The dynamics of complex systems. Studies and applications in computer science and biology

Christophe Guyeux

SPIM

Habilitation à Diriger des Recherches

UFC

école doctorale **sciences pour l'ingénieur et microtechniques**

UNIVERSITÉ DE FRANCHE-COMTÉ

The dynamics of complex systems. Studies and applications in computer science and biology

The dynamics of complex systems

Christophe GUYEUX

SPIM

Habilitation à Diriger des Recherches

école doctorale **sciences pour l'ingénieur et microtechniques**

UNIVERSITÉ DE FRANCHE-COMTÉ

HABILITATION À DIRIGER DES RECHERCHES

de l'Université de Franche-Comté

Spécialité : **Informatique**

présentée par

Christophe GUYEUX

The dynamics of complex systems. Studies and applications in computer science and biology

The dynamics of complex systems

Soutenue le 11 décembre 2013 devant le Jury composé de :

Jacques M. BAHI	Supervisor	Professor at the Université de Franche-Comté
Hamamache KHEDDOUCI	Reviewer	Professor at the Université de Lyon 1
Pierre SPITÉRI	Reviewer	Emeritus professor at IRIT-ENSEEIHT
Christian PARISOT	Reviewer	Privat Docent at the Université de Neuchâtel
Sylvie RUETTE	Examinator	Assistant professor (HDR) at the Université Paris Sud
Didier HOCQUET	Examinator	Professor at the Université de Franche-Comté
Christian MAIRE	Examinator	Professor at the Université de Franche-Comté

CONTENTS

REMERCIEMENTS

Je souhaite avant toutes choses à remercier mon directeur de thèse et d'hdr, le Professeur Jacques M. Bahi, pour ses conseils, sa disponibilité et son amitié. Il a su, malgré un emploi du temps bien chargé, toujours être présent à mes côtés, et me faire profiter de son expérience, son intelligence, et de sa connaissance si fine des objets de ma recherche. Le travail que j'ai pu mener et ce document ne seraient pas ce qu'ils sont sans sa confiance en mes recherches et la pertinence de ses remarques. Ce fut un grand plaisir de travailler avec lui, et j'espère pouvoir continuer à le faire longtemps encore.

Je tiens également à remercier Éric Filiol, Didier Hocquet, Hamamache Kheddouci, Christian Maire, Christian Parisot, Pierre Spitéri et Jean-Marc Steyaert, qui ont bien voulu être membres de mon jury d'HDR. Et plus particulièrement, merci à Hamamache Kheddouci, Christian Parisot, Jean-Marc Steyaert et Pierre Spitéri qui m'ont fait l'honneur d'être les rapporteurs de cette habilitation à diriger les recherches.

Je tiens aussi à remercier tous les membres de l'équipe AND pour leur amitié, leurs conseils, nos travaux en communs, et la bonne ambiance qu'ils contribuent à créer. Je remercie notamment Jean-François Couchot, Raphaël Couturiel, Morad Hakem, David Laiymani, et Michel Salomon pour leurs relectures de qualité et leurs conseils. Enfin, je remercie le Mésocentre de calculs de Franche-Comté et son équipe: sans eux, plusieurs travaux de ce manuscrit n'auraient jamais pu aboutir.

Je ne remercierai jamais assez mes parents et mon frère (et sa petite famille), pour avoir toujours été présents, m'avoir toujours aidé et soutenu. Sans eux, leur gentillesse, leurs encouragements et leur dévouement, je n'en serais pas là. Malgré la distance, ils sont toujours présents à mes côtés.

1

INTRODUCTION

1.1/ GENERAL PRESENTATION

During our thesis [Guy10], we have committed to study chaos of a certain restricted class of discrete dynamical systems, namely the chaotic iterations, whose topological behavior have never been studied, and to provide applications of such complex and so characteristic dynamics in some domains of information security (hash functions and digital watermarking). These researches are summed up in Chapter 2. Since then, we broadened our investigations field at the level of both theory and applications.

At theoretical level, we have not limited our research to the study of a particular class of discrete dynamical systems used by our team, with the mathematical theory of chaos, but we became increasingly interested to the complex dynamics as a whole, for discrete spaces and times. These dynamics can be complex due to their randomness, their chaos, or their complexity to the same named theory, etc. The studied dynamical systems, for their part, can be variations of chaotic iterations but also systems from computer science (sensor networks, neural networks, pseudorandom number generators...) and from biology (protein folding, genomes evolution...) Our approach consists in tracking down, modeling, and studying theoretically these complex dynamics that occur in biology and computer science, and to take benefits at applications level.

We have explained in our thesis how to construct finite state machines having a truly chaotic behavior. The key idea consists in decompartimentalizing the machine by using at each iterate new provided inputs when computing the output. By this process, even when the machine has a finite number of states, it not always enter into a loop, as the input is not necessarily periodic. Such thing has been formalized in our thesis using Turing machines whose behavior is chaotic, in the way that it is impossible to predict the effects of a slight alteration of the tape provided to the machine. Since then, we have continued to move forward with these chaotic Turing machines, by notably proposing a characterization of chaotic Moore machines and by developing applications on information hiding and digital watermarking, hash functions, and pseudorandom number generation. At each time, our machine for information security receives some data as input: an image to hash, a video stream to watermark, a pseudorandom sequence to rework, and so on. We can thus make that this treatment is chaotic in the mathematical sense of the term, and that an adversary cannot predict what will be the hash value, the watermarked media, the next bit of the generator, etc., knowing the past behavior of the machine. These applications have been deepened these three last years.

