a pure strategy  $s(t)$  according to  $x(t)$
- 4     The player gets his payoff vector  $v(s(t))$
- 5     The player updates his mixed strategy  $x$  via the recursion

$$\begin{aligned} y(t+1) &= y(t) + \frac{\sqrt{S}}{T} v(t), \\ x(t+1) &= \Lambda(y_{t+1}) \end{aligned} \quad (\text{HEDGE})$$

**end for**

Littlestone and Warmuth showed, in 1994, [84] that if the utility is always in  $[0, 1]$ , such an algorithm guarantees an external regret of at most  $2\sqrt{T \log(S)} = o(T)$  and hence it is a no-external-regret algorithm. Many variants of this algorithm have been developed to consider cases when the time horizon is unknown or in bandit setting [81].

We will study this algorithm and a variant in a multi-players setting in Chapter 5 and 6. However, instead of analyzing the regret of each player, we will analyze the convergence of the players strategies. We show that while individually playing a no-regret algorithm, players collectively converge towards a Nash Equilibrium.

## Chapter 3

# RNA conformation explorations with a robotics approach

### Contents

---

<b>3.1</b>	<b>Molecule representation</b> . . . . .	<b>36</b>
3.1.1	Kinematic graph . . . . .	36
3.1.2	Secondary structure conservation . . . . .	37
<b>3.2</b>	<b>Clash-free directed motion</b> . . . . .	<b>40</b>
3.2.1	Dynamic Clash-avoiding Constraints . . . . .	40
3.2.2	Directed motion . . . . .	41
3.2.2.1	Least square motion . . . . .	41
3.2.2.2	Comparison with nullspace projection . . . . .	42
3.2.3	Procedure overview and dataset . . . . .	47
3.2.3.1	Procedure overview . . . . .	47
3.2.3.2	Dataset . . . . .	48
<b>3.3</b>	<b>Results</b> . . . . .	<b>48</b>
3.3.1	Benchmark results . . . . .	48
3.3.1.1	Comparison to iMOD . . . . .	48
3.3.1.2	Performances using every fifth C5 atom . . . . .	50
3.3.1.3	Maintaining secondary structure between substates . . . . .	51
3.3.1.4	Distance to the heavy atom goal positions . . . . .	52
3.3.1.5	Study of 3D helical arrangements . . . . .	54
3.3.2	Conformational exchange of riboswitch <i>glnA</i> . . . . .	55
3.3.3	Run time . . . . .	56
<b>3.4</b>	<b>Perspectives and conclusions</b> . . . . .	<b>57</b>
3.4.1	Relative distances . . . . .	58
3.4.1.1	Relative distances approach . . . . .	58
3.4.1.2	Randomization . . . . .	58
3.4.1.3	Preliminary results . . . . .	58
3.4.1.4	Projected applications . . . . .	58
3.4.2	Conclusions . . . . .	59

---

In this chapter we present a kinematics-based procedure to morph an RNA molecule between conformational substates, while avoiding inter-atomic clashes. This procedure led to a publication [85]. Except for Figure 3.1, the figures of this chapter are taken from this publication.

RNAs and especially *non-coding Ribonucleic Acids* (ncRNAs) are extremely dynamic. Their three dimensional structure, called also conformation, fluctuates a lot and can access different conformational substates. This ability allows them to modulate their interaction with others molecules. The flexibility of ncRNAs provides a challenge for probing their complex three dimensions conformational landscape.

However, despite their conformational diversity, many ncRNAs preserve their secondary structure throughout their conformational changes. Notable counterexamples are studied in secondary structures prediction (Section 1.1.2), but here we focus on RNAs that preserve their secondary structure almost perfectly (a few base pair changes can be manually considered).

Our method is based on the observation that junctions between RNA helices are highly flexible and modulate the tertiary structure. It takes three inputs: the full atom initial conformation, its secondary structure and a few atom positions of the goal conformation. Atoms with a known goal position are called marker atoms. The RNA is represented by a 3D unweighted graph, i.e., lengths of edges are fixed and play no role in the procedure however vertices have 3D coordinates which directly correspond to the position of atoms in the molecule. The trajectory is build iteratively, step by step starting from the initial conformation. The final conformation gets remarkably close to the full atom goal conformation even when the marker atoms represent less than 1% of the number of atoms.

The first section presents the graph approach, *Kino-Geometric Sampling* (KGS), based on previous work mainly for proteins [86]–[88] but also for RNAs [89]. The second section presents the computational approach, a least square approach and compares it to the previous gradient descent approach. The dynamic clash-avoiding constraints procedure is also described at the end of Section 2. The last section presents the dataset and the results of the morphing method developed as well as unpublished extensions.

## 3.1 Molecule representation

Although the representation holds for proteins as well as for RNAs, we will only consider RNAs as applications presented at the end of the chapter only concern RNAs. This is a semi coarse-grained approach: all atoms are represented and taken into account, but only main *degrees of freedom* are considered. This allows us to have the precision of an all atoms trajectory, avoiding collisions between any atoms. However we sensibly reduce the dimension of the conformational space by freezing minor degrees of freedom like bond length fluctuations or rotations of double bonds. In addition, the methods maintains secondary structure which tremendously reduces the accessible space.

### 3.1.1 Kinematic graph

We first represent a RNA molecule by a kinematic graph, i.e, a set of vertices connected by edges. Unlike usual graphs in computer sciences, vertices have 3-dimension coordinates. Each atom is a vertex and each bond of the molecule is an edge of the kinematic graph. Bonds lengths and planar angles are fixed to their initial value (value in the initial conformation). The only degrees of freedom are torsions around bonds.

Therefore, bonds considered non-rotatable are removed and their end-vertices are collapsed. A vertex is also called a *rigid body*. Atoms in the same vertex do not move relatively to each others. Non-rotatable bonds are partial double bonds, bonds with one extremity that is linked to no other vertices, and bonds in rings (ribose). This first step represents the RNA molecule by a graph (Figure 3.1a and c).

Then we transform the graph into a tree (Figure 3.1b), i.e a directed graph with no cycles, by choosing a root and directing each edge so that it goes from the root

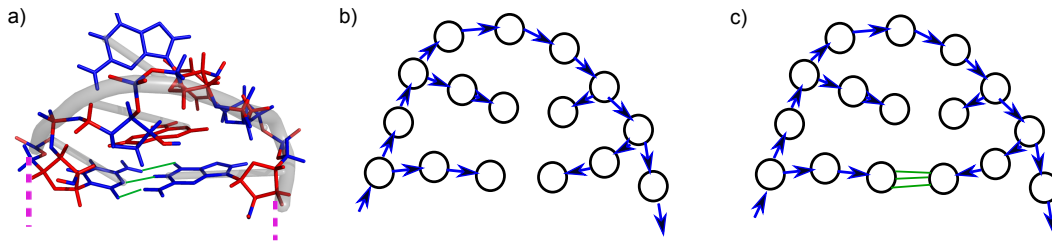


FIGURE 3.1: a) Kinematic graph on an RNA. Atoms in a same rigid body are connected and with the same color: red or blue. b) Kinematic tree c) Directed graph with hydrogen bonds (green edges).

to leaves. RNAs are first of all linear sequence of nucleotides (see Section 1.1), this guarantees that the resulting graph is a tree: it does not contains cycles.

Until now, the secondary structure is not taken into account. We add secondary structure information obtained from the initial conformation with RNAView [17]. A pair of nucleotides is formed by two or three hydrogen bonds between the two nucleotides. Hydrogen bonds are formed between a donor atom and an acceptor. The donor has an hydrogen and "share" it with the acceptor. Only hydrogen bonds participating in a WC base pairs are added to the tree. This creates cycles and transforms the tree into a directed graph (Figure 3.1c).

**Notations** Let  $n$  be the number of edges<sup>1</sup> of the tree, also called degrees of freedom. They are rotatable bonds. Let  $m$  be the number of hydrogen bonds. The number of edges in the final graph (Figure 3.1c) is  $n+m$ . A conformation  $q$  is fully characterized by a vector of  $n$  angles,  $q \in \mathbb{R}^n$ . Let  $f_I(q) \in \mathbb{R}^3$  be the forward endpoint map for atom  $I$ , i.e,  $f_I(q)$  is the position of atom  $I$  in the conformation characterized by  $q \in \mathbb{R}^n$ .

Each hydrogen bond adds five constraints: two constraints on angles and three constraints to fix the distance between the donor and the acceptor (Figure 3.2).

### 3.1.2 Secondary structure conservation

The mouvement is performed by rotation around edges. A rotation only affects atoms that are in rigid bodies below the rotated edge. The nearest common ancestor of two rigid bodies is their closest common parent. This is an important notion as any rotation between the root and a common ancestor has the same effect on both rigid bodies. An hydrogen bond closes a circle (Figure 3.2), which includes the vertex of the acceptor,  $V_A$ , the vertex of the donor,  $V_H$  and their nearest common ancestor, vertex  $V_{NCA}$ . Note that  $V_A$  and  $V_H$  are now connected by the hydrogen bond. Therefore the left rotations, in Figure 3.2, from vertex  $V_{NCA}$  to vertex  $V_A$  as well as the right rotations from vertex  $V_{NCA}$  to vertex  $V_H$  affect all of their atoms. Let  $f_I^R(q) \in \mathbb{R}^3$  (resp.  $f_I^L(q) \in \mathbb{R}^3$ ) be the forward endpoint map for atom  $I$  that could be: the acceptor ( $A$ ), the acceptor neighbor ( $AA$ ), the hydrogen ( $H$ ) or the hydrogen donor ( $D$ ), relatively to the right (R) (resp. left (L)) rotations. For example  $f_H^R(q)$  is the position of the hydrogen atom after rotations from the right side of the circle and  $f_H^L(q)$  is its position after rotations from the left side of the circle.

The angles constraints can be written:

$$E1(q) = (f_A^R(q) - f_H^R(q))^T (f_H^L(q) - f_D^L(q)) = \cos(\theta)d(A, H)d(H, D) = c \quad (3.1)$$

<sup>1</sup>Note that these notations are not the usual ones in graph theory.

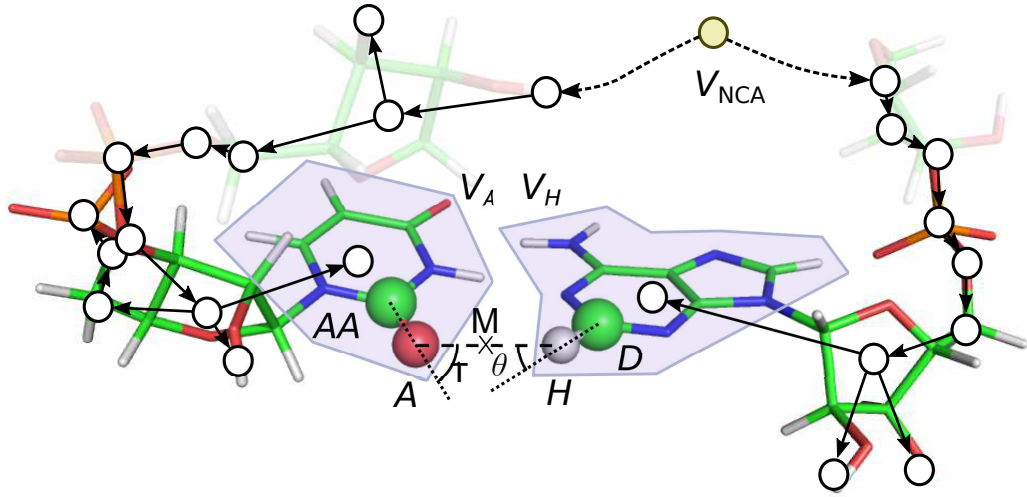


FIGURE 3.2: Portion of the kinematic graph. Angle and bond length constraints of an hydrogen bond.  $V_A$  and  $V_H$  are respectively the vertices of the acceptor and the hydrogen atoms.  $H$  is the hydrogen atom, and  $D$  the donor,  $A$  is the acceptor atom and  $AA$  is its first neighbor. Angles  $\tau$  and  $\theta$  are planar angles between these four atoms. The hydrogen bond imposes a coordinate motion to preserve angles  $\tau$  and  $\theta$  and for the middle point  $M$  to have the same position with the left side rotations and the right side rotations.  $V_{NCA}$  is the nearest common ancestor of  $V_A$  and  $V_H$ .

$$E2(q) = (f_H^L(q) - f_A^L(q))^T (f_A^R(q) - f_{AA}^R(q)) = \cos(\tau)d(A, H)d(A, AA) = t \quad (3.2)$$

where  $d(A, H)$  is the distance between atoms  $A$  and  $H$ .  $c$  and  $t$  are two constants because planar angles and bond lengths are preserved. This gives us two equations that preserve the orientation of the segment  $AH$ .

The three constraints for the equality of the middle point between  $A$  and  $H$  from the right and the left side are:

$$\begin{cases} E3(q) = f_H^L(q).x + f_A^L(q).x - f_H^R(q).x - f_A^R(q).x = 0 \\ E4(q) = f_H^L(q).y + f_A^L(q).y - f_H^R(q).y - f_A^R(q).y = 0 \\ E5(q) = f_H^L(q).z + f_A^L(q).z - f_H^R(q).z - f_A^R(q).z = 0 \end{cases} \quad (3.3)$$

These 5 equations have to be always satisfied for all bonds. Let  $J$  be the  $5m \times n$  jacobian matrix of all constraints (i.e. hydrogen bonds in WC interactions). A perturbation  $\delta q \in \mathbb{R}^n$  is admissible if it satisfies:

$$J\delta q = \begin{pmatrix} \frac{\partial E1_1}{\partial q_0} & \frac{\partial E1_1}{\partial q_1} & \cdots & \frac{\partial E1_1}{\partial q_n} \\ \frac{\partial E2_1}{\partial q_0} & \frac{\partial E2_1}{\partial q_1} & \cdots & \frac{\partial E2_1}{\partial q_n} \\ \frac{\partial E3_1}{\partial q_0} & \frac{\partial E3_1}{\partial q_1} & \cdots & \frac{\partial E3_1}{\partial q_n} \\ \frac{\partial E4_1}{\partial q_0} & \frac{\partial E4_1}{\partial q_1} & \cdots & \frac{\partial E4_1}{\partial q_n} \\ \frac{\partial E5_1}{\partial q_0} & \frac{\partial E5_1}{\partial q_1} & \cdots & \frac{\partial E5_1}{\partial q_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial E5_m}{\partial q_0} & \frac{\partial E5_m}{\partial q_1} & \cdots & \frac{\partial E5_m}{\partial q_n} \end{pmatrix} \delta q = 0 \quad (3.4)$$

when  $Ei_j, i \in [1, 5], j \in [1, m]$  corresponds to equation  $Ei$  of hydrogen bond  $j$ , ensuring that all constraints are satisfied at the first order approximation. All displacements are rotations, we ensure that constraints are satisfied at the first order approximation. Therefore the expected error is quadratic on the rotation amplitude (Figure 3.3).

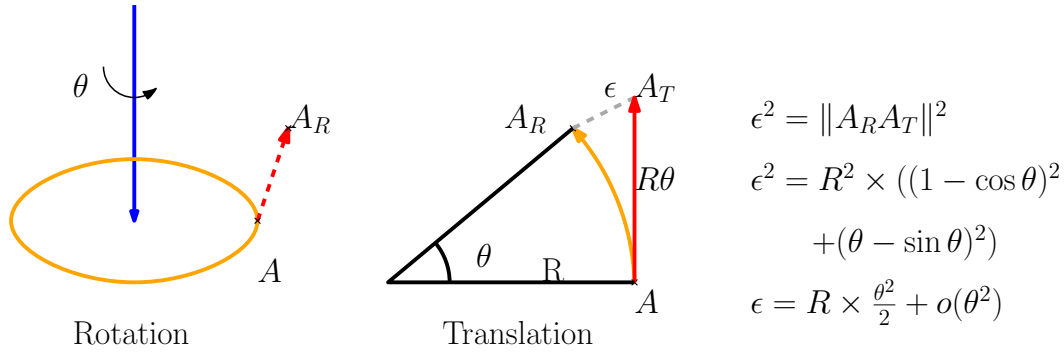


FIGURE 3.3: Illustration of the error between a translation along the tangent vector and a rotation.

Equation 3.4 is equivalent to say that a small perturbation  $\delta q$  is admissible if it lays in the nullspace of the matrix  $J$ , so if

$$\delta q \in \ker J$$

The dimension of the nullspace  $\ker J$  is  $n - r$  where  $r$  is the rank of  $J$ . Both over- and under-constrained scenarios are possible, i.e.,  $5m > n$  or  $5m < n$ , with  $r \leq \min(5m, n)$ . In practice, hydrogen bonds often add redundant constraints. So even if  $5m > n$ ,  $r < n$  leading to  $n - r$  remaining internal degrees of freedom in the nullspace. In rare cases<sup>2</sup>,  $r = n$  and the nullspace is empty, which means that no perturbation will stay in the constraint manifold.

To compute the nullspace,  $N$ , of  $J$  we perform a *Singular Value Decomposition* (SVD)

$$J = U\Sigma V^T \quad (3.5)$$

where  $U$  is a  $5m$  by  $5m$  unitary matrix,  $\Sigma$  is a  $5m$  by  $n$  rectangular diagonal matrix with non-negative real numbers on the diagonal, and  $V$  is a  $n$  by  $n$  unitary matrix. The diagonal entries  $\sigma_i$  of  $\Sigma$  are known as the singular values of  $J$ .

$N \in \mathbb{R}^{n \times (n-r)}$  is the  $n - r$  last column of  $V$ , i.e, the right-singular vectors with vanishing singular values. A small perturbation  $\delta q$  is admissible if it exists a vector  $u \in \mathbb{R}^{n-r}$  such that

$$\delta q = Nu. \quad (3.6)$$

The trajectory is constructed by small steps. A perturbation  $\delta q$  is added to a conformation  $q$  to give a new conformation  $q'$  with  $\delta q, q, q' \in \mathbb{R}^n$ . By choosing a small perturbation parallel to an admissible perturbation, we guarantee that hydrogen bonds, important for the secondary structure, are preserved. Note that the jacobian matrix and its nullspace are updated at every step.

<sup>2</sup>It happens in some case after adding constraints see Section 3.2.1



## 3.2 Clash-free directed motion

Once we have defined the manifold in which the small perturbation needs to be, we can describe equations directing the motion towards the goal position of marker atoms. We want to minimize at each step the distance between the position of marker atoms and their goal positions. However doing so will inevitably lead to clashes between atoms. We therefore only use perturbations in a clash-avoiding submanifold of the admissible perturbation.

### 3.2.1 Dynamic Clash-avoiding Constraints

The dynamic clash-avoiding constraints were developed by Budday [68] and they are used in this method. We define a clash between a pair of atoms whenever the distance between their centers is less than the sum of their van der Waals radii, scaled by a parameter  $c_f$ . To avoid a clash between a pair of atoms  $A$  and  $B$ , we redirect their motion by adding temporary, an one-dimensional constraint to forbid uncoordinated motion along  $n_c$ . Where  $n_c$  is the direction of the line joining the centers  $p_A$  and  $p_B$  of both atoms. This constraint can be written as:

$$n_c^T \left( \frac{\partial f_B}{\partial q} - \frac{\partial f_A}{\partial q} \right) \delta q = 0. \quad (3.7)$$

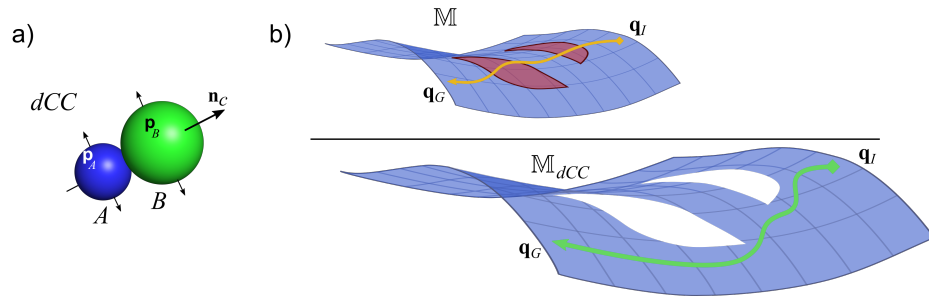


FIGURE 3.4: a) Two colliding atoms, resulting in a dynamic Clash-avoiding Constraint (dCC) that prevents  $A$  and  $B$  from approaching, but allows motion in directions orthogonal to  $n_c$  or jointly along  $n_c$ . b) In the top panel, the blue plane represents the secondary-structure constraint accessible manifold,  $\mathbb{M}$ , with red patches corresponding to sterically unfeasible regions. Moving directly from the initial conformation,  $q_I$ , to the goal,  $q_G$ , will frequently enter colliding regions. In the bottom panel, the clash regions are absent in the resulting lower-dimensional submanifold  $\mathbb{M}_{dCC}$  (cut-out), and the path between  $q_I$  and  $q_G$  is clash-free.

The constraint ensures that  $(p_A - p_B) \cdot n_c$  does not change after the perturbation (Figure 3.4.a), so the distance between  $A$  and  $B$  can only increase. Note that the constraint also permits independent motion of  $A$  and  $B$  on the plane perpendicular to  $n_c$ . The temporary, additional constraints define a subspace of the secondary constraint manifold  $\mathbb{M}$ ; motions in this submanifold ( $\mathbb{M}_{dCC}$ ) avoid clashes. Each perturbation of a conformation starts without temporary constraints, and they are

added as necessary. We call the additional constraints dynamic Clash-avoiding Constraints (dCC); each adds one row to the Jacobian matrix  $J$ . This procedure permits to avoid clashes while moving in the same direction, instead of only reject clashing conformations, or just moving away from the collisions without keeping the same direction.

### 3.2.2 Directed motion

The goal conformation is represented by the coordinates of  $k$  given marker atoms  $A_i^G, i = 1, \dots, k$ . Let  $\Delta(q) \in \mathbb{R}^{3k}$  be the vector of directions between marker atom positions in conformation  $q$  and in the goal conformation (Figure 3.5).

$$\Delta(q) = \begin{pmatrix} A_1^G - f_{A_1}(q) \\ \vdots \\ A_k^G - f_{A_k}(q) \end{pmatrix}. \quad (3.8)$$

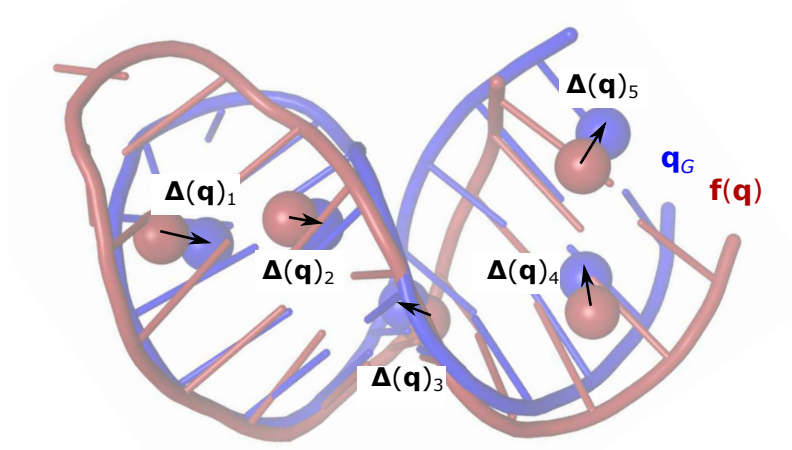


FIGURE 3.5: 3D representation of an intermediate ( $f(q)$ , red) conformation and the goal conformation (blue). From the goal conformation,  $q_G$ , only the positions of marker atoms (blue spheres) are used to drive the transition. Black arrows indicate directions  $\Delta(q)_i$  from the intermediate to the goal position of marker atom  $i$ .

An small perturbation  $\delta q$  of a conformation  $q$  leads to a displacement of atom  $A_i$  positions  $\delta f_{A_i} = \sum_{j=0}^{n-1} \frac{\partial f_{A_i}}{\partial q_j} \delta q_j$ .

We define  $M \in \mathbb{R}^{3k \times n}$ , the Jacobian matrix of the vector  $f_A = (f_{A_1}(q), f_{A_2}(q), \dots, f_{A_k}(q))$  as follows:

$$M = \begin{pmatrix} \frac{\partial f_{A_1}}{\partial q_0} & \frac{\partial f_{A_1}}{\partial q_1} & \dots & \frac{\partial f_{A_1}}{\partial q_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial f_{A_k}}{\partial q_0} & \frac{\partial f_{A_k}}{\partial q_1} & \dots & \frac{\partial f_{A_k}}{\partial q_n} \end{pmatrix}. \quad (3.9)$$

Let  $M\delta q$  be an infinitesimal displacement of the marker atoms.

#### 3.2.2.1 Least square motion

Our objective is to obtain a new conformation with marker atoms as close as possible to their goal positions. This is equivalent to minimizing the difference between the

displacement of marker atoms  $M\delta q$  and  $\Delta(q)$

$$\min_{\delta q} \|M\delta q - \Delta(q)\|_2, \quad (3.10)$$

at each step. In addition, to satisfy constraints,  $\delta q$  should also be an admissible perturbation, i.e. it should follow (3.6),  $\delta q = Nu$ . Substituting into (3.10) we obtain:

$$\min_u \|MNu - \Delta(q)\|_2. \quad (3.11)$$

A least square solution to (3.10) is given by  $u_{ls} = (MN)^\dagger \Delta(q)$ , where  $(MN)^\dagger$  is a pseudo-inverse.<sup>3</sup> Therefore

$$\delta q_{ls} = N(MN)^\dagger \Delta(q) \quad (3.12)$$

If the norm of  $\delta q_{ls}$  is small enough, then  $\delta q = \delta q_{ls}$ , and the goal positions are reached exactly. Else,  $\delta q = \epsilon \delta q_{ls}$  with  $\epsilon$  being a scaling factor. While the minimizing direction, i.e., the infinitesimal perturbation along  $\delta q_{ls}$  is calculated exactly, our first-order approximation to the constraint manifold dictates an iterative approach along this direction using small steps. This adds a complication, as  $M$ ,  $N$ ,  $\Delta(q)$ , and therefore  $u_{ls}$  must be updated at each iteration.

### 3.2.2.2 Comparison with nullspace projection

The previous work on KGS [86]–[89], used a projection into the nullspace of a gradient descent instead of choosing the perturbation in the nullspace.

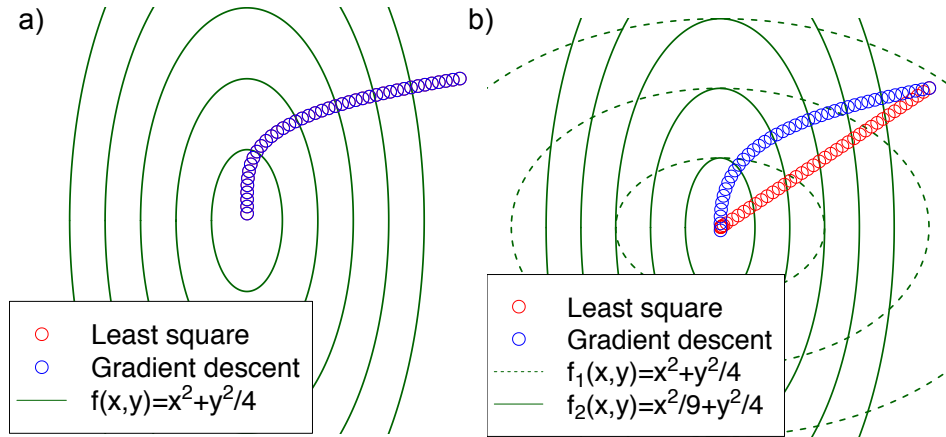


FIGURE 3.6: Comparison between least square and gradient methods. a) There is only one function to minimize. The gradient descent and the least square method follow the same path. b) There are two functions to minimize at the same time ( $f_1$  and  $f_2$ ). The least square method follows a straight line and gets close to the optimum a few steps before the gradient descent method.

<sup>3</sup>A pseudo-inverse of  $A$  is a matrix  $B$  such that  $ABA=A$ ,  $BAB=B$ ,  $AB$  and  $BA$  are Hermitian.

Here we compare these two methods. First, we compute a step in the gradient method. The function to minimize is:

$$\begin{aligned} F(q) &= \|\Delta(q)\|_2^2 \\ &= (A^G - f_A(q))^T (A^G - f_A(q)) \\ &= (A^G)^T A^G - 2(f_A(q))^T A^G + f_A(q)^T f_A(q) \end{aligned} \quad (3.13)$$

Its gradient is:

$$\begin{aligned} \nabla F(q) &= 2\nabla(f_A(q))(f_A(q) - A^G) \\ &= 2M^T(f_A(q) - A^G) \\ &= -2M^T \Delta(q) \end{aligned} \quad (3.14)$$

Therefore the perturbation is along  $-\nabla F(q) = 2M^T \Delta(q)$ . However we only accept admissible perturbation, so the perturbation is projected onto the nullspace with the projection matrix  $NN^T$ :

$$\delta q_g = \gamma NN^T M^T \Delta(q) \quad (3.15)$$

With  $\gamma$  being the chosen step size.

**Projection into the nullspace** Note that for the descent gradient method, working in the nullspace by finding  $u$  such that  $\delta q = Nu$  (3.6) is completely equivalent to projecting the full space gradient into the nullspace. In deed the gradient in the nullspace is

$$u_g = \gamma(MN)^T \Delta(q) \quad (3.16)$$

Therefore  $\delta q_g = \gamma NN^T M^T \Delta(q)$ .

For the least square method there is a difference. The full space least square solution is:

$$v_{ls} = (M)^\dagger \Delta(q) \quad (3.17)$$

Which gives  $NN^T(M)^\dagger \Delta(q)$  once projected into the nullspace.  $N^T$  and  $N^\dagger$  are usually not equal.

**Error comparison** We now compute the distance to marker atoms at the next step. Recall that  $M$  is Jacobian matrix of the vector  $f_A = (f_{A_1}(q), f_{A_2}(q), \dots, f_{A_k}(q))$  so it corresponds to the first order approximation: translations instead of rotations.

$$\begin{aligned} \Delta(q(t+1)) &= A^G - f_A(q(t+1)) \\ &= A^G - f_A(q(t)) - M\delta(q(t)) + o(\delta(q(t))^2) \\ &= \Delta(q(t)) - M\delta(q(t)) + o(\delta(q(t))^2) \end{aligned} \quad (3.18)$$

In order to compare the two methods, we compare the distance to marker atoms after one step for both methods. In the least square case we have

$$\Delta(q(t+1))_{ls} = \Delta(q(t)) - \epsilon MN(MN)^\dagger \Delta(q) + o(\|\epsilon N(MN)^\dagger \Delta(q)\|^2). \quad (3.19)$$

In the gradient method case we have

$$\Delta(q(t+1))_g = \Delta(q(t)) - \gamma MN(MN)^T \Delta(q) + o(\|\gamma N(MN)^T \Delta(q)\|^2). \quad (3.20)$$

Note that to do a fair comparison we need to assume that both steps have the same norm.

$$\|\delta q_{ls}\|_2 = \|\delta q_g\|_2 = \alpha \text{ so } \frac{\epsilon}{\gamma} = \frac{\|NN^T M^T \Delta(q)\|_2}{\|NN^\dagger M^\dagger \Delta(q)\|_2} \quad (3.21)$$

Let  $U'\Sigma'V'^T$  be the SVD decomposition of  $MN$ . We have  $MN = U'\Sigma'V'^T$ ,  $(MN)^T = V'\Sigma'U'^T$ , and  $(MN)^\dagger = V'\Sigma'^\dagger U'^T$ .  $\Sigma'^\dagger$  is the pseudo inverse of  $\Sigma'$  so it is a diagonal matrix with the inverse singular values as  $\Sigma'$  when  $\Sigma'$  has non zero singular values and 0 elsewhere. Let  $Id^*$  be the identity matrix of size  $3k$  by  $3k$  with lines of 0 at the bottom such that  $\text{rank}(Id^*) = \text{rank}(MN) = \text{rank}(\Sigma')$ .

Therefore

$$\begin{aligned} \|\Delta(q(t+1))_{ls}\|_2 &= \|\Delta(q(t)) - \epsilon MN(MN)^\dagger \Delta(q) + o(\alpha^2)\|_2 \\ &= \|U'U'^T \Delta(q(t)) - \epsilon U'Id^*U'^T \Delta(q) + o(\alpha^2)\|_2 \\ &= \|U'(Id - \epsilon Id^*)U'^T \Delta(q(t)) + o(\alpha^2)\|_2 \end{aligned} \quad (3.22)$$

At first order approximation we have

$$\begin{aligned} \|\Delta(q(t+1))_{ls}\|_2 &= \|(Id - \epsilon Id^*)U'^T \Delta(q(t))\|_2 \\ &= \|(Id - \gamma \frac{\|NV'\Sigma'U'^T \Delta(q)\|_2}{\|NV'\Sigma'^\dagger U'^T \Delta(q)\|_2} Id^*)U'^T \Delta(q(t))\|_2 \end{aligned} \quad (3.23)$$

Note that for the least square algorithm to follow a straight line (Figure 3.6b) we need  $\Delta(q(t+1))_{ls}$  and  $\Delta(q(t))$  to be parallel. This implies that  $Id^* = Id$ , that is to say, if  $\text{rank}(MN) = 3k$  so  $MN$  is full rank and has more columns than rows and that  $\Delta(q(t))$  is a singular vector of  $U'$ .

$$\begin{aligned} \|\Delta(q(t+1))_g\|_2 &= \|\Delta(q(t)) - \gamma MN(MN)^T \Delta(q) + o(\alpha^2)\|_2 \\ &= \|U'U'^T \Delta(q(t)) - \gamma U'\Sigma'^2 U'^T \Delta(q) + o(\alpha^2)\|_2 \\ &= \|U'(Id - \gamma \Sigma'^2)U'^T \Delta(q(t)) + o(\alpha^2)\|_2 \end{aligned} \quad (3.24)$$

At first order approximation we have

$$\|\Delta(q(t+1))_g\|_2 = \|(Id - \gamma \Sigma'^2)U'^T \Delta(q(t))\|_2 \quad (3.25)$$

The rest of the comparison is done in a first order approximation. We now study a particular case: assume that  $N$  is the full space (i.e. there is no constraint or we only consider the constrained manifold), that  $M$  is invertible, and that  $U'^T \Delta(q) = (1\dots 1)^T$ .  $\sigma'_i$  are the singular value of  $MN$ , i.e., the value on  $\Sigma'$  diagonal. Using these hypotheses, Equation 3.21 becomes:

$$\frac{\epsilon}{\gamma} = \frac{\|\Sigma'U'^T \Delta(q)\|_2}{\|\Sigma'^{-1}U'^T \Delta(q)\|_2} = \frac{\sqrt{\sum_{i=1}^{3k} \sigma_i'^2}}{\sqrt{\sum_{i=1}^{3k} \frac{1}{\sigma_i'^2}}}$$

and,

$$\begin{aligned}\|\Delta(q(t+1))_{ls}\|_2 &= \left\| \left( Id - \gamma \frac{\sqrt{\sum_{i=1}^{3k} \sigma_i'^2}}{\sqrt{\sum_{i=1}^{3k} \frac{1}{\sigma_i'^2}}} Id \right) U'^T \Delta(q(t)) \right\|_2 \\ &= \sqrt{3k} \left\| 1 - \gamma \frac{\sqrt{\sum_{i=1}^{3k} \sigma_i'^2}}{\sqrt{\sum_{i=1}^{3k} \frac{1}{\sigma_i'^2}}} \right\|_2\end{aligned}$$

$$\begin{aligned}\|\Delta(q(t+1))_g\|_2 &= \left\| (Id - \gamma \Sigma'^2) U'^T \Delta(q(t)) \right\|_2 \\ &= \sqrt{\sum_{i=1}^{3k} (1 - \gamma \sigma_i'^2)^2}\end{aligned}$$

Let  $E = \|\Delta(q(t+1))_{ls}\|_2^2 - \|\Delta(q(t+1))_g\|_2^2$  be the difference between the two methods, if  $E < 0$ , then the least square method has a smaller error than the gradient method and if  $E > 0$ , then it is the contrary. Note that the assumption that  $MN$  is invertible does not change the value of  $E$ . Indeed  $\Sigma'$  and  $Id^*$  have the same rank so the same zeros on the diagonal, the corresponding coordinates in  $\Delta(q(t+1))_{ls}$  and  $\Delta(q(t+1))_g$  are thus equal.

$$\begin{aligned}E &= 3k \left( 1 - \gamma \frac{\sqrt{\sum_{i=1}^{3k} \sigma_i'^2}}{\sqrt{\sum_{i=1}^{3k} \frac{1}{\sigma_i'^2}}} \right)^2 - \sum_{i=1}^{3k} (1 - \gamma \sigma_i'^2)^2 \\ &= 3k - 6k\gamma \frac{\sqrt{\sum_{i=1}^{3k} \sigma_i'^2}}{\sqrt{\sum_{i=1}^{3k} \frac{1}{\sigma_i'^2}}} + 3k\gamma^2 \frac{\sum_{i=1}^{3k} \sigma_i'^2}{\sum_{i=1}^{3k} \frac{1}{\sigma_i'^2}} - 3k + 2\gamma \sum_{i=1}^{3k} \sigma_i'^2 - \gamma^2 \sum_{i=1}^{3k} \sigma_i'^4 \quad (3.26) \\ &= \gamma \left( -6k \frac{\sqrt{\sum_{i=1}^{3k} \sigma_i'^2}}{\sqrt{\sum_{i=1}^{3k} \frac{1}{\sigma_i'^2}}} + 2 \sum_{i=1}^{3k} \sigma_i'^2 + \gamma \left( 3k \frac{\sum_{i=1}^{3k} \sigma_i'^2}{\sum_{i=1}^{3k} \frac{1}{\sigma_i'^2}} - \sum_{i=1}^{3k} \sigma_i'^4 \right) \right) \\ &= \gamma (A + B\gamma)\end{aligned}$$

We first study the sign of  $A = -6k \frac{\sqrt{\sum_{i=1}^{3k} \sigma_i'^2}}{\sqrt{\sum_{i=1}^{3k} \frac{1}{\sigma_i'^2}}} + 2 \sum_{i=1}^{3k} \sigma_i'^2$

$$\begin{aligned}A &= 2 \frac{\sqrt{\sum_{i=1}^{3k} \sigma_i'^2}}{\sqrt{\sum_{i=1}^{3k} \frac{1}{\sigma_i'^2}}} \left( -3k + \sqrt{\sum_{i=1}^{3k} \sigma_i'^2 \sum_{j=1}^{3k} \frac{1}{\sigma_j'^2}} \right) \\ &= 2 \frac{\sqrt{\sum_{i=1}^{3k} \sigma_i'^2}}{\sqrt{\sum_{i=1}^{3k} \frac{1}{\sigma_i'^2}}} \left( -3k + \sqrt{\sum_{i=1}^{3k} \sum_{j=1}^{3k} \frac{\sigma_i'^2}{\sigma_j'^2}} \right) \quad (3.27) \\ &= 2 \frac{\sqrt{\sum_{i=1}^{3k} \sigma_i'^2}}{\sqrt{\sum_{i=1}^{3k} \frac{1}{\sigma_i'^2}}} \left( -3k + \sqrt{3k + \sum_{i=1}^{3k} \sum_{j=i+1}^{3k} \frac{\sigma_i'^2}{\sigma_j'^2} + \frac{\sigma_j'^2}{\sigma_i'^2}} \right)\end{aligned}$$

Remark that  $\frac{\sigma_i'^2}{\sigma_i'^2} > 0$  and  $x + \frac{1}{x} \geq 2$  when  $x > 0$ , therefore,

$$\begin{aligned}
A &\geq 2 \frac{\sqrt{\sum_{i=1}^{3k} \sigma_i'^2}}{\sqrt{\sum_{i=1}^{3k} \frac{1}{\sigma_i'^2}}} \left( -3k + \sqrt{3k + \sum_{i=1}^{3k} \sum_{j=i+1}^{3k} 2} \right) \\
&\geq 2 \frac{\sqrt{\sum_{i=1}^{3k} \sigma_i'^2}}{\sqrt{\sum_{i=1}^{3k} \frac{1}{\sigma_i'^2}}} \left( -3k + \sqrt{3k + \sum_{i=1}^{3k} (3k - i)2} \right) \\
&\geq 2 \frac{\sqrt{\sum_{i=1}^{3k} \sigma_i'^2}}{\sqrt{\sum_{i=1}^{3k} \frac{1}{\sigma_i'^2}}} \left( -3k + \sqrt{3k + 3k(3k - 1)} \right) \\
&\geq 2 \frac{\sqrt{\sum_{i=1}^{3k} \sigma_i'^2}}{\sqrt{\sum_{i=1}^{3k} \frac{1}{\sigma_i'^2}}} (-3k + 3k) \\
&\geq 0
\end{aligned} \tag{3.28}$$

Now we study the sign of  $B = 3k \frac{\sum_{i=1}^{3k} \sigma_i'^2}{\sum_{i=1}^{3k} \frac{1}{\sigma_i'^2}} - \sum_{i=1}^{3k} \sigma_i'^4$

$$\begin{aligned}
B &= \frac{1}{\sum_{j=1}^{3k} \frac{1}{\sigma_j'^2}} \sum_{i=1}^{3k} \left( \sum_{j=1}^{3k} \sigma_j'^2 - \sum_{j=1}^{3k} \frac{\sigma_i'^4}{\sigma_j'^2} \right) \\
&= \frac{1}{\sum_{j=1}^{3k} \frac{1}{\sigma_j'^2}} \sum_{i=1}^{3k} \left( \sum_{j=1}^{3k} \frac{\sigma_j'^4 - \sigma_i'^4}{\sigma_j'^2} \right) \\
&= \frac{1}{\sum_{j=1}^{3k} \frac{1}{\sigma_j'^2}} \sum_{i=1}^{3k} \left( \sum_{j=1}^{3k} \frac{(\sigma_j'^2 - \sigma_i'^2)(\sigma_j'^2 + \sigma_i'^2)}{\sigma_j'^2} \right) \\
&= \frac{1}{\sum_{j=1}^{3k} \frac{1}{\sigma_j'^2}} \sum_{i=1}^{3k} \left( \sum_{j=i+1}^{3k} (\sigma_i'^2 + \sigma_j'^2) \left( 1 - \frac{\sigma_i'^2}{\sigma_j'^2} + 1 - \frac{\sigma_j'^2}{\sigma_i'^2} \right) \right) \\
&= \frac{1}{\sum_{j=1}^{3k} \frac{1}{\sigma_j'^2}} \sum_{i=1}^{3k} \left( \sum_{j=i+1}^{3k} (\sigma_i'^2 + \sigma_j'^2) \left( 2 - \frac{\sigma_i'^2}{\sigma_j'^2} - \frac{\sigma_j'^2}{\sigma_i'^2} \right) \right) \\
&\leq 0
\end{aligned} \tag{3.29}$$

because  $\frac{\sigma_j'^2}{\sigma_i'^2} > 0$  and  $x + \frac{1}{x} \geq 2$  when  $x > 0$ .

Therefore  $E < 0$  implies  $A + B\gamma < 0$  and  $\gamma > -\frac{A}{B}$ . For a step size big enough, the least square methods has a smaller error, however for a small step size, the gradient method has a smaller error.

Note that if all singular values are equal,  $E = 0$  and the two methods lead to the same results.

However for the last step, when  $\|N(MN)^\dagger \Delta(q)\|_2 < \alpha$ , the least square method can reach exactly the goal whereas the gradient method needs to adapt its step size in order to get closer to the goal.

**Computational time** The projected gradient method uses the transpose matrix whereas the least square method uses the pseudo inverse. The transpose matrix computation (linear) is faster than the pseudo inverse (more than quadratic on each dimension). In addition, if the nullspace changes (Section 3.2.1), then the least square method needs to inverse a new matrix whereas the gradient method only needs to project the direction into the new nullspace. Thus, the computational time of a step is faster in the gradient method approach. However if the step size is not small enough, the least square method needs less steps and can reach the goal exactly.

### 3.2.3 Procedure overview and dataset

#### 3.2.3.1 Procedure overview

The procedure starts with the initial full atoms  $q_I$  conformation and a few atoms from the goal  $q_G$  conformation. Hydrogen bonds between WC pairs are preserved by guaranteeing that the perturbation belongs to the nullspace of the matrix of constraints Jacobian. A minimizing perturbation  $\delta u$  from  $q_0 = q_I$  to  $q_G$  is computed from (3.12) and scaled to  $\|\delta u\|_\infty = 0.01$  to obtain a sufficiently small step size that maintains hydrogen bonds. We perturb  $q_0$  to obtain  $q'_1 = q_0 + \delta q$ , and check for clashes. The following two cases can occur:

1. The new conformation  $q'_1$  does not have clashes. It is accepted into the sampling pool.
2. The new conformation  $q'_1$  has clashes. We do not accept  $q'_1$ , but instead we add a clash-avoiding constraints to the Jacobian. We repeat the procedure to compute a new conformation  $q''_1$ , and iterate a fixed number of times.

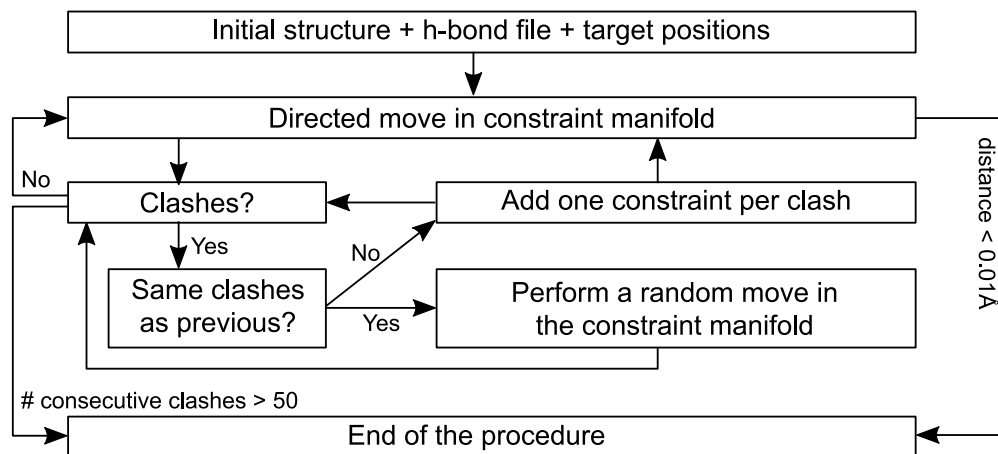


FIGURE 3.7: Flowchart of the morphing procedure.

If, in rare cases, the same atoms are still clashing, but no additional clashes occurred, the dCC is violated owing to our first order approximation, and the desired direction is sterically inaccessible. Therefore, we perform a random move in  $\mathbb{M}_{dCC}$  instead of moving towards marker atom goal positions. Note that clash constraints cannot be added indefinitely, since they gradually rigidify the molecule.

Finally, a new clash-free conformation  $q_1$  is accepted, the clash constraints are removed and a new optimal perturbation is computed starting from  $q_1$ . Unless otherwise stated, the procedure is repeated until the RMSD distance to the goal is less than  $0.5\text{\AA}$ , or  $0.01\text{\AA}$  when we omit clash avoidance. The van der Waals scaling parameter is fixed at  $c_f = 0.6$ , allowing some degree of overlap. Stereochemical constraints



are often relaxed in conformational sampling, balancing computational speed and structure quality [90].

### 3.2.3.2 Dataset

We selected a dataset of RNA molecules from the PDB with one chain and between 15 to 200 nucleotides in length (Table 3.1). All were solved by NMR spectroscopy, and all had more than one model with identical atoms. We used NMR because it gives a bundle of conformations for the same sequence of atoms, which allows us to test our method from one conformation to another. RNA molecules with nucleotides other than A, U, C, and G were excluded. We selected the first model as the initial conformation, and selected the model with the highest RMSD (with Biopython [91]) from the first model as the goal conformation. We limited our dataset to RNAs with an RMSD between initial and goal greater than 2Å. This resulted in a dataset of 78 multi-model molecules, with average heavy atom RMSD between the initial and goal conformations of 4.14Å (maximum RMSD 28.80Å, minimum RMSD 2.01Å).

## 3.3 Results

We applied three different protocols on the 78 RNAs dataset.

1. To compare our results with iMOD, a NMA-based morphing procedure [70] which does not explicitly avoid collisions, we first compute conformational transitions while ignoring clashes.
2. We evaluate the performance of the clash avoiding procedure using a hierarchical approach. We first selected a small subset of atoms as marker atoms to guide the initial conformation toward the goal conformation. We further optimized the conformational transition using all heavy atoms.
3. We used our procedure to understand to what degree bulges and higher-order junctions govern 3D helical arrangement.

Finally, we applied our procedure to the *Synechococcus elongatus* L-glutamine riboswitch, which undergoes a large transition between its free and ligand binding conformations [92].

### 3.3.1 Benchmark results

#### 3.3.1.1 Comparison to iMOD

The iMOD morphing tool uses torsion angles and NMA to iteratively minimize the distance between all pairs of heavy atoms in the initial and goal conformations. We used the iMOD default parameters setting, except that we used 90% instead of 10% of modes, since that generally led to lower RMSD to the goal conformation. Conformational transitions computed with iMOD are not restricted to the secondary structure constraint manifold. Instead, all non-bonded atoms are interconnected by harmonic springs within a 10Å radius [70]. iMOD does not check for clashes, but its ENM limits the introduction of new clashes for small amplitude motions. To perform a direct comparison with iMOD we disabled clash avoidance in KGS, while instructing it to minimize the distance between all pairs of heavy atoms. We executed KGS and iMOD transitions on each of the 78 RNA molecules. We terminated KGS after 1,000 iterations or when the heavy-atom RMSD was under 0.01Å.

iMOD finished in a few seconds which is faster than KGS which finished in average on 34 minutes. However, KGS dramatically reduced the RMSD to the goal

PDB ID	# of residue	# of Hbond	DoF	Initial RMSD (Å)	No dCC RMSD (Å)	iMOD RMSD (Å)	Stage1 RMSD (Å)	Stage2 RMSD (Å)	Full RMSD (Å)	hC5 RMSD (Å)	PDB ID	# of residue	# of Hbond	DoF	Initial RMSD (Å)	No dCC RMSD (Å)	iMOD RMSD (Å)	Stage1 RMSD (Å)	Stage2 RMSD (Å)	Full RMSD (Å)	hC5 RMSD (Å)
2NIQ	155	112	1153	3.70	0.27	0.68	1.37	1.32	2.23	1.26	1YNG	31	20	226	4.20	0.60	2.09	2.76	1.61	1.72	NC
2LKR	111	66	812	5.72	0.96	3.17	4.95	4.93	5.55	5.55	21DT	31	25	231	8.15	0.97	1.57	4.74	4.71	5.05	5.76
1S9S	101	84	757	28.80	0.73	1.04	21.20	20.17	27.14	26.24	1NBR	29	26	217	2.35	0.82	1.45	2.27	1.30	1.35	NC
1PSO	77	50	568	3.69	0.26	1.21	1.40	1.03	3.04	1.10	1SCL	29	20	214	2.81	0.35	0.62	1.26	0.50	0.50	1.12
2MQT	68	54	509	2.37	0.33	1.19	1.94	1.54	1.13	NC	2JWV	29	19	212	2.67	0.25	0.43	0.75	0.50	0.50	0.78
2N3Q	62	59	470	7.50	0.36	0.75	1.90	0.50	0.50	1.77	2K5Z	29	21	214	2.12	0.24	1.34	3.43	0.99	1.35	NC
1PSM	55	44	410	4.22	0.28	1.05	1.68	0.84	2.24	1.63	2M24	29	20	212	3.78	0.49	1.00	1.91	0.50	1.19	NC
2KZL	55	34	404	2.05	0.27	0.82	1.44	0.89	0.80	NC	28SP	28	20	207	2.15	0.54	1.26	2.05	0.99	1.11	1.30
2L00	49	39	364	8.02	0.49	1.06	2.64	1.61	0.82	5.84	2LUN	28	25	210	5.63	0.63	0.66	2.81	0.50	1.50	0.74
2KE6	48	40	356	2.46	0.27	0.60	1.26	0.50	1.20	NC	21XY	27	28	205	3.01	0.72	1.15	3.90	2.42	1.25	NC
2KUU	48	40	356	3.22	0.32	0.63	2.02	1.73	1.53	2.12	21JJ	27	19	200	2.58	0.50	1.02	1.77	0.92	0.90	NC
2KUV	48	44	359	3.52	0.33	1.01	2.51	1.77	2.40	NC	1QC8	25	18	184	6.97	1.56	1.27	4.91	1.85	2.08	4.00
2KUUW	48	35	354	2.98	0.24	0.47	1.33	0.50	0.92	0.71	1NC0	24	22	180	2.74	0.64	1.67	1.84	1.63	1.52	NC
1YMO	47	39	351	2.07	0.39	0.75	0.94	0.68	1.00	NC	1SYZ	24	21	179	2.15	0.37	1.46	2.10	1.65	1.22	NC
2MTJ	47	39	352	7.80	0.55	0.69	2.96	0.50	2.18	5.78	21V0	24	18	177	2.14	0.44	0.92	1.63	1.13	1.16	NC
1Z2J	45	50	344	2.68	0.22	0.72	1.57	1.30	1.11	0.81	1S34	23	19	171	3.13	0.43	1.09	3.04	2.03	2.12	NC
21ZJ	42	40	317	2.33	0.25	0.53	0.72	0.50	0.50	NC	11LR	23	17	170	3.17	0.27	0.97	2.94	1.39	1.14	NC
1ZC5	41	40	310	2.75	0.28	0.57	1.17	0.50	0.50	0.71	2M12	23	14	168	3.98	0.55	2.19	3.97	1.90	1.59	NC
2HUA	40	35	301	2.46	0.25	0.59	0.86	0.50	0.50	0.62	2N20	23	14	168	7.28	1.02	2.20	5.90	0.60	2.46	NC
2MXL	39	28	288	4.94	0.50	1.59	2.66	0.59	3.25	2.60	2N2P	23	14	168	9.75	1.68	2.40	6.26	1.28	2.70	NC
1DXS	38	33	285	2.01	0.48	1.05	1.34	0.70	0.56	NC	3PHP	23	21	172	4.19	0.35	0.70	4.37	0.50	1.04	2.14
2KHY	38	21	277	2.32	0.34	0.81	1.91	0.50	0.50	NC	1OSW	22	17	140	3.27	0.50	1.51	2.88	0.73	0.90	2.93
2LHP	37	38	281	4.29	0.59	1.08	3.06	1.87	3.68	1.69	1PHY	22	25	167	2.68	0.91	1.15	1.78	1.10	1.10	1.25
2LUB	37	39	282	2.21	0.67	1.24	1.33	1.27	1.60	1.23	11JZ	22	18	164	6.12	0.89	0.96	5.82	1.28	2.38	2.53
2LSE	35	31	262	2.85	0.49	1.27	1.80	0.98	2.06	1.94	2HNS	22	22	166	2.41	0.26	0.46	1.52	0.50	0.50	0.81
2M57	35	26	259	5.30	0.56	1.71	3.01	0.71	4.30	3.03	17RA	21	17	156	3.97	0.20	0.49	1.89	0.50	0.50	0.80
2PCV	35	15	252	10.57	0.59	1.39	3.26	1.45	3.42	3.13	2RPT	20	21	152	2.58	0.34	0.51	1.06	0.50	0.50	0.80
1PSN	34	20	248	2.68	0.35	1.39	2.00	1.20	0.92	NC	1SLP	19	17	141	2.13	0.60	0.68	2.20	0.60	0.77	NC
1R2P	34	27	252	4.80	0.61	0.87	3.12	1.12	2.19	1.54	1UUU	19	16	141	2.34	0.34	0.88	2.40	0.50	0.50	NC
1R7W	34	33	255	6.75	0.76	1.58	8.54	7.70	5.49	3.63	2RUU	19	19	143	2.20	0.37	0.62	2.16	0.54	1.65	NC
1R7Z	34	28	252	3.68	0.46	0.91	2.40	0.50	0.53	2.33	2CVO	18	10	130	2.02	0.37	0.81	1.35	0.50	0.50	NC
2EUY	34	26	252	5.52	0.55	1.05	2.08	1.15	3.06	2.99	1WKS	17	15	126	2.40	0.41	0.66	2.94	0.93	0.50	NC
2JTP	34	34	257	2.50	0.45	0.75	1.47	0.51	0.51	1.02	1YN1	17	16	127	2.02	0.32	0.45	1.41	0.50	0.50	NC
2KPV	34	34	257	2.17	0.69	0.88	3.41	1.16	1.24	NC	2KPC	17	13	125	2.52	0.44	0.72	2.54	1.39	0.77	NC
2LPT	34	32	256	2.12	0.33	0.76	1.78	0.50	0.50	NC	2KRP	17	14	125	3.82	0.32	0.70	1.95	0.50	0.50	1.64
2L14	32	33	240	6.04	0.30	0.47	5.41	3.42	0.50	1.30	2KVN	17	13	125	2.98	0.27	0.55	1.37	0.50	0.50	NC
1YLG	31	23	228	4.61	0.78	1.23	2.69	2.00	2.62	3.65	2MNC	16	11	115	4.73	0.82	1.07	5.16	0.72	0.72	2.65
1YNC	31	23	228	4.30	0.58	1.83	3.41	1.47	2.00	NC	MEAN	36.01	28.90	267.46	4.14	0.51	1.06	2.83	1.47	1.90	2.77
1YNE	31	23	228	4.21	0.77	2.11	3.53	3.28	2.56	NC	MEDIAN	31.00	24.00	228.00	3.07	0.45	0.98	2.13	0.95	1.17	1.69

TABLE 3.1: Dataset details and results summary. For each of the 78 single-strand RNA molecules solved by NMR the sequence length, number of hydrogen bonds (Hbond), number of degrees of freedom (DoF) and heavy atom RMSD between initial and goal conformations are reported. The "No dCC", "iMOD", "Stage1", "Stage2", "Full" and "hC5" columns represent final heavy atom RMSD distances based on all heavy atom information (without clash-avoidance), iMOD, the first stage of sparse information (with clash-avoidance), the second stage of sparse information (with clash-avoidance), the all heavy atom experiment with clash-avoidance, and experiments base on C5 atoms in WC base pairs respectively.

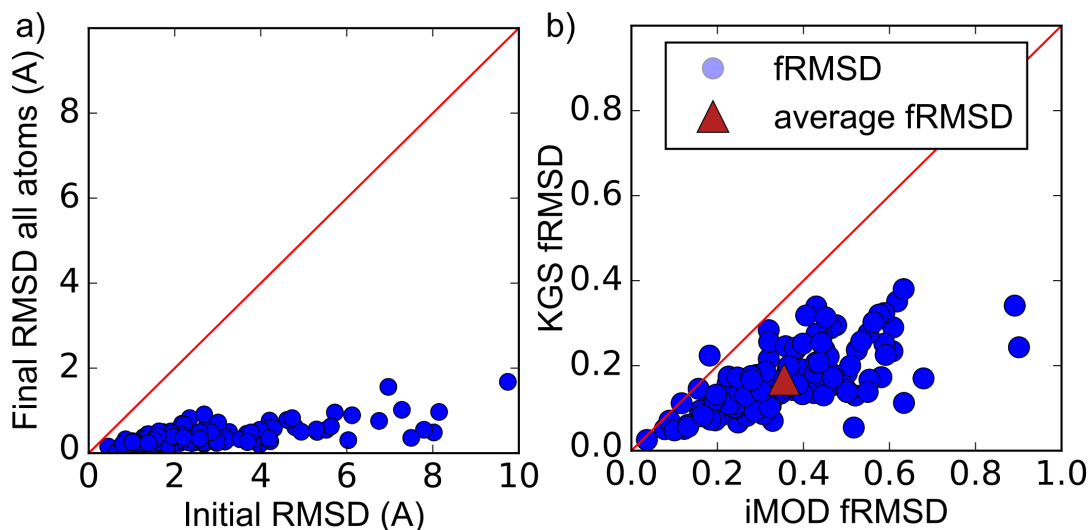


FIGURE 3.8: Conformational transitions on the secondary structure constraint manifold, using all heavy atoms as markers but ignoring clashes. a) The initial RMSD in Å ( $x$ -axis) versus the final RMSD ( $y$ -axis) to the goal conformation obtained by KGS. Data points below the red diagonal indicate improvement towards the goal conformation. b) Comparison of the performance of iMOD and KGS. The  $x$ -axis shows the fractional RMSD (fRMSD: the RMSD between the final conformation and the goal conformation divided by the RMSD between initial and goal conformations) obtained by iMOD. The  $y$ -axis shows the corresponding ratio for KGS. Data points below the red diagonal indicate a lower fRMSD and superior performance for KGS than for iMOD. The red triangle indicates the average fRMSD value over all 78 data points.

conformation. On average by 86%, from 4.14Å RMSD to 0.51Å, compared to 69% for iMOD (Figure 3.8). That represents an average improvement of 0.55Å RMSD of KGS over iMOD. Strikingly, for nearly every RNA molecule, 77 out of 78 in the benchmark dataset, KGS approached the goal conformation closer in heavy atom RMSD than iMOD. In one out of 78 cases iMOD approached the target conformation 0.29Å closer than KGS (PDB ID 1QC8). Although, KGS had not converged within 1,000 steps, and did get closer than iMOD in 2,000 steps.

In many cases, KGS performed significantly better than iMOD. For example, the initial RMSD for the highly dynamic 23-nucleotide ID3 stem loop of domain 1 of the ai5 $\gamma$  group II intron (PDB ID 2M12), is 3.98Å. KGS came within 0.55Å of the goal conformation, 1.64Å closer than iMOD. In particular, iMOD struggled to fit the large amplitude motions of hairpin loop bases that do not participate in WC interaction.

### 3.3.1.2 Performances using every fifth C5 atom

As helical structure is mostly preserved between RNA substates, we expected that the dimensionality reduction encoded in our secondary structure constraint manifold would be effective in guiding large fragments of the molecule to their goal conformation. We exploited this insight to avoid over-constraining the system, adopting a hierarchical approach for clash-avoiding transitions. We selected the C5 atom of every fifth nucleotide to guide the initial to the goal conformation. The C5 atoms therefore emulate a sparse experimental data set—fewer than 1% of the total number of atoms. Then, starting from the best conformation at this stage, in a second stage

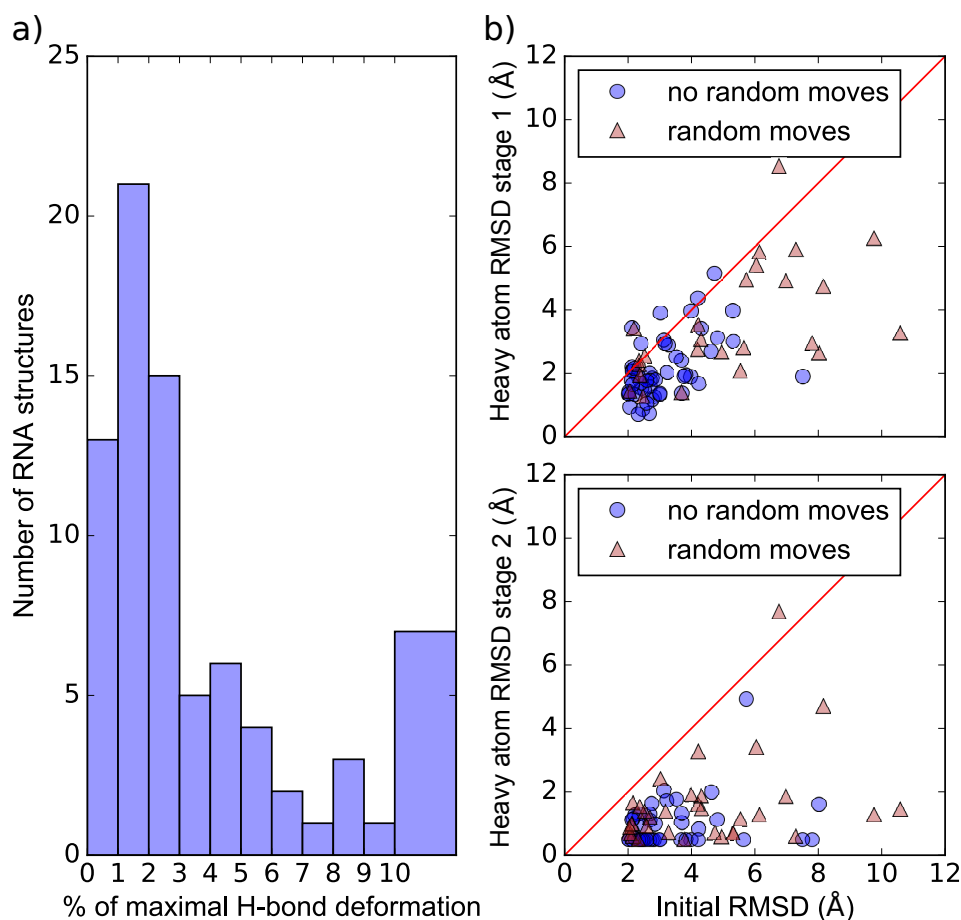


FIGURE 3.9: Hydrogen bond deformation and heavy atom RMSD minimization in KGS clash-free conformational transitions. (a) Distribution of largest acceptor-to-hydrogen distance deformation for WC-constraints in the benchmark set. (b) Top, heavy atom RMSD improvement for morphing with clash-avoidance and one C5 marker atom every five residues (first stage). Triangles indicate RNAs where morphing required random moves. Bottom, heavy atom RMSD improvement after switching to all heavy atoms (second stage). PDB ID 1S9S is omitted. It has an initial RMSD of  $28.80\text{\AA}$ , its first stage RMSD is  $21.20\text{\AA}$  and its second stage RMSD is  $20.17\text{\AA}$ .

we selected all heavy atoms as markers to further reduce the distance to the goal conformation.

We first evaluated the RMSD between the final marker atom positions and their corresponding goal positions. Out of the 78 RNA structures in our data set, 70 (90%) reached the goal positions of marker atoms extremely close, to within  $0.5\text{\AA}$  (Table 3.1). For the remaining eight RNA molecules that did not reach their goal positions to within  $0.5\text{\AA}$ , clashes could not be fully resolved, resulting in an excess of random moves.

### 3.3.1.3 Maintaining secondary structure between substates

We then verified that KGS properly maintained helical structure between clash-avoiding conformational transitions. Our mathematical model maintains distance constraints exactly by calculating admissible velocities in the constraint manifold

(Subsection 3.2.2). However, the finite step size of the perturbation respects distance constraints only to first order approximation. We therefore monitored deformation of the hydrogen bond distances in conformational transitions. In 91% of the RNA molecules the maximal hydrogen bond length deformation was below 10% (Figure 3.9a). These values are well within the range of expected fluctuations, and suggest that our conformational transitions stay close to the secondary structure constraint manifold. This is important as breaking WC base pairs is energetically unfavorable, making it less likely to occur in a trajectory.

An analysis with *Defining the Secondary Structure of RNA* (DSSR) [93] confirmed that A-form helices were maintained along the trajectory. Out of 1275 WC base pairs among the initial conformations, 486 were classified as A-form, 6 as B-form, and 783 could not be classified (Table 3.2). For the KGS final conformation, 440 were A-form, 7 B-form, and 840 could not be classified. Just over 9% of WC pairs could no longer be classified at A-form, despite maintaining their constraints. Two pairs picked up A-form classification. We did not observe A-form to B-form or reverse transitions.

#### 3.3.1.4 Distance to the heavy atom goal positions

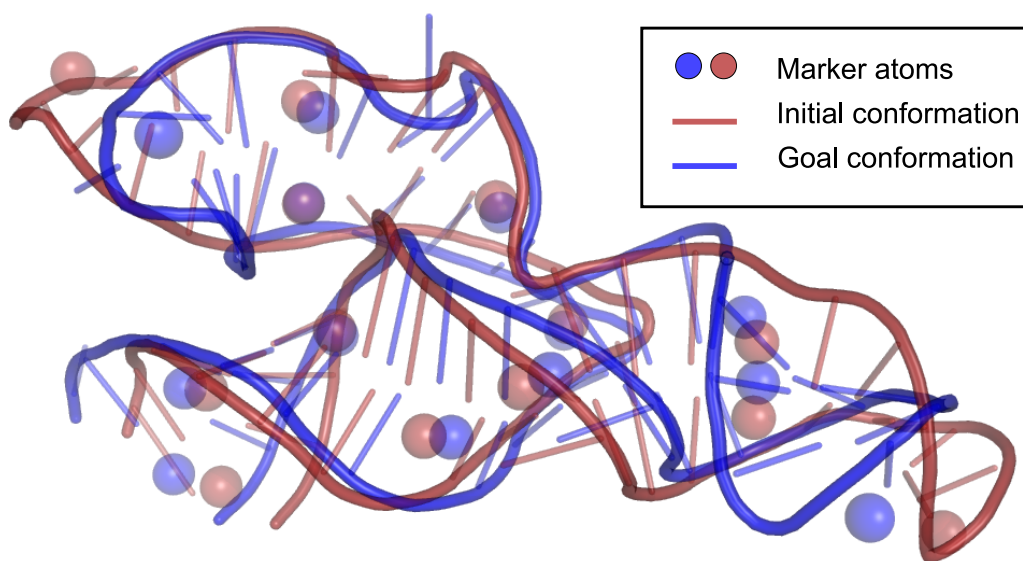


FIGURE 3.10: Initial and goal conformations of a 62-nucleotide II-III-VI three-way junction (PDB ID 2N3Q) with one C5 marker atoms selected every five nucleotides. A clash-free morphing movie can be found at <https://youtu.be/VwAXm-UaxJg>. The heavy atom RMSD between the initial (tan) and the goal (blue) conformation is 7.50Å. The heavy atom RMSD between the final and goal conformation is 1.90Å.

To understand to what extent RNA secondary structure guides their all-atom 3D structure, we calculated the heavy atom RMSD between the best conformation determined by the marker goal positions and the (true) goal conformation. Figure 3.9b shows that KGS morphing significantly reduced the heavy atom RMSD to the goal conformation. Strikingly, using fewer than one percent of atoms as markers, the average heavy atom RMSD from the best KGS conformation to the goal conformation was 2.83Å, compared to 4.14Å between the initial and goal conformations. 66% of

	initial	goal	KGS final	# of WC bp
	o	o	o	576
	A	A	A	370
	x	x	x	69
	o	A	o	62
	A	o	o	49
	A	o	A	39
	A	A	o	27
	o	A	A	15
	o	o	A	12
	o	x	x	9
	o	-	o	6
	x	o	o	6
	-	x	x	5
	o	x	o	4
	o	o	x	3
	o	o	B	3
	x	x	o	3
	x	A	x	2
	x	A	A	2
	x	o	x	2
	B	o	B	2
	B	B	B	2
	-	o	o	2
	B	o	o	2
	o	o	-	2
	o	-	-	2
	x	-	x	2
	o	A	x	1
	-	o	-	1
	-	A	-	1
	x	o	A	1
	A	A	-	1
	x	x	-	1
	-	A	A	1
	-	x	-	1
A	486	482	440	
B	6	2	7	
Z	0	0	0	
x	88	92	93	
o	695	700	737	
-	11	10	9	
# of WC	1275	1276	1277	

TABLE 3.2: WC base pairs classification by DSSR [93]. "A", "B" and "Z" stand for the three helical structures, "o" stands for unclassified base pairs, "x" stands for a backbone gap and "-" means that the base pair is not a WC base pair in that conformation. The upper part of the table shows the number of WC base pairs according to their classification in the initial, goal and the KGS final conformations. The bottom part of the table represents the number of WC base pairs of each type.

the benchmark set improved the heavy atom RMSD by at least 20%. In some cases, the improvement is dramatic. For example, the heavy atom RMSD between the initial (tan) and goal (blue) conformation for the 62-nucleotide VS ribozyme II-III-VI

three-way junction is reduced from 7.50Å, to 1.90Å (Figure 3.10). The final marker atoms RMSD is less than 0.01Å.

We then selected all heavy atoms as marker atoms to further refine the distance to the goal conformation. We used an RMSD threshold of 0.5Å. The final, average RMSD to the goal conformation was reduced to 1.47Å (from 4.14Å, fRMSD = 0.34). Figure 3.9b bottom panel, and Table 3.1 summarize the results. By contrast, computing the conformational transitions using heavy atoms as markers directly resulted in a final RMSD of 1.90Å, corresponding to an fRMSD of 0.41. With the hierarchical approach, 64% of the benchmark set improved the heavy atom RMSD by at least 60% compared to 53% using heavy atoms directly.

Our results suggest that the constraint manifold efficiently reduces dimensionality of conformation space, while retaining key structural information that is largely accessible through a sparse set of marker atoms. A hierarchical approach for conformational morphing capitalizing on this insight is more efficient than a direct approach.

### 3.3.1.5 Study of 3D helical arrangements

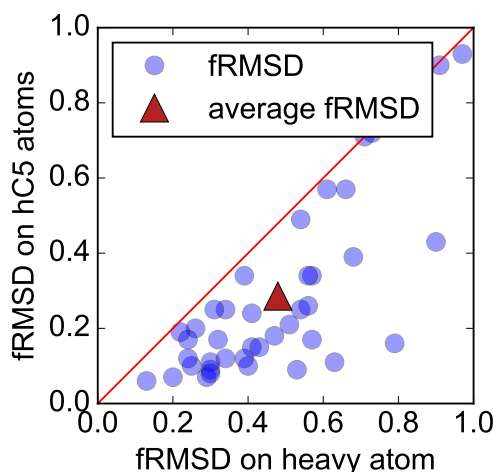


FIGURE 3.11: Conformational transitions using all hC5 (C5 atoms in WC pairs) atoms as markers with clash-avoidance. Heavy atom fRMSD ( $x$ -axis) versus hC5 fRMSD ( $y$ -axis) suggest heavy atom fRMSD is only weakly correlated to 3D helical arrangement. The red triangle indicates the average fRMSD value over all 41 data points.

Next, to probe the role of bulges and higher-order junctions in 3D helical arrangement, we used only 'helical' C5 (hC5) atoms as markers, i.e., all C5 atoms in WC base pairs. The hC5 initial RMSD reports on differences in helical arrangement between conformations of our benchmark set. We selected a subset of 41 structures from our benchmark set for which helices of the initial and goal conformations were separated by at least 2Å. The average hC5 RMSD is 4.28Å. Clash-avoiding morphing reduced the hC5 RMSD to, on average, 1.69Å, corresponding to an fRMSD of 0.29. Note that these clash-free motions take place on the secondary structure constraint manifold  $\mathbb{M}_{dCC}$ , which is precisely the accessible conformation space when helical structures are preserved. The small values of helical C5 fRMSD therefore suggest

that single-stranded motifs play a significant role in modulating 3D helical arrangements through intra-junction, coordinated motions on  $M_{dCC}$ .

However, hC5 helical arrangement leaves a substantial fraction of heavy atom fRMSD unaccounted for. To examine the flexibility of the single-stranded motifs, we also calculated the overall heavy atom RMSD to the goal conformation. We found that, using only the hC5 atom markers to rearrange helices, in half of the cases in the benchmark set the overall heavy atom RMSD was reduced by at least 57%. This suggests that conformational variability of intra-junction regions remained elevated for a substantial fraction of the benchmark set, and that their conformations only weakly correlate with helical arrangements (Figure 3.11). Junctions often bind ligands and are responsible for most function in RNA. While they modulate 3D helical arrangements, these regions must remain flexible to exchange between conformational substates and interact with multiple partners.

### 3.3.2 Conformational exchange of riboswitch *glnA*

Riboswitches are non-coding domains of messenger RNA, often located in untranslated regions, that bind to a partner and regulate transcription and/or translation [94]. The L-glutamine-binding riboswitch *glnA* is a three-stem junction, with helices P1, P2, and P3. L-glutamine-binding riboswitches are implicated in nitrogen metabolism [95], and found exclusively in cyanobacteria and marine metagenomic sequences. In a 61 nucleotides construct of the *Synechococcus elongatus glnA* sensing domain, binding of L-glutamine results in a large, 13.11Å conformational change from a 'tuning-fork'-shaped, free conformation to an 'L'-shaped bound conformation [92]. The bound form is stabilized by a long-distance base pair G23-C60. In the absence of L-glutamine, the riboswitch samples the free conformation and a minor conformation similar to the bound conformation. A molecular dynamic study [92] suggested, that the exchange between the free and ligand-bound conformations is accompanied by partial melting of the P1 helix, which may be required to access the bound conformation.

To test the significance of partial P1 melting for exchanging between substates, we computed clash-free conformational transitions with and without its two dissociating WC base pairs (C1-G59 and G2-C58) as constraints (Figure 3.12a). Once more, we took a hierarchical approach. First we selected six C5 atoms as markers, three each in the apical loops of helices P2 and P3 (Figure 3.12b). Using these marker atoms, the goal marker positions were reached to within 0.01Å with and without C1-G59 and G2-C58 as constraints. The heavy atom fRMSD was 0.52 when C1-G59 and G2-C58 were included as constraints, and 0.47 when they were not.

For the second stage, we selected all C5 atoms (one per nucleotide) as markers. When C1-G59 and G2-C58 were omitted as constraints, our procedure found a clash-free conformational transition to within 1.73Å heavy atom RMSD of the bound conformation. Bases G23 and C60 approach each other in this less-constrained transition, suggesting a base-pair could form. By contrast, when C1-G59 and G2-C58 were included, the RMSD remained elevated at 4.44Å. Interestingly, these constraints were among the most deformed during the transition, with strain exceeding 5%.

Our results suggest that partial melting of P1, proposed in a previous study [92], is required to adopt the ligand-bound state. While *glnA* samples a minor conformation similar to the bound conformation, binding of L-glutamine may lower the free-energy barrier to make the bound state accessible.