A last work on the theoretical study of chaotic finite state machines consisted of the effective construction of chaotic neural networks on the one hand, and of the demonstration that it is possible to prove that a given neural network is either chaotic or not on the other hand: they are simply finite state machines that receive new inputs at each iterate, and whose outputs can be either predicted or not depending on the complexity of the dynamic generated by the associated iterative system. Finally, artificial intelligence tools play an important role in some branches of information security like steganalysis: the detection of the presence of secrete information inside images is currently realized by using support vector machines or neural networks that learn to make the distinction between true natural images and steganographied ones. Using this learning, they must then be able to detect sleazy images in a given channel. We have shown that multilayer perceptrons (some neural networks) are not able to learn really chaotic dynamics, and have concluded for application purpose that steganalysers can be put into default using chaotic hiding methods. These researches are summarized in Chapter 3.

The application of complex dynamics to the information hiding field has been deepened in various directions these last three years. New hiding algorithms have been proposed, each of them having its own particularities: digital watermarking (without extraction) or steganography, robust of fragile, chaotic or not, inserting only one bit or a large amount of data, coupled with our pseudorandom generator, using or not the data contained in the host cover (via Canny filters for instance), and so on. Indeed, since our thesis, we have tried to develop another cryptographic approach for steganography. Except a few noteworthy theoretical works realized by the french school (Barbier, Filiol, Fontaine, Cayre, or Bas: see, *e.g.*, [BFM06, BA08, BM08b, BAM09, CB08b, KBB$^+$13]), this discipline was almost consisting in producing hiding methods and then to check if the hidden data is detectable with steganalyzers. Used tools were signal processing (to define insertion locations, and the features to take into account during steganalyzers conception), code theory, and artificial intelligence (to construct steganalyzers). In particular, to the best of our knowledge, no security proofs were produced (except for the stego-security [CB08b], a notion far from the rigor commonly in use in cryptology), and no security notion were clearly established since the ones of Barbier and Filiol [BM08b, BAM09]. As artificial intelligence (and thus steganalizers) seems to have difficulties to deal with chaos, we have started by studying the topological properties of algorithms we have previously proposed and of other algorithms that can be found in the state of the art. Then we have proven that almost all our algorithms are stego-secure, the only ones currently available with the natural watermarking of parameter $\eta = 1$. Finally, these security notions being not fully formed in our opinion, we have more recently introduced the beginning of a formal framework in which the security of a stego-system is defined with distinguishers in the complexity theory, and we have proposed the first security proofs in this rigorous framework close to usual standards. This additional knowledge is summarized in Chapter 5.

We were also interested in pseudorandom number generation (Chapter 4), a topic not investigated in our thesis. The starting point of our research in using chaotic dynamics for pseudorandom generation is that, in practice, the random character of generators is verified with statistical batteries of tests like DieHARD or TestU01: the tests embedded in these libraries try to find biases close to properties defined in the theory of chaos. For instance, the system must be intrinsically complicated to be chaotic according to Devaney: it cannot be separated in two subsystems more easy to study, which implies that orbits must visit the whole phase space. The same concern applies for generators, as various statistical tests aim at regarding whether some value sets are less produced than

other ones, or even are never returned by the generator. Similarly a chaotic system is supposed to have elements of regularity, while the good occurrence of regular orbits is commonly checked by various tests in the batteries mentioned above. In other words, if a recurrent sequence is such that it varies a little when its first term is a little changed (topological proof of non-sensitiveness), then statistical biases appear when the seed of the generator is modified. Our approach consists in taking one or more random sources as input (like a pseudorandom number generator or a physical source of entropy) and realizing a chaotic iterations based post-treatment on it, in such a way that the resulting chaotic iterations based generator possesses some provable properties of chaos. When the generators provided as inputs are defective, we have verified several times and on large data sets that the resulting generator has much better results on statistical tests. On the other hand, we have also proven that some interesting properties of the inputted generators are preserved by this post-treatment: their speed for instance, but also their cryptographically secure property. So we are able to preserve the good properties of the inputs, to add proven properties of chaos, while improving their statistical profile and without degradation of speed. Implementations on FPGA and GPU have been also proposed as well as a physical coupling with a chaotic laser.

In the same way, always concerning the applications of discrete complex systems to information security, we have pursued the study of the chaotic iterations based hash functions, which has just been initiated in our thesis. Contrarily to what has been realized in our thesis, we have not attempted to propose a new complete hash function but, as for the pseudorandom number generation, to realize a post-treatment of chaotic iterations on preexisting hash functions. By doing so, we obtain new hash functions proven as chaotic, and when the provided hash functions are defective (among other thing regarding their diffusion, confusion, or avalanche effect), the chaos properties have the effect of improving them, repairing these flaws. Additionally, we have proven that, when considering cryptographic hash functions, the new hash function resulting of this post-treatment continues to possess some security properties possibly possessed by the inputted function, like the first and second preimage resistance, or the resistance against collisions, which are defined in the complexity theory framework. Lastly, the hash functions we propose now are keyed ones, and they can be coupled with our generators based on chaotic iterations. As these research works are simple improvements of our thesis, we have only summarized them in Appendix C.

In all the previous applications, we have at each time tried to take benefits of complex dynamics, by proving their presence or by adding them when they were absent. We have also taken advantage on many occasions of these complex dynamics for sensor networks. A second field of investigations appeared as very interesting at the end of our thesis, due to our specialization in this domain: the modeling, the study, and the simulation of complex systems found in other disciplines than the computer science one. We have acquired particular skills on complex dynamics having the following form: an operation picked in a set of possible functions, and applied only on a variable subset of coordinates of the system. We found out that these particular complex dynamics appeared naturally and frequently in molecular biology, and more particularly in protein folding and in the evolution of genomes over time.

We have started by rewriting the model of protein folding usually used in conformation prediction tools, namely the 2D/3D HP square lattice model, thanks to a discrete dynamical system, and we have proven that this system exhibited various properties of chaos. Such a complex dynamic raises numerous questions. Firstly, the problem of predicting

the 3D conformation of a protein being proven as a NP-complete one, such a prediction is currently realized by using artificial intelligence tools. However, we have proven that at least some of these tools are not able to tackle some complex dynamics, like these we have found in the HP model. We can thus ask the question of the quality of the predicted conformations, especially since these conformations are infrequently compared to the reality due to cost reasons. Furthermore, we have discovered that some prediction tools were consisting in finding the 3D conformation ("protein") that minimizes the free enthalpy of self-avoiding walks (SAW) built by elongation, while other tools operate by folding the straight line having the size of the protein. We have proven that these two sets were disjoints, and that the proof of NP-completeness was true only for the first set. Thus, in the second case, dynamics seems too complex to be predicted by artificial intelligence tools, and the use of such tools is not a priori justified, as the problem is not currently proven as difficult in that case. To learn more on the difficulty of an optimization problem on the whole folded SAWs, we have deeply studied these particular walks. Among other things, we have: proposed various ways to formalize these folded walks, exhibited a characterization, proven that they are infinite in number, provided the shortest unfoldable walks currently known, bounded their number given a number of steps, proposed algorithms to study or generate them (with brute force, Monte-Carlo, and backtracking approaches deployed on the Mésocentre de calcul de Franche-Comté). Finally, we have computed a graphical software to study them, which can be freely downloaded. Let us remark that up-to-date biological researches tend to show that proteins are mainly "intrinsically disordered" and that our study of the chaos in protein folding should help people working in this field.

The second biological iterative system subjected to local modifications we have regarded is the genomes, which are modified during the course of their evolution, due to mutations, insertions-deletions of nucleotides, by changes with boarder amplitude (inversion, or simple copy or deletion of large DNA strains), or by other modifications specific to repetitions (segmentar duplication, tandem repeats, and move of transposable elements TEs). It has become evident to us that these various operations can be modified using the iterative systems we regularly used, and that a biomathematical modeling of evolution has not yet been realized, except for some types of mutations and some elementary models of transpositions of TEs. However we believed that being able to predict such an evolution, in the future as in the past, should be interesting for various reasons, like predicting the evolution of viruses such as HIV or influenza, reconstructing the past history or the common ancestor of bacteria strongly aggressive for our species, to better combat them (*M.tuberculosis*, *Y.pestis*, etc.), or to help the development of synthesis biology by simplifying *in silico* studies. We thus have proceeded to the modeling of various operations of genomic rearrangement, to their complexity study, and we have proven that, beside being complex, such dynamics can be predicted in some extend. We thus have rewritten and studied the nucleotides mutations as a discrete dynamical system and have generalized the so-called GTR mutation model. We have proposed mutation matrix models having 6 parameters and have completed the theoretical study of it with a colleague of Chrono-environnement. We have used these models on concrete cases of evolution on *S.cerevisiae*, and we now study, with colleagues of the Laboratoire de Mathématiques de Besançon (LMB), the possibility to infer a probability law of mutation on genes scale using graphical models, to take into account the fact that closed neighbors impact more largely the mutation probability than distant neighbors.

Concerning the move of TEs, inspired by transport equations and helped by colleagues of

the LMB, we have written partial differential equations to describe the density evolution of transposons (cut and paste mechanism of move) and retrotransposons (copy and paste), and we have proposed a branching model of these latter. These models require the knowledge of the initial conditions and reliable values for their sets of parameters, in order to use them in numerical simulations. This is why we have realized deep researches of TEs on *Drosophila* genomes, using home made dating and discovery algorithms. Other concrete bioinformatics applications have consolidated and enriched our knowledge of the genomes evolution. They will help us to refine our models and to populate bases of knowledge usable for predicting such evolution. For the sake of concision, only some of them will be evoked at the end of this manuscript.

1.2/ Publications Related to our Research Works

In what follows are listed all the publications related to our investigations. A star * means that authors are listed in alphabetic order. We have listed only publications posterior to 2011.

1.2.1/ Book

1. Jacques Bahi and Christophe Guyeux*. Discrete Dynamical Systems and Chaotic Machines: Theory and Applications. CRC Press, Juin 2013. 223 pages.