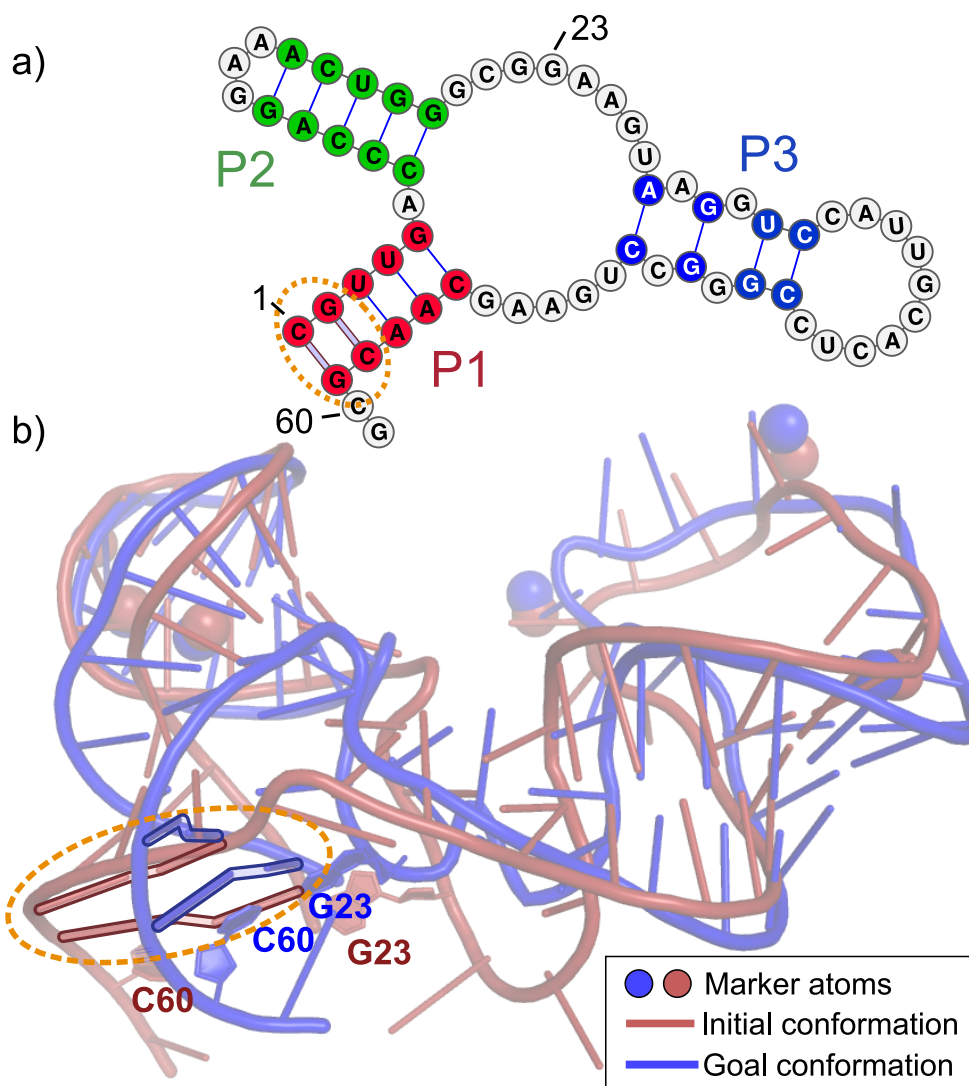


FIGURE 3.12: (a) Secondary structure representation of the L-glutamine riboswitch. Helix P1 is shown in red; the two WC pairs C1-G59 and G2-C58 involved in partial melting are encircled in orange. (b) Superposition of the 3D conformations of the L-glutamine riboswitch. The initial conformation (tan) is the unbound state while the goal conformation (blue) is the bound state. The marker atoms are shown in both conformations as spheres. An animation of the motion with partial melting of P1 is at <https://youtu.be/BhJW0DXLR4>.

### 3.3.3 Run time

The run time of our procedure depends on the size of the molecule (Figure 3.13, top row) and the number of constraints (Figure 3.13, bottom row), as these values determine the size of the Jacobian matrix. While the singular value decomposition of the Jacobian is highly optimized by parallel solvers *Intel's Math Kernel Library* (MKL), it has a significant computational complexity and often dominates run time.

Without dCC, the vast majority of transitions finish within seconds (Figure 3.13,

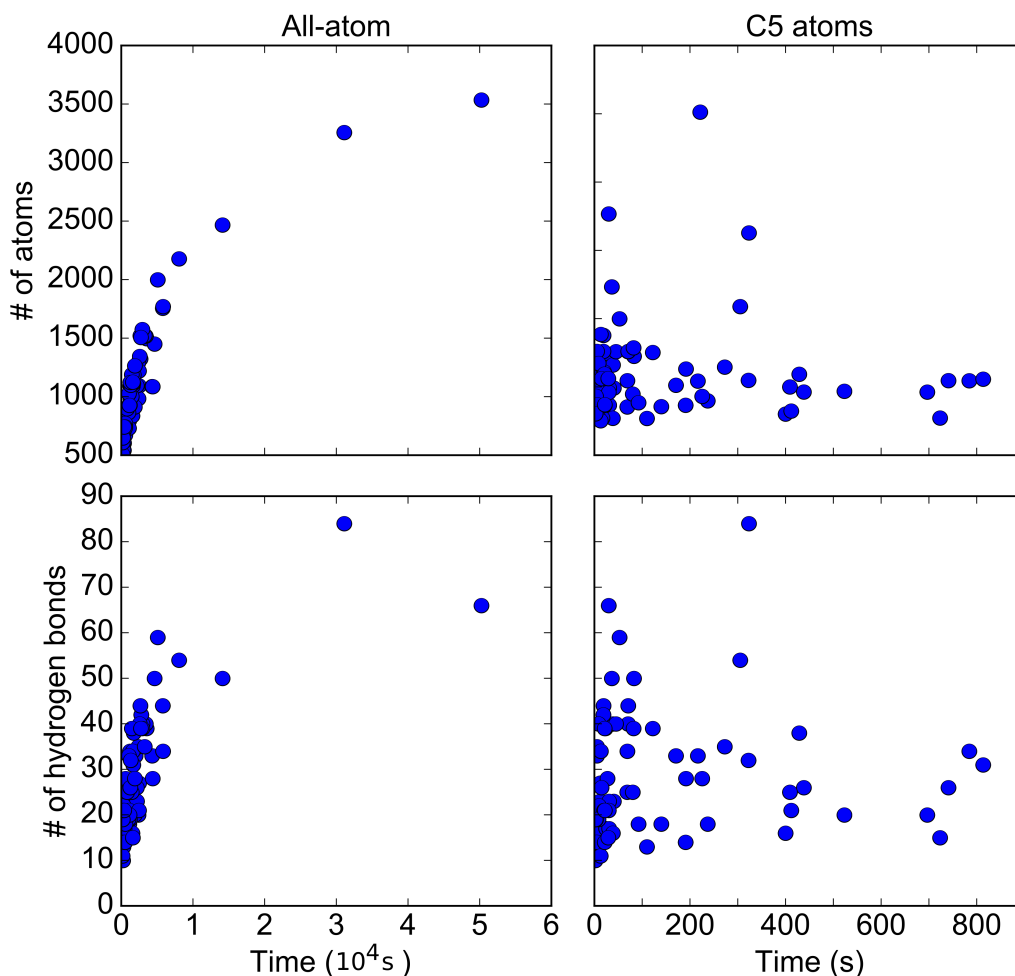


FIGURE 3.13: Run times of our procedure vs. the number of atoms (top) and the number of hydrogen bonds (bottom). Run times using all-atom goal conformations is shown on the left, and run times for every fifth C5 marker atom and dCC clash avoidance is shown on the right.

left column). Adding dCC clash-avoidance, but keeping the size of the Jacobian small by using a sparse set of marker atoms results in reasonable run times for all systems (Figure 3.13, right column). dCC clash-avoidance in combination with heavy atom marker atoms results in long run times, up to several hours for some RNA in our benchmark set.

### 3.4 Perspectives and conclusions

In this section we present a development that will soon be applied to experimental data. First, the presented directed motion uses absolute atoms position as goal. Here we present an other approach that uses distances between pair of atoms as goal. This approach allows more possibilities and can be applied to distance distributions obtained from DEER experiments.

### 3.4.1 Relative distances

Instead of having  $k$  marker atoms and their goal positions, we have  $k$  pair of atoms  $\{A_{1,1}, A_{1,2}, \dots, A_{k,1}, A_{k,2}\}$  and their goal distances  $\{d_1, \dots, d_k\}$ . Therefore we do not have anymore a goal conformation. The initial conformation is perturbed to reach goal distances, and its secondary structure is preseved. The approach stays the same as in Section 3.2.2.

#### 3.4.1.1 Relative distances approach

The matrix  $M$  is Jacobian matrix of the vector  $(f_{A_{1,1}}(q), f_{A_{1,2}}(q), \dots, f_{A_{k,2}}(q))$ . Only the definition of  $\Delta(q)$  changes. The optimal displacement we want to obtain is such that

$$\begin{aligned} \overrightarrow{A_{i,1}A_{i,2}} + \Delta(q)_{A_{i,1}} \Delta(q)_{A_{i,2}} &= \overrightarrow{A_{i,1}A_{i,2}} \times \frac{d_i}{\|\overrightarrow{A_{i,1}A_{i,2}}\|} \\ \Delta(q)_{A_{i,1}} \Delta(q)_{A_{i,2}} &= \overrightarrow{A_{i,1}A_{i,2}} \times \frac{d_i - \|\overrightarrow{A_{i,1}A_{i,2}}\|}{\|\overrightarrow{A_{i,1}A_{i,2}}\|} \end{aligned} \quad (3.30)$$

with  $i \in [1, k]$  where  $\Delta(q)_{A_{i,1}}$  (resp.  $\Delta(q)_{A_{i,2}}$ ) is the desired movement of atom  $A_{i,1}$  (resp.  $A_{i,2}$ ), and  $d_i$  the goal distance between the two atoms of the couple  $i$ . This can be obtained with:

$$\begin{aligned} \Delta(q)_{A_{i,1}} &= \overrightarrow{A_{i,1}A_{i,2}} \times \frac{d_i - \|\overrightarrow{A_{i,1}A_{i,2}}\|}{2\|\overrightarrow{A_{i,1}A_{i,2}}\|} \\ \Delta(q)_{A_{i,2}} &= -\overrightarrow{A_{i,1}A_{i,2}} \times \frac{d_i - \|\overrightarrow{A_{i,1}A_{i,2}}\|}{2\|\overrightarrow{A_{i,1}A_{i,2}}\|} \end{aligned} \quad (3.31)$$

The rest of the computation is the same as in Section 3.2.2.

#### 3.4.1.2 Randomization

Just as the morphing method presented in Section 3.2.2, this approach lacks of randomization. The movement is deterministic except for the random moves when it is stuck in collisions. To circumvent this problem we can blend the directed perturbation with a random perturbation drawn randomly in the constraints manifold.

This randomization has two main benefits. First, it is more likely to bypass clashes. Second, multiple runs give different conformations observing the same distances. We can thus obtain an ensemble of result conformations.

#### 3.4.1.3 Preliminary results

We applied this method to simulate the opening motion of the alpha domain of GDP-bound G protein. We use the distance distribution of a previous DEER [96] experiment to generate an ensemble of conformations that could explain the distribution (Figure 3.14).

#### 3.4.1.4 Projected applications

We develop this approach in the idea of applying it to DEER data (see Section 1.2.1). The benefits can be twofold. An *a priori* analysis, testing multiple set of couple of

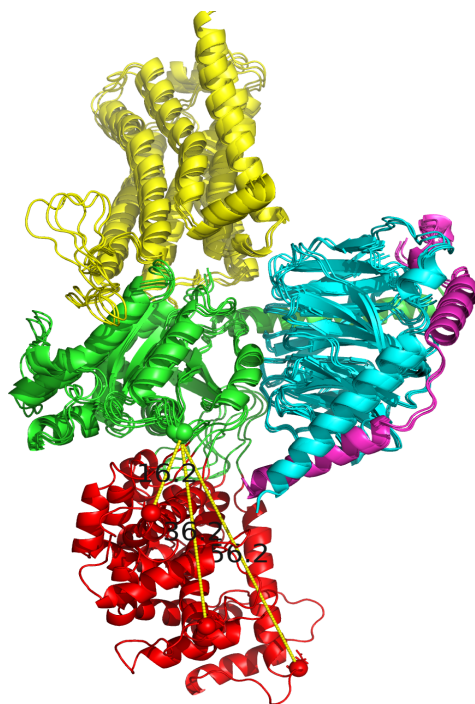


FIGURE 3.14: Three GDP-bound G protein conformation colored by domain. The conformations are from the generated ensemble of conformations of corresponding to the observed distance distribution [96] between residues 261 and 112 (yellow dashed lines above the red domain). The alpha domain in red moves away from the rest of the protein.

atoms with different range of distances will guide experimentalists in their choices for positioning labels. Indeed with such simulations, one can test which couple of atoms have distances that are likely to significantly change.

An *a posteriori* analysis, knowing exactly to which atoms the labels were attached and what are the distributions of distances will help understanding the correlation between distances. Indeed, DEER data give no insight on the simultaneity of distances, the distance distributions are not temporally linked. We do not know from this data what is the distance between two probes when two others were at a given distance. Our relative distance algorithm is able to determine which set of distance can lead to a feasible conformation and moreover the feasibility also depends on the initial conformation. That will lead to possible path observing all distance distributions.

### 3.4.2 Conclusions

We presented an efficient computational procedure to obtain clash-free structural transition pathways between conformational substates of RNA. Our procedure is mathematically rooted in preserving RNA secondary structure by calculating admissible perturbations in a lower-dimensional constraint manifold. The secondary structure constraint manifold helps guiding the initial conformation to the goal conformation from few marker atoms. Stable tertiary motifs, e.g., kissing loops or pseudo-knots, that further govern conformational flexibility are trivially accommodated as constraints in our framework. Despite the near deterministic nature of the move sets, our algorithm performed extremely well. Nonetheless, coupling to more

sophisticated motion planners to explore the conformational landscape [69], [97] will likely increase the performance, and result in a broader ensemble of transition pathways.

Why is it important to efficiently explore helical rearrangements? RNA receptors can accommodate a diverse set of small molecules binding to interhelical junctions [98], which often selectively stabilizes helical arrangements [99]. Our results suggest that helical rearrangement poorly correlates with junction conformation, indicating functional roles for junctions beyond governing tertiary structure. Mapping sequence identity and conformational variability of junctions to helical arrangements and their interconversions can help us understand the molecular mechanisms of RNA-ligand interactions.

Our algorithm is uniquely positioned to start addressing these important questions. In contrast to MD simulation, conformational sampling-based approaches can quickly interrogate the effect of junction sequence identities, conformations, mutations or insertions for thousands of structures. Our L-glutamine-binding riboswitch *glnA* example suggests that certain transitions may require Watson-Crick pairs to break. Likewise, Watson-Crick pairs could (transiently) form. Our approach currently provides two mechanisms to facilitate evolving constraints.

1. Watson-Crick pairs can be excluded or included explicitly in the input files for the software, and
2. In practice, we often find that many singular values of the constraint matrix are very small ( $10^{-4}$ ), but non-vanishing.

Selecting right-singular vectors corresponding to these small, non-vanishing singular values as additional basis vectors for the null-space allows for small violations of constraints. In fact, there is a direct correspondence between the magnitude of the singular value and the magnitude of the constraint violations it allows. The singular value cut-off is a user-defined parameter for the software. The *glnA* application suggests small constraint violations occur even when working in the true nullspace, owing to first order approximation.

Furthermore, dCC constraints could act like transient Watson-Crick pairs. While dCC constraints would unlikely result in fully formed Watson-Crick pairs, it is conceivable that a (one-dimensional) dCC forms between GC or AU pairs. Thus, monitoring constraint formations (dCC) or violations along a pathway could report on Watson-Crick pairs that form or break to facilitate conformational transitions.

Our procedure can have applications in helping to interpret sparse data, or exploring conformational landscapes of dynamic ensembles. Conformational transitions can be used as starting models for detailed, but expensive MD simulations. Adapting our procedure to rely on pair-wise distances between atoms instead of absolute distances will make it a valuable tool for interpreting, for example, DEER or FRET experiments.

## Chapter 4

# Finding alternate conformations of ligands

### Contents

<b>4.1</b>	<b>Introduction</b>	<b>61</b>
4.1.1	Fitting the density for proteins	61
4.1.2	Alternate conformation for ligands	62
<b>4.2</b>	<b>Method presentation</b>	<b>62</b>
4.2.1	Ligand sampling	62
4.2.2	Fitting the density	64
<b>4.3</b>	<b>Preliminary results and Discussions</b>	<b>65</b>
4.3.1	Results	65
4.3.2	Discussion	65

Most experimental methods for solving protein structures need a post-processing step to retrieve atoms positions. For example X-ray diffraction experimental methods give a density map which is a spatial and temporal average of the protein structure. Fitting methods are developed to retrieve atoms positions from the density map. However they often focus on finding one position per atom, disregarding the possibility of alternative positions. An ensemble of distinct conformations can often explain better the density rather than only one conformation [100]. This is also true for ligands in the protein binding pocket. While a few softwares attempt to find alternative conformations for proteins, to our knowledge nothing is done for ligands. We present here an ongoing project for finding ligand's alternative conformations based on qFit [101]. qFit finds alternative conformations for proteins, and we extend the workflow to ligand. In the first section, we present qFit for proteins and the new challenge for ligands. In the second section, we explain how we sample ligand conformations and give quick overview of fitting the density.<sup>1</sup> Then in the last section we present our first results and discuss limitations.

## 4.1 Introduction

### 4.1.1 Fitting the density for proteins

For proteins, alternative side-chains are easy to spot in the density map but they often involve backbone deviations which are more subtle. qFit not only samples side chains but also backbone conformations. The algorithm starts without side chains (only the  $C_\beta$  atom, that is the first atom of the side chain). For each AA, six trial positions for the  $C_\alpha$  atoms (atom in the backbone) are chosen and torsional backbone

<sup>1</sup>My contribution to this project is only the ligand sampling. I present the rest of the method as a context.

angles are adjusted to reach those positions. Then starting from a rotamer<sup>2</sup> each torsional angle of the side chain is sampled one by one. After each sampling step, a *Multi-integer Quadratic Programming* (MIQP) algorithm selects the optimal subset of a few side chain conformations to fit the density.

The last step consists on building the protein from the different possible conformations of all AAs. The MIQP algorithm is run on each subset of a few consecutive residues to select collective motions. Then subsets of consecutive residues are gathered into a few alternative protein's conformations by a Monte-Carlo algorithm.

qFit showed impressive results in automatically finding alternative conformations. However it does not deal yet with ligands.

### 4.1.2 Alternate conformation for ligands

Proteins often bind ligands that are crystalized with them. Finding alternative conformations for ligands is crucial because it will give a better insight on the binding interaction and stabilization of protein structure [102]. Unlike AAs this is no rotamers library to gather possible conformations of the ligand, we need to sample the ligand conformations.

However ligand sampling rises new problems. First of all, alternative conformation of ligands are often colliding with water. Water is usually added to fill unexplained density. Therefore density corresponding to alternative ligand conformation might be filled with water.

In addition, ligands are a bit more complex to sample because there are not always obvious ways to build ligands and sample them. For AAs, the side chain is build hierarchically from the backbone, but ligands can have forks that make this order less natural. Also ligands do not have obvious anchor point, like  $C_\beta$  for side chain. Therefore we need to perform a larger three-dimensional search for global rotations and translations.

## 4.2 Method presentation

We present the ligand sampling method and an overview of the fitting step. The overall workflow iteratively uses both methods as illustrated in Figure 4.2.

### 4.2.1 Ligand sampling

A ligand can have many degrees of freedom. We consider bonds that are not part of a cycle as rotatable bonds and so, degrees of freedom.

The ligand is constructed hierarchically starting from a *root* which is chosen at an extremity. First we try global rotations and translations to find the best position for the ligand, in term of density fitting and collision with the protein receptor. A collision occurs when the center of two atoms are too close. Because of the combinatorial explosion of samples, we cannot sample all degrees of freedom at the same time. However, it is important to sample at least two degrees of freedom together to allow coordinated moves.

Therefore we sample degrees of freedom two by two with an overlap such that each non-extremal degree of freedom is sampled twice: once with the degree of freedom before and once with the degree of freedom after. This setup allows for more flexibility. Sampling is done by step of  $10^\circ$ .

---

<sup>2</sup>A rotamer is a possible conformation of a side-chain.

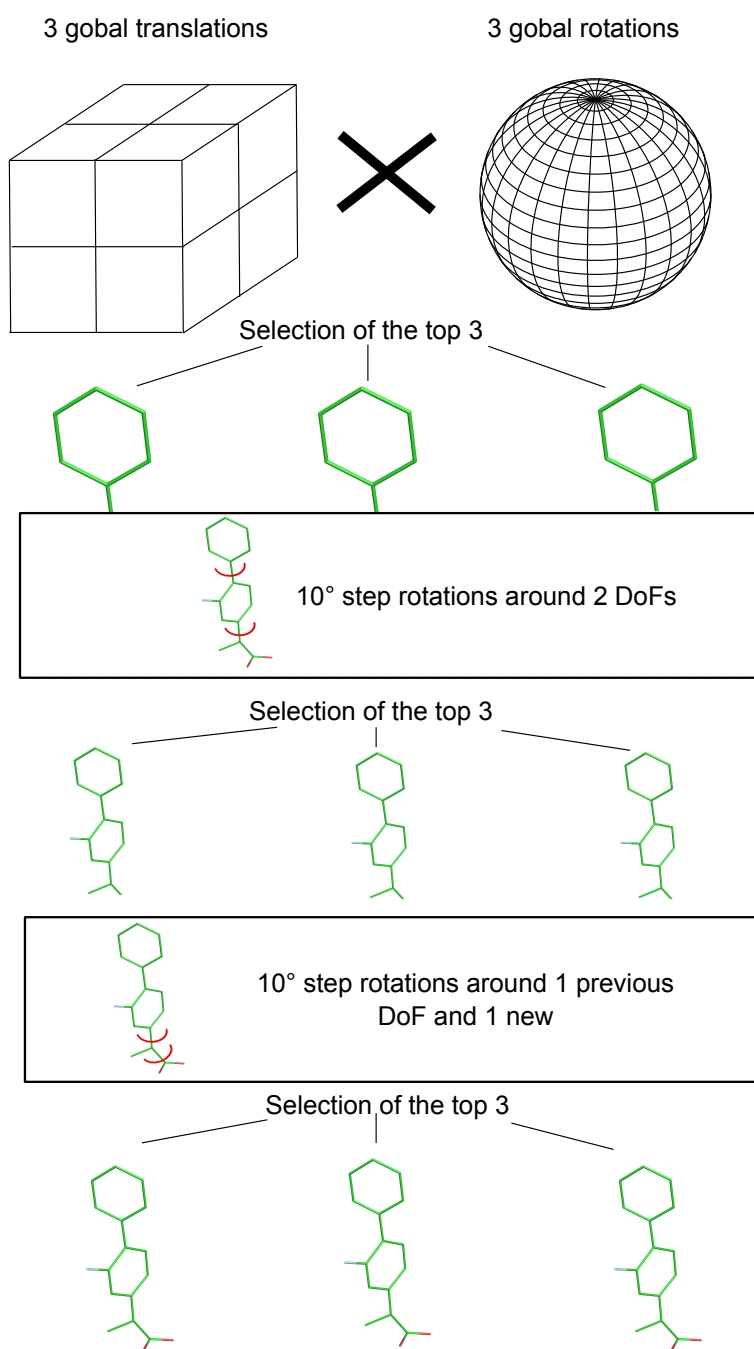


FIGURE 4.1: Workflow for finding ligand's alternate conformation. The ligand is first globally perturbed with global rotations and translations. From the resulting conformations we select the  $k$  best in term off avoiding collision with the receptor and fitting the density map. For the figure simplicity  $k = 3$  but for experiments we use  $k = 5$ . Then for each of the  $k$  conformations we sample 2 degrees of freedom with 10° step ( $36^2$  total) and again select the  $k$  best. Then for each of the  $k$  conformations we sample 2 degrees of freedom : one new and one already sampled the step before, and we select the  $k$  best. And so on until no more degrees of freedom remain to be sampled.



For non-linear ligand, we first sample one side of the ligand and then the other side. After each sampling of two degrees of freedom, the ligand is checked for collision. Because we construct the ligand hierarchically, we only consider the atoms that will not be affected by others rotations for the collision.

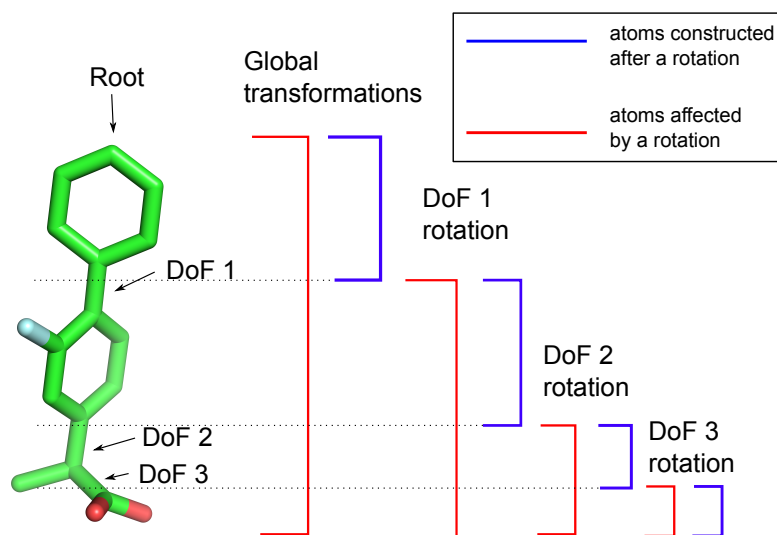


FIGURE 4.2: Sampling of degree of freedom. The ligand is considered as a tree, the root is one of the extremities of the ligand. Each rotation of a rotatable bond affects all atoms below the bond. Atoms that are not affected by the next rotation are called *constructed* and are considered for density fitting and collisions checking.

We reject any colliding conformation, others conformations are fed to fitting program.

#### 4.2.2 Fitting the density

The experimental density map is taken as an input. However to choose the best set of AA conformations we compute their density maps with clipper [103].

Then we use the MIQP solver from IBM CPLEX [104] to retrieve the best combination. MIQP finds the best occupancy vector  $q$  which attributes an occupancy<sup>3</sup> to each conformation generated. The occupancy vector should only contains values between 0 and 1, have a sum of 1, and have less non zero values than a predetermined integer  $k$  (we used 5). So  $q$  is such that

$$q = \arg \min_q (\|\rho^\circ - \sum_i q_i \rho_i\|_2) \quad \text{s.t. } q_i \geq 0 \quad \sum_i q_i = 1 \quad \sum_i \mathbb{1}_{q_i > 0} \leq k \quad (4.1)$$

where  $\rho^\circ$  is the observed electron density map, and  $\rho_i$  is the computed density map for the conformations  $i$  and  $k$  is the maximal number of non-zero values.

<sup>3</sup>Occupancy can be understood as a probability for the AA to be in that conformation.

## 4.3 Preliminary results and Discussions

### 4.3.1 Results

We tested this method on a few examples with good results: one or more alternative conformations are found and improved the density fit. Figure 4.3 shows the result for the protein which PDB id is 2xu3. In this example, the resolution of 0.9 gives precious information that permits to detect an alternative conformation. The blue ligand conformation is the alternative conformation found. It is mostly similar to the native ligand (green) but different by the position of one 5-member ring. We can note that the far right extremity of the ligand reach an unexplained density.

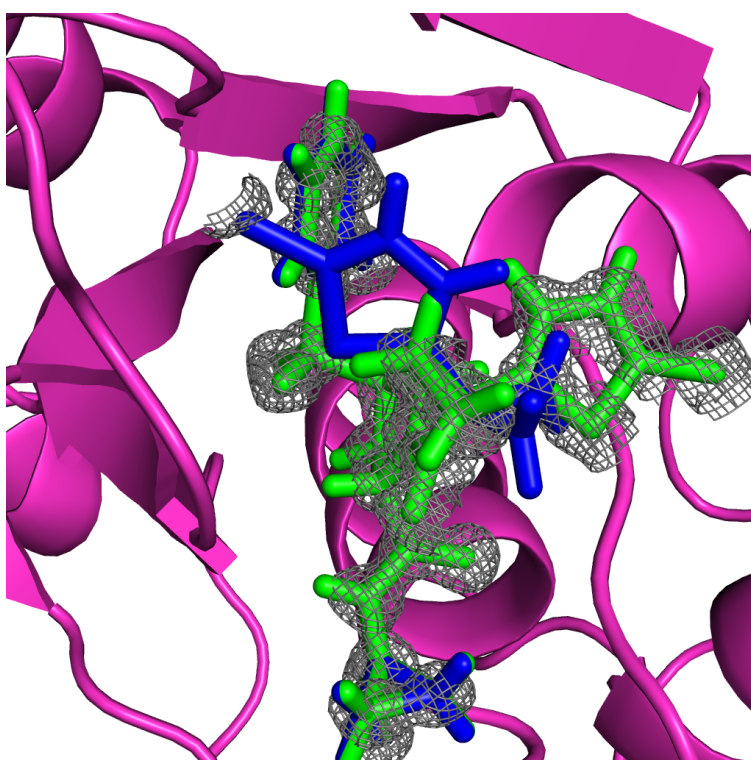


FIGURE 4.3: Alternative ligand for the protein ligand complex 2xu3. The resolution of 0.9 allows for very precise details that permit to detect and alternative conformation. The protein is in pink, the main ligand conformation given in the structure file is in green. The ligand experimental density in shown in gray mesh. The alternative ligand's conformation we found is in blue.

### 4.3.2 Discussion

Our first results are promising, we can find alternative conformations of ligands that explain better the density. A validation step would be to test the method on synthetic data. That is to say to retrieve alternative conformations from density maps that are generated from a ligand with 2 or more alternate conformations. After that step we need to select a dataset of interesting complexes, with medical applications, and with high resolution (any work on density needs a good resolution) to run our method on.

An important point that we did not address yet is overfitting. How can we know if alternative conformations correspond to actual ligand conformations or noise on

the density map? And how can we know how many conformations we are looking for? We are trying to address this problem by trying different maximal numbers of conformations (from 1 to 5) and comparing the result set. If they are globally the same, i.e. for example, if the set of 2 different conformations is included in the set of 3 different conformations, then we assume that the 2 conformations were not due to noise.

An other perspective is to combine both the protein and ligand fitting. For now we suppose that the protein is in one conformation to find multiple conformations of the ligand. However we can easily imagine a collective motion of the ligand and side chains of the proteins. So alternative protein conformations correspond to alternative ligand conformations. This is the next thing we want to do.

## Chapter 5

# Multi-player exponential learning with full information

### Contents

---

<b>5.1 Preliminaries</b> . . . . .	<b>68</b>
5.1.1 Notations . . . . .	68
5.1.2 HEDGE algorithm . . . . .	70
<b>5.2 Convergence in generic games</b> . . . . .	<b>71</b>
5.2.1 Elimination of dominated strategies . . . . .	71
5.2.1.1 Technical Lemma . . . . .	71
5.2.1.2 Results . . . . .	73
5.2.1.3 Discussions . . . . .	74
5.2.2 Local convergence towards strict equilibrium . . . . .	74
5.2.2.1 Logit map properties . . . . .	74
5.2.2.2 Result . . . . .	75
5.2.2.3 Discussions . . . . .	76
5.2.3 Global convergence towards an unique strict equilibrium . .	76
5.2.3.1 Kullback-Leiber divergence properties . . . . .	77
5.2.3.2 Results . . . . .	78
5.2.3.3 Discussion . . . . .	80
<b>5.3 Global convergence in generic potential games</b> . . . . .	<b>80</b>
5.3.1 Asymptotic Pseudotrajectory . . . . .	81
5.3.2 Global convergence in the continuous dynamics . . . . .	82
5.3.3 Global convergence in the potential game . . . . .	83
5.3.4 Convergence Rate . . . . .	84
<b>5.4 Conclusions</b> . . . . .	<b>86</b>

---

In this Chapter we study a special case of multi-player EW learning algorithms call HEDGE. These algorithms are broadly studied in regret minimization where only the average matters, not the final state see (Chapter 2). We show here that, in generic games, such algorithms have the rational property of eliminating dominated strategies at exponential rate. This property leads the algorithm to converge toward a Nash Equilibrium provided that it has been initialized close enough, or that the NE is unique. Furthermore by restricting our analysis to potential games we show that the mixed strategies profile converges toward a NE regardless of initial conditions. In the first section we introduce notations and the algorithm. The second section focuses on the asymptotic local convergence, it corresponds to a paper recently accepted at SAGT 2017 (*Symposium on Algorithmic Game Theory*). The third section brings stronger properties with the global convergence toward equilibria, restricting the analysis to the class of potential games.

## 5.1 Preliminaries

### 5.1.1 Notations

We recall definitions introduced in Chapter 2. We focus on games that are played by a finite set  $\mathcal{N} = \{1, \dots, N\}$  of  $N$  players (or agents). Each player  $i \in \mathcal{N}$  is assumed to have a finite set of actions (or pure strategies)  $\mathcal{S}_i$ , and the players' preferences for one action over another are represented by each action's utility (or payoff). Specifically, as players interact with each other, the individual payoff of each player is given by a function  $u_i: \mathcal{S} \equiv \prod_i \mathcal{S}_i \rightarrow \mathbb{R}$  of all players' actions, and each agent seeks to maximize the utility  $u_i(s_i; s_{-i})$  of his chosen action  $s_i \in \mathcal{S}_i$  against the action profile  $s_{-i}$  of his opponents.<sup>1</sup> A game is then called *generic* if a small change to any of the payoff does not introduce new Nash equilibria nor remove existing ones. In particular, in generic games, there are no unilateral payoff ties in the support of Nash Equilibria, i.e. if  $u_i(s_i; s_{-i}) \neq u_i(s'_i; s_{-i})$  for all Nash Equilibrium  $x^*$ ,  $s_i \in \text{supp}(x_i^*)$ ,  $s'_i \in \mathcal{S}_i$ ,  $i \in \mathcal{N}$ .

Players can also use *mixed strategies* by playing probability distributions  $x_i = (x_{is_i})_{s_i \in \mathcal{S}_i} \in \Delta(\mathcal{S}_i)$  over their action sets  $\mathcal{S}_i$ . The resulting probability vector  $x_i$  is called the *mixed strategy* of the  $i$ -th player and the set  $\mathcal{X}_i = \Delta(\mathcal{S}_i)$  is the corresponding mixed strategy space of player  $i$ . Based on this, we write  $\mathcal{X} \equiv \prod_i \mathcal{X}_i$  for the game's *strategy space*, i.e. the space of all mixed strategy profiles  $x = (x_i)_{i \in \mathcal{N}}$ . In this context (and in a slight abuse of notation), the expected payoff of the  $i$ -th player in the mixed strategy profile  $x = (x_1, \dots, x_N)$  is

$$u_i(x) = \sum_{s_1 \in \mathcal{S}_1} \cdots \sum_{s_N \in \mathcal{S}_N} u_i(s_1, \dots, s_N) x_{1s_1} \cdots x_{Ns_N}. \quad (5.1)$$

Accordingly, if player  $i$  plays the pure strategy  $s_i \in \mathcal{S}_i$ , we will write

$$v_{is_i}(x) = u_i(s_i; x_{-i}) = u_i(x_1, \dots, s_i, \dots, x_N) \quad (5.2)$$

for the payoff corresponding to  $s_i$ , and  $v_i(x) = (v_{is_i}(x))_{s_i \in \mathcal{S}_i}$  for the resulting *payoff vector* of player  $i$ . A player's expected payoff can thus be written as

$$u_i(x) = \sum_{s_i \in \mathcal{S}_i} x_{is_i} v_{is_i}(x) = \langle v_i(x) | x_i \rangle, \quad (5.3)$$

where  $\langle v_i(x) | x_i \rangle$  denotes the canonical bilinear pairing between  $v_i(x)$  and  $x_i$ .

A fundamental rationality principle in the analysis of finite games is that, assuming full knowledge of the game, a player would have no incentive to play an action that always yields suboptimal payoffs with respect to another (fixed) action. To formalize this,  $s_i \in \mathcal{S}_i$  is called (*strictly*) *dominated* by  $s'_i$  (and written  $s_i \prec s'_i$ ) if

$$u_i(s_i; s_{-i}) < u_i(s'_i; s_{-i}) \quad \text{for all } s_{-i} \in \mathcal{S}_{-i} \equiv \prod_{j \neq i} \mathcal{S}_j, i \in \mathcal{N}. \quad (5.4)$$

Extending the notion of strategic dominance, the most widely used solution concept in game theory is that of a NE, i.e. a state  $x^* \in \mathcal{X}$  which is unilaterally stable in the sense that

$$u_i(x_i^*; x_{-i}^*) \geq u_i(x_i; x_{-i}^*) \quad \text{for all } x_i \in \mathcal{X}_i, i \in \mathcal{N}, \quad (\text{NE})$$

<sup>1</sup>In the above  $(s_i; s_{-i})$  is shorthand for  $(s_1, \dots, s_i, \dots, s_N)$ , used here to highlight the action of player  $i$  against that of all other players.

or, equivalently, writing  $\text{supp}(x)$  for the support of  $x$ :

$$v_{is_i}(x^*) \geq v_{is'_i}(x^*) \quad \text{for all } s_i \in \text{supp}(x_i^*) \text{ and all } s'_i \in \mathcal{S}_i, i \in \mathcal{N}. \quad (5.5)$$

If  $x^*$  is *pure* (i.e.  $\text{supp}(x_i^*) = \{s_i^*\}$  for some  $s_i^* \in \mathcal{S}_i$  and all  $i \in \mathcal{N}$ ), then it is called a *pure equilibrium*. In generic games, a pure equilibrium satisfies (5.5) as a strict inequality for all  $s'_i \notin \text{supp}(x_i^*)$ ,  $i \in \mathcal{N}$ , so we sometimes refer to pure equilibria in generic games as *strict*.

Such equilibria will play a key role in our analysis, so we provide a convenient variational characterization below:

**Proposition 5.1.** *In generic games,  $x^*$  is a pure equilibrium if and only if there exists a neighborhood  $U$  of  $x^*$  and*

$$\langle v(x)|x - x^* \rangle \leq -\frac{1}{2}a\|x - x^*\| \quad \text{for some } a > 0 \text{ and for all } x \in U, \quad (5.6)$$

where  $\|x\| = \sum_i \sum_{s \in \mathcal{S}_i} |x_{is}|$  denotes the  $L^1$ -norm of  $x$ .

*Proof.* Assume that  $x^*$  is a pure equilibrium. Then, for all  $i \in \mathcal{N}$ , we have

$$\begin{aligned} \langle v_i(x)|x_i - x_i^* \rangle &= u_i(x_i; x_{-i}) - u_i(x_i^*; x_{-i}) \\ &= \sum_{s_i \neq s_i^*} x_{is_i} u_i(s_i; x_{-i}) + x_{is_i^*} u_i(s_i^*; x_{-i}) - u_i(s_i^*; x_{-i}) \\ &= \sum_{s_i \neq s_i^*} x_{is_i} [u_i(s_i; x_{-i}) - u_i(s_i^*; x_{-i})], \end{aligned} \quad (5.7)$$

where the first line is a consequence of (5.3) while the last one follows by noting that  $\sum_{s_i \neq s_i^*} x_{is_i} = 1 - x_{is_i^*}$  and rearranging. Equation (5.5) is a strict inequality for all  $s'_i \notin \text{supp}(x_i^*)$ ,  $i \in \mathcal{N}$  for pure equilibrium in generic games. Therefore  $u_i(s_i^*; x_{-i}^*) - u_i(s_i; x_{-i}^*) > 0$  for all  $s_i \in \mathcal{S}_i \setminus \{s_i^*\}$ . Now, by continuity there exists a real number  $a > 0$  and a neighborhood  $U$  of  $x^*$  in  $\mathcal{X}$  such that for all  $s_i \in \mathcal{S}_i \setminus \{s_i^*\}$ ,  $u_i(s_i^*; x_{-i}) - u_i(s_i; x_{-i}) \geq a > 0$  for all  $x \in U$ . Therefore:

$$\langle v_i(x)|x_i - x_i^* \rangle \leq -a \sum_{s_i \neq s_i^*} x_{is_i} \quad (5.8)$$

Hence, combining Eqs. (5.7) and (5.8), we get the bound

$$\langle v(x)|x - x^* \rangle = \sum_{i \in \mathcal{N}} \langle v_i(x)|x_i - x_i^* \rangle \leq -a \sum_{i \in \mathcal{N}} \sum_{s_i \neq s_i^*} x_{is_i} \leq -\frac{a}{2} \sum_{i \in \mathcal{N}} \|x_i - x_i^*\|, \quad (5.9)$$

where the last inequality follows from the fact that  $x_{is_i^*}^* = 0$  if  $s_i \neq s_i^*$  so  $\|x_i - x_i^*\| = 1 - x_{is_i^*} + \sum_{s_i \neq s_i^*} x_{is_i} = 2 \sum_{s_i \neq s_i^*} x_{is_i}$ .

Assume now that  $x^*$  satisfies (5.6) but is not a strict NE, so  $v_{is_i}(x^*) \leq v_{is'_i}(x^*)$  for some  $s_i \in \text{supp}(x_i^*)$ ,  $s'_i \in \mathcal{S}_i \setminus \{s_i\}$ ,  $i \in \mathcal{N}$ . Then, if we take  $x = x^*$  except for two coordinates  $x_{is'_i} = x_{is'_i}^* + \lambda$  and  $x_{is_i} = x_{is_i}^* - \lambda$  with  $\lambda > 0$  small enough such that  $x \in U$ , we get

$$\langle v(x)|x - x^* \rangle = \langle v_i(x)|x_i - x_i^* \rangle = \lambda v_{is'_i}(x^*) - \lambda v_{is_i}(x^*) \geq 0, \quad (5.10)$$

in contradiction to (5.6) which yields  $\langle v(x)|x - x^* \rangle < 0$ .  $\square$

Clearly, if (5.6) holds for all  $x \in \mathcal{X}$ , then  $x^*$  is the unique NE of the game; in particular, it is easy to verify that this is the case in the Prisoner's Dilemma and its variants, potential games with a unique equilibrium, etc.

### 5.1.2 HEDGE algorithm

The algorithm that we examine is the so-called *HEDGE* variant of the EW algorithm [83]. In a nutshell, the main idea of the algorithm is as follows: at each stage  $t = 1, 2, \dots$  of the process, players maintain and update a *performance score* for each of their actions (pure strategies) based on each action's cumulative payoff up to stage  $t$ . These scores are then converted to mixed strategies by assigning exponentially higher probabilities to actions with higher scores; subsequently, a new action is drawn based on these mixed strategies, and the process repeats.

More precisely, this process can be encoded as follows:

**Algorithm 5.1** HEDGE with variable step-size  $\gamma_t$

- 1 Each player  $i \in \mathcal{N}$  has an initial score vector  $y_i(1)$  and plays with initial mixed strategy  $x_i(1) = \Lambda_i(y_i(1))$  where the logit map  $\Lambda_i$  is defined as

$$\Lambda_i(y_i) = \frac{1}{\sum_{s \in \mathcal{S}_i} \exp(y_{is})} (\exp(y_{is}))_{s \in \mathcal{S}_i}. \quad (5.11)$$

- 2 **for** each round  $t$
- 3     Each player  $i \in \mathcal{N}$  draws a pure strategy  $s_i(t)$  according to  $x_i(t)$
- 4     Each player  $i \in \mathcal{N}$  gets an estimate  $\hat{v}_i(t)$  of his payoff vector  $v_i(s(t))$
- 5     Each player  $i \in \mathcal{N}$  updates his mixed strategy  $x_i$  via the recursion

$$\begin{aligned} y_i(t+1) &= y_i(t) + \gamma_t \hat{v}_i(t), \\ x_i(t+1) &= \Lambda_i(y_i(t+1)) \end{aligned} \quad (\text{HEDGE})$$

**end for**

Motivated by practical implementation issues (especially in large networks and telecommunication systems), the formulation of (HEDGE) above tacitly assumes that players have *imperfect knowledge* of their payoff vectors  $v_i(x(t))$  at each iteration of the algorithm – for instance, contaminated by measurement errors or other uncertainty factors. To formalize this assumption, we will focus on the general noisy feedback model

$$\hat{v}_i(t) = v_i(s(t)) + \xi_i(t), \quad (5.12)$$

where the error process  $\xi = (\xi_i)_{i \in \mathcal{N}}$  satisfies the statistical hypotheses

1. *Zero-mean*:

$$\mathbb{E}[\xi(t) | \mathcal{F}_{t-1}] = 0 \quad \text{for all } t = 1, 2, \dots \text{ (a.s.)} \quad (\text{H1})$$

2. *Finite Mean Square Error (MSE)*: there exists some  $\sigma > 0$  such that

$$\mathbb{E}[\|\xi(t)\|_\infty^2 | \mathcal{F}_{t-1}] \leq \sigma^2 \quad \text{for all } t = 1, 2, \dots \text{ (a.s.)} \quad (\text{H2})$$

In the above, the expectation  $\mathbb{E}[\cdot]$  is taken with respect to the randomness induced by the players' mixed strategies and the error process  $\xi$ , while  $\mathcal{F}_t$  denotes the history of  $(y(t), x(t), s(t), \hat{v}(t))$  up to stage  $t$ .<sup>2</sup> Put differently, Hypotheses (H1) and (H2)

<sup>2</sup>In formal mathematical language,  $\mathcal{F}_t$  is the natural filtration induced by  $(y(t), x(t), s(t), \hat{v}(t))$  [105].

simply mean that the players' estimates  $\hat{v}_i$  are *conditionally unbiased and bounded in mean square*, i.e.

$$\mathbb{E}[\hat{v}(t)|\mathcal{F}_{t-1}] = v(x(t)), \quad \text{for all } t = 1, 2, \dots \text{ (a.s.)} \quad (5.13a)$$

$$\mathbb{E}[\|\hat{v}(t)\|_\infty^2|\mathcal{F}_{t-1}] \leq L^2, \quad \text{for all } t = 1, 2, \dots \text{ (a.s.)} \quad (5.13b)$$

where  $L > 0$  is a finite positive constant (obviously, in the noiseless case  $\xi = 0$ , the constant  $L$  is simply a bound on the players' maximum absolute payoff). Note here that (5.13a) is phrased in terms of the players' *mixed* strategy profile  $x(t)$ , not the action profile  $s(t) = (s_i(t); s_{-i}(t))$  which is chosen based on  $x(t)$  at stage  $t$ . To see that (H1) indeed implies (5.13a) in this context, simply observe that for all  $i \in \mathcal{N}$ ,  $s_i \in \mathcal{S}_i$ , we have

$$\begin{aligned} \mathbb{E}[\hat{v}_{is_i}(t)|\mathcal{F}_{t-1}] &= \sum_{s_{-i} \in \mathcal{S}_{-i}} [u_i(s_i; s_{-i})x_{s_{-i}}(t) + \mathbb{E}[\xi_{is_i}(t)|\mathcal{F}_{t-1}]] \\ &= u_i(s_i; x_{-i}(t)) = v_{is_i}(x(t)). \end{aligned} \quad (5.14)$$

Thus, Hypotheses (H1) and (H2) allow for a broad range of noise distributions, including all compactly supported, (sub-)Gaussian, (sub-)exponential and log-normal distributions.

## 5.2 Convergence in generic games

In this section, we provide our three convergence results for the algorithm (HEDGE); for simplicity, we assume throughout the analysis that the games under study are *generic*, in particular there are no payoff ties between a player's actions. For each result we will follow the same presentation: we start with a technical property needed for the proof, then the statement of the result and its proof and we finish with a discussion on hypotheses.

### 5.2.1 Elimination of dominated strategies

We will often use a martingale property that allows the convergence even in presence of noise.

#### 5.2.1.1 Technical Lemma

The convergence is based on the increasing difference between scores,  $y(t)$ , of the strategies in equilibria's support and other strategies. As defined in the algorithm  $y_s(t)$  is the weighted sum of the estimated payoff,  $\hat{v}_s(t)$ .  $\hat{v}_s(t)$  is subjected to noise, so the estimated payoff of strategies in the support of an equilibrium can be smaller than the estimates payoff of other strategies.

If the noise follows the hypotheses (H1) and (H2), we show that the difference of the weighted sum of the estimated payoff will increase when the difference of the actual payoff (without noise) is positive.

**Lemma 5.2.** *Suppose that (HEDGE) is run with a step-size sequence such that  $\lim_{n \rightarrow \infty} U_n = \infty$ , with  $U_n = \sum_{j=1}^n \gamma_j$ , and noisy payoff observations satisfying Hypotheses (H1) and (H2). If there exists some  $a > 0$  such that  $v_{s'}(x) - v_s(x) \geq a$  for all  $x \in \mathcal{X}$  then for all  $c \in (0, a)$ , there exists some  $t_0$  such that  $y_{s'}(t+1) - y_s(t+1) \geq c \sum_{i=1}^t \gamma_i$  for all  $t \geq t_0$  (a.s.).*



The proof of this lemma lays on the Theorem 2.18 of Hall and Heyde 1980 [105], in particular the equation (2.17). We report here a version of this part of the theorem adapted to our notations with  $\zeta_j = \hat{v}_{s'}(j) - v_{s'}(x(j)) - [\hat{v}_s(j) - v_s(x(j))]$ .

**Theorem 5.3.** [105, Theorem 2.18] *If  $\{S_n = \sum_{j=1}^n \gamma_j \zeta_j, \mathcal{F}_n, n \geq 1\}$  is a martingale and  $U_n = \sum_{j=1}^n \gamma_j$  is a nondecreasing sequence of positive random variable such that  $U_n$  is  $\mathcal{F}_{n-1}$ -measurable for each  $n$  and  $\lim_{n \rightarrow \infty} U_n = \infty$ ,  $\sum_{j=1}^{\infty} \frac{\mathbb{E}(\|\gamma_j \zeta_j\|^2 | \mathcal{F}_{j-1})}{U_j^2} < \infty$  then*

$$\lim_{n \rightarrow \infty} U_n^{-1} S_n = \lim_{n \rightarrow \infty} \frac{\sum_{j=1}^n \gamma_j \zeta_j}{\sum_{j=1}^n \gamma_j} = 0 \quad (a.s.)$$

*Proof.* Proof of Lemma (5.2)

Recall  $\zeta_j = \hat{v}_{s'}(j) - v_{s'}(x(j)) - [\hat{v}_s(j) - v_s(x(j))]$ . By assumption there exists  $a > 0$  such that  $v_{s'}(x) - v_s(x) \geq a$  for all  $x \in \mathcal{X}$ . Then,

$$\begin{aligned} y_{s'}(t+1) - y_s(t+1) &= y_{s'}(1) - y_s(1) + \sum_{j=1}^t \gamma_j (\hat{v}_{s'}(j) - \hat{v}_s(j)) \\ &= y_{s'}(1) - y_s(1) + \sum_{j=1}^t \gamma_j [v_{s'}(x(j)) - v_s(x(j))] + \sum_{j=1}^t \gamma_j \zeta_j \\ &\geq y_{s'}(1) - y_s(1) + \sum_{j=1}^t \gamma_j \left[ a + \frac{\sum_{j=1}^t \gamma_j \zeta_j}{\sum_{j=1}^t \gamma_j} \right]. \end{aligned} \tag{5.15}$$

Now we will use Theorem (5.3) to prove that  $\frac{\sum_{j=1}^t \gamma_j \zeta_j}{\sum_{j=1}^t \gamma_j} \rightarrow_{t \rightarrow \infty} 0$ .

The reformulation of Hypothesis (H1) gives :

$$\begin{aligned} \mathbb{E}[\zeta_j | \mathcal{F}_{j-1}] &= \mathbb{E}[\xi_{s'}(j) + v_{s'}(s(j)) - v_{s'}(x(j)) - \xi_s(j) - v_s(s(j)) + v_s(x(j)) | \mathcal{F}_{j-1}] \\ &= \mathbb{E}[\xi_{s'}(j) - \xi_s(j) | \mathcal{F}_{j-1}] + \mathbb{E}[v_{s'}(s(j)) | \mathcal{F}_{j-1}] - v_{s'}(x(j)) \\ &\quad - \mathbb{E}[v_s(s(j)) | \mathcal{F}_{j-1}] + v_s(x(j)) \\ &= 0 \end{aligned}$$

$\zeta_j$  is  $\mathcal{F}_j$ -measurable, meaning that it is fully determined by the information of  $\mathcal{F}_j$ . With  $S_n = \sum_{i=1}^n \gamma_i \zeta_i$ , it follows that

$$\mathbb{E}[S_j | \mathcal{F}_{j-1}] = \gamma_j \mathbb{E}[\zeta_j | \mathcal{F}_{j-1}] + \mathbb{E}[S_{j-1} | \mathcal{F}_{j-1}] = S_{j-1}$$

Therefore  $\{S_n = \sum_{j=1}^n \gamma_j \zeta_j, \mathcal{F}_n, n \geq 1\}$  is a martingale, and as  $\gamma_t$  depends only on  $t$  and it is positive,  $U_n = \sum_{j=1}^n \gamma_j$  is a nondecreasing sequence of positive random variable such that  $U_n$  is  $\mathcal{F}_{n-1}$ -measurable for each  $n \in \mathbb{N}$ . In addition  $\beta \leq 1$  gives us that  $\lim_{n \rightarrow \infty} U_n = \infty$ .

We focus now on proving the last hypothesis of Theorem (5.3), i.e.,  $\sum_{j=1}^{\infty} \frac{\mathbb{E}(\|\gamma_j \zeta_j\|^2 | \mathcal{F}_{j-1})}{U_j^2} < \infty$ . First we show that  $\mathbb{E}[\|\zeta(j)\|^2 | \mathcal{F}_{j-1}] \leq 8\sigma^2 + 32D^2$  where  $D = \max_{x \in \mathcal{X}} \|v(x)\|_{\infty}$ :

$$\begin{aligned} \mathbb{E}[\|\zeta(j)\|^2 | \mathcal{F}_{j-1}] &= \mathbb{E}[\|(\xi_{s'}(j) + v_{s'}(s(j)) - v_{s'}(x(j)) - \xi_s(j) - v_s(s(j)) + v_s(x(j)))\|^2 | \mathcal{F}_{j-1}] \\ &\leq 4 \max_{s \in \mathcal{S}} \mathbb{E}[\|(\xi_s(j) - v_s(s(j)) + v_s(x(j)))\|^2 | \mathcal{F}_{j-1}] \\ &\leq 8\sigma^2 + 8 \max_{s \in \mathcal{S}} \mathbb{E}[\|(v_s(s(j)) + v_s(x(j)))\|^2 | \mathcal{F}_{j-1}] \\ &\leq 8\sigma^2 + 32 \max_{s \in \mathcal{S}} \mathbb{E}[\|v_s(x(j))\|^2 | \mathcal{F}_{j-1}] \\ &\leq 8\sigma^2 + 32D^2, \end{aligned} \tag{5.16}$$

according to Hypothesis (H2). Second,  $\gamma_t$  is decreasing so  $U_n = \sum_{j=1}^n \gamma_j \geq n\gamma_n$  and  $U_n^{-2} \leq \frac{1}{n^2 \gamma_n^2}$ . Therefore

$$\sum_{j=1}^{\infty} \frac{\mathbb{E}(\|\gamma_j \zeta_j\|^2 | \mathcal{F}_{j-1})}{U_j^2} < \sum_{j=1}^{\infty} \frac{\gamma_j^2 (8\sigma^2 + 32D^2)}{j^2 \gamma_j^2} = (8\sigma^2 + 32D^2) \sum_{j=1}^{\infty} \frac{1}{j^2} < \infty$$

Therefore all hypotheses are fulfilled and

$$\lim_{n \rightarrow \infty} \frac{\sum_{j=1}^n \gamma_j \zeta_j}{\sum_{j=1}^n \gamma_j} = 0 \quad (a.s.)$$

So for all  $c \in (0, a)$  there exist some  $t_0$  such that  $-\frac{\sum_{j=1}^t \gamma_j \zeta_j}{\sum_{j=1}^t \gamma_j} \leq \frac{y_{s'}(1) - y_s(1)}{\sum_{j=1}^t \gamma_j} + a - c$  for all  $t > t_0$ . Putting that in (5.15) we have :

$$y_{s'}(t+1) - y_s(t+1) \geq c \sum_{j=1}^t \gamma_j \quad \text{for all } t \geq t_0 \text{ (a.s.)}$$

□

### 5.2.1.2 Results

We begin our asymptotic analysis with the elimination of dominated strategies:

**Theorem 5.4.** *Suppose that (HEDGE) is run with a step-size sequence of the form  $\gamma_t \propto 1/t^\beta$  for some  $\beta \leq 1$  (not necessarily positive), and noisy payoff observations satisfying Hypotheses (H1) and (H2). If  $s_i \in \mathcal{S}_i$  is dominated, there exists some  $c > 0$  such that*

$$x_{is_i}(t) = \mathcal{O}(\exp(-c \sum_{j=1}^{t-1} \gamma_j)) \quad \text{with probability 1.} \tag{5.17}$$

In particular, if  $\beta < 1$ ,  $s_i$  becomes extinct exponentially fast (a.s.).

*Proof.* Proof of Theorem (5.4).

Suppose that  $s_i$  is dominated by  $s'_i$  for some  $s'_i \in \mathcal{S}_i$ . There exists some  $a > 0$  such that  $v_{is'_i}(x) - v_{is_i}(x) \geq a$  for all  $x \in \mathcal{X}$ . We can then apply Lemma (5.2) and obtain  $y_{is'_i}(t) - y_{is_i}(t) \geq c \sum_{j=1}^{t-1} \gamma_j$  for all  $t \geq t_0$  (a.s.). We thus get

$$x_{is_i}(t) = \frac{e^{y_{is_i}(t)}}{\sum_{s''_i} e^{y_{is''_i}(t)}} \leq \frac{e^{y_{is_i}(t)}}{e^{y_{is'_i}(t)}} = e^{y_{is_i}(t) - y_{is'_i}(t)} \leq e^{-c \sum_{j=1}^{t-1} \gamma_j} \tag{5.18}$$

and our proof is complete. □

### 5.2.1.3 Discussions

It should be noted here that the elimination of dominated strategies with imperfect knowledge of the game's payoffs is by no means a given. For instance, if players play a greedy best response scheme at each round and the payoff observation errors are not supported on a small, compact set, dominated strategies will be played infinitely often (simply because at each round, any strategy could be erroneously perceived as a best response) [75]. With this in mind, the fact that the rate of elimination (5.17) *improves* with more aggressive – even *increasing* – step-size sequences  $\gamma_t$  is somewhat surprising because it suggests that players can employ (HEDGE) in a very greedy fashion and achieve fast dominated strategy extinction rates, even in the presence of arbitrarily high estimation errors.

Regarding the game's dimensionality (i.e. the number of players and actions per player), our proof also shows that  $c$  depends only on the player's payoffs – specifically, we can take  $c = \frac{1}{2} \min_{s_{-i} \in \mathcal{S}_{-i}} [u_i(s'_i; s_{-i}) - u_i(s_i; s_{-i})] > 0$ . In other words, the algorithm's half-life is asymptotically *independent* of the size of the game, and only depends on the players' relative payoff differences.

## 5.2.2 Local convergence towards strict equilibrium

Before addressing the local convergence towards strict equilibrium we show an important property of the logit map:

### 5.2.2.1 Logit map properties

We show that if  $x^*$  is a pure equilibrium, for any  $x$  close enough to  $x^*$ , for all players  $i$ , the difference between score  $y_{s_i^*}(t)$  (with  $s_i^*$  being the support of  $x^*$ ) of  $s_i^*$  and any other pure strategy of player  $i$  can be large as we want.

**Proposition 5.5.** *Let  $\mathcal{S} = \{1, \dots, n\}$  be a finite set and let  $\Delta \equiv \Delta(\mathcal{S})$  denote the  $(n - 1)$ -dimensional simplex spanned by  $\mathcal{S}$ . Then, if  $x^* \in \Delta$  is pure (i.e.  $\text{supp}(x^*) = \{s^*\}$  for some  $s^* \in \mathcal{S}$ ), the set  $U_M = \{x = \Lambda(y) : y_s - y_{s^*} \leq -M\}$  for all  $s \neq s^*$  is a neighborhood of  $x^*$  in  $\Delta^\circ$ ; furthermore, if  $M$  is sufficiently large,  $U_M$  is contained in any  $\|\cdot\|$ -ball centered at  $x^*$ .*

*Proof.* We assume to the contrary that  $U_M$  is not a neighborhood of  $x^*$  in  $\mathcal{X}^\circ$ . So there exists a sequence  $x_k = \Lambda(y_k)$  in  $\mathcal{X}^\circ$  that converges to  $x^*$ , but  $x_k \notin U_M$  for all  $k$ . By passing to a subsequence if necessary, there exists some  $s \in \mathcal{S}$  such that  $y_{k,s} - y_{k,s^*} > -M$  and  $y_{k,s} \geq y_{k,s'}$  for all  $s'$  (simply pick a constant subsequence of  $\arg \max_{s' \in \mathcal{S}} \{y_{k,s'} : y_{k,s'} - y_{k,s^*} > -M\}$  if needed). We then get

$$x_{k,s} = \frac{e^{y_{k,s}}}{\sum_{s'} e^{y_{k,s'}}} = \frac{1}{e^{y_{k,s^*} - y_{k,s}} + \sum_{s' \neq s^*} e^{y_{k,s'} - y_{k,s}}} \geq \frac{1}{e^M + n - 1}, \quad (5.19)$$

contradicting the original assumption  $x_{k,s} \rightarrow 0$  (since  $x_k \rightarrow x^*$ ).

For the converse implication (namely that  $U_M$  is contained in a ball around  $x^*$ ), fix some  $\delta > 0$  and let  $z_s = y_s - y_{s^*}$ ,  $s \in \mathcal{S} \setminus \{s^*\}$ . Then, letting  $x = \Lambda(y)$  for some  $y \in U_M$ , we have

$$x_{s^*} = \frac{e^{y_{s^*}}}{\sum_{s \in \mathcal{S}} e^{y_s}} = \frac{1}{1 + \sum_{s \neq s^*} e^{z_s}} \geq 1 - \sum_{s \neq s^*} e^{z_s} \geq 1 - (n - 1)e^{-M}. \quad (5.20)$$

Thus, for  $M > \frac{|\log \delta|}{2(n-1)}$ , we obtain  $\|x - x^*\| = 2(1 - x_{s^*}) \leq 2(n - 1)e^{-M} \leq \delta$ , implying that  $U_M$  is contained in the ball  $B_\delta = \{x : \|x - x^*\| \leq \delta\}$ .  $\square$

### 5.2.2.2 Result

We now turn to local convergence properties of (HEDGE) in generic games that admit pure NE.

**Theorem 5.6.** *Fix a confidence level  $\varepsilon > 0$  and suppose that (HEDGE) is run with a small enough (depending on  $\varepsilon$ ) step-size  $\gamma_t$  satisfying  $\sum_{t=1}^{\infty} \gamma_t^2 < \infty$  and  $\sum_{t=1}^{\infty} \gamma_t = \infty$  and imperfect payoff observations satisfying Hypotheses (H1) and (H2). If  $x^*$  is a pure equilibrium and (HEDGE) is initialized not too far from  $x^*$ , we have*

$$\mathbb{P}\left(\|x(t+1) - x^*\| \leq C'e^{-c\sum_{j=1}^t \gamma_j} \text{ for all } t\right) \geq 1 - \varepsilon, \quad (5.21)$$

where  $c > 0$  is a constant that only depends on the game and  $C' > 0$  is a constant that depends on the initialization of (HEDGE). In particular, under the stated assumptions,  $x(t) \rightarrow x^*$  with probability at least  $1 - \varepsilon$ .

*Proof.* Proof of Theorem (5.6).

Write  $x^* = (s_1^*, \dots, s_N^*)$  for the pure equilibrium under study.  $x^*$  is a pure equilibrium so there exist  $a > 0$  and a neighborhood  $U$  of  $x^*$  in  $\mathcal{X}$  such that for all  $s_i \in \mathcal{S}_i \setminus \{s_i^*\}$ ,  $u_i(s_i^*; x_{-i}) - u_i(s_i; x_{-i}) \geq a$ . By Proposition (5.5) there is a  $M > 0$  such the  $U_M \subset U$ . Let set  $z_{is_i} = y_{is_i} - y_{is_i^*}$  then, we have

$$z_{is_i}(t+1) = z_{is_i}(t) + \gamma_t [v_{is_i}(x(t)) - v_{is_i^*}(x(t))] + \gamma_t \zeta_{is_i}(t), \quad (5.22)$$

where  $\zeta_{is_i}(t) = \hat{v}_{is_i}(t) - v_{is_i}(x(t)) - (\hat{v}_{is_i^*}(t) - v_{is_i^*}(x(t)))$ . Thus, assuming that (HEDGE) is initialized in  $U_{2M}$  and telescoping, we get

$$z_{is_i}(t+1) \leq -2M + \sum_{j=1}^t \gamma_j [v_{is_i}(x(j)) - v_{is_i^*}(x(j))] + \sum_{j=1}^t \gamma_j \zeta_{is_i}(j). \quad (5.23)$$

We now claim that, if  $\gamma_t$  is chosen appropriately, we have

$$\mathbb{P}(\sup_t \sum_{j=1}^t \gamma_j \zeta_{is_i}(j) \leq M) \geq 1 - \varepsilon/(N(S_i - 1)), \quad (5.24)$$

where  $S_i = |\mathcal{S}_i|$ . Indeed, let  $X_{is_i}(t) = \sum_{j=1}^t \gamma_j \zeta_{is_i}(j)$  and let  $E_i(t)$  denote the event  $\sup_{1 \leq j \leq t} |X_{is_i}(j)| \geq M$ . By Hypothesis (H1),  $X_{is_i}(t)$  is a martingale (see proof of Lemma (5.2)) so Doob's maximal inequality [105, Theorem 2.1] yields

$$\mathbb{P}(E_i(t)) \leq \frac{\mathbb{E}[X_{is_i}(t)^2]}{M^2} \leq \frac{(8\sigma^2 + 32D^2) \sum_{j=1}^t \gamma_j^2}{M^2}, \quad (5.25)$$

where we used the noise variance estimate

$$\mathbb{E}[\zeta_{is_i}^2(t)] \leq 8\sigma^2 + 32D^2,$$

proved in the proof of Lemma (5.2) and the fact that  $\mathbb{E}[\mathbb{E}[\zeta_{is_i}(t)\zeta_{is_i}(t')|\mathcal{F}_t]] = 0$  if  $t > t'$ . Since  $E_i(t+1) \subseteq E_i(t) \subseteq \dots$ , it follows that the event  $E_i = \cap_{t=1}^{\infty} E_i(t)$  occurs with probability  $\mathbb{P}(E_i) \leq (8\sigma^2 + 32D^2)\Gamma_2/M^2$  where  $\Gamma_2 = \sum_{t=1}^{\infty} \gamma_t^2 < \infty$ . Thus, if  $\gamma_t$  is chosen so that  $\Gamma_2 \leq \varepsilon M^2/(N(S_i - 1)(8\sigma^2 + 32D^2))$ , we get  $\mathbb{P}(X_{is_i}(t) \geq M \text{ for all } t) \leq \varepsilon/(N(S_i - 1))$ .

Assume therefore that  $\Gamma_2 \leq \varepsilon M^2/(N(S_i - 1)(8\sigma^2 + 32D^2))$ . Then, we obtain

$$\mathbb{P}(\max_{i \in \mathcal{N}, s_i \in \mathcal{S}_i} \sup_t X_{is_i}(t) \geq M) \leq \sum_{i \in \mathcal{N}} \sum_{s_i \neq s_i^*} \frac{\varepsilon}{N(S_i - 1)} \leq \varepsilon. \quad (5.26)$$

Hence, going back to (5.23), an induction shows that  $x(t) \in U_M$  for all  $t$  with probability at least  $1 - \varepsilon$ . As  $x^*$  is a pure equilibrium there exist  $a > 0$  and a neighborhood  $U$  of  $x^*$  in  $\mathcal{X}$  such that  $v_{is_i}(x(j)) - v_{is_i^*}(x(j)) < -a$  for all  $s_i \neq s_i^*$  and  $x \in U$ . Proposition (5.5) ensures that with  $M$  large enough  $U_M \subset U$ . When this occurs, we also have

$$\mathbb{P}(z_{is_i}(t+1) \leq -M - a \sum_{j=1}^t \gamma_j \text{ for all } t) \leq 1 - \varepsilon. \quad (5.27)$$

$$\begin{aligned} x_{is_i^*}(t+1) &= \frac{1}{1 + \sum_{s_i \neq s_i^*} \exp(z_{is_i}(t+1))} \\ &\geq 1 - \sum_{s_i \neq s_i^*} \exp(z_{is_i}(t+1)) \geq 1 - \sum_{s_i \neq s_i^*} e^{-M} e^{-a \sum_{j=1}^t \gamma_j}, \end{aligned} \quad (5.28)$$

with probability at least  $1 - \varepsilon$ . Therefore, since  $\|x_i - x_i^*\| = 1 - x_{is_i^*} + \sum_{s_i \neq s_i^*} x_{is_i} = 2(1 - x_{is_i^*})$ , rearranging (5.28) yields

$$\mathbb{P}\left(\|x(t+1) - x^*\| \leq 2 \sum_{i \in \mathcal{N}} \sum_{s_i \neq s_i^*} e^{-M} e^{-a \sum_{j=1}^t \gamma_j}\right) \geq 1 - \varepsilon,$$

and our assertion follows.  $\square$

More specifically if the step-size is of the form  $\gamma_t = \gamma/t^\beta$ , we obtain an expression for the convergence rate.

**Corollary 5.7.** *With assumptions as above, if (HEDGE) is run with a step-size of the form  $\gamma_t = \gamma/t^\beta$  for some sufficiently small  $\gamma > 0$  and  $\beta \in ]1/2, 1[$ , we have*

$$\mathbb{P}\left(\|x(t) - x^*\| = \mathcal{O}\left(e^{-c\gamma t^{1-\beta}}\right)\right) \geq 1 - \varepsilon, \quad (5.29)$$

### 5.2.2.3 Discussions

We note here that, in contrast to Theorem (5.4), the summability requirements  $\sum_{t=1}^{\infty} \gamma_t^2 < \infty$  and  $\sum_{t=1}^{\infty} \gamma_t = \infty$  constrain the admissible step-size policies that lead to pure equilibrium (for instance, constant step-size policies are no longer admissible). In particular, the most aggressive step-size that can be used in the context of Theorem (5.6) is  $\gamma_t \propto t^{-\beta}$  for some  $\beta$  close (but not equal) to  $1/2$ , leading to a convergence rate of  $\lambda^{t^{1-\beta}}$  for some  $\lambda < 1$  (cf. Corollary (5.7)).

The main idea behind the proof of Theorem (5.6) is to use Doob's maximal inequality for martingales to show that the probability of  $x(t)$  escaping the basin of attraction of a pure NE  $x^*$  can be made arbitrarily small if the algorithm's step-size is chosen appropriately. Building on this, we now analyse the global convergence when the NE is unique. If  $x^*$  satisfies the variational inequality (5.6) throughout  $\mathcal{X}$ , we have the stronger results.

### 5.2.3 Global convergence towards an unique strict equilibrium

We now study the case where there is an unique strict equilibrium that satisfies (5.6) for all  $x \in \mathcal{X}$ . Any strict equilibrium is a pure equilibrium.

### 5.2.3.1 Kullback-Leiber divergence properties

As opposed to Theorems (5.4) and (5.6), the global convergence toward a pure equilibrium relies heavily on the so-called *Kullback–Leibler* (KL) divergence [106], defined here as

$$D_{\text{KL}}(x^*, x) = \sum_{i \in \mathcal{N}} \sum_{s_i \in \mathcal{S}_i} x_{is_i}^* \log \frac{x_{is_i}^*}{x_{is_i}} \quad \text{for all } x \in \mathcal{X}^\circ. \quad (5.30)$$

The KL divergence is a positive-definite, asymmetric distance measure that is particularly well-adapted to the analysis of the replicator dynamics [107]–[109]. Before addressing the problem of global convergence towards an unique strict equilibrium we show an important property of the KL divergence.

**Proposition 5.8.** *Let  $\mathcal{S} = \{1, \dots, n\}$  be a finite set and let  $\Delta \equiv \Delta(\mathcal{S})$  denote the  $(n - 1)$ -dimensional simplex spanned by  $\mathcal{S}$ . Let  $x = \Lambda(y)$ ,  $x' = \Lambda(y')$  for some  $y, y' \in \mathbb{R}^n$  and  $x^* \in \mathcal{X}$ . Then, we have*

$$D_{\text{KL}}(x^*, x') \leq D_{\text{KL}}(x^*, x) + \langle y' - y | x - x^* \rangle + \frac{1}{2} \|y' - y\|_\infty^2. \quad (5.31)$$

*Proof.* Let  $h(x) = \sum_s x_s \log(x_s)$ ,  $x \in \Delta$ , and let  $h^*(y) = \max_{x \in \mathcal{X}} \{\langle y | x \rangle - h(x)\} = \log(\sum_s e^{y_s})$  (see Appendix A for the calculation) denote the convex conjugate of  $h$  [110], [111]. Then, the derivation yields

$$\frac{\partial h^*}{\partial y_s} = \frac{\exp(y_s)}{\sum_{s'} \exp(y_{s'})} = \Lambda_{s'}(y), \quad (5.32)$$

so, by the properties of Legendre transformations [110] (see Appendix A), we get  $\Lambda(y) = \arg \max \{\langle y | x \rangle - h(x)\}$ . Therefore, taking  $x = \Lambda(y)$ , the KL divergence becomes

$$\begin{aligned} D_{\text{KL}}(x^*, x) &= \sum_s x_s^* \log(x_s^*) - \sum_s x_s^* \log(x_s) \\ &= h(x^*) - \sum_s x_s \log(x_s) + \sum_s ((x_s - x_s^*) \log(x_s)) \\ &= h(x^*) - h(x) + \langle \nabla h(x) | x - x^* \rangle \\ &= h(x^*) + \langle y | x \rangle - h(x) - \langle y | x \rangle + \langle \nabla h(x) | x - x^* \rangle \\ &= h(x^*) + h^*(y) - \langle y | x^* \rangle - \langle y | x \rangle + \langle y | x^* \rangle + \langle \nabla h(x) | x - x^* \rangle \\ &= h(x^*) + h^*(y) - \langle y | x^* \rangle + \langle \nabla h(x) - y | x - x^* \rangle \\ &= h(x^*) + h^*(y) - \langle y | x^* \rangle =: F(x^*, y), \end{aligned} \quad (5.33)$$

where  $F(x^*, y)$  is the so-called Fenchel coupling [112], and we used the fact that  $y = \nabla h(x) + \lambda \mathbf{1}$  (recall that  $x = \Lambda(y)$  is defined as the maximizer of the quantity  $\langle y | x \rangle - h(x)$ ).

With this in mind, it suffices to show that

$$F(x^*, y') \leq F(x^*, y) + \langle y' - y | \Lambda(y) - x^* \rangle + \frac{1}{2} \|y' - y\|_\infty^2. \quad (5.34)$$

However, since  $h$  is 1-strongly convex with respect to the  $L^1$  norm [111, p. 135], it follows that its convex conjugate  $h^*$  is 1-strongly smooth with respect to the  $L^\infty$

norm (the dual of the  $L^1$ -norm) [111, p. 148]. Specifically, this implies that

$$h^*(y') \leq h^*(y) + \langle y' - y | \nabla h^*(y) \rangle + \frac{1}{2} \|y' - y\|_\infty^2 \quad (5.35)$$

Eq. (5.34) then follows by writing out the definition of  $F(x^*, y')$  and then using (5.35) and (5.32).  $\square$

### 5.2.3.2 Results

We show that  $x^*$  is a *recurrent point* of the process  $x(t)$ , i.e.  $x(t)$  visits any neighborhood of  $x^*$  infinitely many times. We then use an argument similar to the proof of Theorem (5.4) to show that the process actually converges to  $x^*$  at an asymptotic rate of  $\mathcal{O}(e^{-c \sum_{j=1}^t \gamma_j})$ .

**Theorem 5.9.** *Suppose that the algorithm (HEDGE) is run with a step-size  $\gamma_t$  such that  $\sum_{t=1}^\infty \gamma_t^2 < \infty$ ,  $\sum_{t=1}^\infty \gamma_t = \infty$  and imperfect payoff observations satisfying Hypotheses (H1) and (H2). If  $x^*$  satisfies (5.6) for all  $x \in \mathcal{X}$ , then:*

1.  $x(t) \rightarrow x^*$  (a.s.).
2. There exists a (deterministic) constant  $c > 0$  depending only on the game such that

$$\|x(t+1) - x^*\| = \mathcal{O}(e^{-c \sum_{j=1}^t \gamma_j}). \quad (5.36)$$

*Proof.* Since  $\sum_{t=1}^\infty \gamma_t = \infty$ , it suffices to prove (5.36). Then, given that  $x^* = (s_1^*, \dots, s_N^*)$  is pure, we have:

$$\begin{aligned} D_{\text{KL}}(x^*, x) &= \sum_{i \in \mathcal{N}} \sum_{s_i \in \mathcal{S}_i} x_{s_i}^* \log \frac{x_{is_i}^*}{x_{is_i}} \\ &= \sum_{i \in \mathcal{N}} x_{s_i^*}^* \log \frac{x_{is_i^*}^*}{x_{is_i^*}} \\ &= - \sum_{i \in \mathcal{N}} \log x_{is_i^*} \\ &= - \sum_{i \in \mathcal{N}} \log(1 - (1 - x_{is_i^*}^*)) \\ &= - \sum_{i \in \mathcal{N}} \log(1 - \|x_i - x_i^*\|/2) \geq \frac{1}{2} \|x - x^*\|, \end{aligned} \quad (5.37)$$

so it suffices to show that  $D_{\text{KL}}(x^*, x(t)) \rightarrow 0$ . With this in mind, let  $D_t = D_{\text{KL}}(x^*, x(t))$ . Then, Proposition (5.8) yields

$$D_{t+1} \leq D_t + \gamma_t \langle v(x(t)) | x(t) - x^* \rangle + \gamma_t \psi_t + \frac{1}{2} \gamma_t^2 \|\hat{v}(t)\|_\infty^2, \quad (5.38)$$

where we have set  $\psi_t = \langle \xi(t) + v(s(t)) - v(x(t)) | x(t) - x^* \rangle$ . Using this bound, we will show that  $x(t)$  visits any neighborhood  $U$  of  $x^*$  infinitely many times.

Indeed, assume on the contrary that this is not the case, so for  $t$  large enough there is a  $\alpha > 0$ , such that  $\|x(t) - x^*\| > \alpha$ . Then, by Proposition (5.1), there exists some  $\delta > 0$  such that  $\langle v(x(t)) | x(t) - x^* \rangle \leq -\alpha\delta$  for all sufficiently large  $t$ . Hence,

telescoping (5.38) yields

$$\begin{aligned} D_{t+1} &\leq D_1 - \alpha\delta \sum_{j=1}^t \gamma_j + \sum_{j=1}^t \gamma_j \psi_j + \frac{1}{2} \sum_{j=1}^t \gamma_j^2 \|\hat{v}(j)\|_\infty^2 \\ &\leq D_1 - \sum_{j=1}^t \gamma_j \left[ \alpha\delta - \frac{\sum_{j=1}^t \gamma_j \psi_j}{\sum_{j=1}^t \gamma_j} - \frac{\sum_{j=1}^t \gamma_j^2 \|\hat{v}(j)\|_\infty^2}{2 \sum_{j=1}^t \gamma_j} \right]. \end{aligned} \quad (5.39)$$

Given that

$$\begin{aligned} \mathbb{E}[\psi_j | \mathcal{F}_{j-1}] &= \mathbb{E}[\langle \xi(t) + v(s(t)) - v(x(t)) | x(j) - x^* \rangle | \mathcal{F}_{j-1}] \\ &= \langle \mathbb{E}[\xi_j | \mathcal{F}_{j-1}] | x(j) - x^* \rangle + \langle \mathbb{E}[v(s(t)) | \mathcal{F}_{j-1}] - v(x(t)) | x(j) - x^* \rangle = 0 \end{aligned}$$

and

$$\begin{aligned} \mathbb{E}[\psi_j^2 | \mathcal{F}_{j-1}] &\leq 2\mathbb{E}[\|\xi_j\|_\infty^2 | \mathcal{F}_{j-1}] + 2\mathbb{E}[\|v(s(j)) - v(x(j))\|_\infty^2 | \mathcal{F}_{j-1}] \\ &\leq 2\sigma^2 + 8\max_{s \in \mathcal{S}} \|v(x(j))\|_\infty = 2\sigma^2 + 8D^2 \end{aligned}$$

according to Hypothesis (H2). It follows that  $\sum_{t=1}^\infty \frac{\mathbb{E}[\|\gamma_t \psi_t\|^2 | \mathcal{F}_{t-1}]}{(\sum_{j=1}^t \gamma_j)^2} \leq \sum_{t=1}^\infty \frac{\gamma_t^2 (2\sigma^2 + 8D^2)}{t^2 \gamma_t^2} < \infty$

Hence, by Theorem (5.3), it follows that  $\frac{\sum_{j=1}^t \gamma_j \psi_j}{\sum_{j=1}^t \gamma_j} \rightarrow 0$  (a.s.). Likewise, if we let  $S_t = \sum_{j=1}^t \gamma_j^2 \|\hat{v}(j)\|_\infty^2$ , we get  $\mathbb{E}[S_t] = \mathbb{E}[\mathbb{E}[S_t | \mathcal{F}_t]] \leq L^2 \sum_{j=1}^t \gamma_j^2 \leq \Gamma_2 L^2$ , where  $\Gamma_2 = \sum_{j=1}^\infty \gamma_j^2$ . Hence, by Doob's martingale convergence theorem [105, Theorem 2.5],  $S_t$  converges to some (random) finite value (a.s.). Combining the above, we conclude that the term in the brackets of (5.39) converges to  $\alpha\delta$  (a.s.). In turn, this implies that  $D_{t+1} \rightarrow -\infty$ , a contradiction as the KL divergence is positive.