1.2.2/ Peer reviewed international journals and book chapters

1.2.2.1/ Book chapters

1. Raphaël Couturier and Christophe Guyeux*. Pseudorandom number generator on GPU. Designing Scientific Applications on GPUs, CRC press. *(*):***-***, 2013. Note: Accepted manuscript. To appear.

2. Christophe Guyeux and Jacques Bahi. A Topological Study of Chaotic Iterations. Application to Hash Functions. In CIPS, Computational Intelligence for Privacy and Security, volume 394 of Studies in Computational Intelligence, pages 51-73. Springer, 2012. Note: Revised and extended journal version of an IJCNN best paper (Core's rank: A).

1.2.2.2/ International journals

1. Jacques Bahi, Christophe Guyeux, and Abdallah Makhoul*. Two Security Layers for Hierarchical Data Aggregation in Sensor Networks. IJAACS, International Journal of Autonomous and Adaptive Communications Systems, 7(3):***-***, 2014.

2. Jacques M. Bahi, Christophe Guyeux, Kamel Mazouzi, and Laurent Philippe*. Computational investigations of folded self-avoiding walks related to protein folding. Computational biology and chemistry (Elsevier, I.F. 1.793). 22 pages. Accepted paper, to appear.

3. Christophe Guyeux, Nathalie M.-L. Côté, Wojciech Bienia, and Jacques Bahi. Is protein folding problem really a NP-complete one? first investigations. Journal of bioinformatics and computational biology (JBCB), 25 pages. Accepted paper, to appear.

4. Xiaole Fang, Benjamin Wetzel, Jean-Marc Merolla, John M. Dudley, Laurent Larger, Christophe Guyeux, Jacques M. Bahi. Noise and chaos contributions in fast random bit sequence generated from broadband optoelectronic entropy sources. IEEE Transactions on Circuits and Systems I (I.F. 2.24). *(*): ***-***, 2013. Note: 13 pages, Accepted manuscript. To appear.

5. Jacques Bahi, Xiaole Fang, Christophe Guyeux, and Laurent Larger*. FPGA Design for Pseudorandom Number Generator Based on Chaotic Iteration used in Information Hiding Application. Applied Mathematics & Information Sciences (I.F. 0.731), 7(6):2175-2188, 2013.

6. Jacques Bahi, Xiaole Fang, Christophe Guyeux, and Qianxue Wang*. Suitability of chaotic iterations schemes using XORshift for security applications. JNCA, Journal of Network and Computer Applications (Elsevier, I.F. 1.467), *(*):***-***, 2013. Note: Accepted manuscript. To appear.

7. Jacques M. Bahi, Christophe Guyeux, Antoine Perasso*. Predicting the Evolution of two Genes in the Yeast Saccharomyces Cerevisiae. (Elsevier Procedia Computer Science). 11: 4-16 (2012). Proceedings of CSBIO12 conference.

8. Jacques Bahi, Jean-François Couchot, and Christophe Guyeux*. Quality analysis of a chaotic proven keyed hash function. International Journal On Advances in Internet Technology, 5(1):26-33, 2012.

9. Jacques Bahi, Jean-François Couchot, and Christophe Guyeux. Steganography: a class of secure and robust algorithms. The Computer Journal (I.F. 0.755), 55(6):653-666, 2012.

10. Jacques Bahi, Jean-François Couchot, Christophe Guyeux, and Michel Salomon*. Neural Networks and Chaos: Construction, Evaluation of Chaotic Networks, and Prediction of Chaos with MultiLayer Feedforward Network. AIP Chaos, An Interdisciplinary Journal of Nonlinear Science (I.F. 2.188), 22(1):013122-1 - 013122-9, March 2012. Note: 9 pages.

11. Jacques Bahi, Nathalie Côté, Christophe Guyeux, and Michel Salomon. Protein Folding in the 2D Hydrophobic-Hydrophilic (HP) Square Lattice Model is Chaotic. Cognitive Computation (Springer, I.F. 0.867), 4(1):98-114, 2012.

12. Jacques Bahi, Christophe Guyeux, Abdallah Makhoul, and Congduc Pham*. Low Cost Monitoring and Intruders Detection using Wireless Video Sensor Networks. International Journal of Distributed Sensor Networks (I.F. 0.727), 2012, 2012. Note: 11 pages.

13. Jacques Bahi, Xiaole Fang, Christophe Guyeux, and Qianxue Wang*. Evaluating Quality of Chaotic Pseudo-Random Generators. Application to Information Hiding. IJAS, International Journal On Advances in Security, 4(1-2):118-130, 2011.

1.2.3/ PEER REVIEWED INTERNATIONAL CONFERENCES AND WORKSHOPS

1.2.3.1/ INTERNATIONAL CONFERENCES

1. Jacques Bahi, Jean-François Couchot, Nicolas Friot, Christophe Guyeux, and Kamel Mazouzi*. Quality Studies of an Invisible Chaos-Based Watermarking Scheme with Message Extraction. In IIH-MSP'13, 9th Int. Conf. on Intelligent Information Hiding and Multimedia Signal Processing, Beijing, China, pages ***-***, October 2013. Note: To appear.

2. Jacques Bahi, Nicolas Friot, and Christophe Guyeux*. Topological study and Lyapunov exponent of a secure steganographic scheme. In Javier Lopez and Pierangela Samarati, editors, SECRYPT'2013, Int. Conf. on Security and Cryptography. SECRYPT is part of ICETE - The International Joint Conference on e-Business and Telecommunications, Reykjavik, Iceland, pages ***-***, July 2013. SciTePress. Note: 8 pages. To appear. (acceptation rate: 13%)

3. Jacques Bahi, Christophe Guyeux, and Pierre-Cyrille Héam*. A Cryptographic Approach for Steganography. In IIH-MSP'13, 9th Int. Conf. on Intelligent Information Hiding and Multimedia Signal Processing, Beijing, China, pages ***-***, October 2013. Note: To appear.

4. Jacques Bahi, Jean-François Couchot, Nicolas Friot, and Christophe Guyeux*. A Robust Data Hiding Process Contributing to the Development of a Semantic Web. In INTERNET'2012, 4-th Int. Conf. on Evolving Internet, Venice, Italy, pages 71-76, June 2012.

5. Jacques Bahi, Xiaole Fang, and Christophe Guyeux*. An optimization technique on pseudorandom generators based on chaotic iterations. In INTERNET'2012, 4-th Int. Conf. on Evolving Internet, Venice, Italy, pages 31-36, June 2012.

6. Jacques Bahi, Nicolas Friot, and Christophe Guyeux*. Lyapunov exponent evaluation of a digital watermarking scheme proven to be secure. In IIH-MSP'2012, 8-th Int. Conf. on Intelligent Information Hiding and Multimedia Signal Processing, Piraeus-Athens, Greece, pages 359-362, July 2012. IEEE Computer Society. (acceptation rate: 15%)

7. Jacques Bahi, Christophe Guyeux, and Antoine Perasso*. Predicting the Evolution of Gene ura3 in the Yeast Saccharomyces Cerevisiae. In CSBio 2012: 3rd Int. Conf. on Computational Systems-Biology and Bioinformatics, volume 11 of Procedia Computer Science, Bangkok, Thailand, pages 4-16, October 2012.

8. Jacques Bahi, Jean-François Couchot, and Christophe Guyeux*. Performance Analysis of a Keyed Hash Function based on Discrete and Chaotic Proven Iterations. In INTERNET 2011, the 3-rd Int. Conf. on Evolving Internet, Luxembourg, Luxembourg, pages 52-57, June 2011. Note: Best paper award.

9. Jacques Bahi, Jean-François Couchot, and Christophe Guyeux*. Steganography: a Class of Algorithms having Secure Properties. In IIH-MSP-2011, 7-th Int. Conf. on Intelligent Information Hiding and Multimedia Signal Processing, Dalian, China, pages 109-112, October 2011.

10. Jacques Bahi, Jean-François Couchot, Christophe Guyeux, and Adrien Richard*. On the Link Between Strongly Connected Iteration Graphs and Chaotic Boolean Discrete-Time Dynamical Systems. In FCT'11, 18th Int. Symp. on Fundamentals of Computation Theory, volume 6914 of LNCS, Oslo, Norway, pages 126-137, August 2011. (Core's rank: A)

11. Jacques Bahi, Jean-François Couchot, Christophe Guyeux, and Qianxue Wang*. Class of Trustworthy Pseudo Random Number Generators. In INTERNET 2011, the 3-rd Int. Conf. on Evolving Internet, Luxembourg, Luxembourg, pages 72-77, June 2011.

12. Jacques Bahi, Nathalie Côté, and Christophe Guyeux*. Chaos of Protein Folding. In IJCNN 2011, Int. Joint Conf. on Neural Networks, San Jose, California, United States, pages 1948-1954, July 2011 (Core's rank: A).

13. Jacques Bahi, Xiaole Fang, Christophe Guyeux, and Qianxue Wang*. On the design of a family of CI pseudo-random number generators. In WICOM'11, 7th Int. IEEE Conf. on Wireless Communications, Networking and Mobile Computing, Wuhan, China, pages 1-4, September 2011.

14. Jacques Bahi, Christophe Guyeux, Abdallah Makhoul, and Congduc Pham*. Secure scheduling of wireless video sensor nodes for surveillance applications. In ADHOCNETS 11, 3rd Int. ICST Conference on Ad Hoc Networks, volume 89 of LNICST, Paris, France, pages 1-15, September 2011. Springer.

15. Jacques Bahi, Christophe Guyeux, and Michel Salomon*. Building a Chaotic Proven Neural Network. In ICCANS 2011, IEEE Int. Conf. on Computer Applications and Network Security, Maldives, Maldives, pages ***-***, May 2011.

16. Nicolas Friot, Christophe Guyeux, and Jacques Bahi. Chaotic Iterations for Steganography - Stego-security and chaos-security. In Javier Lopez and Pierangela Samarati, editors, SECRYPT'2011, Int. Conf. on Security and Cryptography. SECRYPT is part of ICETE - The International Joint Conference on e-Business and Telecommunications, Sevilla, Spain, pages 218-227, July 2011. SciTePress. (Acceptation rate: 14%)

1.2.3.2/ INTERNATIONAL WORKSHOPS

1. Jacques Bahi, Xiaole Fang, and Christophe Guyeux*. State-of-the-art in Chaotic Iterations based pseudorandom numbers generators Application in Information Hiding. In IHTIAP'2012, 1-st Workshop on Information Hiding Techniques for Internet Anonymity and Privacy, Venice, Italy, pages 90-95, June 2012.