We have thus shown that  $x(t)$  visits infinitely many times every neighborhood  $U$  of  $x^*$  – and hence, in particular, the neighborhood  $U_{2M}$  defined in Proposition (5.5). Since  $x(t)$  remains in  $U_{2M}$  with positive probability, it follows that the probability that  $x(t)$  exits  $U_{2M}$  infinitely many times is zero. We thus get  $x(t) \in U_{2M}$  for all  $t$  greater than some random (but finite)  $t_0$ ; hence, telescoping

$$y_{is_i}(t+1) - y_{is_i^*}(t+1) = y_{is_i}(t) - y_{is_i^*}(t) + \gamma_t [v_{is_i}(x(t)) - v_{is_i^*}(x(t))] + \gamma_t \zeta_{is_i}(t), \quad (5.40)$$

where  $\zeta_{is_i}(t) = \hat{v}_{is_i}(t) - v_{is_i}(x(t)) - (\hat{v}_{is_i^*}(t) - v_{is_i^*}(x(t)))$ . Furthermore,  $x^*$  is a pure equilibrium so there exist  $a > 0$  and a neighborhood  $U$  of  $x^*$  in  $\mathcal{X}$  such that such that for all  $s_i \in \mathcal{S}_i \setminus \{s_i^*\}$ ,  $u_i(s_i^*; x_{-i}) - u_i(s_i; x_{-i}) \geq a$ . By Proposition (5.5) there is a  $M > 0$  such that  $U_{2M} \subset U$ .

$$y_{is_i}(t+1) - y_{is_i^*}(t+1) \leq -2M - \sum_{j=t_0}^t \gamma_j \left[ a - \left( \sum_{j=t_0}^t \gamma_j \right)^{-1} \sum_{j=t_0}^t \gamma_j \zeta_{is_i}(j) \right]. \quad (5.41)$$

Just as in the proof of Theorem (5.4), the Theorem (5.3) shows that the term in the brackets of (5.41) converges to  $a$ . Our claim then follows as in the proof of Theorem (5.4) to show that  $x_{is_i}(t) = \mathcal{O}(\exp(-c \sum_{j=1}^t \gamma_j))$  if  $s_i \neq s_i^*$ .  $\square$

**Corollary 5.10.** *With assumptions as above, if (HEDGE) is run with a step-size of the form  $\gamma_t = \gamma/t^\beta$  some  $\beta \in (1/2, 1)$ , we have  $\|x(t) - x^*\| = \mathcal{O}(e^{-\frac{\alpha\gamma}{1-\beta} t^{1-\beta}})$ .*



### 5.2.3.3 Discussion

The step-size assumption in the statement of Theorem (5.9) is key in achieving this, but it is important to note it can be relaxed to the lighter requirement  $\sum_{j=1}^t \gamma_j^2 / \sum_{j=1}^t \gamma_j \rightarrow 0$  if the players' feedback noise is bounded (for instance, if players have access to their actual pure payoff information). When this is the case, it is possible to achieve a convergence rate of the form  $\mathcal{O}(e^{-ct^{1-\beta}})$  for any  $\beta > 0$  by using a step-size sequence of the form  $\gamma_t \propto 1/t^\beta$ . Finally, we should also note that the multiplicative constant in (5.36) is  $\mathcal{O}(\sum_{i \in \mathcal{N}} |\mathcal{S}_i|)$ , i.e. it is linear in the dimensionality of the game (just as in the case of Theorem (5.6)). As for the constant  $c > 0$ , our proof shows that it can be chosen in the same way as the respective coefficient of Theorem (5.4), showing that the rate of elimination of dominated strategies is the same as that of convergence to pure equilibria.

## 5.3 Global convergence in generic potential games

In this section we will only study potential games.

**Definition 5.1** (Potential Game). A game is called a *continuous potential game* if there exists a  $\mathcal{C}^1$  function  $F : K \rightarrow \mathbb{R}$  such that for all  $i, s$  and  $x$ ,

$$\frac{\partial F}{\partial x_{is}}(x) = u_i(s; x). \quad (5.42)$$

Here we prove that algorithm (HEDGE) ran on a potential game with Hypotheses (H1) and (H2') (see below) and condition on the step-size  $\gamma(t) = \frac{\gamma}{t^\beta}$  with  $\beta \in [\frac{1}{2}, 1]$  converges to a NE of the game.

Our proof is based on Benaim's study of stochastic approximations [113]. It follows 4 points:

1. Show that  $x$  is an asymptotic pseudo trajectory of a continuous dynamics (Theorem (5.11)),
2. Show that the potential function of the game is strict Lyapunov function of the dynamics (Theorem (5.12)),
3. Rest point of the dynamics are restricted NE (Corollary (5.13)),
4. Show that if  $x$  converges toward a point it is a NE (Theorem (5.14)).

We need a stronger control on the variance,<sup>3</sup> such that:

$$\mathbb{P}[\|\hat{v}(t)\|_\infty^2 \geq z | \mathcal{F}_{t-1}] = \mathcal{O}(\frac{1}{z^q}) \text{ for all } t = 1, 2, \dots \text{ with } q > 2. \quad (\text{H2}')$$

Hypothesis (H2') is stronger than (5.13b) indeed, it implies that

$$\mathbb{P}[\|\hat{v}(t)\|_\infty^2 = z | \mathcal{F}_{t-1}] \leq \frac{K}{z^q} \text{ for } z \text{ big enough}$$

Therefore  $\mathbb{E}[\|\hat{v}(t)\|_\infty^2 | \mathcal{F}_{t-1}] \leq z + K \int_z^\infty \frac{t}{t^q} dt = z + \frac{1}{(2-q)z^{q-1}} < \infty$ .

As Hypothesis (H2) and (5.13b), Hypothesis (H2') implies the same kind of bound for the noise variance.  $\|\xi(t)\|_\infty^2 \leq 2\|\hat{v}(t)\|_\infty^2 + 2\|v(t)\|_\infty^2$ , as the game is finite,  $\|v(t)\|_\infty^2 \leq$

<sup>3</sup> This is important to show that  $x$  is an asymptotic pseudo trajectory of a continuous dynamics, in particular to show that  $x$  is an approximate Robbins-Monro algorithm.

D. Therefore

$$\mathbb{P}[\|\xi(t)\|_\infty^2 \geq 2z + 2D|\mathcal{F}_{t-1}] \leq \mathbb{P}[\|\hat{v}(t)\|_\infty^2 \geq 2z|\mathcal{F}_{t-1}] = \mathcal{O}\left(\frac{1}{(2z)^q}\right) \text{ for all } t = 1, 2, \dots \quad (5.43)$$

### 5.3.1 Asymptotic Pseudotrajectory

We start with the first step and show that  $x(t)$  defined as the interpolated process of the sequence  $(x(n))_{n \in \mathbb{N}}$  in algorithm (HEDGE) is an asymptotic pseudotrajectory of a continuous dynamic.

**Definition 5.2** (Asymptotic Pseudotrajectories). Given a flow  $\phi : \mathbb{R} \times M \rightarrow M, (t, x) \rightarrow \phi(t, x) = \phi_t(x)$  such that  $\phi_0 = \text{Identity}$  and  $\phi_{t+s} = \phi_t \circ \phi_s$ , a continuous function  $X : \mathbb{R} \rightarrow M$  is an asymptotic pseudotrajectory if

$$\lim_{t \rightarrow \infty} \sup_{0 \leq h \leq T} d(X(t+h), \phi_h(X(t))) = 0 \text{ for any } T > 0$$

Thus for  $t$  large enough, the curve  $[0, T] \rightarrow \mathcal{M} : h \rightarrow X(t+h)$  is as close as we want to the  $\phi$ -trajectory of the point  $X(h)$  over the interval  $[0, T]$ .

**Theorem 5.11.** Assume that Hypotheses (H1) and (H2') are true and if the step size is of the form  $\gamma(t) = \frac{\gamma}{t^\beta}$  with  $\beta \in [\frac{1}{2}, 1]$ , the interpolated process of the sequences  $(x(n))_{n \in \mathbb{N}}$  is an asymptotic pseudo trajectory of the solutions of the following ordinary differential equation<sup>4</sup>

$$\dot{x}_{is}(t) = x_{is}(t) \left( v_{is}(x(t)) - \sum_{s' \in \mathcal{S}_i} v_{is'}(x(t)) x_{is'}(t) \right) \quad (5.44)$$

*Proof.* First, we will check whether the stochastic process  $(x(n))_{n \in \mathbb{N}}$  given by (HEDGE) is an approximate Robbins-Monro algorithm. Note that  $x_{is} = \Lambda_{is}(y_i) = \frac{\exp(y_{is})}{\sum_{s \in \mathcal{S}_i} \exp(y_{is})}$ ,

$\frac{\partial \Lambda_{is}(y_i)}{\partial y_{is'}} = x_{is}(\mathbb{1}_{s=s'} - x_{is'})$ , and

$$\frac{\partial^2 \Lambda_{is}(y_i)}{\partial y_{is'} \partial y_{is''}} = x_{is}(\mathbb{1}_{s=s'=s''} - \mathbb{1}_{s=s'} x_{is''} - x_{is'}(\mathbb{1}_{s=s''} + \mathbb{1}_{s'=s''} - 2x_{is''}))$$

Using Taylor's Remainder Theorem, we obtain:

$$\begin{aligned} x_{is}(n+1) &= \Lambda_{is}(y_i(n+1)) \\ &= \Lambda_{is}(y_i(n) + \gamma_n \hat{v}_i(n)) \\ &= \Lambda_{is}(y_i(n)) + \gamma_n \left( \nabla \Lambda_{is}^T(y_i(n)) \hat{v}_i(n) + \frac{1}{2} \gamma_n \hat{v}_i^T(n) \text{Hess} \Lambda_{is}(\psi_i(n)) \hat{v}_i(n) \right) \\ &= x_{is}(n) + \gamma_n \left( \nabla \Lambda_{is}^T(y_i(n)) \hat{v}_i(n) + \frac{\gamma_n}{2} \hat{v}_i^T(n) \text{Hess} \Lambda_{is}(\psi_i(n)) \hat{v}_i(n) \right) \end{aligned}$$

where  $\nabla \Lambda_{is}$  is the gradient vector of  $\Lambda_{is}$ ,  $\nabla \Lambda_{is}^T$  is its transposed,  $\text{Hess} \Lambda_{is}$  is the Hessian matrix of  $\Lambda_{is}$ , and  $\psi_i(n)$  is in the line segment going out from  $y_i(n)$  to the point  $y_i(n+1)$ .

<sup>4</sup>This is also known as the Replicator Dynamics.

$$\begin{aligned}
x_{is}(n+1) &= x_{is}(n) \\
&+ \gamma_n \left( x_{is}(n)(\hat{v}_{is}(n) - \sum_{s' \in \mathcal{S}_i} x_{is'}(n)\hat{v}_{is'}(n)) + \frac{\gamma_n}{2} \hat{v}_i^T(n) \text{Hess} \Lambda_{is}(\psi_i(n)) \hat{v}_i(n) \right) \\
&= x_{is}(n) + \gamma_n \left( x_{is}(n)(\hat{v}_{is}(n) - \sum_{s' \in \mathcal{S}_i} x_{is'}(n)\hat{v}_{is'}(n)) + \gamma_n a_n \right)
\end{aligned}$$

where  $a_n = \frac{1}{2} \hat{v}_i^T(n) \text{Hess} \Lambda_{is}(\psi_i(n)) \hat{v}_i(n)$ .

By definition of  $\frac{\partial^2 \Lambda_{is}(y_i)}{\partial y_{is'} \partial y_{is''}}$ , all components of  $\text{Hess} \Lambda_{is}(\psi_i(n))$  are bounded. Let  $E_{is,n}$  be the event  $\|\hat{v}_{is}(n)\|^2 \geq n^\alpha$  for  $\frac{1}{q} < \alpha < \frac{1}{2}$ , with  $q$  defined in Hypothesis (H2'). Hypothesis (H2') gives us that

$$\sum_{n=0}^{\infty} \mathbb{P}(E_{is,n}) = \sum_{n=0}^{\infty} \mathbb{P}(\|\hat{v}_{is}(n)\|^2 \geq n^\alpha | \mathcal{F}_{n-1}) = \sum_{n=0}^{\infty} \mathcal{O}\left(\frac{1}{n^{q\alpha}}\right) < \infty \quad (5.45)$$

The Borel-Cantelli theorem gives us that  $E_{is,n}$  is true for only a finite number of  $n \in \mathbb{N}$ . Therefore for  $n > \max_{m \in \mathbb{N}} \{m; \exists i \in \mathcal{N}, s \in \mathcal{S}_i, E_{is,m} \text{ is true}\}$ ,  $\|\hat{v}_{is}(n)\|^2 < n^\alpha$ . By hypothesis  $\gamma_n = o(n^b)$  for any  $b > -1/2$ . In particular,  $\gamma_n = o(n^{-\alpha})$  so  $\lim_{n \rightarrow \infty} a_n \gamma_n = 0$ .

$$\begin{aligned}
x_{is}(n+1) &= x_{is}(n) + \gamma_n \left( x_{is}(n)(v_{is}(x(n)) - \sum_{s' \in \mathcal{S}_i} x_{is'}(n)v_{is'}(x(n))) \right) \\
&+ \gamma_n \left( x_{is}(n)(\hat{v}_{is}(n) - v_{is}(x(n)) - \sum_{s' \in \mathcal{S}_i} x_{is'}(n)[\hat{v}_{is'}(n) - v_{is'}(x(n))]) + \gamma_n a_n \right)
\end{aligned}$$

Let  $U_{i,n} = x_{is}(n)(\hat{v}_{is}(n) - v_{is}(x(n)) - \sum_{s' \in \mathcal{S}_i} x_{is'}(n)[\hat{v}_{is'}(n) - v_{is'}(x(n))])$ , using (5.13a) and (5.13b), we get

1.  $\mathbb{E}[U_{i,n} | \mathcal{F}_{n-1}] = 0$  for all  $n$
2.  $\mathbb{E}[\|U_{i,n}\|^2] < \infty$  for all  $n$

Remark 4.5 and propositions 4.2 and 4.1 of [113] allow us to conclude the proof (theses propositions and remark are written in Appendix B).  $\square$

### 5.3.2 Global convergence in the continuous dynamics

In this subsection,  $x(t)$  is not anymore the discret variable defined in the algorithm but a solution of the dynamics (5.44). We are now going to show that it converges to rest points of the dynamics (5.44). Let  $F$  be the potential function.

**Definition 5.3.** Given a flow  $\phi$ , a continuous function  $F : M \rightarrow \mathbb{R}$  is called a strict increasing *Lyapounov* function if  $F(\phi_t(y))$  weakly increases for all  $y$  and is constant if and only if  $y$  is a rest point of  $\phi$ .

**Theorem 5.12.** *The potential function  $F$  of the game is a strict increasing Lyapounov function of the flow inducted by the dynamics (5.44).*

*Proof.* We consider the variation of  $F$ . We have

$$\begin{aligned}
\dot{F}(x) &= \sum_{i \in \mathcal{N}} \sum_{s \in \mathcal{S}_i} \frac{\partial F}{\partial x_{is}}(x) \dot{x}_{is} \\
&= \sum_{i \in \mathcal{N}} \sum_{s \in \mathcal{S}_i} v_{is}(x) x_{is} \left( v_{is}(x) - \sum_{s' \in \mathcal{S}_i} v_{is'}(x) x_{is'} \right) \\
&= \sum_{i \in \mathcal{N}} \sum_{s \in \mathcal{S}_i} \sum_{s' \in \mathcal{S}_i} v_{is}(x) x_{is} (v_{is}(x) x_{is'} - v_{is'}(x) x_{is'}) \\
&= \sum_{i \in \mathcal{N}} \sum_{s \in \mathcal{S}_i} \sum_{s' \in \mathcal{S}_i} x_{is} x_{is'} (v_{is}(x)^2 - v_{is}(x) v_{is'}(x)) \\
&= \sum_{i \in \mathcal{N}} \sum_{s \in \mathcal{S}_i} \sum_{\substack{s' \in \mathcal{S}_i \\ s' > s}} x_{is} x_{is'} [v_{is}(x) - v_{is'}(x)]^2
\end{aligned}$$

So, we can conclude that  $\dot{F}(x) \geq 0$ .

$\dot{F}(x) = 0$  on all rest point of the dynamic.  $\dot{F}(x) = 0$  implies  $\forall i \in \mathcal{N}, \forall s, s' \in \mathcal{S}_i, x_{is} = 0$  or  $x_{is'} = 0$  or  $v_{is}(x) = v_{is'}(x)$ .

$\dot{x}_{is} = x_{is} (v_{is}(x) - \sum_{s' \in \mathcal{S}_i} v_{is'}(x) x_{is'})$  so if  $x_{is} = 0$  then  $\dot{x}_{is} = 0$ . Otherwise,  $v_{is}(x) = v_{is'}(x)$  for all  $s' \in \mathcal{S}_i$  such that  $x_{is} \neq 0$  and  $x_{is'} \neq 0$ , we have then  $\dot{x}_{is} = x_{is} (v_{is}(x) - v_{is}(x) \sum_{\substack{s' \in \mathcal{S}_i \\ x_{is'} \neq 0}} x_{is'}) = 0$

To conclude,  $F$  is increasing and its derivative is null if and only if its evaluated on a rest point of the dynamics (5.44).  $F$  is a strict increasing Lyapunov function of the dynamics (5.44).  $\square$

**Corollary 5.13.** *Rest points of the dynamics are restricted NE*

*Proof.* The dynamics (5.44) admits a strict increasing Lyapunov function. We are now going to show that the rest points of the dynamics correspond to NE. Going back to the Lyapunov proof,  $x(t)$  is a rest point of the dynamics if and only if  $\dot{F}(x) = \sum_{i \in \mathcal{N}} \sum_{s \in \mathcal{S}_i} \sum_{s' \in \mathcal{S}_i \wedge s' > s} x_{is} x_{is'} [v_{is}(x) - v_{is'}(x)]^2 = 0$ . That condition is equivalent to  $\forall i \in \mathcal{N}, \forall s, s' \in \text{supp}(x_i), v_{is}(x) = v_{is'}(x)$ .  $x$  respects that condition if and only if it is a NE of the game restricted to the support of  $x$ , that is to say  $x$  is a restricted NE.

Therefore the only rest points of the dynamics are restricted NE.  $\square$

### 5.3.3 Global convergence in the potential game

In this section,  $(x(n))_{n \in \mathbb{N}}$  is the discret process defined in algorithm (HEDGE).

In finite generic potential game the number of NE is finite [114] and isolated. The number of restricted game of a finite game is finite, so the number of restricted NE is finite. We can apply Corollary 6.6 of [113] (see Appendix B) to show that the discret process  $x(n)$  induced by (HEDGE) converges to a rest point of the dynamics (5.44) which are restricted NE.

We now prove that if  $x(n)$  converges it is only toward a non-restricted NE.

**Theorem 5.14.** *Assume that Hypotheses (H1) and (H2') are true, if the step size is of the form  $\gamma(t) = \frac{\gamma}{t^\beta}$  with  $\beta \in [\frac{1}{2}, 1]$ , and the game is a potential game then  $x(n)$ , defined in (HEDGE), converges to a NE*

*Proof.* We proved that the flow induced by the dynamics (5.44) admits a strict Lyapunov function and has countable rest points. We can apply Corollary 6.6 of [113] that gives  $x(n)$  converges to a restricted NE.

The last step is now to prove that if  $x(n) \rightarrow x^*$  then  $x^*$  is a NE.

We show by contradiction that  $(x(n))_{n \in \mathbb{N}}$  converges to  $x^*$  a NE. Assume that  $x^*$  is not a NE.

$\exists i \in \mathcal{N}, \exists s' \in \mathcal{S}_i, s' \notin \text{supp}(x_i^*), s.t., v_{is'}(x^*) > v_{is}(x^*), \forall s \in \text{supp}(x_i^*)$ .

By continuity of  $u$ , there is a neighborhood  $U$  of  $x^*$  and  $a > 0$  such that:

$\exists i \in \mathcal{N}, \exists s' \in \mathcal{S}_i, s' \notin \text{supp}(x_i^*), s.t., v_{is'}(x) - v_{is}(x) > a, \forall s \in \text{supp}(x_i^*), x \in U$ .

Using Lemma (5.2), for  $n_0$  big enough and  $n \geq n_0$ :

$$y_{is'}(n) - y_{is}(n) \geq C + b \sum_{t=n_0}^{n-1} \gamma_t \rightarrow \infty \quad (5.46)$$

Thus  $\frac{x_{is'}(n)}{x_{is}(n)} = \exp(y_{is'}(n) - y_{is}(n)) \rightarrow \infty$

$x_{is}(n) \rightarrow 0$  and  $s$  is not in the support of  $x_i^*$  which is a contradiction.

Therefore  $x^*$  is a NE.  $\square$

### 5.3.4 Convergence Rate

**Assumption 5.1.** Assume that, except for a null set of initial condition,  $x(n)$  converges to a pure Nash equilibrium.

We did not prove this assumption but we intuitively consider that it is true. Figure 5.1 illustrates the evolution of mixed strategy in a two players game. Each player has two strategies ( $s_1$  and  $s_2$ ) and players share the same utility functions.

Any initial conditions give a trajectory that converges towards a pure NE: either  $(s_1, s_1)$  or  $(s_2, s_2)$  except for the diagonal between  $(s_1, s_2)$  and  $(s_2, s_1)$ . Indeed, initial conditions in the diagonal between  $(s_1, s_2)$  and  $(s_2, s_1)$  give a trajectory that converges towards the mixed equilibria.

**Theorem 5.15.** Under the same assumption, if  $x(n)$  converges to  $x^* = (s_1^*, \dots, s_N^*)$ . Then, for some constants  $a > 0$  and  $C > 0$

$$1 - x_{s^*}(n) \leq C \exp\left(-\sum_{k=n_0}^{n-1} \gamma_k a\right)$$

*Proof.* By continuity of  $u$ , there is a neighborhood  $U$  of  $x^*$  and  $a' > 0$  such that:

$\forall i \in \mathcal{N}, \forall s' \in \mathcal{S}_i, s' \neq s_i^*, v_{is_i^*}(x) - v_{is'}(x) > a', x \in U$ . Using Lemma (5.2) for  $n_0$  big enough:

$$y_{is^*}(n) - y_{is}(n) \geq C + b \sum_{t=n_0}^{n-1} \gamma_t$$

So, by computation, we can deduce that

$$\sum_{s \in \mathcal{S}_i, s \neq s^*} \exp(y_{is}(n) - y_{is^*}(n)) \leq \sum_{s \in \mathcal{S}_i, s \neq s^*} \exp(-C_{n_0} - \sum_{k=n_0}^{n-1} \gamma_k a) \leq C \exp\left(-\sum_{k=n_0}^{n-1} \gamma_k a\right)$$

where we set  $C = |\mathcal{S}_i| \exp(-C_{t_0})$ . Now, we will focus on  $x_{s^*}(n)$ .

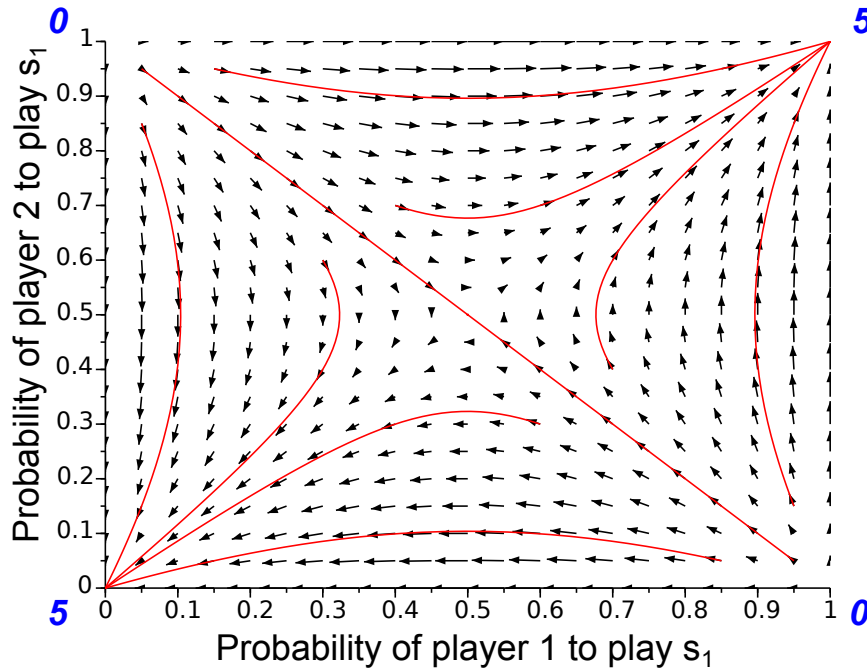


FIGURE 5.1: Example of a flow and a few trajectories. Player 1 and 2 have each two strategies  $s_1$  and  $s_2$ , the x-axis shows the probability of Player 1 to play its  $s_1$  and the y-axis shows the probability of Player 2 to play its  $s_1$ . The utility are written in blue  $u(s_1, s_1) = u(s_2, s_2) = 5$  and  $u(s_1, s_2) = u(s_2, s_1) = 0$ . All initial conditions give a trajectory that converges toward  $(s_1, s_1)$  or  $(s_2, s_2)$ , except for the diagonal between  $(s_1, s_2)$  and  $(s_2, s_1)$  which leads to a trajectory that converged toward  $([\frac{1}{2}s_1, \frac{1}{2}s_2], [\frac{1}{2}s_1, \frac{1}{2}s_2])$ . Figure done with [115].

$$\begin{aligned} x_{s^*}(n) &= \frac{\exp(y_{is^*}(n))}{\sum_{s \in \mathcal{S}} \exp(y_{is}(n))} = \frac{1}{\sum_{s \in \mathcal{S}_i} \exp(y_{is}(n) - y_{is^*}(n))} \\ &= \frac{1}{1 + \sum_{s \in \mathcal{S}_i, s \neq s^*} \exp(y_{is}(n) - y_{is^*}(n))} \end{aligned}$$

$$x_{s^*}(n) \geq \frac{1}{1 + C \exp(-\sum_{k=n_0}^n \gamma_k a)}$$

Since for any  $z > 0$ ,  $\frac{1}{1+z} \geq 1 - z$ , we obtain

$$1 - x_{s^*}(n) \leq C \exp(-\sum_{k=n_0}^{n-1} \gamma_k a)$$

(For example If  $\gamma_n = \frac{1}{\sqrt{n}}$ , then  $\ln(1 - x_{s^*}(n)) = \Omega(\sqrt{n})$ )

□

## 5.4 Conclusions

We showed that, in generic games, EW algorithms have the rational property of eliminating dominated strategies at exponential rate. In addition, the mixed strategy profile converges toward NE if the NE is unique or if the initial conditions are close enough.

Then, by restricting our analysis to potential games and using a stronger hypothesis on the noise control, we showed that mixed strategy profile converges toward a NE regardless of initialization.

The HEDGE algorithm is an example of dual-averaging regularized algorithms. The next chapter study the convergence of mixed strategy profile for any multi-player dual-averaging regularized algorithm.

## Chapter 6

# Extension to regularized learning and partial information

### Contents

<b>6.1</b>	<b>Dual-averaging regularized learning</b>	<b>87</b>
6.1.1	Preliminaries	87
6.1.1.1	Penalty functions	87
6.1.1.2	Algorithm	89
6.1.2	Convergence in potential games	89
6.1.2.1	Asymptotic Pseudotrajectory	90
6.1.2.2	Global convergence with the continuous dynamics	91
6.1.3	Global convergence in the potential game	93
6.1.4	Convergence time	93
<b>6.2</b>	<b>Bandit setting</b>	<b>94</b>
6.2.1	$\epsilon$ -dynamic and $\epsilon$ -equilibrium	94
6.2.2	Global convergence	96
6.2.2.1	Asymptotic pseudo trajectory	96
6.2.2.2	Global convergence with the continuous dynamics	97
6.2.3	Global convergence in the potential game	99
6.2.4	Convergence time	99
<b>6.3</b>	<b>Conclusions</b>	<b>100</b>

The previous chapter shows convergence guarantees with a particular algorithm called HEDGE and a full information payoff. In this chapter we show that the previous results in potential games can be extended to a larger class of algorithms, the dual-averaging regularized algorithms. These algorithms are broadly studied [111], [112], [116], [117] to cite a few, so it is important to extend our previous results to this class of algorithms. We also study algorithms with bandit setting, when a player has only access to the payoff of the chosen strategy.

## 6.1 Dual-averaging regularized learning

### 6.1.1 Preliminaries

#### 6.1.1.1 Penalty functions

The greedy method for choosing a player's best strategy is to choose:

$$x_i(t+1) = \arg \max_{x_i \in \mathcal{X}_i} \langle y_i(t) | x_i(t) \rangle, \quad \text{for all players } i \in \mathcal{N} \quad (6.1)$$

However the maximum of  $\langle y_i | x_i \rangle$  could often be outside  $\mathcal{X}_i$ . And the maximum restricted to  $\mathcal{X}_i$  might be multiple or in the border of  $\mathcal{X}_i$ .



To avoid this problem we introduce a *penalty* function  $h_i : \mathcal{X}_i \rightarrow \mathbb{R}$  and define the *choice map* by:

$$x_i(t+1) = Q_i(y_i(t)) = \arg \max_{x_i \in \mathcal{X}_i} (\langle y_i(t) | x_i \rangle - h_i(x_i)). \quad (6.2)$$

In order to have an unique solution the penalty function needs to be convex, stronger conditions define a penalty function.

**Definition 6.1.** Let  $\mathcal{C}$  be a compact convex subset of a finite-dimensional normed space  $\mathcal{V}$ . We say that  $h : \mathcal{C} \rightarrow \mathbb{R}$  is a *penalty function* (or *regularizer*) on  $\mathcal{C}$  if:

1.  $h$  is continuous,
2.  $h$  is strongly convex, i.e., there exists some  $K > 0$  such that

$$h(tx + (1-t)x') \leq th(x) + (1-t)h(x') - \frac{1}{2}Kt(1-t)\|x' - x\|^2$$

for all  $x, x' \in \mathcal{C}$  and all  $t \in [0, 1]$ .

For simplicity of the proofs, we add four hypotheses that  $h_i$  need to satisfy for each player  $i \in \mathcal{N}$ :

1.  $h_i$  is steep, i.e.  $\|\nabla(h_i(x_i))\| \rightarrow \infty$  when  $x_i \rightarrow \partial\mathcal{X}_i$  where  $\partial\mathcal{X}_i$  is the boundary of  $\mathcal{X}_i$ ,
2.  $h'_i$  is continuous on  $]0, \infty[$ ,
3.  $\|Hess(Q_i)\|_\infty < \infty$ , where  $Hess(Q_i)$  is the Hessian matrix of  $Q_i$ ,
4.  $h_i$  is decomposable, i.e.  $h_i(x_i) = \sum_{s_i \in \mathcal{S}_i} h_{is_i}(x_i)$ .

The first hypothesis makes sure that  $x$  is always in the interior of  $\mathcal{X}$  although it can convergence to the boundary where pure equilibria belong.

The second hypothesis also implies that  $\|Jac(Q_i)\|_\infty < \infty$ , where  $Jac(Q_i)$  is the Jacobian matrix of  $Q_i$ .

We believe that the first two hypotheses are not mandatory but we did not do the proves without theses hypotheses.

We start by presenting a possible penalty function.

**Gibbs entropy and the logit map** A classical example, studied in Chapter 5, is the *Gibbs entropy*:

$$h_i(x) = \sum_{s_i \in \mathcal{S}_i} x_{is_i} \log(x_{is_i}) \quad \text{for all } i \in \mathcal{N}. \quad (6.3)$$

$h_i$  verify all the conditions: it is continuous decomposable and strongly convex (see Appendix A.1.2) and  $\nabla(h_i(x_i))_{s_i} = 1 + \log(x_{is_i}) \rightarrow_{x_{is_i} \rightarrow 0} \infty$  so  $h_i$  is steep.

As shown in Appendix A, the corresponding choice map is the previously studied *logit map*:

$$Q_i(y(t)) = \left( \frac{\exp(y_{is}(t))}{\sum_{s' \in \mathcal{S}_i} \exp(y_{is'}(t))} \right)_{s \in \mathcal{S}_i}$$

Let  $x_{is} = Q_{is}(y_i) = \frac{\exp(y_{is})}{\sum_{s' \in \mathcal{S}_i} \exp(y_{is'})}$ . We have  $\frac{\partial Q_{is}(y_i)}{\partial y_{is'}} = x_{is}(\mathbf{1}_{s=s'} - x_{is'})$ , and  $\frac{\partial^2 Q_{is}(y_i)}{\partial y_{is'} \partial y_{is''}} = x_{is}(\mathbf{1}_{s=s'=s''} - \mathbf{1}_{s=s'}x_{is''} - x_{is'}(\mathbf{1}_{s=s''} + \mathbf{1}_{s'=s''} - 2x_{is''}))$ .

Therefore  $Q_i$  also verifies its condition as  $|HessQ_{i,ss'}| \leq 3$ .

### 6.1.1.2 Algorithm

The dual-averaging regularized algorithm can be written as follows:

**Algorithm 6.1** Dual-averaging regularized algorithm with variable step-size  $\gamma_t$

- 1 Each player  $i \in \mathcal{N}$  has an initial score vector  $y_i(1)$  and plays with initial mixed strategy  $x_i(1) = Q_i(y_i(1))$  where the choice map  $Q_i$  is defined from the penalty function as

$$Q_i(y_i(n)) = \arg \max_{x_i \in \mathcal{X}_i} (\langle y_i(n) | x_i(n) \rangle - h_i(x_i(n))). \quad (6.4)$$

- 2 **for** each round  $n$
- 3     Each player  $i \in \mathcal{N}$  draws a pure strategy  $s_i(n)$  according to  $x_i(n)$
- 4     Each player  $i \in \mathcal{N}$  gets an estimate  $\hat{v}_i(n)$  of their payoff vector  $v_i(s(n))$
- 5     Each player  $i \in \mathcal{N}$  updates their mixed strategy  $x_i$  via the recursion

$$\begin{aligned} y_i(n+1) &= y_i(n) + \gamma_n \hat{v}_i(n), \\ x_i(n+1) &= Q_i(y_i(n+1)) \end{aligned} \quad (\text{DUAL AVERAGING})$$

**end for**

We keep the same noise model as in the previous chapter:

$$\hat{v}_i(n) = v_i(s(n)) + \xi_i(n), \quad (6.5)$$

where the error process  $\xi = (\xi_i)_{i \in \mathcal{N}}$  satisfies the statistical hypotheses

1. *Zero-mean:*

$$\mathbb{E}[\xi(n) | \mathcal{F}_{n-1}] = 0 \quad \text{for all } n = 1, 2, \dots \text{ (a.s.)}. \quad (\text{H1})$$

2. *Finite MSE:* there exists some  $\sigma > 0$  such that

$$\mathbb{E}[\|\xi(n)\|_\infty^2 | \mathcal{F}_{n-1}] \leq \sigma^2 \quad \text{for all } n = 1, 2, \dots \text{ (a.s.)}. \quad (\text{H2})$$

## 6.1.2 Convergence in potential games

Just like in Chapter 5 to study the global converge to the set of Nash Equilibria, we limit our study to potential games (see Definition 5.1 Chapter 5).

We also need a stronger control on the variance such that:

$$\mathbb{P}[\|\hat{v}(n)\|_\infty^2 \geq z | \mathcal{F}_{n-1}] = \mathcal{O}\left(\frac{1}{z^q}\right) \text{ for all } n = 1, 2, \dots \text{ with } q > 2. \quad (\text{H2}')$$

We show here that we can follow the same steps as with algorithm HEDGE.

1. Show that  $x$  is an asymptotic pseudo trajectory of a continuous dynamic,
2. Show that the potential function of the game is strict Lyapunov function of the continuous dynamic,
3. Show that  $x$  converges toward a rest point of the continuous dynamic,
4. Show that if  $x$  converges toward a point it is a Nash Equilibrium.

### 6.1.2.1 Asymptotic Pseudotrajectory

**Theorem 6.1.** *Under Hypotheses (H1) and (H2'), if the step size is of the form  $\gamma(n) = \frac{\gamma}{n^\beta}$  with  $\beta \in [\frac{1}{2}, 1]$ , the interpolated process of the sequences  $(x_i(n))_{n \in \mathbb{N}}$  is an asymptotic pseudo trajectory of the solutions of the following ordinary differential equation*

$$\begin{aligned} \dot{x}_{is}(t) &= \sum_{s' \in \mathcal{S}_i} \nabla(Q_{is})_{s'}(y_i(t))v_{is'}(x(t)) = \nabla Q_{is}^T(y_i(t))v_i(x(t)) \\ \dot{x}_i(t) &= \text{Jac}(Q_i)(y_i(t))v_i(x(t)), \end{aligned} \quad (6.6)$$

where  $\nabla Q_{is}$  is the gradient vector of  $Q_{is}$  and  $\text{Jac}(Q_i)$  the Jacobian matrix of  $Q_i$ .

*Proof.* First, we will check whether the stochastic process  $(x_i(t))$  given by (DUAL AVERAGING) is an approximate Robbins-Monro algorithm. Using Taylor's Remainder Theorem, we obtain:

$$\begin{aligned} x_{is}(n+1) &= Q_{is}(y_i(n+1)) \\ &= Q_{is}(y_i(n) + \gamma_n \hat{v}_i(n)) \\ &= Q_{is}(y_i(n)) + \gamma_n \left( \nabla Q_{is}^T(y_i(n))\hat{v}_i(n) + \frac{1}{2} \gamma_n \hat{v}_i^T(n) \text{Hess}(Q_{is})(\psi_i(n))\hat{v}_i(n) \right) \\ &= x_{is}(n) + \gamma_n \left( \nabla Q_{is}^T(y_i(n))\hat{v}_i(n) + \frac{\gamma_n}{2} \hat{v}_i^T(n) \text{Hess}(Q_{is})(\psi_i(n))\hat{v}_i(n) \right) \\ &= x_{is}(n) + \gamma_n (\nabla Q_{is}^T(y_i(n))v_i(x(n)) + \nabla Q_{is}^T(y_i(n))(\hat{v}_i(n) - v_i(x(n))) + \gamma_n a_n) \end{aligned}$$

where  $\nabla Q_{is}$  is the gradient vector of  $Q_{is}$ ,  $\nabla Q_{is}^T$  is its transposed,  $\text{Hess}(Q_{is})$  is the Hessian matrix of  $Q_{is}$ ,  $\psi_i(n)$  is in the line segment going out from  $y_i(n)$  to the point  $y_i(n+1)$ , and  $a_n = \frac{1}{2} \hat{v}_i^T(n) \text{Hess}(Q_{is})(\psi_i(n))\hat{v}_i(n)$ .