2. Jacques Bahi, Jean-François Couchot, Nicolas Friot, and Christophe Guyeux*. Application of Steganography for Anonymity through the Internet. In IHTIAP'2012, 1-st Workshop on Information Hiding Techniques for Internet Anonymity and Privacy, Venice, Italy, pages 96-101, June 2012.

1.2.4/ SOFTWARE PATENTS

1. Jacques Bahi, Nicolas Friot, and Christophe Guyeux*. CIS-5, Programme sécurisé de stéganographie basé sur les itérations chaotiques, January 2012. Note: Produit logiciel. Numéro de dépôt APP : IDDN.FR.001.040023.00.S.P.2012.000.10800 (logiciel oeuvre de l'Université de Franche-Comté).

2. Jacques Bahi and Christophe Guyeux*. Black Cat, Fonction de hachage basée sur les itérations chaotiques, February 2011. Note: Produit logiciel. Numéro de dépôt APP : IDDN 11-090012-000 (logiciel oeuvre de l'Université de Franche-Comté).

3. Jacques Bahi and Christophe Guyeux*. CIPRNG, Chaotic Iteration based Pseudo-Random Number Generator, July 2011. Note: Produit logiciel. Numéro de dépôt APP : IDDN.FR 001.270007.000.S.P.2011.000.10800 (logiciel œuvre de l'Université de Franche-Comté).

4. Jacques Bahi and Christophe Guyeux*. TSC, Tatouage Sûr par Chaos, July 2011. Note: Produit logiciel. Numéro de dépôt APP : IDDN.FR 001.270010.000.S.P.2011.000.10800 (logiciel œuvre de l'Université de Franche-Comté).

1.2.5/ INVITED TALKS

1.2.5.1/ INVITATIONS TO INTERNATIONAL CONFERENCES

1. Christophe Guyeux. Discrete dynamical systems for predicting genomes' evolution. SDEDE 2014, Symposium on differential equations and difference equations, Homburg, Germany, 5th - 8th September 2014.

2. Christophe Guyeux. A Cryptographic Approach of Security in Wireless Sensor Networks. Theory and Practice. Invited talk at IFS 2012, Indo-French Symposium on Sensors Technologies and Systems, Delhi, India, March 2012.

1.2.5.2/ SEMINAR AND ORAL COMMUNICATIONS

1. Christophe Guyeux. Utilisation de systèmes dynamiques chaotiques pour la génération de nombres pseudo-aléatoires. Application en cryptographie. Deuxième journée LMB/FEMTO, 2 juillet 2013 à l'UFR ST, Besançon.

2. Christophe Guyeux. Approches bioinformatiques pour la reconstruction du génome ancestral de Mycobacterium tuberculosis. Le 28 mars 2013 à l'Institut Monod, Paris VI.

3. Christophe Guyeux. Premiers résultats dans la reconstruction du génome ancestral de Mycobacterium tuberculosis. Le 11 mars 2013, à l'institut de génétique et microbiologie de l'université Paris Sud.

4. Jacques Bahi, Jean-François Couchot, and Christophe Guyeux*. Steganography: secure and robust algorithms. Journées Codes et Stéganographie, Hôtel de la Monnaie, Rennes, France, March 2012.

5. Jacques Bahi, Christophe Guyeux, and Pierre-Cyrille Héam*. A Complexity Approach for Steganalysis. Journées Codes et Stéganographie, Hôtel de la Monnaie, Rennes, France, March 2012.
6. Nicolas Friot, Christophe Guyeux, and Jacques Bahi. A new secure process for steganography: CI2. Stego and topological security. Journées Codes et Stéganographie, Hôtel de la Monnaie, Rennes, France, March 2012.
7. Christophe Guyeux. Les itérations chaotiques et asynchrones : étude topologique et applications informatiques. Séminaire invité dans l'équipe "Topologie Dynamique" du département de mathématiques d'Orsay, Université Paris XI, June 2012.
8. Christophe Guyeux. Les modèles d'évolution des génomes : l'existant et ses améliorations possibles. Séminaire invité à l'Institut de Génétique et Microbiologie, UMR CNRS 8621, Paris, France, March 2012.
9. Christophe Guyeux. Quelques avancées pour une approche topologique en sécurité informatique. Séminaire invité CCA, Codage, Cryptologie, Algorithmes, Paris, France, February 2012.
10. Christophe Guyeux. Utilisation d'itérations chaotiques pour la génération de nombres pseudo-aléatoires. Séminaire invité dans l'équipe "Protection de l'information, cryptographie et codes" du département mathématiques et informatique de l'institut Xlim, Université de Limoges, May 2012.
11. Christophe Guyeux and Jacques Bahi. Étude topologique de l'étalement de spectre. Journées Codes et Stéganographie, Écoles Militaires de Saint-Cyr, Coëtquidan, January 2011.