By hypotheses all components of  $\text{Hess}(Q_{is})(\psi_i(n))$  are bounded. Let  $E_{is,n}$  be the event  $\|\hat{v}_{is}(n)\|^2 \geq n^\alpha$  for  $\frac{1}{q} < \alpha < \frac{1}{2}$ , with  $q$  defined in Hypothesis (H2'). Hypothesis (H2') gives us that

$$\sum_{n=0}^{\infty} \mathbb{P}(E_{is,n}) = \sum_{n=0}^{\infty} \mathbb{P}(\|\hat{v}_{is}(n)\|^2 \geq n^\alpha | \mathcal{F}_{n-1}) = \sum_{n=0}^{\infty} \mathcal{O}\left(\frac{1}{n^{q\alpha}}\right) < \infty \quad (6.7)$$

The Borel-Cantelli theorem gives us that  $E_{is,n}$  is true for only a finite number of  $n \in \mathbb{N}$ . Therefore for  $n > \max_{m \in \mathbb{N}} \{m; \exists i \in \mathcal{N}, s \in \mathcal{S}_i, E_{is,m} \text{ is true}\}$ ,  $\|\hat{v}_{is}(n)\|^2 < n^\alpha$ . By hypothesis  $\gamma_n = o(n^b)$  for any  $b > -1/2$ . In particular,  $\gamma_n = o(n^{-\alpha})$  so  $\lim_{n \rightarrow \infty} a_n \gamma_n = 0$ .

Let  $U_{i,n} = \nabla Q_{is}^T(y_i(n))(\hat{v}_i(n) - v_i(x(n)))$ , using (5.13a) and (5.13b) (from Hypotheses (H1) and (H2)) we obtain  $\|\nabla Q_{is}^T(y_i(n))\|_\infty < \infty$  for all  $n$  and the fact that  $\nabla Q_{is}^T(y_i(n))$  is  $\mathcal{F}_{n-1}$ -measurable (because the information of  $y(n)$  is included in  $\mathcal{F}_{n-1}$ ), we get

1.  $\mathbb{E}[U_{i,n} | \mathcal{F}_{n-1}] = 0$  for all  $n$
2.  $\mathbb{E}[\|U_{i,n}\|^2] < \infty$  for all  $n$

Remark 4.5 and propositions 4.2 and 4.1 of [113] allow us to conclude the proof (these propositions and remark are written in Appendix B).  $\square$

**Lemma 6.2.** *With the same assumptions, the interpolated process of the sequences  $(y_i(n))_{n \in \mathbb{N}}$  is an asymptotic pseudo trajectory of the solutions of the following ordinary differential equation*

$$\dot{y}_i(t) = v_i(x(t)) \quad (6.8)$$

*Proof.* Recall that

$$y_i(t+1) = y_i(t) + \gamma_t \hat{v}_i(t) = y_i(t) + \gamma_t v_i(x(t)) + \gamma_t (\hat{v}_i(t) - v_i(x(t)))$$

Using Hypotheses (H1) and (H2) in propositions 4.2 and 4.1 of [113] allow us to conclude the proof.  $\square$

### 6.1.2.2 Global convergence with the continuous dynamics

In this subsection,  $x(t)$  is not anymore the discrete variable defined in the algorithm but a solution of the dynamics (6.6).

We are now going to show that  $x(t)$  converges to rest points of (6.6). Let  $F$  be the potential function of the game.

**Theorem 6.3.** *The potential function  $F$  of the game is a strict increasing Lyapunov function (see Definition 5.3) of the flow induced by the dynamics (6.6).*

*Proof.* We consider the variation of  $F$ . We have

$$\begin{aligned} \dot{F}(x(t)) &= \sum_{i \in \mathcal{N}} \frac{\partial F}{\partial x_i}(x(t)) \dot{x}_i(t) \\ &= \sum_{i \in \mathcal{N}} v_i^T(x(t)) \dot{x}_i(t) \\ &= \sum_{i \in \mathcal{N}} \sum_{s \in \mathcal{S}_i} v_{is}(x(t)) \dot{x}_{is}(t) \end{aligned} \quad (6.9)$$

We are now going to find an other expression of  $\dot{x}_i(t)$ , to that end we use the Lagrange multiplier. Recall that  $Q_i(y_i) = \arg \max_{x_i \in \mathcal{X}_i} (\langle y_i(t) | x_i(t) \rangle - h_i(x_i(t)))$ , so  $Q_i$  can be seen as an optimisation of  $\langle y_i(t) | x_i(t) \rangle - h_i(x_i(t))$  when  $\sum_{s \in \mathcal{S}_i} x_{is}(t) = 1$ . For simplicity we drop the "(t)".

$$\mathcal{L}_i(x_i, \lambda_i) = \sum_{s \in \mathcal{S}_i} (x_{is} y_{is} - h_{is}(x_{is})) + \lambda_i \left( \sum_{s \in \mathcal{S}_i} x_{is} - 1 \right) \quad (6.10)$$

By derivation we obtain

$$\frac{\partial \mathcal{L}_i}{\partial \lambda_i} = 0 = \sum_{s \in \mathcal{S}_i} x_{is} - 1 \quad (6.11)$$

and, noting  $h'_{is}(x_{is}) = \frac{dh_{is}}{dx_{is}}(x_{is})$  we have

$$\frac{\partial \mathcal{L}_i}{\partial x_{is}} = 0 = y_{is} - h'_{is}(x_{is}) + \lambda_i, \quad \text{for all } s \in \mathcal{S}_i. \quad (6.12)$$

Taking the derivative again we get

$$\dot{y}_{is} = h''_{is}(x_{is}) \dot{x}_{is} - \dot{\lambda}_i, \quad \text{for all } s \in \mathcal{S}_i. \quad (6.13)$$

Let  $H_i$  be the Hessian matrix of  $h_i$ . As  $h$  is strongly convex, its Hessian matrix is strictly positive and invertible, in particular  $h''_{is}(x_{is}) > 0$ . Let  $d_{is} = h''_{is}(x_{is})$ , we

obtain:

$$\begin{aligned} \dot{x}_{is} &= \frac{\dot{y}_{is}}{d_{is}} + \frac{\dot{\lambda}_i}{d_{is}}, \quad \text{for all } s \in \mathcal{S}_i. \\ 0 &= \sum_{s \in \mathcal{S}_i} \dot{x}_{is} = \sum_{s \in \mathcal{S}_i} \left( \frac{\dot{y}_{is}}{d_{is}} + \frac{\dot{\lambda}_i}{d_{is}} \right) \end{aligned} \quad (6.14)$$

Because  $\sum_{s \in \mathcal{S}_i} x_{is} = 1$ , we can express  $\dot{\lambda}_i$  and  $\dot{x}_{is}$ .

$$\begin{aligned} \dot{\lambda}_i &= - \frac{\sum_{s \in \mathcal{S}_i} \frac{\dot{y}_{is}}{d_{is}}}{\sum_{s \in \mathcal{S}_i} \frac{1}{d_{is}}} \\ \dot{x}_{is} &= \frac{\dot{y}_{is}}{d_{is}} - \frac{1}{d_{is}} \frac{\sum_{s' \in \mathcal{S}_i} \frac{\dot{y}_{is'}}{d_{is'}}}{\sum_{s' \in \mathcal{S}_i} \frac{1}{d_{is'}}} = \sum_{s' \in \mathcal{S}_i} \frac{\dot{y}_{is} - \dot{y}_{is'}}{d_{is'} d_{is} \sum_{s' \in \mathcal{S}_i} \frac{1}{d_{is'}}} \end{aligned} \quad (6.15)$$

Injecting that into (6.9) and using  $\dot{y}_{is}(t) = v_{is}(x(t))$  (Lemma (6.2)):

$$\begin{aligned} \dot{F}(x(t)) &= \sum_{i \in \mathcal{N}} \sum_{s \in \mathcal{S}_i} \sum_{s' \in \mathcal{S}_i} \frac{v_{is}^2(x(t)) - v_{is'}(x(t))v_{is}(x(t))}{d_{is'} d_{is} \sum_{s' \in \mathcal{S}_i} \frac{1}{d_{is'}}} \\ \dot{F}(x(t)) &= \sum_{i \in \mathcal{N}} \sum_{s \in \mathcal{S}_i} \sum_{s' \in \mathcal{S}_i, s' \neq s} \frac{(v_{is}(x(t)) - v_{is'}(x(t)))^2}{d_{is'} d_{is} \sum_{s' \in \mathcal{S}_i} \frac{1}{d_{is'}}} \\ \dot{F}(x(t)) &\geq 0 \end{aligned} \quad (6.16)$$

If  $x(t)$  is a rest point,  $\dot{x}_{is}(t) = 0$  for all  $i \in \mathcal{N}, s \in \mathcal{S}_i$  and  $\dot{F}(x(t)) = 0$  using the Equation (6.9). Let show the reverse, i.e., that if  $\dot{F}(x(t)) = 0$  then  $x(t)$  is a rest point. First we remark that  $\dot{F}(x(t)) = 0$  if each term of the sum is nul. At least one of three conditions needs to be true for a term to be null:

1.  $\frac{1}{d_{is}} = 0$ ,
2.  $\frac{1}{d_{is'}} = 0$ ,
3.  $v_{is}(x(t)) = v_{is'}(x(t))$ .

Recall that  $h$  is steep, its derivative goes to the infinity when reaching the boundary of  $\mathcal{X}$ . Therefore the second derivative of  $h$  also goes to infinity when reaching the boundary and  $d_{is} \rightarrow_{x_{is} \rightarrow 0} \infty$ .

So those three conditions can be gathered in one condition :

$$\dot{F}(x(t)) = 0 \Leftrightarrow \forall i \in \mathcal{N}, v_{is}(x(t)) = v_{is'}(x(t)) \forall s, s' \in \mathcal{S}_i \text{ such that } \frac{1}{d_{is}} \neq 0 \text{ and } \frac{1}{d_{is'}} \neq 0$$

Recall that  $y_{is}(t) = v_{is}(x(t))$  (Lemma (6.2)) so this condition applied to (6.15) gives us that  $\dot{x}_{is}(t) = 0 \quad \forall i \in \mathcal{N}, \forall s, s' \in \mathcal{S}_i$ . Therefore  $\dot{F}(x(t)) = 0$  if and only if  $x(t)$  is a rest point of the dynamics (6.6). □

The corollary (5.13) applies as well in the dual-averaging regularized algorithm because  $d_{is} \rightarrow_{x_{is} \rightarrow 0} \infty$  so rest points of the dynamics are Nash Equilibria. And equilibria are countable and isolated because the potential game is finite and generic.

### 6.1.3 Global convergence in the potential game

In this section,  $x(n)$  is again the discrete variable defined in the algorithm.

Just like for (HEDGE) the Corollary 6.6 of [113] allows to conclude that  $x(n)$  converges to a Nash Equilibrium.

We show now that  $x(n)$  converges to a Nash Equilibrium of the game (not restricted).

**Theorem 6.4.** *If  $x(n) \rightarrow x^*$ ,  $x^*$  is a Nash Equilibrium.*

*Proof.* We show by contradiction that  $(x(n))_{n \in \mathbb{N}}$  converges to  $x^*$  a Nash Equilibrium. Assume that  $x^*$  is not a Nash Equilibrium.

$\exists i \in \mathcal{N}, \exists s' \in \mathcal{S}_i, s' \notin \text{supp}(x_i^*), s.t., v_{is'}(x^*) > v_{is}(x^*), \forall s \in \text{supp}(x_i^*)$ .

By continuity of  $u$ , there is a neighborhood  $U$  of  $x^*$  and  $a > 0$  such that:

$\exists i \in \mathcal{N}, \exists s' \in \mathcal{S}_i, s' \notin \text{supp}(x_i^*), s.t., v_{is'}(x) - v_{is}(x) > a, \forall s \in \text{supp}(x_i^*), x \in U$ .

Using Lemma 5.2, for  $n_0$  big enough and  $n \geq n_0$ :

$$y_{is'}(n) - y_{is}(n) \geq C + b \sum_{t=n_0}^{n-1} \gamma_t \rightarrow \infty \quad (6.17)$$

Thus, using (6.12)  $y_{is'}(n) - y_{is}(n) = h'_{is'}(x_{is'}(n)) - h'_{is}(x_{is}(n))$  we have

$$h'_{is'}(x_{is'}(n)) - h'_{is}(x_{is}(n)) \geq C + b \sum_{t=n_0}^{n-1} \gamma_t$$

And

$$h'_{is'}(x_{is'}(n)) - h'_{is}(x_{is}(n)) \rightarrow_{n \rightarrow \infty} \infty$$

That is a contradiction because  $h'_{is'}(x_{is'}(n)) \rightarrow -\infty$  because  $s'$  is not in the support of  $x^*$  and  $x_{is'}(n) \rightarrow 0$  and  $h'_{is}(x_{is}(n))$  is bounded.

Therefore  $x^*$  is a Nash Equilibrium.  $\square$

### 6.1.4 Convergence time

**Assumption 6.1.** Assume that except for a nullset of initial conditions,  $x(t)$  converges to a pur Nash equilibrium.

**Theorem 6.5.** *Assume that  $x(t) \rightarrow x^* = (s_1^*, \dots, s_N^*)$ . Then,*

$$1 - x_{s_i^*}(n) \leq \sum_{s \in \mathcal{S}_i, s \neq s_i^*} h'^{-1}_{is}(\mathcal{O}(\sum_{k=n_0}^{n-1} \gamma_k))$$

*Proof.* By continuity of  $u$ , there is a neighborhood  $U$  of  $x^*$  and  $a' > 0$  such that:

$\forall i \in \mathcal{N}, \forall s' \in \mathcal{S}_i, s' \neq s_i^*, v_{is'}(x) - v_{is}(x) > a', x \in U$ . Using Lemma 5.2 for  $n_0$  big enough and  $s \in \mathcal{S}_i, s \neq s_i^*$ :

$$y_{is^*}(n) - y_{is}(n) \geq C + b \sum_{t=n_0}^{n-1} \gamma_t$$

So, using (6.12)  $y_{is^*}(n) - y_{is}(n) = h'_{is^*}(x_{is^*}(n)) - h'_{is}(x_{is}(n))$ , and that  $h'_{is}(1) < \infty$  because  $h'$  is continuous on  $]0, \infty[$ , we can deduce that

$$\begin{aligned}
h'_{i s^*}(x_{i s^*}(n)) - h'_{i s}(x_{i s}(n)) &\geq C + b \sum_{t=n_0}^{n-1} \gamma_t \\
h'_{i s}(x_{i s}(n)) &\leq h'_{i s^*}(x_{i s^*}(n)) - C - b \sum_{t=n_0}^{n-1} \gamma_t \\
h'_{i s}(x_{i s}(n)) &\leq h'_{i s^*}(1) - C - b \sum_{t=n_0}^{n-1} \gamma_t \\
x_{i s}(n) &\leq h'^{-1}_{i s} \left( \mathcal{O} \left( \sum_{t=n_0}^{n-1} \gamma_t \right) \right)
\end{aligned} \tag{6.18}$$

□

## 6.2 Bandit setting

We consider now the extension to bandit setting. In this setting, a player has only access to the noisy payoff of the strategy he chooses. This is a more realistic setting but implies modification in the algorithm.

The payoff received by a player is now :

$$\hat{v}_{i s_i}(n) = \mathbb{1}_{s_i=s_i(n)} \frac{v_{i s_i}(s(n))}{x_{i s_i}(n)} + \xi_{i s_i}(n) \tag{6.19}$$

With such a setting, Hypothesis (H2) does not imply (5.13b) anymore and the noise is not bounded. To overpass this problem, the algorithm is slightly changed such that  $x_{i s_i}(n) > \epsilon$  with a  $\epsilon > 0$  parameter of the algorithm. Fixing such an  $\epsilon$  also allow us to use Hypothesis (H2').

### 6.2.1 $\epsilon$ -dynamic and $\epsilon$ -equilibrium

The modified algorithm is as follows, with  $S_i$  being the number of pure strategies in  $\mathcal{S}_i$ .

**Algorithm 6.2** Epsilon dual-averaging regularized algorithm with variable step-size  $\gamma_t$  and fixed  $\epsilon_i$  that can differ between players.

- 1 Each player  $i \in \mathcal{N}$  has an initial score vector  $y_i(1)$  and plays with initial mixed strategy  $x_i^\epsilon(1) = \frac{\epsilon_i}{S_i} + (1 - \epsilon_i)Q_i(y_i(1))$  where the choice map  $Q_i$  is defined from the penalty function as

$$Q_i(y_i(n)) = \arg \max_{x_i \in \mathcal{X}_i} (\langle y_i(n) | x_i(n) \rangle - h_i(x_i(n))). \tag{6.20}$$

- 2 **for** each round  $n$
- 3     Each player  $i \in \mathcal{N}$  draws a pure strategy  $s_i(n)$  according to  $x_i^\epsilon(n)$
- 4     Each player  $i \in \mathcal{N}$  gets an estimate  $\hat{v}_i(n)$  of their payoff vector  $v_i(s(n))$
- 5     Each player  $i \in \mathcal{N}$  updates their mixed strategy  $x_i^\epsilon$  via the recursion

$$\begin{aligned}
y_i(n+1) &= y_i(n) + \gamma_n \hat{v}_i(n), \\
x_{i s}(n+1) &= Q_i(y_i(n+1)), \\
x_{i s}^\epsilon(n+1) &= \frac{\epsilon_i}{S_i} + (1 - \epsilon_i)x_{i s}(n+1)
\end{aligned} \tag{ $\epsilon$ -DA}$$

**end for**

**Observation 6.1.** For any  $t \in \mathbb{N}$ ,  $\|x^\epsilon(n) - x(n)\|_2 \leq N\epsilon$  where  $\epsilon = \max_{i \in \mathcal{N}} \frac{\epsilon_i}{\sqrt{S_i}}$ ,  $N$  is the number of players and  $S_i$  the number of pure strategies of player  $i$ .

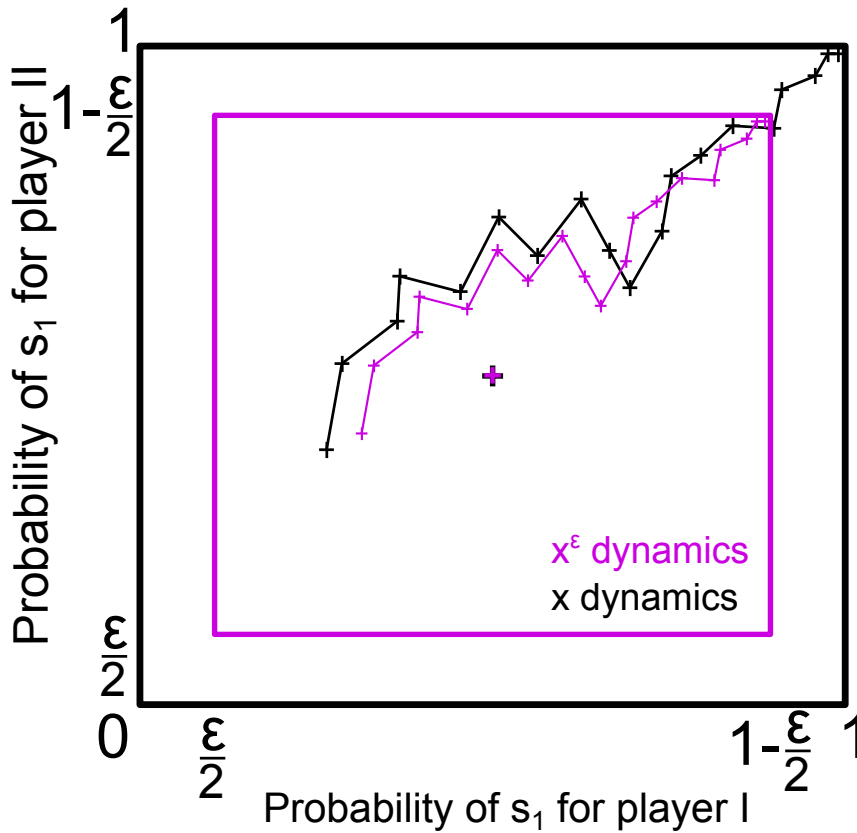


FIGURE 6.1: Evolution of  $x(n)$  and  $x^\epsilon(n)$  in a two player game with two strategies each. The x-axis is the probability of player one to play its first strategy which is  $x_{1s_1}(n)$  (resp.  $x_{1s_1}^\epsilon(n)$ ), the y-axis is the probability of player two to play its first strategy which is  $x_{2s_1}(n)$  (resp.  $x_{2s_1}^\epsilon(n)$ ).

This ensures that  $x_{iS}^\epsilon(n+1) > \epsilon$  so variances of payoffs are bounded. The  $\epsilon$ -dynamics followed by  $x^\epsilon$  is different than previously studied dynamics as it cannot converge to pure equilibria. Figure 6.1 represents the simultaneous evolutions of  $x(n)$  (in black) and  $x^\epsilon(n)$  (in violet). The middle point, corresponding to an uniformly probability of drawing strategies, is the only point equal in both dynamics.

This rises the notion of  $\epsilon$ -equilibrium.

**Definition 6.2.** A strategy profile  $x^*$  is said to be an  $\epsilon$ -Nash equilibrium if it satisfies the following condition:

$$u_i(x_i^*; x_{-i}^*) + \epsilon \geq u_i(x_i; x_{-i}^*) \quad \forall x_i \in \mathcal{X}_i, i \in \mathcal{N}. \quad (\epsilon\text{-NE})$$

The link between the trajectory of  $x(n)$  and  $x^\epsilon(n)$  appears clearly in Figure 6.1. But it is important to state the relation between the convergence of the two.

**Lemma 6.6.** If  $(x(n))_{n \in \mathbb{N}}$  converges to  $x^*$  a Nash equilibrium, then  $(x^\epsilon(n))_{n \in \mathbb{N}}$ , defined in  $(\epsilon\text{-DA})$ , converges to a  $\delta(\epsilon)$ -Nash equilibrium with  $\delta(\epsilon) \rightarrow 0$  when  $\epsilon \rightarrow 0$ , with  $\epsilon = \max_{i \in \mathcal{N}} \frac{\epsilon_i}{\sqrt{S_i}}$ .



*Proof.*  $x^\epsilon$  is uniquely defined from  $x$  by a continuous equation, so  $(x^\epsilon(n))_{n \in \mathbb{N}}$  also converges to a limit  $x^{\epsilon*}$ . Observation (6.1) gives us that  $\|x^\epsilon(n) - x(n)\|_2 \leq N\epsilon$  so  $\|x^{\epsilon*} - x^*\|_2 \leq N\epsilon$ .

We show that if  $\|x - x'\|_2 \leq \epsilon$  then  $\|u_i(x) - u_i(x')\|_2 \leq \frac{\delta(\epsilon)}{2} \rightarrow_{\epsilon \rightarrow 0} 0$

$$\begin{aligned}
|u_i(x) - u_i(x')| &= \left| \sum_{s_1 \in \mathcal{S}_1} \cdots \sum_{s_N \in \mathcal{S}_N} u_i(s_1, \dots, s_N) (x_{1s_1} \cdots x_{Ns_N} - x'_{1s_1} \cdots x'_{Ns_N}) \right| \\
&\leq \sum_{s_1 \in \mathcal{S}_1} \cdots \sum_{s_N \in \mathcal{S}_N} u_i(s_1, \dots, s_N) |(x_{1s_1} \cdots x_{Ns_N} - x'_{1s_1} \cdots x'_{Ns_N})| \\
&\leq \sum_{s_1 \in \mathcal{S}_1} \cdots \sum_{s_N \in \mathcal{S}_N} u_i(s_1, \dots, s_N) |(x_{1s_1} \cdots x_{Ns_N} - (x_{1s_1} - \epsilon) \cdots (x_{Ns_N} - \epsilon))| \\
&\leq \sum_{s_1 \in \mathcal{S}_1} \cdots \sum_{s_N \in \mathcal{S}_N} u_i(s_1, \dots, s_N) \left| \sum_{k=1}^N (-\epsilon)^k \binom{N}{k} \right| \\
&= \frac{\delta(\epsilon)}{2} \\
&\rightarrow_{\epsilon \rightarrow 0} 0
\end{aligned} \tag{6.21}$$

$x^*$  is a Nash Equilibrium so  $u_i(x_i^*; x_{-i}^*) \geq u_i(x_i; x_{-i}^*)$  for all  $x_i \in \mathcal{X}_i, i \in \mathcal{N}$ . In addition we have  $\|x^{\epsilon*} - x^*\|_2 \leq N\epsilon$  and  $\|\{x_i, x_{-i}^{\epsilon*}\} - \{x_i, x_{-i}^*\}\|_2 \leq N\epsilon$ .

$$\begin{aligned}
u_i(x_i^{\epsilon*}; x_{-i}^{\epsilon*}) + \frac{\delta(\epsilon)}{2} &\geq u_i(x_i^*; x_{-i}^*) \geq u_i(x_i; x_{-i}^*) \geq u_i(x_i; x_{-i}^{\epsilon*}) - \frac{\delta(\epsilon)}{2} \\
u_i(x_i^{\epsilon*}; x_{-i}^{\epsilon*}) + \delta(\epsilon) &\geq u_i(x_i; x_{-i}^{\epsilon*}) \text{ for all } x_i \in \mathcal{S}_i, i \in \mathcal{N}
\end{aligned} \tag{6.22}$$

□

This result allows us to focus on the convergence of  $x(n)$  towards Nash Equilibrium. We will proceed like in Section 6.1.2, the redundant calculations are not written again.

## 6.2.2 Global convergence

The first step is to show that the linear interpolation  $x(t)$  of  $(x(n))_{n \in \mathbb{N}}$  is an asymptotic pseudotrajectory of a dynamics.

### 6.2.2.1 Asymptotic pseudo trajectory

**Theorem 6.7.** *Under Hypotheses (H1) and (H2'), if the step size is of the form  $\gamma(t) = \frac{\gamma}{t^\beta}$  with  $\beta \in [\frac{1}{2}, 1]$ , the interpolated process of the sequences  $(x_i(n))_{n \in \mathbb{N}}$  is an asymptotic pseudo trajectory of the solutions of the following ordinary differential equation*

$$\begin{aligned}
\dot{x}_{is}(t) &= \sum_{s' \in \mathcal{S}_i} \nabla(Q_{is})_{s'}(y_i(t)) v_{is'}(x^\epsilon(t)) = \nabla Q_{is}^T(y_i(t)) v_i(x^\epsilon(t)) \\
\dot{x}_i(t) &= \text{Jac}(Q_i)(y_i(t)) v_i(x^\epsilon(t)),
\end{aligned} \tag{6.23}$$

where  $\nabla Q_{is}$  is the gradient vector of  $Q_{is}$  and  $\text{Jac}(Q_i)$  the Jacobian matrix of  $Q_i$ .

*Proof.* First, we will check that the stochastic process  $(x_i(n))$  given by  $(\epsilon\text{-DA})$  is an approximate Robbins-Monro algorithm. Using Taylor's Remainder Theorem, we obtain:

$$\begin{aligned}
x_{is}(n+1) &= Q_{is}(y_i(n+1)) \\
&= Q_{is}(y_i(n) + \gamma_n \hat{v}_i(n)) \\
&= Q_{is}(y_i(n)) + \gamma_n \left( \nabla Q_{is}^T(y_i(n)) \hat{v}_i(n) + \frac{1}{2} \gamma_n \hat{v}_i^T(n) \text{Hess}(Q_{is})(\psi_i(n)) \hat{v}_i(n) \right) \\
&= x_{is}(n) + \gamma_n \left( \nabla Q_{is}^T(y_i(n)) \hat{v}_i(n) + \frac{\gamma_n}{2} \hat{v}_i^T(n) \text{Hess}(Q_{is})(\psi_i(n)) \hat{v}_i(n) \right) \\
&= x_{is}(n) + \gamma_n \left( \nabla Q_{is}^T(y_i(n)) v_i(x^\epsilon(n)) + \nabla Q_{is}^T(y_i(n)) (\hat{v}_i(n) - v_i(x^\epsilon(n))) + \gamma_n a_n \right)
\end{aligned}$$

The following of the proof is the exact same as in (DUAL AVERAGING).  $\square$

We also have the same result with  $y(t)$ .

**Lemma 6.8.** *With the same assumptions, the interpolated process of the sequences  $(y_i(n))_{n \in \mathbb{N}}$  is an asymptotic pseudo trajectory of the solutions of the following ordinary differential equation*

$$\dot{y}_i(t) = v_i(x^\epsilon(t)) \quad (6.24)$$

*Proof.* Recall that

$$y_i(t+1) = y_i(t) + \gamma_t \hat{v}_i(t) = y_i(t) + \gamma_t v_i(x^\epsilon(t)) + \gamma_t (\hat{v}_i(t) - v_i(x^\epsilon(t)))$$

Using Hypotheses (H1) and (H2) in propositions 4.2 and 4.1 of [113] allow us to conclude the proof.  $\square$

### 6.2.2.2 Global convergence with the continuous dynamics

In this subsection,  $x(t)$  is not anymore the discrete variable defined in the algorithm but a solution of the dynamics (6.23).

We are now going to show that it converges to rest points of (6.6). Let  $F$  be the potential function of the game.

**Theorem 6.9.** *The potential function  $F$  of the game is a strict increasing Lyapunov function (see Definition (5.3)) of the flow induced by the dynamics (6.6).*

*Proof.* The calculus are slightly different than in (DUAL AVERAGING) because here the dynamics is on  $x$  but the payoff is evaluated on  $x^\epsilon$ . We consider the variation of  $F$ . We have

$$\begin{aligned}
\dot{F}(x^\epsilon(t)) &= \sum_{i \in \mathcal{N}} \frac{\partial F}{\partial x_i}(x^\epsilon(t)) \dot{x}_i(t) \\
&= \sum_{i \in \mathcal{N}} v_i^T(x^\epsilon(t)) \dot{x}_i(t) \\
&= \sum_{i \in \mathcal{N}} \sum_{s \in \mathcal{S}_i} v_{is}(x^\epsilon(t)) \dot{x}_{is}(t)
\end{aligned} \quad (6.25)$$

We are now going to find another expression of  $\dot{x}_i(t)$ , to that end we use the Lagrange multiplier. We don't write the calculus that are the exact same as in (DUAL AVERAGING).

$$\dot{x}_{is} = \frac{\dot{y}_{is}}{d_{is}} - \frac{1}{d_{is}} \frac{\sum_{s' \in \mathcal{S}_i} \frac{\dot{y}_{is'}}{d_{is'}}}{\sum_{s' \in \mathcal{S}_i} \frac{1}{d_{is'}}} = \sum_{s' \in \mathcal{S}_i} \frac{\dot{y}_{is} - \dot{y}_{is'}}{d_{is'} d_{is} \sum_{s' \in \mathcal{S}_i} \frac{1}{d_{is'}}} \quad (6.26)$$

With  $d_{is} = \frac{d^2 h_{is}}{d^2 x_{is}}(x_{is})$ .

Injecting that into (6.25) and using  $\dot{y}_{is}(t) = v_{is}(x^\epsilon(t))$  (Lemma (6.8)):

$$\begin{aligned} \dot{F}(x^\epsilon(t)) &= \sum_{i \in \mathcal{N}} \sum_{s \in \mathcal{S}_i} \sum_{s' \in \mathcal{S}_i} \frac{v_{is}^2(x^\epsilon(t)) - v_{is'}(x^\epsilon(t))v_{is}(x^\epsilon(t))}{d_{is'} d_{is} \sum_{s' \in \mathcal{S}_i} \frac{1}{d_{is'}}} \\ \dot{F}(x^\epsilon(t)) &= \sum_{i \in \mathcal{N}} \sum_{s \in \mathcal{S}_i} \sum_{s' \in \mathcal{S}_i, s' \neq s} \frac{(v_{is}(x^\epsilon(t)) - v_{is'}(x^\epsilon(t)))^2}{d_{is'} d_{is} \sum_{s' \in \mathcal{S}_i} \frac{1}{d_{is'}}} \\ \dot{F}(x^\epsilon(t)) &\geq 0 \end{aligned} \quad (6.27)$$

It is important to remark here that whereas payoff  $v$  are evaluated on  $x^\epsilon$ ,  $d_{is}$  corresponds to the second derivative of  $h_{is}$  evaluated on  $x$ . This is because payoffs are evaluated on the probability used to draw strategies whereas  $d_{is}$  comes from the analysis of the choice map that determines  $x$ . The consequences are that  $d_{is}$  converges to  $\infty$  when  $x_{is} \rightarrow 0$  because  $h$  is steep, but  $x_{is}^\epsilon$  does not converge to 0.

If  $x^\epsilon(t)$  is an equilibrium,  $\dot{x}_{is}^\epsilon(t) = \frac{\dot{x}_{is}(t)}{1-\epsilon_i} = 0$  for all  $i \in \mathcal{N}$ ,  $s \in \mathcal{S}_i$  and  $\dot{F}(x^\epsilon(t)) = 0$  using the Equation (6.9). Let show the reverse, meaning that is  $\dot{F}(x^\epsilon(t)) = 0$  then  $x^\epsilon(t)$  is an equilibrium. First we remark that  $\dot{F}(x^\epsilon(t)) = 0$  if each term of the sum is null. At least one of three conditions needs to be true for a term to be null:

1.  $\frac{1}{d_{is}} = 0$ ,
2.  $\frac{1}{d_{is'}} = 0$ ,
3.  $v_{is}(x^\epsilon(t)) = v_{is'}(x^\epsilon(t))$ .

Recall that  $h$  is steep, its derivative goes to the infinity when reaching the boundary of  $\mathcal{X}$ , as  $\mathcal{X}$  is bounded. Therefore the second derivative of  $h$  does the same and  $d_{is} \rightarrow_{x_{is} \rightarrow 0} \infty$ .

So those three conditions can be gathered in one condition :

$$\dot{F}(x^\epsilon(t)) = 0 \Leftrightarrow \forall i \in \mathcal{N} v_{is}(x^\epsilon(t)) = v_{is'}(x^\epsilon(t)) \forall s, s' \in \mathcal{S}_i \text{ such that } \frac{1}{d_{is}} \neq 0 \text{ and } \frac{1}{d_{is'}} \neq 0$$

Recall that  $\dot{y}_{is}(t) = v_{is}(x^\epsilon(t))$  (Lemma (6.2)) so this condition applied to (6.26) gives us that  $\dot{x}_{is}(t) = 0 \quad \forall i \in \mathcal{N}, \forall s \in \mathcal{S}_i$  so  $\dot{x}_{is}^\epsilon(t) = 0$ . Therefore  $\dot{F}(x^\epsilon(t)) = 0$  if and only if  $(x^\epsilon(t))$  is an equilibrium of the dynamics (6.23). □

The corollary (5.13) does not apply here because  $\frac{1}{d_{is}} \rightarrow 0$  when  $x_{is} \rightarrow 0$  but the payoff is evaluated on  $x^\epsilon$  which coordinates are never null. However, finding the number of rest points of the dynamics is equivalent of finding  $x \in \mathcal{X}$  such that

$$\forall i \in \mathcal{N} v_{is}(x^\epsilon(t)) = v_{is'}(x^\epsilon(t)) \forall s, s' \in \mathcal{S}_i \text{ such that } x_{is} > 0 \text{ and } x_{is'} > 0$$

Let  $sup_i$  be the cardinal of the support of  $x_i$ . Because the game is generic, we obtain,  $sup_i - 1$  independent equations per player ( $v_{is'}(x^\epsilon(t)) = v_{is''}(x^\epsilon(t))$  is a combination of  $v_{is}(x^\epsilon(t)) = v_{is''}(x^\epsilon(t))$  and  $v_{is}(x^\epsilon(t)) = v_{is'}(x^\epsilon(t))$ ) and by definition of the support we also have  $sup_i - 1$  unknowns per player. Therefore we obtain the same

number of unknowns and independent equations. We thus have one solution by different support of  $x(t)$  which is a finite number of rest points.

### 6.2.3 Global convergence in the potential game

In that section  $x(n)$  is again the discrete variable defined in the algorithm.

Just like for (HEDGE) the Corollary 6.6 of [113] allows us to conclude that  $x^\epsilon(n)$  converge to a rest point of the dynamics (6.23).

Then Lemma (6.6) shows that  $x^\epsilon(n)$  converges if and only if  $x(n)$  converges.

**Theorem 6.10.** *If  $x^\epsilon(n) \rightarrow x^{*\epsilon}$ ,  $x^{*\epsilon}$  is a  $\epsilon$ -Nash Equilibrium and  $x(n) \rightarrow x^*$  with  $x^*$  a Nash Equilibrium, given that  $\epsilon$  is sufficiently small.*

*Proof.* When  $x^\epsilon(n)$  converges,  $x(n)$  converges too. Applying Lemma 6.6 it suffices to show that if  $x(n) \rightarrow x^*$  then  $x^*$  is Nash Equilibrium. We show by contradiction that  $(x(n))_{n \in \mathbb{N}}$  converges to  $x^*$  a Nash Equilibrium. Assume that  $x^*$  is not a Nash Equilibrium.

$\exists i \in \mathcal{N}, \exists s' \in \mathcal{S}_i, s' \notin \text{supp}(x_i^*), s.t., v_{is'}(x^*) > v_{is}(x^*), \forall s \in \text{supp}(x_i^*)$ .

By continuity of  $u$ , there is a neighborhood  $U$  of  $x^*$  and  $a > 0$  such that:

$\exists i \in \mathcal{N}, \exists s' \in \mathcal{S}_i, s' \notin \text{supp}(x_i^*), s.t., v_{is'}(x) - v_{is}(x) > a, \forall s \in \text{supp}(x_i^*), x \in U$ .

For  $\epsilon$  small enough and for all  $n$  big enough,  $x^\epsilon(n) \in U$  because  $\|x^\epsilon(n) - x^*\| \leq \|x^\epsilon(n) - x(n)\| + \|x(n) - x^*\| \leq N\epsilon + \|x(n) - x^*\|$ . So,  $\exists i \in \mathcal{N}, \exists s' \in \mathcal{S}_i, s' \notin \text{supp}(x_i^*), s.t., v_{is'}(x^\epsilon(n)) - v_{is}(x^\epsilon(n)) > a, \forall s \in \text{supp}(x_i^*)$  Using Lemma 5.2, for  $n_0$  big enough and  $n \geq n_0$ :

$$y_{is'}(n) - y_{is}(n) \geq C + b \sum_{t=n_0}^{n-1} \gamma_t \rightarrow \infty \quad (6.28)$$

Thus, using (6.12)  $y_{is'}(n) - y_{is}(n) = h'_{is'}(x_{is'}(n)) - h'_{is}(x_{is}(n))$  we have

$$h'_{is'}(x_{is'}(n)) - h'_{is}(x_{is}(n)) \geq C + b \sum_{t=n_0}^{n-1} \gamma_t$$

and

$$h'_{is'}(x_{is'}(n)) - h'_{is}(x_{is}(n)) \rightarrow_{n \rightarrow \infty} \infty$$

That is a contradiction because  $h'_{is'}(x_{is'}(n)) \rightarrow -\infty$  as  $s'$  is not in the support of  $x^*$  and  $x_{is'}(n) \rightarrow 0$  and  $h'_{is}(x_{is}(n))$  is bounded. So  $s$  is not in the support of  $x_i^*$  which is a contradiction.

Therefore  $x^*$  is a Nash Equilibrium and  $x^{*\epsilon}$  is a  $\epsilon$ -Nash Equilibrium. □

### 6.2.4 Convergence time

The asymptotical convergence rate in the bandit setting is the same as in the semi-bandit setting because the convergence of  $(x^\epsilon(n))$  is equivalent to the convergence of  $(x(n))$ . However the time from which the convergence rate is reached changes between the semi-bandit and bandit setting. That is to say the  $n_0$  such that  $x^\epsilon(n_0) \in U$  in the bandit setting and  $x(n_0) \in U$  in the semi-bandit setting. In the bandit setting  $n_0$  is larger than in the semi-bandit game because  $x(n)$  needs to be in a smaller neighborhood for  $x^\epsilon(n)$  to be in  $U$ .

### 6.3 Conclusions

We showed that dual-averaging regularized algorithms in potential games and under specific assumption on the step size converge toward NE in the semi-bandit setting and towards  $\epsilon$ -NE in the bandit setting. Furthermore, asymptotical convergence rate is exponential in both settings using HEDGE algorithm. For more general dual-averaging regularized algorithms, the asymptotical converge rate depends on the penalty function used. As a perspective work, we would like to extend our results in the bandit setting to a more general noise instance, when only the payoff of the chosen strategy is affected by noise. We would also like to study the case of  $\epsilon$ -algorithm with  $\epsilon(t) \rightarrow_{t \rightarrow \infty} 0$ . We believe that under some assumption on  $\epsilon(t)$ ,  $x^\epsilon$  will converge to a NE instead of a  $\epsilon$ -NE as we shown. The next chapter presents an application of HEDGE algorithm to protein folding.

## Chapter 7

# From game theory to protein folding

### Contents

---

<b>7.1</b>	<b>Game setting</b>	<b>101</b>
7.1.1	Dataset	102
7.1.1.1	Secondary structure	102
7.1.1.2	Type of amino acids	102
7.1.2	Players	102
7.1.3	Strategies	103
7.1.3.1	Strategies in the "all AAs" setting	104
7.1.3.2	Strategies in the "tripeptide" setting	104
7.1.4	Utility functions	104
7.1.4.1	Root Mean Squared Distance	105
7.1.4.2	BC-score	105
7.1.4.3	Distance based	106
<b>7.2</b>	<b>Algorithm and implementation</b>	<b>107</b>
7.2.1	Algorithm	107
7.2.2	Implementation	107
<b>7.3</b>	<b>Results and discussions</b>	<b>107</b>
7.3.1	Results	108
7.3.2	Discussion	109

---

Protein folding, i.e, the prediction of the protein structure, can be seen as an optimisation problem, a search of equilibria. Folded proteins correspond to minima of the energy landscape [118], [119]. Using the same approach as in [55] with RNA folding, we attempt to fold proteins. However folding a protein is a much harder problem than an RNA. Not only are protein sequences much more diverse because proteins have 20 AAs and RNAs have only 4 nucleotides, so there are much more parameters to consider, but proteins are also more compact. Also, whereas non-local interactions are well characterized for RNA (base pairs), side chain interactions can be very transient. An approximated fold can contain a lot of collisions (atoms overlapping) and be rejected because it is not biologically correct.

### 7.1 Game setting

We use a dataset of protein chains to determine how to define players, their strategies and utility.

### 7.1.1 Dataset

As we wanted to obtain a distribution of distances and angles corresponding to the distribution over all proteins, we chose a low percentage of identity among sequence. Thus, all protein chains we studied have less than 20% of sequence identity. We also restricted the dataset to only keep proteins with at least 2.0 Å of resolution. The tool, CulledPDB [120], allowed us to easily retrieve all protein chains satisfying these conditions.

We then cleaned the 5,880 chains to keep chains with no gap in the sequence. We also removed alternate conformations.<sup>1</sup>

We obtained a 5,836 protein chains dataset on which we performed various analysis. The number of AAs per chain in the dataset is very disparate, from 6 to 500 with an average of 206.6 and a median of 185 AAs per chain. For the statistics on distances between AAs we did not consider distances larger than 20 Å as we only find them in big proteins. We also did not consider the two AAs directly after and before in the protein chain (the four closest neighbors) because these distances are the same for all AAs in any protein.

For our analysis, we made two distinctions - the secondary structure and the AAs type.

#### 7.1.1.1 Secondary structure

We use DSSP [11] to assign a secondary structure to each AA from the dataset. DSSP classifies secondary structure into eleven categories, distinguishing between  $\pi$ -helix,  $3_{10}$ -helix and  $\alpha$ -helix. As we want to be able to apply our method to protein with unknown structure, we need our classification to correspond to what PSIPRED [12] can classify. We therefore only distinguished between three secondary structures - helices (corresponding to "H" in DSSP), strands of  $\beta$ -sheet (corresponding to "E" in DSSP) and loop (corresponding to all the rest).

When measuring distances between AAs we did not consider AAs in the same element of secondary structure, for example of the same helix, or the same loop. We took this measure because our main focus is to determine how the secondary structures arrange themselves among others.

#### 7.1.1.2 Type of amino acids

AAs have different physico-chemical characteristics that are essential for the folding of the protein. Indeed a few mutations, i.e. changes in the AA sequence, can result in a completely different fold [121]. We used the three groups classification described in Table 1.2. This classification between, charged, polar and hydrophobic AAs keep the main physico-chemical information. Hydrophobic AAs are usually buried inside the protein to avoid any interaction with the solvent. Polar AAs participate in hydrogen bonds and charged AAs can also interact with each other.

### 7.1.2 Players

We work with two different definitions of players. Both definitions are related to amino acids. We use a *coarse-grained* approach, each AAs is represented by only 4 (or 3 for the glycine) of its atoms - N,  $C_{\alpha}$ , C and  $C_{\beta}$  (except for the glycine which

<sup>1</sup>Alternate conformations appear when an atom can have multiple positions. In that case, all potential positions are written in the structure file with their probability. By removing alternate conformations we mean keeping the most likely.

does not have that atom). Bonds length and angles in the backbone of the protein are almost fixed. We computed them in the dataset and found the values illustrated in Figure 7.1.

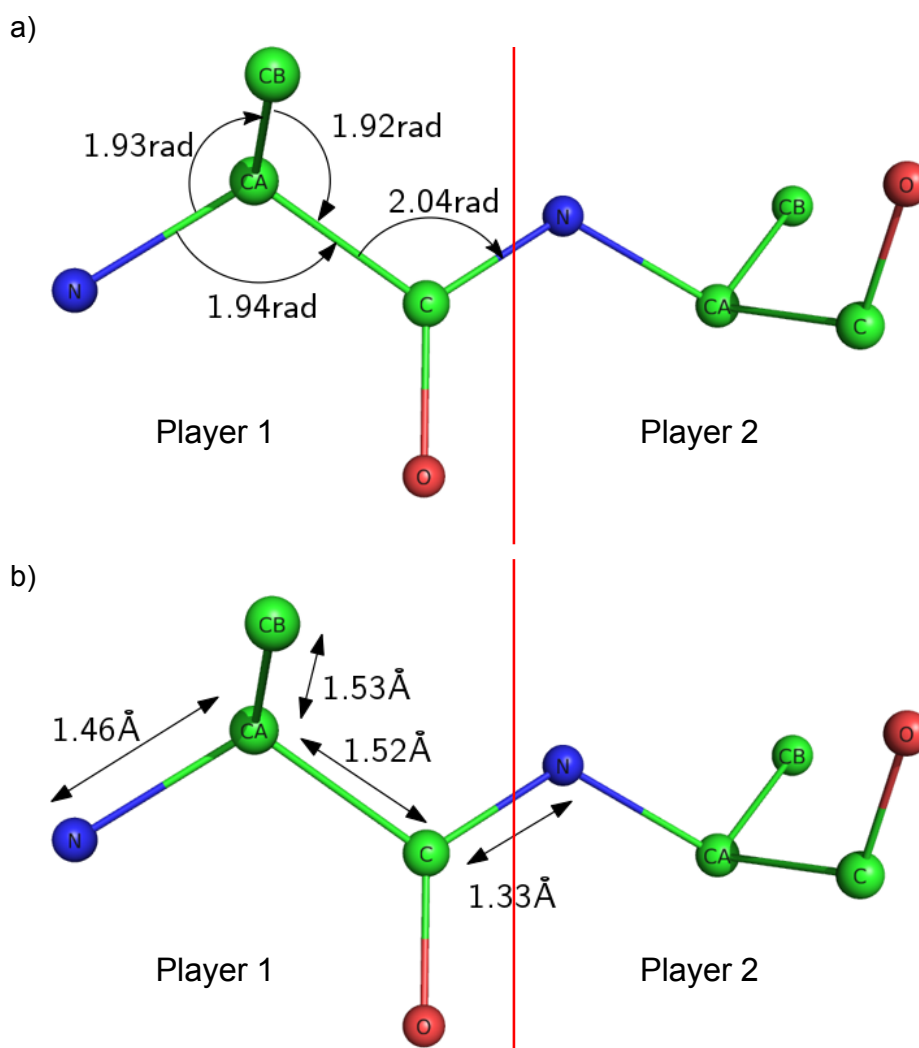


FIGURE 7.1: Representation of a player and its 4 atoms. the oxygen atom (O) is represented for a better understanding of the figure but it is not in the player description. a) Plan angles between atoms of the AA. b) Bond lengths.

These fixed values allow us to construct the backbone of the protein only from its sequence.

In the "all AAs" setting each amino acid is a player. In the "tripeptide" setting a player is a set of three AAs, with no overlap. If the number of AAs is not a multiple of 3, the last or the two last AAs do not belong to a player.

### 7.1.3 Strategies

Strategies are the moves that players can perform to fold the protein. We only consider *torsional angles*, i.e., rotation around bonds. As presented in Chapter 1 two



torsional angles<sup>2</sup> are allowed in each AA backbone. Therefore in the "all AAs" setting a strategy is a couple of two angles  $(\phi, \psi)$  and in the "tripeptide" setting a strategy is a tuple of six angles  $(\phi_1, \psi_1, \phi_2, \psi_2, \phi_3, \psi_3)$ , corresponding to the two torsional angles of three AAs. The strategy set of each player is defined differently for the two settings. In both settings the strategy sets of players depends on the secondary structure of the AA.

### 7.1.3.1 Strategies in the "all AAs" setting

In this setting we choose the strategy sets according to the accessible torsional angles in the Ramachandran plot [10]. Ramachandran plots represent the possible couples of  $(\phi, \psi)$  angles for an AA. We chose to use bins of 1 degree and computed Ramachandran plots over our dataset for helices, strands and loops (Figure 7.2) by counting the number of occurrences of each couple.

We chose strategy sets by randomly drawing 100 to 10000 (depending on experiments) couples of angles from each Ramachandran plots. We draw couples of angles according to the distribution observed in the dataset but without drawing twice the same couple.

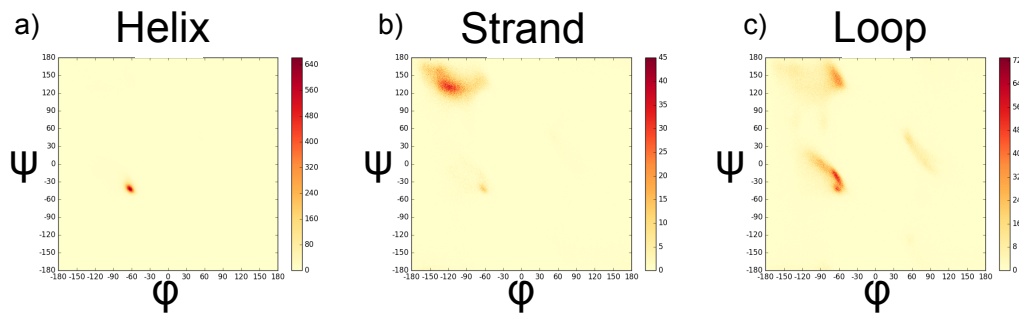


FIGURE 7.2: Ramachandran plots for (a) helices, (b) strands, (c) loop. Each point correspond to a couple of  $(\phi, \psi)$  angles. The color scale indicates the number of AAs in the dataset having a certain couple of torsional angles.

### 7.1.3.2 Strategies in the "tripeptide" setting

For the tripeptide setting, we used a database [122] of tripeptide (three AAs) based on the SCOP database [31]. For each tripeptide possible sequence ( $20^3$  possibilities but we only use the ones present in the protein under study) the database contains the tuples of angles found in the SCOP database as well as their secondary structures.

For each player we use as strategies the tuples in the database that correspond to the same secondary structures.

## 7.1.4 Utility functions

Now that we have chosen players and their strategies, the last step is to define an utility function. We will mainly use the algorithm from section 5.3 without noise. Therefore we want our game to be a potential game. A natural way of choosing the utility function such that the game is a potential game is to choose the same utility

<sup>2</sup>As presented in Chapter 1 we always set the angle  $\omega$  to  $\pi$ .

function for all players. Then the utility function is itself the potential function. The local minima of the utility function are NE, and thus they are potential limit points of the algorithm. Therefore the utility function should give high utility to folded protein. We tried three different functions. The first two are test functions that use the native (i.e., the folded protein) as input.

#### 7.1.4.1 Root Mean Squared Distance

The natural utility function is the RMSD to the native structure. This utility function can not be used for prediction as it needs the native structure. However, it is useful to assess the quality of our method independently of the quality of the utility function itself. To compute the RMSD we first need to superimpose the two structures. That includes translating them so their barycenters are the same point and then find the best rotation. Once that is done, position of all atoms are compared one by one. The RMSD between two aligned structures  $S_1$  and  $S_2$  having the same atoms  $\{a_1, \dots, a_n\}$  is:

$$RMSD(S_1, S_2) = \sqrt{\frac{1}{n} \sum_{i=1}^n \left( (a_{ix}^{S_1} - a_{ix}^{S_2})^2 + (a_{iy}^{S_1} - a_{iy}^{S_2})^2 + (a_{iz}^{S_1} - a_{iz}^{S_2})^2 \right)} \quad (7.1)$$

where  $a_{ix}^{S_1}$  the  $x$  coordinate of atom  $a_i$  in the structure  $S_1$ .

However this would give a minimal utility to the native structure so we used  $100 - RMSD(native, S)$  as a utility function.<sup>3</sup> A drawback of such a function is that it is only meaningful for small values. For relatively different structures (with an RMSD greater than 15 Å to 10 Å) having an higher RMSD does not mean that the structures are more different. This is partly due to the fact that RMSD is highly sensitive to outliers.

#### 7.1.4.2 BC-score

The *Binet-Cauchy score* [123] measures similarity between two proteins but does not use superimposition. The only pre-computing step is to translate both structures so that their barycenters are at the origin  $(0, 0, 0)$ . This score is normalized between 1 for exact similarity to  $-1$  for exact anti-similarity (the mirror structure). It can be seen as a correlation coefficient. The BC-score only considers  $C\alpha$  atoms in the protein (that is one atom per residue). It compares the volumes of all tetrahedra formed by the origin and three  $C\alpha$  atoms (there are  $\binom{n}{3}$  such tetrahedra in a protein of  $n$  AAs). The volume of such a tetrahedra is  $\frac{1}{6}$  of the determinant of the  $3 \times 3$  matrix which rows are coordinates of  $C\alpha$  atoms. To compare tetrahedra volumes in the two structures, the BC-score computes the scalar product of two  $\binom{n}{3}$ -vectors each containing determinants of the  $3 \times 3$  matrices. We note  $S_1^\alpha$  and  $S_2^\alpha$  the  $n \times 3$  matrices of structure  $S_1$  and  $S_2$ , each row is a  $C\alpha$  atom's coordinates. This calculations have a complexity of  $N^3$ .

However BC-score uses the *Binet-Cauchy* theorem to simplify the computation. *Binet-Cauchy* theorem gives that:

$$\sum_{R \subset \{1, \dots, n\}} \det(S_{1R}^\alpha) \det(S_{2R}^\alpha) = \det(S_1^{\alpha T} S_2^\alpha) \quad (7.2)$$

where the sum is over all subset of three rows.

<sup>3</sup>100 is arbitrary chosen to be greater than any RMSD we could evaluate.

As we normalize the BC-score, its formula becomes:

$$BC(S_1^\alpha, S_2^\alpha) = \frac{\det(S_1^{\alpha T} S_2^\alpha)}{\det(S_1^{\alpha T} S_1^\alpha) \det(S_2^{\alpha T} S_2^\alpha)}$$

So computing a BC-score is linear on the number of AAs.

### 7.1.4.3 Distance based

We construct a utility function based on our dataset. Unlike the two previous functions, this function does not need the native structure. We computed over the dataset the probability of having a given distance (by steps of 0.1 Å) for any couple of two AAs. We did not consider AAs in the same portion of secondary structure, nor the two neighbors before and after in the protein's sequence. We distinguished between the type of AAs (hydrophobic, polar and charged) and secondary structures (helix, strand or loop). Then we divided each of the type and secondary structure specific distribution by the overall distribution in order to highlight particularities (Figure 7.3).

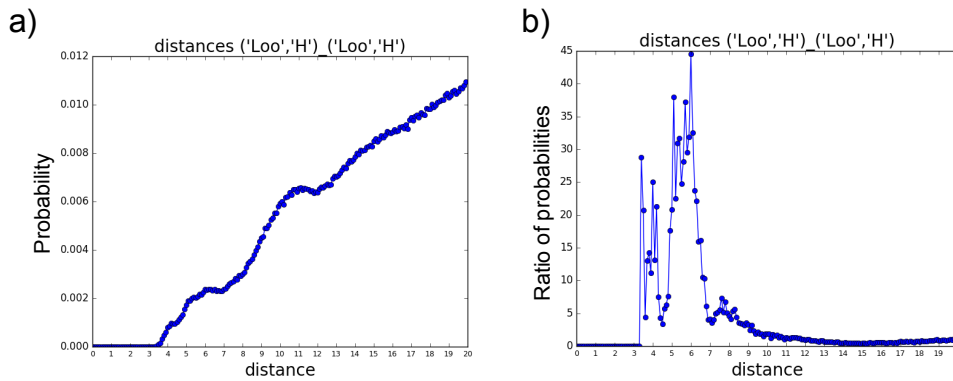


FIGURE 7.3: Distribution of distances between two  $C\beta$  atoms of AAs hydrophobic in a loop. a) The distribution. b) The same distribution divided by the distribution over all pairs of AAs.

Also, in order to avoid excessively high values for uncommon distances we gather all distances under 3.5 Å in the same bin. Then the formula of this utility function is the sum over all couple of atoms of the probability of their distance given their type and secondary structure divided by the probability of their distance :

$$U(S) = \sum_{i=1}^{n-2} \sum_{\substack{j=i+2 \\ s_i \neq s_j}}^n \frac{\mathbb{P}(d(a_i, a_j) | t_i, t_j, s_i, s_j)}{\mathbb{P}(d(a_i, a_j))} \mathbb{1}_{(d(a_i, a_j) < 20)} \quad (7.3)$$

where  $t_i, t_j$  and  $s_i, s_j$  are the types and secondary structures of atoms  $a_i$  and  $a_j$ . The numerator corresponds to the probability of finding  $d(a_i, a_j)$  between two players of types and secondary structures  $t_i, t_j, s_i, s_j$ . The denominator corresponds to the probability of finding  $d(a_i, a_j)$  between two players of any type. This allows to highlight distances specific to a given couple of players and reduce the noise in high distances due to the globular shape.

**Collisions** None of these three utility functions avoid collisions. However a structure with collisions, i.e. overlapping atoms, can not be relevant. Thus we give a negative utility to structures with collisions. In order to encourage the resolution of collisions, a colliding utility is also proportional to the number of colliding atoms.

## 7.2 Algorithm and implementation

### 7.2.1 Algorithm

We use the HEDGE algorithm as studied in Chapter 5 with a step size  $\gamma_t = \frac{1}{\sqrt{t}}$  and the same utility function for all players. In the "all AAs" setting, we start by randomly drawing the strategies for each kind of players (helix, strand and loop) according to the distribution in the Ramachandran plot.

Then in both settings each player randomly initializes its score vector  $y_i(1)$  and computes its mixed strategy vector with the logit map.

At each step, each player draws a pure strategy according to its mixed strategy vector. Then the protein is deformed according to the chosen strategies. Each player receives the utility of the protein for all of its strategies and updates its score vector and its mixed strategy for the next step.

### 7.2.2 Implementation

We use OpenMPI (*Message Passing Interface*) parallelization to speed-up the computation. The natural way to do this is by parallelizing utility computation. Each player tries all its strategies and computes its utility. That step is the more time consuming because utility is computed many times, but it can also be easily parallelized by distributing players between nodes.

Thus, in the master node, players choose their pure strategy and the protein is deformed accordingly. Then the master sends all atom's positions to all the nodes (BCAST) so that they can set protein's conformation. Nodes also receive the number id of players they are responsible off. Each node computes the payoff vector of players it is responsible. To do that it just deforms the protein according to each of the players's strategies, and it computes the utility. Then each node sends the payoff vectors it computed to the master (SCATTER). Only the master updates the score vector of all players and computes their new mixed strategy.

## 7.3 Results and discussions

Here, we present our first results with the RMSD utility function. The distance based and the BC-score utility functions did not give good result yet. The BC-score is very slow to converge, we discuss the possible reasons for that in the discussion part.

We did experiments on the Inria cluster Tompouce, each experiment was parallelized on 24 processes for a few hours (from 12h to 48h). Every experiments was launched from 2 to 10 times with different seeds.

We tried different settings, but for the results presented here, we used 1000 strategies per players in the "All AAs" setting for loop and strands and 100 for helix. In both settings, we used a collision threshold of 1 Å. The collision threshold is used to determine if atoms are colliding: if the center of two atoms are closer than the threshold, the two atoms are colliding. The secondary structures are found using PSIPRED.

### 7.3.1 Results

We tested the algorithm on two structures from CASP12 [39]: T0892 domain 1 and T0898 domain 2. In order to compare our result with the result of CASP experiment, we also computed the *GDT\_TS* score (GlobalDistanceTest\_TotalScore). This score is used by CASP to rank structure prediction. It is based on the percentage of AAs under a certain cutoff distance from the native structure.

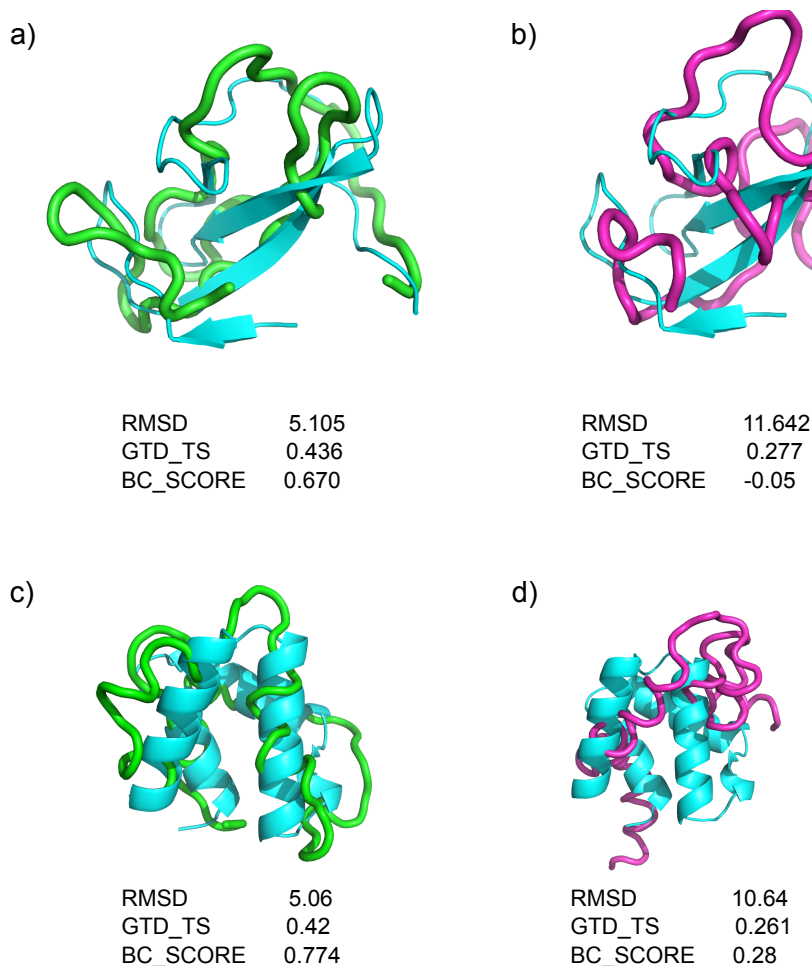


FIGURE 7.4: Proteins folded using the all AAs setting, native structure is in cyan, results are in green and violet. a) and b) represent results for T0898. a) is with RMSD as utility function and b) with the distance based utility. c) and d) represent results for T0892. c) is with RMSD as utility function and d) with distance based utility. Scores of each structure are written underneath them.

**T0898 domain 2** This target is a domain of 55AAs with a  $\beta$ -sheet. In CASP12 the best GDT\_TS score was 0.68, the worst was 0.18. Over all groups that submitted a model in CASP12 our structure determined with RMSD utility would have been 20<sup>th</sup> out of 109 groups and with our distance based utility 78<sup>th</sup> (see Figure 7.4).

**T0892 domain 1** This target is mainly a four helix domain of 69AAs. In CASP12 the best GDT\_TS score was 0.83, the worst was 0.098. Over all groups that submitted a model in CASP12 our structure determined with RMSD utility would have been 56<sup>th</sup> out of 103 groups and with our distance based utility 96<sup>th</sup> (see Figure 7.4).

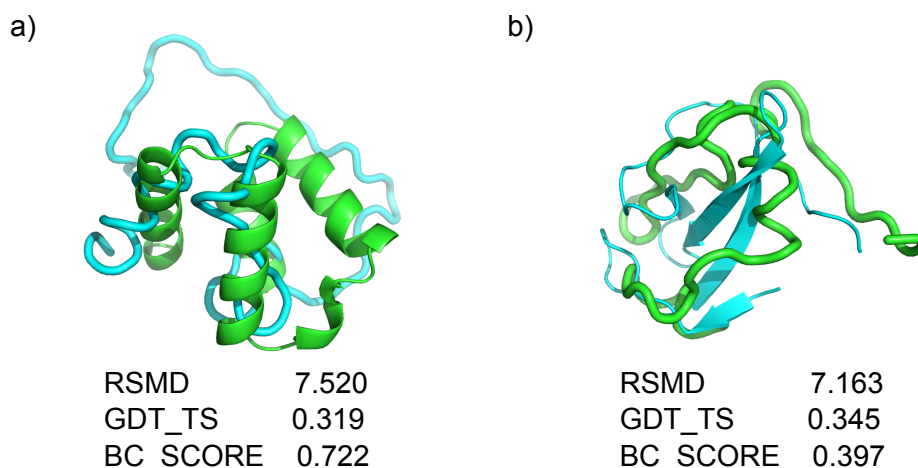


FIGURE 7.5: Proteins folded using the tripeptide setting and RMSD as utility function, native structure is in cyan, results are in green and violet. a) Represent the best result for T0892 and b) for T0898. Scores of each structure are written underneath them.

Our experiments with the tripeptide setting also leads to good results, in particular with the RMSD utility function (Figure 7.5). Our best structure would have arrived 81<sup>th</sup> and 50<sup>th</sup> respectively on T0892 and T0898.

Our results are promising, we discuss a few way of improving.

### 7.3.2 Discussion

We need to identify what we can improve and how.

First it is important to note, that the algorithm converged to local minima and not the global minima. For instance in the all AAs setting, the closest structure to the native was less than 2 Å of RMSD from the native. But we did not reach it.

Increasing the number of strategies available improved the quality of the global minima, but reduced the quality of our results. This could be because the more strategies we add, the more local minima there are. And the more likely our algorithm will converge to a non-optimal conformation.

The problem with the tripeptide setting was different. The closest structure possible to the native had collisions. A possible lead would be to increase the number of possible strategies by duplicating them and perturbing the duplicated by a small value. When we attempted to fold a globular protein, the native structure atoms are not colliding but they are close. The structure is therefore highly sensible to small angle deviations. We need to find a trade-off between increasing the quality of the global minimum and keeping the average RMSD of the final structure low.

A second point is the convergence speed. As studied in Chapters 5 and 6 the convergence speed depends on the difference between utility values. This could explain why the BC-score was long to converge in 48h, its values are small, with

small changes. We need to scale utility function in order to obtain a good convergence rate, but not too fast to make sure the algorithm is allotted sufficient time for a comprehensive exploration.

## Chapter 8

# Conclusions and perspectives

This thesis presents research results motivated by the same general idea of applying new theoretical results to structural bioinformatics problems. We can divide our results in two main research works.

First, we extended a known graph approach (KGS) to the problem of morphing between two conformations. We developed a kinematics-based procedure to morph an RNA molecule between conformational substates, while avoiding inter-atomic clashes. We used a known approach to represent an RNA as a kinematic linkage, and maintains RNA secondary structure by treating hydrogen bonds between base pairs as constraints. However, we developed a novel approach by addressing the problem of finding transition even when we have only sparse information on the goal conformation. Our results confirm that maintaining secondary structures is an efficient and reliable way of reducing the number of degrees of freedom. In addition, our results suggest that molecular junctions can modulate 3D structural rearrangements, while secondary structure elements guide large parts of the molecule along the transition to the correct final conformation.

We also worked on extending the density fitting approach (qFit) for proteins to protein-ligand complexes. Although we do not have final results, our first results are very promising. We can already find alternative ligand conformations that correspond well to the electron density.

Secondly, we try to prove and apply new convergence results in iterative games to protein folding. *Ab-initio* protein folding is a very difficult problem which does not have satisfying answer yet. We started by focusing on theoretical results that are necessary to start applying the algorithm to our protein problem. We even showed stronger results than what we needed. Indeed, we stated the convergence of a class of no-regret algorithms toward Nash Equilibria even in presence of noise (under a few assumptions). We proved the local (global is the equilibrium is unique) convergence toward Nash Equilibria in generic games with semi-bandit feedback. We showed global convergence in semi-bandit setting when the game is a potential and generic game. Finally we proved the convergence to an  $\epsilon$ -Nash Equilibrium in the case of bandit feedback. Most of our convergence results have an exponential rate of convergence.

Besides raising the problem of convergence in the HEDGE algorithm and its extensions, protein folding is very challenging in itself. We developed a protein folding method based on game theory. We represented the folding problem as an equilibrium search with a finite number of players playing a no-regret algorithm in the context of other players. Our method gives encouraging results and finds local equilibria. However we need to improve the method to make it less sensible to local minima.



**Perspectives** Our kinematics-based procedure to morph an RNA molecule is arising interest among experimentalists. We will extend it so that it can satisfy distances between multiple atom pairs instead or in addition to exact atom positions. This will allow us to collaborate with experimentalists that are eager to have a tool to suggest interesting positions for their experimental probes.

Concerning the problem of finding of alternative ligand conformations, we first want to validate our first results by running the method on synthetic data. Then we will need to apply it to a few examples of interest and determine how well it can help experimentalists. We have no doubt that automatizing the "fitting the density" step for ligand will be a major improvement because this step is still often done manually, and overwhelmingly time-consuming.

For the convergence results on EW algorithms we would like to work on removing a few assumptions as well as extending our results to more games. We also want to generalize the  $\epsilon$ -dynamics to a decreasing  $\epsilon$  so that this algorithm would also converge toward Nash Equilibria instead of  $\epsilon$ -Nash Equilibria.

To address the protein folding problem, we first need to avoid systematically converging towards a local minima that can be far from the global minimum. We could do that, by using a more explorative algorithm. We also need to develop a more accurate score function to evaluate the fold quality.

# Acronyms

**AA** *Amino Acid*. 4–7, 10, 11, 18, 61, 62, 64, 101–109

**AMBER** *Assisted Model Building and Energy Refinement*. 18

**CAPRI** *Critical Assessment of PRediction of Interactions*. 11, 14

**CASP** *Critical Assessment of Techniques for Protein Structure Prediction*. 11, 14–17, 108, 109

**CCE** *Coarse Correlated Equilibrium*. 31, 32

**CHARMM** *Chemistry at HARvard Macromolecular Mechanics*. 18

**DEER** *Double Electron-Electron Resonance*. 12, 14, 57–59

**DNA** *Deoxyribonucleic Acid*. 3, 4

**DSSP** *Dictionary of Protein Secondary Structure*. 7, 102

**DSSR** *Defining the Secondary Structure of RNA*. 52

**EM** *Electronic Microscopy*. 12, 13

**ENM** *Elastic Network Model*. 21, 48

**EW** *Exponential Weights*. 33, 67, 70, 86, 112

**HMM** *Hidden Markov Model*. 7, 10

**KGS** *Kino-Geometric Sampling*. 36, 42, 48, 50–52, 111

**KL** *Kullback–Leibler*. 77, 79

**MIQP** *Multi-integer Quadratic Programming*. 62, 64

**MKL** *Intel's Math Kernel Library*. 56

**MSE** *Mean Square Error*. 70, 89

**ncRNA** *non-coding Ribonucleic Acid*. 35, 36

**NE** *Nash Equilibrium*. 29–31, 67–70, 75, 76, 80, 83, 84, 86, 100, 105

**NMA** *Normal Mode Analysis*. 21, 23, 48

**NMR** *Nuclear Magnetic Resonance*. 12–14, 48

**PDB** *Protein Data Bank*. 10, 12, 14, 16, 48, 65

**RMSD** *Root Mean Square Deviation.* 23, 47–55, 105, 107–109

**RNA** *Ribonucleic Acid.* 3–6, 8–15, 17, 18, 35–37, 48, 101, 111, 112

**RRT** *Rapidly exploring Random Tree.* 21–23

**SAXS** *Small Angles X-Ray Scattering.* 12, 13

**SVD** *Singular Value Decomposition.* 39

**WC** *Watson-Crick.* 5, 37, 38, 47, 50, 52, 54, 55

# Bibliography

- [1] J. S. Mattick and I. V. Makunin, “Non-coding RNA”, en, *Hum. Mol. Genet.*, vol. 15, no. suppl 1, R17–R29, Apr. 2006, ISSN: 0964-6906, 1460-2083. DOI: 10.1093/hmg/ddl046.
- [2] W. R. Taylor, “The classification of amino acid conservation”, *Journal of Theoretical Biology*, vol. 119, no. 2, pp. 205–218, Mar. 1986, ISSN: 0022-5193. DOI: 10.1016/S0022-5193(86)80075-3.
- [3] J. D. Watson, F. H. Crick, and others, “Molecular structure of nucleic acids”, *Nature*, vol. 171, no. 4356, pp. 737–738, 1953.
- [4] C. I. Branden and others, *Introduction to protein structure*. Garland Science, 1999.
- [5] D. Eisenberg, “The discovery of the  $\alpha$ -helix and  $\beta$ -sheet, the principal structural features of proteins”, en, *PNAS*, vol. 100, no. 20, pp. 11 207–11 210, Sep. 2003, ISSN: 0027-8424, 1091-6490. DOI: 10.1073/pnas.2034522100.
- [6] L. Pauling, R. B. Corey, and H. R. Branson, “The structure of proteins: Two hydrogen-bonded helical configurations of the polypeptide chain”, en, *PNAS*, vol. 37, no. 4, pp. 205–211, Apr. 1951, ISSN: 0027-8424, 1091-6490. DOI: 10.1073/pnas.37.4.205.
- [7] L. Schrödinger, *The PyMOL Molecular Graphics System, Version 1.8 Schrödinger, LLC*. 2016.
- [8] D. Pal and P. Chakrabarti, “Cis peptide bonds in proteins: Residues involved, their conformations, interactions and locations<sup>1</sup>”, *Journal of Molecular Biology*, vol. 294, no. 1, pp. 271–288, Nov. 1999, ISSN: 0022-2836. DOI: 10.1006/jmbi.1999.3217.
- [9] G. N. Ramachandran, C. Ramakrishnan, and V. Sasisekharan, “Stereochemistry of polypeptide chain configurations”, *Journal of Molecular Biology*, vol. 7, no. 1, pp. 95–99, 1963.
- [10] G. t. Ramachandran and V. Sasisekharan, “Conformation of polypeptides and proteins”, *Advances in Protein Chemistry*, vol. 23, pp. 283–437, 1968.
- [11] R. P. Joosten, T. A. te Beek, E. Krieger, M. L. Hekkelman, R. W. Hooft, R. Schneider, C. Sander, and G. Vriend, “A series of PDB related databases for everyday needs”, *Nucleic Acids Res*, vol. 39, no. Database issue, pp. D411–D419, Jan. 2011, ISSN: 0305-1048. DOI: 10.1093/nar/gkq1105.
- [12] L. J. McGuffin, K. Bryson, and D. T. Jones, “The PSIPRED protein structure prediction server”, eng, *Bioinformatics*, vol. 16, no. 4, pp. 404–405, Apr. 2000, ISSN: 1367-4803.
- [13] K. Asai, S. Hayamizu, and K. Handa, “Prediction of protein secondary structure by the hidden Markov model”, *Bioinformatics*, vol. 9, no. 2, pp. 141–146, Apr. 1993, ISSN: 1367-4803. DOI: 10.1093/bioinformatics/9.2.141.

- [14] F. A. P. Vendeix, A. M. Munoz, and P. F. Agris, "Free energy calculation of modified base-pair formation in explicit solvent: A predictive model", en, *RNA*, vol. 15, no. 12, pp. 2278–2287, Dec. 2009, ISSN: 1355-8382, 1469-9001. DOI: 10.1261/rna.1734309.
- [15] F. H. Crick, "Codon—anticodon pairing: The wobble hypothesis", *Journal of Molecular Biology*, vol. 19, no. 2, pp. 548–555, 1966.
- [16] K. Darty, A. Denise, and Y. Ponty, "VARNA: Interactive drawing and editing of the RNA secondary structure.", *Bioinformatics*, vol. 25, no. 15, pp. 1974–5, 2009.
- [17] H. Yang, F. Jossinet, N. Leontis, L. Chen, J. Westbrook, H. Berman, and E. Westhof, "Tools for the automatic identification and classification of RNA base pairs", eng, *Nucleic Acids Res.*, vol. 31, no. 13, pp. 3450–3460, Jul. 2003, ISSN: 1362-4962.
- [18] N. B. Leontis and E. Westhof, "Geometric nomenclature and classification of RNA base pairs.", *RNA*, vol. 7, no. 4, pp. 499–512, 2001.
- [19] R. Nussinov, G. Pieczenik, J. Griggs, and D. Kleitman, "Algorithms for Loop Matchings", *SIAM J. Appl. Math.*, vol. 35, no. 1, pp. 68–82, Jul. 1978, ISSN: 0036-1399. DOI: 10.1137/0135006.
- [20] R. B. Lyngsø and C. N. Pedersen, "RNA pseudoknot prediction in energy-based models", eng, *J. Comput. Biol.*, vol. 7, no. 3-4, pp. 409–427, 2000, ISSN: 1066-5277. DOI: 10.1089/106652700750050862.
- [21] M Zuker and P Stiegler, "Optimal computer folding of large RNA sequences using thermodynamics and auxiliary information.", *Nucleic Acids Res*, vol. 9, no. 1, pp. 133–148, Jan. 1981, ISSN: 0305-1048.
- [22] R. Lorenz, M. T. Wolfinger, A. Tanzer, and I. L. Hofacker, "Predicting RNA secondary structures from sequence and probing data", eng, *Methods*, vol. 103, pp. 86–98, Jul. 2016, ISSN: 1095-9130. DOI: 10.1016/j.ymeth.2016.04.004.
- [23] D. H. Mathews and D. H. Turner, "Prediction of RNA secondary structure by free energy minimization", eng, *Curr. Opin. Struct. Biol.*, vol. 16, no. 3, pp. 270–278, Jun. 2006, ISSN: 0959-440X. DOI: 10.1016/j.sbi.2006.05.010.
- [24] J. E. Tabaska, R. B. Cary, H. N. Gabow, and G. D. Stormo, "An RNA folding method capable of identifying pseudoknots and base triples.", *Bioinformatics*, vol. 14, no. 8, pp. 691–699, Jan. 1998, ISSN: 1367-4803. DOI: 10.1093/bioinformatics/14.8.691.
- [25] M. Dorn, M. B. e Silva, L. S. Buriol, and L. C. Lamb, "Three-dimensional protein structure prediction: Methods and computational strategies", *Computational Biology and Chemistry*, vol. 53, pp. 251–276, 2014.
- [26] C. Laing and T. Schlick, "Analysis of four-way junctions in RNA structures", *Journal of Molecular Biology*, vol. 390, no. 3, pp. 547–559, 2009.
- [27] A. Lescoute and E. Westhof, "Topology of three-way junctions in folded RNAs", *RNA*, vol. 12, no. 1, pp. 83–93, 2006.
- [28] M. Djelloul and A. Denise, "Automated motif extraction and classification in RNA tertiary structures", *RNA*, vol. 14, no. 12, pp. 2489–2497, 2008.