1.2.6/ SUBMITTED ARTICLES

1.2.6.1/ PEER REVIEWED INTERNATIONAL JOURNALS

1. Christophe Guyeux, Jean-François Couchot, and Jacques M. Bahi. On the interest and realization of chaos-based information hiding schemes: a review. Journal of Chaos (Hindawi). Submitted on 30/10/13.
2. Qianxue Wang, Christophe Guyeux, and Jacques M. Bahi. The CIPRNGs family of chaotic iteration based post-treatments for pseudorandom number generators. International Journal of Bifurcation and Chaos.
3. Jacques Bahi, Jean-François Couchot, and Christophe Guyeux*. A chaos-based approach for information hiding security. IEEE Transactions on Signal Processing. Submitted on 30/08/13.
4. Jean-François Couchot, Raphael Couturier, and Christophe Guyeux*. STABYLO: a lightweight edge-based steganographic approach. The computer Journal. Submitted on 16/07/13. 5 pages double colonne.
5. Jacques M. Bahi, Christophe Guyeux, and Abdallah Makhoul*. A Security Framework for Wireless Sensor Networks: Theory and Practice. The Scientific World Journal. Submitted on 08/08/13, 13 pages double colonne.

6. Xiaole Fang, Christophe Guyeux, Jacques Bahi, and Qianxue Wang. An Efficient Design of Pseudorandom Number generator Based On Chaotic Iteration Lookup Table For Information Hiding Application. Security and Communication Networks (Wiley). Submitted on 31/07/13.

7. Wiem Elghazel, Kamal Medjaher, Christophe Guyeux, Mourad Hakem, Nourredine Zerhouni, and Jacques M. Bahi. Dependability of Sensor Networks for Prognostics and Health Management. Computers in industry (Elsevier). 29 pages.

8. Jacques M Bahi and Christophe Guyeux*. A Review of Chaotic Iteration Based Pseudorandom Number Generators. Journal of Chaos (Hindawi). Submitted on 30/10/13, 41 pages.

9. Jacques M. Bahi, Christophe Guyeux, and Antoine Perasso*. Chaos in DNA Evolution. International Journal of Biomathematics. Submitted on 06/02/13, 15 pages.

10. Jacques M. Bahi, Christophe Guyeux, and Antoine Perasso*. Relaxing the Symmetric Hypothesis in Genome Evolutionary Models. Journal of Biological Systems. Submitted on 23/05/13, 28 pages.

11. Jacques M. Bahi, Christophe Guyeux, Jean-Marc Nicod, and Laurent Philippe*. Protein structure prediction software generate two different sets of conformations Or the study of unfoldable self-avoiding walks. IEEE/ACM Transactions on Computational Biology and Bioinformatics. Submitted on 05/07/13, 26 pages.

12. Jacques M. Bahi, Raphaël Couturier, Christophe Guyeux, and Pierre-Cyrille Héam*. Efficient and Cryptographically Secure Generation of Chaotic Pseudorandom Numbers on GPU. Parallel computing (Elsevier). Submitted on 30/10/12, 28 pages.

13. Christophe Guyeux, Jacques M. Bahi, and Raphaël Couturier. Introducing the truly chaotic finite state machines and theirs applications in security field. International Journal of Unconventional Computing. Submitted on 08/09/13, 12 pages.

14. Christophe Guyeux, Abdallah Makhoul, and Jacques M. Bahi. Epidemiological approaches for data survivability in unattended wireless sensor networks: considering the sensors lifetime. New Generation Computing (Springer). Submitted on 30/09/13, 18 pages.

15. Caroline Bréchet, Julie Plantin, Marlène Sauget, Michelle Thouverez, Pascal Cholley, Christophe Guyeux, Didier Hocquet, Xavier Bertrand. The wastewater treatment plant's outflow and sludge release ESBL-producing Escherichia coli in the environment. Clinical infectious disease. Submitted on 30/09/13, 27 pages.

1.2.6.2/ PEER REVIEWED INTERNATIONAL CONFERENCE

1. Guyeux, C. and Couchot, J.-F. and Roland, J.-Y. and Al Kindy, B and Bahi, J.M. Reconstructing the last common ancestor of the Mycobacterium tuberculosis complex: a position paper. Submitted to Bioinformatics, the 5th international conference on bioinformatics models, methods and algorithms.

1.3/ NOTATIONS

In the whole document, to prevent from any conflicts and to avoid unreadable writings, we have considered the following notations, usually in use in discrete mathematics.

$\sharp X$ is the Cardinality of a set X and $\mathcal{P}(X)$ is the set of subsets of X. $\mathbb{B}$ stands for the set $\{0;1\}$ with its usual algebraic structure (Boolean addition, multiplication, and negation), while the symbols $\wedge$, $\vee$, and $\oplus$ mean respectively the AND, OR, and XOR Boolean operations. $\mathbb{N}$, $\mathbb{Z}$, $\mathbb{Q}$, and $\mathbb{R}$ are the usual notations of the following respective sets: natural numbers, integers (the natural numbers, zero, and the negatives of the natural numbers), rational numbers, and real numbers. $(\mathcal{M}_{m,n}(\mathbb{A}), +, \times, .)$ stands for the matrices algebra (m rows and n columns) on $\mathbb{A}$. $\mathcal{X}^{\mathcal{Y}}$ is the set of applications from $\mathcal{Y}$ to $\mathcal{X}$, and thus $\mathcal{X}^{\mathbb{N}}$ means the set of sequences belonging in $\mathcal{X}$. The set of congruence classes modulo n, for its part, is denoted as $\mathbb{Z}/n\mathbb{Z}$, while $[\![a;b]\!] = \{a, a+1, \ldots, b\}$ is the set of integers between a and b. Additionally,

- the $n-$th term of the sequence s is denoted by s^n,
- the $i-$th component of vector v is v_i,
- the $k-$th composition of function f is denoted by f^k. Thus $f^k = f \circ f \circ \ldots \circ f$, k times,
- the derivative of f is f'.