- [29] G. M. Salem, E. G. Hutchinson, C. A. Orengo, and J. M. Thornton, "Correlation of observed fold frequency with the occurrence of local structural motifs", eng, *J. Mol. Biol.*, vol. 287, no. 5, pp. 969–981, Apr. 1999, ISSN: 0022-2836. DOI: 10.1006/jmbi.1999.2642.
- [30] C. A. Orengo, J. E. Bray, D. W. A. Buchan, A. Harrison, D. Lee, F. M. G. Pearl, I. Sillitoe, A. E. Todd, and J. M. Thornton, "The CATH protein family database: A resource for structural and functional annotation of genomes", eng, *Proteomics*, vol. 2, no. 1, pp. 11–21, Jan. 2002, ISSN: 1615-9853.
- [31] A. G. Murzin, S. E. Brenner, T. Hubbard, and C. Chothia, "SCOP: A structural classification of proteins database for the investigation of sequences and structures", eng, *J. Mol. Biol.*, vol. 247, no. 4, pp. 536–540, Apr. 1995, ISSN: 0022-2836. DOI: 10.1006/jmbi.1995.0159.
- [32] R. D. Finn, P. Coghill, R. Y. Eberhardt, S. R. Eddy, J. Mistry, A. L. Mitchell, S. C. Potter, M. Punta, M. Qureshi, A. Sangrador-Vegas, G. A. Salazar, J. Tate, and A. Bateman, "The Pfam protein families database: Towards a more sustainable future", *Nucleic Acids Res*, vol. 44, no. D1, pp. D279–D285, Jan. 2016, ISSN: 0305-1048. DOI: 10.1093/nar/gkv1344.
- [33] H. M. Berman, J. Westbrook, Z. Feng, G. Gilliland, T. N. Bhat, H. Weissig, I. N. Shindyalov, and P. E. Bourne, "The Protein Data Bank", eng, *Nucleic Acids Res.*, vol. 28, no. 1, pp. 235–242, Jan. 2000, ISSN: 0305-1048.
- [34] J. J. Gray, S. Moughon, C. Wang, O. Schueler-Furman, B. Kuhlman, C. A. Rohl, and D. Baker, "Protein-protein docking with simultaneous optimization of rigid-body displacement and side-chain conformations", eng, *J. Mol. Biol.*, vol. 331, no. 1, pp. 281–299, Aug. 2003, ISSN: 0022-2836.
- [35] G. C. P. van Zundert, J. P.G.L. M. Rodrigues, M. Trellet, C. Schmitz, P. L. Kastiris, E. Karaca, A. S. J. Melquiond, M. van Dijk, S. J. de Vries, and A. M.J. J. Bonvin, "The HADDOCK2.2 Web Server: User-Friendly Integrative Modeling of Biomolecular Complexes", *Journal of Molecular Biology, Computation Resources for Molecular Biology*, vol. 428, no. 4, pp. 720–725, Feb. 2016, ISSN: 0022-2836. DOI: 10.1016/j.jmb.2015.09.014.
- [36] H.-X. Zhou and Y. Shan, "Prediction of protein interaction sites from sequence profile and residue neighbor list", en, *Proteins*, vol. 44, no. 3, pp. 336–343, Aug. 2001, ISSN: 1097-0134. DOI: 10.1002/prot.1099.
- [37] D. W. Ritchie, "Recent progress and future directions in protein-protein docking", eng, *Curr. Protein Pept. Sci.*, vol. 9, no. 1, pp. 1–15, Feb. 2008, ISSN: 1389-2037.
- [38] J. Janin, S. J. Wodak, M. F. Lensink, and S. Velankar, "Assessing Structural Predictions of Protein-Protein Recognition: The CAPRI Experiment", *Reviews in Computational Chemistry*, vol. 28, pp. 137–173, 2015.
- [39] J. Moult, K. Fidelis, A. Kryshtafovych, T. Schwede, and A. Tramontano, "Critical assessment of methods of protein structure prediction: Progress and new directions in round XI", en, *Proteins*, vol. 84, pp. 4–14, Sep. 2016, ISSN: 1097-0134. DOI: 10.1002/prot.25064.
- [40] A. McPherson, "Introduction to protein crystallization", eng, *Methods*, vol. 34, no. 3, pp. 254–265, Nov. 2004, ISSN: 1046-2023. DOI: 10.1016/j.ymeth.2004.03.019.

- [41] D. Marion, "An Introduction to Biological NMR Spectroscopy", *Mol Cell Proteomics*, vol. 12, no. 11, pp. 3006–3025, Nov. 2013, ISSN: 1535-9476. DOI: 10.1074/mcp.O113.030239.
- [42] M. Billeter, "A Consensus on Protein Structure Accuracy in NMR?", *Structure*, vol. 23, no. 2, pp. 255–256, Feb. 2015, ISSN: 0969-2126. DOI: 10.1016/j.str.2015.01.007.
- [43] X.-c. Bai, G. McMullan, and S. H. W. Scheres, "How cryo-EM is revolutionizing structural biology", *eng, Trends Biochem. Sci.*, vol. 40, no. 1, pp. 49–57, Jan. 2015, ISSN: 0968-0004. DOI: 10.1016/j.tibs.2014.10.005.
- [44] E. Callaway, "The revolution will not be crystallized", *Nature*, vol. 525, no. 7568, p. 172, 2015.
- [45] S. Yang, M. Parisien, F. Major, and B. Roux, "RNA Structure Determination Using SAXS Data", *en, The Journal of Physical Chemistry B*, vol. 114, no. 31, pp. 10 039–10 048, Aug. 2010, ISSN: 1520-6106, 1520-5207. DOI: 10.1021/jp1057308.
- [46] D. Franke, C. M. Jeffries, and D. I. Svergun, "Correlation Map, a goodness-of-fit test for one-dimensional X-ray scattering spectra", *en, Nat Meth*, vol. 12, no. 5, pp. 419–422, May 2015, ISSN: 1548-7091. DOI: 10.1038/nmeth.3358.
- [47] B. E. Suzek, Y. Wang, H. Huang, P. B. McGarvey, and C. H. Wu, "UniRef clusters: A comprehensive and scalable alternative for improving sequence similarity searches", *Bioinformatics*, vol. 31, no. 6, pp. 926–932, Mar. 2015, ISSN: 1367-4803. DOI: 10.1093/bioinformatics/btu739.
- [48] C. B. Anfinsen, "Principles that govern the folding of protein chains", *eng, Science*, vol. 181, no. 4096, pp. 223–230, Jul. 1973, ISSN: 0036-8075.
- [49] W. G. Noid, "Perspective: Coarse-grained models for biomolecular systems", *The Journal of chemical physics*, vol. 139, no. 9, 09B201\_1, 2013.
- [50] S. Raman, R. Vernon, J. Thompson, M. Tyka, R. Sadreyev, J. Pei, D. Kim, E. Kellogg, F. DiMaio, O. Lange, L. Kinch, W. Sheffler, B.-H. Kim, R. Das, N. V. Grishin, and D. Baker, "Structure prediction for CASP8 with all-atom refinement using Rosetta", *eng, Proteins*, vol. 77 Suppl 9, pp. 89–99, 2009, ISSN: 1097-0134. DOI: 10.1002/prot.22540.
- [51] T. A. Wassenaar, K. Pluhackova, R. A. Böckmann, S. J. Marrink, and D. P. Tieleman, "Going Backward: A Flexible Geometric Approach to Reverse Transformation from Coarse Grained to Atomistic Models", *J. Chem. Theory Comput.*, vol. 10, no. 2, pp. 676–690, Feb. 2014, ISSN: 1549-9618. DOI: 10.1021/ct400617g.
- [52] L. N. Kinch, W. Li, R. D. Schaeffer, R. L. Dunbrack, B. Monastyrskyy, A. Kryshtafovych, and N. V. Grishin, "CASP 11 target classification", *en, Proteins*, vol. 84, pp. 20–33, Sep. 2016, ISSN: 1097-0134. DOI: 10.1002/prot.24982.
- [53] *Special Issue: Eleventh Meeting on the Critical Assessment of Techniques for Protein Structure Prediction*, *Proteins: Structure, Function, and Bioinformatics*. Wiley-Blackwell, Sep. 2016, vol. 84.
- [54] T. A. Jones and S. Thirup, "Using known substructures in protein model building and crystallography", *eng, EMBO J.*, vol. 5, no. 4, pp. 819–822, Apr. 1986, ISSN: 0261-4189.

- [55] M. Boudard, J. Bernauer, D. Barth, J. Cohen, and A. Denise, "GARN: Sampling RNA 3d Structure Space with Game Theory and Knowledge-Based Scoring Strategies", *PLOS ONE*, vol. 10, no. 8, e0136444, 2015, ISSN: 1932-6203. DOI: 10.1371/journal.pone.0136444.
- [56] H. Kamisetty, S. Ovchinnikov, and D. Baker, "Assessing the utility of coevolution-based residue-residue contact predictions in a sequence- and structure-rich era", *Proc Natl Acad Sci U S A*, vol. 110, no. 39, pp. 15 674–15 679, Sep. 2013, ISSN: 0027-8424. DOI: 10.1073/pnas.1314045110.
- [57] P. Bradley, K. M. S. Misura, and D. Baker, "Toward High-Resolution de Novo Structure Prediction for Small Proteins", en, *Science*, vol. 309, no. 5742, pp. 1868–1871, Sep. 2005, ISSN: 0036-8075, 1095-9203. DOI: 10.1126/science.1113801.
- [58] J. Wang, R. M. Wolf, J. W. Caldwell, P. A. Kollman, and D. A. Case, "Development and testing of a general amber force field", en, *J. Comput. Chem.*, vol. 25, no. 9, pp. 1157–1174, Jul. 2004, ISSN: 1096-987X. DOI: 10.1002/jcc.20035.
- [59] K. Vanommeslaeghe, E. Hatcher, C. Acharya, S. Kundu, S. Zhong, J. Shim, E. Darian, O. Guvench, P. Lopes, I. Vorobyov, and A. D. Mackerell, "CHARMM general force field: A force field for drug-like molecules compatible with the CHARMM all-atom additive biological force fields", en, *J. Comput. Chem.*, vol. 31, no. 4, pp. 671–690, Mar. 2010, ISSN: 1096-987X. DOI: 10.1002/jcc.21367.
- [60] M. Karplus and G. A. Petsko, "Molecular dynamics simulations in biology", en, *Nature*, vol. 347, no. 6294, pp. 631–639, Oct. 1990, ISSN: 0028-0836. DOI: 10.1038/347631a0.
- [61] K. Henzler-Wildman and D. Kern, "Dynamic personalities of proteins", en, *Nature*, vol. 450, no. 7172, pp. 964–972, Dec. 2007, ISSN: 0028-0836. DOI: 10.1038/nature06522.
- [62] M. Tuckerman, *Statistical mechanics: theory and molecular simulation*. Oxford University Press, 2010.
- [63] E. C. Dykeman and O. F. Sankey, "Normal mode analysis and applications in biological physics", en, *J. Phys.: Condens. Matter*, vol. 22, no. 42, p. 423 202, 2010, ISSN: 0953-8984. DOI: 10.1088/0953-8984/22/42/423202.
- [64] S. M. LaValle and J. J. Kuffner Jr, "Rapidly-exploring random trees: Progress and prospects", 2000.
- [65] I. Al-Bluwi, T. Siméon, and J. Cortés, "Motion planning algorithms for molecular simulations: A survey", *Computer Science Review*, vol. 6, no. 4, pp. 125–143, 2012.
- [66] P. Yao, L. Zhang, and J.-C. Latombe, "Sampling-based exploration of folded state of a protein under kinematic and geometric constraints", *Proteins: Structure, Function, and Bioinformatics*, vol. 80, no. 1, pp. 25–43, 2012.
- [67] D. Devaurs, M. Vaisset, T. Siméon, and J. Cortés, "A multi-tree approach to compute transition paths on energy landscapes", in *Workshop on Artificial Intelligence and Robotics Methods in Computational Biology, AAAI'13*, 2013, pp–8.
- [68] D. Budday, R. Fonseca, S. Leyendecker, and H. Van den Bedem, "Frustration-Guided Motion Planning Reveals Conformational Transition in Proteins", *Proteins: Structure, Function, and Bioinformatics*, 2017.



- [69] D. Devaurs, T. Siméon, and J. Cortés, "Optimal Path Planning in Complex Cost Spaces With Sampling-Based Algorithms", *IEEE Transactions on Automation Science and Engineering*, vol. 13, no. 2, pp. 415–424, Apr. 2016, ISSN: 1545-5955. DOI: 10.1109/TASE.2015.2487881.
- [70] J. R. Lopéz-Blanco, J. I. Garzón, and P. Chacón, "iMod: Multipurpose normal mode analysis in internal coordinates", *Bioinformatics*, vol. 27, no. 20, pp. 2843–2850, 2011.
- [71] A. Y. L. Sim, M. Levitt, and P. Minary, "Modeling and design by hierarchical natural moves", en *PNAS*, vol. 109, no. 8, pp. 2890–2895, Feb. 2012, ISSN: 0027-8424, 1091-6490. DOI: 10.1073/pnas.1119918109.
- [72] J. Von Neumann and O. Morgenstern, "Theory of games and economic behavior, 2nd rev", 1947.
- [73] N. Nisan, T. Roughgarden, E. Tardos, and V. V. Vazirani, *Algorithmic game theory*. Cambridge University Press Cambridge, 2007, vol. 1.
- [74] R. Savani and B. von Stengel, "Game Theory Explorer: Software for the applied game theorist", *Computational Management Science*, vol. 12, no. 1, pp. 5–33, 2015.
- [75] Y. Viossat and A. Zapechelnyuk, "No-regret dynamics and fictitious play", *Journal of Economic Theory*, vol. 148, no. 2, pp. 825–842, Mar. 2013, ISSN: 0022-0531. DOI: 10.1016/j.jet.2012.07.003.
- [76] J. F. Nash and others, "Equilibrium points in n-person games", *Proceedings of the national academy of sciences*, vol. 36, no. 1, pp. 48–49, 1950.
- [77] R. J. Aumann, "Subjectivity and correlation in randomized strategies", *Journal of mathematical Economics*, vol. 1, no. 1, pp. 67–96, 1974.
- [78] D. P. Foster and R. Vohra, "Regret in the on-line decision problem", *Games and Economic Behavior*, vol. 29, no. 1, pp. 7–35, 1999.
- [79] S. Hart and A. Mas-Colell, "A Simple Adaptive Procedure Leading to Correlated Equilibrium", en *Econometrica*, vol. 68, no. 5, pp. 1127–1150, Sep. 2000, ISSN: 1468-0262. DOI: 10.1111/1468-0262.00153.
- [80] J. Hannan, "Approximation to Bayes risk in repeated play", *Contributions to the Theory of Games*, vol. 3, pp. 97–139, 1957.
- [81] S. Bubeck and N. Cesa-Bianchi, "Regret Analysis of Stochastic and Nonstochastic Multi-armed Bandit Problems", English, *MAL*, vol. 5, no. 1, pp. 1–122, Dec. 2012, ISSN: 1935-8237, 1935-8245. DOI: 10.1561/22000000024.
- [82] P. Auer, "Using confidence bounds for exploitation-exploration trade-offs", *Journal of Machine Learning Research*, vol. 3, no. Nov, pp. 397–422, 2002.
- [83] Y. Freund and R. E. Schapire, "Adaptive Game Playing Using Multiplicative Weights", *Games and Economic Behavior*, vol. 29, no. 1, pp. 79–103, Oct. 1999, ISSN: 0899-8256. DOI: 10.1006/game.1999.0738.
- [84] N. Littlestone and M. K. Warmuth, "The Weighted Majority Algorithm", *Information and Computation*, vol. 108, no. 2, pp. 212–261, Feb. 1994, ISSN: 0890-5401. DOI: 10.1006/inco.1994.1009.
- [85] A. Héliou, D. Budday, R. Fonseca, and H. van den Bedem, "Fast, clash-free RNA conformational morphing using molecular junctions", *Bioinformatics*, 2017. DOI: 10.1093/bioinformatics/btx127.

- [86] H. Van Den Bedem, I. Lotan, J.-C. Latombe, and A. M. Deacon, "Real-space protein-model completion: An inverse-kinematics approach", *Acta Crystallographica Section D: Biological Crystallography*, vol. 61, no. 1, pp. 2–13, 2005.
- [87] P. Yao, A. Dhanik, N. Marz, R. Propper, C. Kou, G. Liu, H. Van Den Bedem, J.-C. Latombe, I. Halperin-Landsberg, and R. B. Altman, "Efficient algorithms to explore conformation spaces of flexible protein loops", *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, vol. 5, no. 4, pp. 534–545, 2008.
- [88] 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*, vol. 83, pp. 36–47, 2015.
- [89] R. Fonseca, D. V. Pachov, J. Bernauer, and H. Van den Bedem, "Characterizing RNA ensembles from NMR data with kinematic models", *Nucleic Acids Research*, gku707, 2014.
- [90] B. J. Bender, A. Cisneros III, A. M. Duran, J. A. Finn, D. Fu, A. D. Lokits, B. K. Mueller, A. K. Sangha, M. F. Sauer, A. M. Sevy, and others, "Protocols for Molecular Modeling with Rosetta3 and RosettaScripts", *Biochemistry*, vol. 55, no. 34, pp. 4748–4763, 2016.
- [91] P. J. Cock, T. Antao, J. T. Chang, B. A. Chapman, C. J. Cox, A. Dalke, I. Friedberg, T. Hamelryck, F. Kauff, B. Wilczynski, and others, "Biopython: Freely available Python tools for computational molecular biology and bioinformatics", *Bioinformatics*, vol. 25, no. 11, pp. 1422–1423, 2009.
- [92] A. Ren, Y. Xue, A. Peselis, A. Serganov, H. M. Al-Hashimi, and D. J. Patel, "Structural and Dynamic Basis for Low-Affinity, High-Selectivity Binding of L-Glutamine by the Glutamine Riboswitch", English, *Cell Reports*, vol. 13, no. 9, pp. 1800–1813, Dec. 2015, ISSN: 2211-1247. DOI: 10.1016/j.celrep.2015.10.062.
- [93] X.-J. Lu, H. J. Bussemaker, and W. K. Olson, "DSSR: An integrated software tool for dissecting the spatial structure of RNA", *Nucleic Acids Res*, vol. 43, no. 21, e142–e142, Dec. 2015, ISSN: 0305-1048. DOI: 10.1093/nar/gkv716.
- [94] B. J. Tucker and R. R. Breaker, "Riboswitches as versatile gene control elements", *Current opinion in structural biology*, vol. 15, no. 3, pp. 342–348, 2005.
- [95] T. D. Ames and R. R. Breaker, "Bacterial aptamers that selectively bind glutamine", *RNA biology*, vol. 8, no. 1, pp. 82–89, 2011.
- [96] R. O. Dror, T. J. Mildorf, D. Hilger, A. Manglik, D. W. Borhani, D. H. Arlow, A. Philippsen, N. Villanueva, Z. Yang, M. T. Lerch, W. L. Hubbell, B. K. Kobilka, R. K. Sunahara, and D. E. Shaw, "Structural basis for nucleotide exchange in heterotrimeric G proteins", en, *Science*, vol. 348, no. 6241, pp. 1361–1365, Jun. 2015, ISSN: 0036-8075, 1095-9203. DOI: 10.1126/science.aaa5264.
- [97] C.-A. Roth, T. Dreyfus, C. H. Robert, and F. Cazals, "Hybridizing rapidly exploring random trees and basin hopping yields an improved exploration of energy landscapes", eng, *J Comput Chem*, vol. 37, no. 8, pp. 739–752, Mar. 2016, ISSN: 1096-987X. DOI: 10.1002/jcc.24256.
- [98] J. R. Thomas and P. J. Hergenrother, "Targeting RNA with Small Molecules", *Chem. Rev.*, vol. 108, no. 4, pp. 1171–1224, Apr. 2008, ISSN: 0009-2665. DOI: 10.1021/cr0681546.

- [99] M. H. Bailer, X. Sun, and H. M. Al-Hashimi, "Topology Links RNA Secondary Structure with Global Conformation, Dynamics, and Adaptation", en, *Science*, vol. 327, no. 5962, pp. 202–206, Jan. 2010, ISSN: 0036-8075, 1095-9203. DOI: 10.1126/science.1181085.
- [100] N. Furnham, T. L. Blundell, M. A. DePristo, and T. C. Terwilliger, "Is one solution good enough?", en, *Nat Struct Mol Biol*, vol. 13, no. 3, pp. 184–185, Mar. 2006, ISSN: 1545-9993. DOI: 10.1038/nsmb0306-184.
- [101] H. Van Den Bedem, A. Dhanik, J.-C. Latombe, and A. M. Deacon, "Modeling discrete heterogeneity in X-ray diffraction data by fitting multi-conformers", *Acta Crystallographica Section D: Biological Crystallography*, vol. 65, no. 10, pp. 1107–1117, 2009.
- [102] P. D. Adams, K. Aertgeerts, C. Bauer, J. A. Bell, H. M. Berman, T. N. Bhat, J. M. Blaney, E. Bolton, G. Bricogne, D. Brown, S. K. Burley, D. A. Case, K. L. Clark, T. Darden, P. Emsley, V. A. Feher, Z. Feng, C. R. Groom, S. F. Harris, J. Hendle, T. Holder, A. Joachimiak, G. J. Kleywegt, T. Krojer, J. Marcotrigiano, A. E. Mark, J. L. Markley, M. Miller, W. Minor, G. T. Montelione, G. Murshudov, A. Nakagawa, H. Nakamura, A. Nicholls, M. Nicklaus, R. T. Nolte, A. K. Padyana, C. E. Peishoff, S. Pieniazek, R. J. Read, C. Shao, S. Sheriff, O. Smart, S. Soisson, J. Spurlino, T. Stouch, R. Svobodova, W. Tempel, T. C. Terwilliger, D. Tronrud, S. Velankar, S. C. Ward, G. L. Warren, J. D. Westbrook, P. Williams, H. Yang, and J. Young, "Outcome of the First wwPDB/CCDC/D3r Ligand Validation Workshop", eng, *Structure*, vol. 24, no. 4, pp. 502–508, Apr. 2016, ISSN: 1878-4186. DOI: 10.1016/j.str.2016.02.017.
- [103] K. Cowtan, "The clipper project", *Joint CCP4 and ESF-EACBM Newsletter on Protein Crystallography*, vol. 40, 2002.
- [104] I. I. CPLEX, "V12. 1: User's Manual for CPLEX", *International Business Machines Corporation*, vol. 46, no. 53, p. 157, 2009.
- [105] P. Hall and C. C. Heyde, *Martingale limit theory and its application*, en. Academic Press, Jun. 1980, ISBN: 978-0-12-319350-6.
- [106] S. Kullback and R. A. Leibler, "On Information and Sufficiency", EN, *Ann. Math. Statist.*, vol. 22, no. 1, pp. 79–86, Mar. 1951, ISSN: 0003-4851, 2168-8990. DOI: 10.1214/aoms/1177729694.
- [107] J. W. Weibull, *Evolutionary game theory*. MIT press, 1997.
- [108] W. H. Sandholm, *Population games and evolutionary dynamics*. MIT press, 2010.
- [109] R. Laraki and P. Mertikopoulos, "Higher order game dynamics", *Journal of Economic Theory*, vol. 148, no. 6, pp. 2666–2695, Nov. 2013, ISSN: 0022-0531. DOI: 10.1016/j.jet.2013.08.002.
- [110] R. T. Rockafellar, *Convex analysis*, ser. Princeton Landmarks in Mathematics. Princeton University Press, Princeton, NJ, 1970, ISBN: 978-0-691-01586-6.
- [111] S. Shalev-Shwartz and others, "Online learning and online convex optimization", *Foundations and Trends® in Machine Learning*, vol. 4, no. 2, pp. 107–194, 2012.
- [112] P. Mertikopoulos and W. H. Sandholm, "Learning in Games via Reinforcement and Regularization", *Mathematics of OR*, vol. 41, no. 4, pp. 1297–1324, Aug. 2016, ISSN: 0364-765X. DOI: 10.1287/moor.2016.0778.
- [113] M. Benaïm, "Dynamics of stochastic approximation algorithms", in *Seminaire de probabilités XXXIII*, Springer, 1999, pp. 1–68.

- [114] J. C. Harsanyi, "Oddness of the number of equilibrium points: A new proof", en, *Int J Game Theory*, vol. 2, no. 1, pp. 235–250, Dec. 1973, ISSN: 0020-7276, 1432-1270. DOI: 10.1007/BF01737572.
- [115] S. Enterprises and others, "Scilab: Free and Open Source software for numerical computation", *Scilab Enterprises, Orsay, France*, p. 3, 2012.
- [116] J. Hofbauer and W. H. Sandholm, "Evolution in games with randomly disturbed payoffs", *Journal of Economic Theory*, vol. 132, no. 1, pp. 47–69, 2007.
- [117] "Primal-dual subgradient methods for convex problems", *Mathematical Programming*, author = Nesterov, Yurii, year = 2009, pages = 221–259, vol. 120, no. 1,
- [118] M. Levitt and S. Lifson, "Refinement of protein conformations using a macromolecular energy minimization procedure", *Journal of Molecular Biology*, vol. 46, no. 2, pp. 269–279, Dec. 1969, ISSN: 0022-2836. DOI: 10.1016/0022-2836(69)90421-5.
- [119] "Effective energy functions for protein structure prediction", *Current Opinion in Structural Biology*, author = Lazaridis, Themis and Karplus, Martin, year = 2000, pages = 139–145, vol. 10, no. 2,
- [120] G. Wang and R. L. Dunbrack, "PISCES: A protein sequence culling server", *Bioinformatics*, vol. 19, no. 12, pp. 1589–1591, 2003.
- [121] P. C. Ng and S. Henikoff, "SIFT: Predicting amino acid changes that affect protein function", eng, *Nucleic Acids Res.*, vol. 31, no. 13, pp. 3812–3814, Jul. 2003, ISSN: 1362-4962.
- [122] K. Molloy, N. Buhours, M. Vaisset, T. Siméon, E. Ferré, and J. Cortés, "A Reinforcement Learning Approach to Protein Loop Modeling", in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, Hamburg, Germany, Sep. 2015.
- [123] F. Guyon and P. Tufféry, "Fast protein fragment similarity scoring using a Binet-Cauchy kernel", eng, *Bioinformatics*, vol. 30, no. 6, pp. 784–791, Mar. 2014, ISSN: 1367-4811. DOI: 10.1093/bioinformatics/btt618.



## Appendix A

# Convex conjugates

### A.1 Gibbs entropy

#### A.1.1 Convex conjugate

Let  $\mathcal{S} = \{1, 2, \dots, n\}$  be a finite set and let  $\Delta \equiv \Delta(\mathcal{S})$  denote the  $(n - 1)$ -dimensional simplex spanned by  $\mathcal{S}$ . Let  $h(x) = \sum_{s \in \mathcal{S}} x_s \log(x_s)$  be a function defined on  $\Delta$ . Its *convex conjugate* is  $h^*(y) = \max_{x \in \Delta} \{\langle y|x \rangle - h(x)\}$ . We want to find an other expression of  $h^*(y)$ .

Using Lagrange multiplier we obtain  $\mathcal{L}(x, \lambda) = \langle y|x \rangle - h(x) + \lambda(\sum_{s \in \mathcal{S}} x_s - 1)$ . By derivation we have:

$$\begin{aligned} \frac{\partial \mathcal{L}}{\partial x_s}(x, \lambda) &= y_s - 1 - \log(x_s) + \lambda = 0, \quad \text{for all } s \in \mathcal{S} \\ \log(x_s) &= y_s - 1 + \lambda, \quad \text{for all } s \in \mathcal{S} \end{aligned} \tag{A.1}$$

$$x_s = \exp(y_s - 1 + \lambda), \quad \text{for all } s \in \mathcal{S}$$

$$\frac{\partial \mathcal{L}}{\partial \lambda} = \sum_{s \in \mathcal{S}} x_s - 1 = 0 \tag{A.2}$$

Summing (A.1) on  $s \in \mathcal{S}$  we obtain:

$$\sum_{s \in \mathcal{S}} x_s = \sum_{s \in \mathcal{S}} \exp(y_s - 1 + \lambda) = \exp(\lambda - 1) \sum_{s \in \mathcal{S}} \exp(y_s)$$

Combining this with (A.2) we obtain:

$$\exp(1 - \lambda) = \sum_{s \in \mathcal{S}} \exp(y_s)$$

Therefore

$$\arg \max_{x \in \Delta} \{\langle y|x \rangle - h(x)\} = (x_s)_{s \in \mathcal{S}} = \left( \frac{\exp(y_s)}{\sum_{s \in \mathcal{S}} \exp(y_s)} \right)_{s \in \mathcal{S}} = \Lambda(y) \tag{A.3}$$

And

$$\begin{aligned} h^*(y) &= \sum_{s \in \mathcal{S}} \left( y_s \frac{\exp(y_s)}{\sum_{s \in \mathcal{S}} \exp(y_s)} \right) - \sum_{s \in \mathcal{S}} \left( \frac{\exp(y_s)}{\sum_{s \in \mathcal{S}} \exp(y_s)} \right) (y_s - \log(\sum_{s \in \mathcal{S}} \exp(y_s))) \\ &= \log(\sum_{s \in \mathcal{S}} \exp(y_s)) \end{aligned} \tag{A.4}$$

### A.1.2 Strong convexity

Here we show that  $h(x) = x \log(x)$  is 1-strongly convex on  $[0, 1]$ .

Recall that  $h$  is  $K$ -strongly convex if and only if there exists some  $K > 0$  such that

$$h(tx + (1-t)x') \leq th(x) + (1-t)h(x') - \frac{1}{2}Kt(1-t)\|x' - x\|^2$$

First we prove it on  $]0, 1[$  using the Taylor's theorem on  $\log(x)$  which is  $C^\infty$  on  $]0, 1[$ . For  $x, x' \in ]0, 1[$ , let  $x'' = tx + (1-t)x'$  we have:

$$\begin{aligned} h(x'') &= h(tx + (1-t)x') \\ &= (tx + (1-t)x') \log(tx + (1-t)x') \\ &= tx \log(x + (1-t)(x' - x)) + (1-t)x' \log(x' + t(x - x')) \\ &\leq tx [\log(x) + (1-t)(x' - x) \frac{1}{x} - \frac{1}{2} \frac{1}{x^2} (1-t)^2 (x' - x)^2 + (1-t)^2 o((x' - x)^2)] \\ &\quad + (1-t)x' [\log(x') + t(x - x') \frac{1}{x'} - \frac{1}{2} \frac{1}{x'^2} t^2 (x - x')^2 + t^2 o((x' - x)^2)] \\ &= tx \log(x) + (1-t)x' \log(x') + t(1-t)(x' - x) + t(1-t)(x - x') \\ &\quad - \frac{1}{2} (1-t)t(x' - x)^2 \left( \frac{1-t}{x} + \frac{t}{x'} \right) + t(1-t) o((x' - x)^2) \\ &= tx \log(x) + (1-t)x' \log(x') - \frac{1}{2} (1-t)t(x' - x)^2 \left( \frac{1-t}{x} + \frac{t}{x'} \right) + o((x' - x)^2) \\ h(x'') &\leq tx \log(x) + (1-t)x' \log(x') - \frac{1}{2} (1-t)t(x' - x)^2 \end{aligned} \tag{A.5}$$

The last inequality comes from  $\frac{1}{x} \geq 1$  and  $\frac{1}{x'} \geq 1$  so  $\frac{1-t}{x} + \frac{t}{x'} \geq 1$ .  
By continuity of the last inequality in 0,  $h$  is 1-strongly convex on  $[0, 1]$ .

## A.2 General convex conjugates

Let  $\mathcal{S} = 1, 2, \dots, n$  be a finite set and let  $\Delta \equiv \Delta(\mathcal{S})$  denote the  $(n-1)$ -dimensional simplex spanned by  $\mathcal{S}$ . Let  $h(x) : \Delta \rightarrow \mathbb{R}$  a strongly convex function. Its convex conjugate is  $h^*(y) = \max_{x \in \Delta} \{ \langle y | x \rangle - h(x) \}$ .

Using Lagrange multiplier we obtain  $\mathcal{L}(x, \lambda) = \langle y | x \rangle - h(x) + \lambda(\sum_{s \in \mathcal{S}} x_s - 1)$ . By derivation we have:

$$\begin{aligned} \frac{\partial \mathcal{L}}{\partial x_s} &= y_s - \frac{\partial h}{\partial x_s}(x) + \lambda = 0, \quad \text{for all } s \in \mathcal{S} \\ \frac{\partial \mathcal{L}}{\partial \lambda} &= \sum_{s \in \mathcal{S}} x_s - 1 = 0 \\ \nabla h(x) &= y + \lambda \mathbf{1} \end{aligned} \tag{A.6}$$

Where  $\mathbf{1}$  is a vector of ones.

## Appendix B

# Properties from [113]

In the following,  $F : \mathbb{R}^m \rightarrow \mathbb{R}^m$  is a continuous map and  $\{x_n\}_{n \in \mathbb{N}}$  a time process living in  $\mathbb{R}^m$  whose general form can be written as

$$x_{n+1} - x_n = \gamma_{n+1}(F(x_n) + U_{n+1}) \quad (\text{B.1})$$

where  $\{\gamma_n\}_{n \geq 1}$  is a given sequence of nonnegative numbers such that

$$\sum_k \gamma_k = \infty, \quad \lim_{n \rightarrow \infty} \gamma_n = 0.$$

$U_n \in \mathbb{R}^m$  are perturbations.

$\tau_n = \sum_{i=1}^n \gamma_i$  for  $f \geq 1$  and  $\bar{U}$  is the continuous time process defined by

$$\bar{U}(\tau_n + s) = U_{n+1}$$

for all  $n \in \mathbb{N}, 0 \leq s \leq \gamma_{n+1}$ .

**Proposition 4.1** Let  $F$  be a continuous globally integrable vector field. Assume that:

**A 1** For all  $T > 0$

$$\lim_{n \rightarrow \infty} \sup \left\{ \left\| \sum_{i=n}^{k-1} \gamma_{i+1} U_{i+1} \right\| : k = n+1, \dots, m(\tau_n + T) \right\} = 0.$$

or equivalently

$$\lim_{t \rightarrow \infty} \Delta(t, T) = 0$$

with

$$\Delta(t, T) = \sup_{0 \leq h \leq T} \left\| \int_t^{t+h} \bar{U}(s) ds \right\|.$$

**A 2**  $\sup_n \|x_n\| < \infty$

Then the interpolated process  $X$  is an asymptotic pseudotrajectory of the flow  $\Phi$  induced by  $F$ .

**Proposition 4.2** Let  $\{x_n\}$  given by B.1 be a Robbins-Monro algorithm. Suppose that for some  $q \geq 2$

$$\sup_n E(\|U_{n+1}\|^q) < \infty$$

and

$$\sum_n \gamma_n^{1+\frac{q}{2}} < \infty.$$

Then assumption **A 1** of proposition 4.1 holds with probability 1.



**Remark 4.5** Propositions 4.2 and 4.4 assume a Robbins Monro type algorithm. However it is not hard to verify that the conclusions of these propositions continue to hold if  $\{x_n\}$  satisfies the more general recursion

$$x_{n+1} - x_n = \gamma_{n+1}(F(x_n) + U_{n+1} + b_{n+1})$$

where  $U_n$  is a martingale difference noise and  $\lim_{n \rightarrow \infty} b_n = 0$  almost surely.

Let  $X : \mathbb{R} \rightarrow M$  be an asymptotic pseudotrajectory of a semiflow  $\Phi$ . The *limit set*  $L(x)$  of  $X$  is the set of limits of convergent sequences  $X(t_k), t_k \rightarrow \infty$ . That is

$$L(X) = \cap \bar{X}([t, \infty)).$$

**Corollary 6.6** Assume that  $X$  is precompact,  $\Phi$  admits a strict Lyapounov function, and that there are countable many equilibria in  $L(X)$ . Then  $X(t)$  converges to an equilibrium as  $t \rightarrow \infty$ .



## Titre : Conformations moléculaires et théorie des jeux

**Mots clefs :** Structure, transition, protéine, ARN, théorie des jeux, convergence, équilibre

**Résumé :** Les protéines et acides ribonucléiques sont les principaux acteurs de nombreux processus cellulaires. Comprendre leurs fonctions, structures et interactions est un challenge important. Les méthodes expérimentales fournissent des informations sur la structure et la dynamique des molécules. Cependant les méthodes expérimentales sont limitées par le nombre de molécules qu'elles peuvent observer et les moyens qu'elles requièrent. Les méthodes de prédiction permettent d'obtenir des informations structurelles de façon quasi-automatique. Pour s'assurer de leur fiabilité, elles sont testées sur des données expérimentales. Nous présentons une procédure basée sur la cinétique inverse pour trouver une transition entre deux conformations d'un ARN tout en conservant sa structure secondaire. Nous obtenons des résultats comparables à l'état de l'art, ce qui montre que notre sélection des degrés de liberté est pertinente. De

plus, nous utilisons des données partielles, ce qui permet d'utiliser différents types de résultats expérimentaux. Nous abordons aussi le problème du repliement protéique par une approche de théorie des jeux. Nous représentons une protéine par un jeu où les joueurs sont les acides aminés et les stratégies, les angles dièdres. La prédiction de structure peut alors être vue comme la recherche d'un équilibre dans un jeu multi-joueur où les fonctions d'utilité correspondent à la qualité du repliement. Nous montrons que l'algorithme de non-regret, appelé *Hedge*, garantit l'élimination des stratégies dominées et la convergence locale vers un équilibre de Nash. Puis, en limitant notre analyse aux jeux de potentiel, nous montrons qu'une classe plus large d'algorithmes, les algorithmes de régularisation, convergent vers un équilibre de Nash presque sûrement.

## Title : Molecular conformations and game theory

**Keywords :** Structure, transition, protein, RNA, game theory, convergence, equilibrium

**Abstract :** Proteins and Ribonucleic Acids are the workhorses of many cellular processes. Understanding their functions, structures and interactions is an important challenge. Experimental methods provide actual information on structure and dynamics of molecules. However they have limitations: they cannot be applied to all molecules, and they need a lot of resources. Prediction methods are almost automatic ways of obtaining structural information. They are tested on experimental data to attest their reliability. We present, here, approaches tackling different problems. We develop a kinematics-based procedure to morph a RNA molecule between conformations while preserving its secondary structure. We obtain results comparable to state of the art methods showing that our selection of degrees of freedom is efficient. Furthermore

we only use sparse information allowing for various kinds of experimental inputs. We also look at the protein structure prediction problem from a game theory angle. We represent the protein dynamics as a game, in which players are amino acids and strategies are dihedral angles. The structure prediction can thus be seen as finding equilibrium in a multi-players game where all players have utility functions corresponding to the quality of the protein structure. We showed that a well-known no-regret algorithm, called *Hedge*, guarantees dominated strategies to vanish and a local convergence toward Nash equilibria. Furthermore restricting our analysis to potential games we showed that dual-averaging regularized learning algorithms converge toward a Nash equilibrium almost surely.