Example 1. *Let $u : \mathbb{N} \to \mathbb{R}^2$ a sequence of $\mathbb{R}^2$. Then u^0 is the first term of this sequence. This is a vector having two components: u_1^0 and u_2^0.*

We will use the notation $\lfloor x \rfloor$ for the integral part of a real x, that is, the greatest integer lower than x. $\lceil x \rceil$, for its part, will be the smallest integer greater than x.

I

THEORETICAL ASPECTS OF CHAOTIC MACHINES

2

Basic Recalls in Chaotic Finite State Machines

This chapter serves as a foundation for all this HDR manuscript. We recall in it the mathematical theory of chaos and the theoretical results regarding the chaotic finite state machines that have been obtained during our thesis: this approach will serve as canvas in our questioning about complex biological systems. Reader is referred to [BG13] for further investigations or for the proofs of results recalled here.

2.1/ Topological Study of Disorder

2.1.1/ Historical context

Recurrent sequences, also called discrete dynamical systems, of the form

$$u^0 \in \mathbb{R}, u^{n+1} = f(u^n), \tag{2.1}$$

with f continuous, have been well studied since the early years of mathematical analysis. They are widely used, for instance to resolve equations using a Newton method, or when approximating the solutions to differential equations using finite difference equations. The context study was the seek for convergence, which is for instance guaranteed when using monotonic functions or contractions. In the middle of the last century, Coppel has established a link between this desire of convergence and the existence of a cycle in iterations [Cop55]. More precisely, his theorem states that, considering the recurrent sequence with a function $f : I \longrightarrow I$ continuous on the line segment I, the absence of any 2-cycle implies the convergence of the discrete dynamical system.

This theorem establishes a clear link between the existence of a cycle of a given length and the convergence of the system. In other words, between cycles and order. Conversely, Li and Yorke have established in 1975 that the presence of a point of period three implies chaos in the same situation than previously [LY75]. By chaos, they mean the existence of points of any period: this kind of disorder, which is the first occurrence of the term “chaos” in the mathematical literature, is thus related to the multiplicity of periods. Since that time, the mathematical theory of chaos has known several developments to qualify or quantify the richness of chaos presented by a given discrete dynamical system, one of the most famous work, although old, being the one of Devaney [Dev89].

2.1.2/ ITERATIVE SYSTEMS

In the distributed computing community, dynamical systems have been generalized to take into account delay transmission or heterogeneous computational powers. Mathematically, the intended result is often one fixed point resulting from the iterations of a given function over a Boolean vector, considering that:

- at time t, x^t is computed using not only x^{t-1}, but potentially any $x^k, k < t$, due to delay transmission,
- not all the components of x^t are supposed to be updated at each iteration: each component represents a unit of computation, and these units have not the same processing frequency.

Some particular cases of these iterative systems are well documented, namely the serial, parallel, or chaotic modes. In the serial mode, each component is updated one by one, whereas the parallel mode consists in updating all the components at each iteration, leading to an usual discrete dynamical system.

Finally, iterative systems in chaotic mode, simply called *chaotic iterations*, are defined as follows [Rob86].

Definition 1. *Let* $f : \mathbb{B}^{\mathsf{N}} \longrightarrow \mathbb{B}^{\mathsf{N}}$ *and* $S \in \mathcal{P}\left([\![1, \mathsf{N}]\!]\right)^{\mathbb{N}}$ *a sequence called "chaotic strategy".* General chaotic iterations $(f, (x^0, S))$ *are defined by:*

$$\left\{ \begin{array}{l} x^0 \in \mathbb{B}^{\mathsf{N}} \\ \forall n \in \mathbb{N}^*, \forall i \in [\![1, \mathsf{N}]\!], x_i^n = \left\{ \begin{array}{ll} x_i^{n-1} & \text{if } i \notin S^n \\ f(x^{n-1})_i & \text{if } i \in S^n. \end{array} \right. \end{array} \right.$$

In other words, to obtain x^n, we compute $f(x^{n-1})$ and we only update in x^{n-1} the components whose indexes are into the chaotic strategy S^n.

A particular case occurs when the chaotic strategy is constituted by singletons: at each iteration, only one component is updated. Such "unary chaotic iterations" can thus be defined by $f : \mathbb{B}^{\mathsf{N}} \longrightarrow \mathbb{B}^{\mathsf{N}}$, $S \in \mathbb{S} = [\![1, \mathsf{N}]\!]^{\mathbb{N}}$, and

$$\left\{ \begin{array}{l} x^0 \in \mathbb{B}^{\mathsf{N}} \\ \forall n \in \mathbb{N}^*, \forall i \in [\![1, \mathsf{N}]\!], x_i^n = \left\{ \begin{array}{ll} x_i^{n-1} & \text{if } i \neq S^n \\ f(x^{n-1})_i & \text{if } i = S^n. \end{array} \right. \end{array} \right.$$

Example 2. *When considering the Boolean negation in unary chaotic iterations, and two integer sequences* p *and* q*, a pseudorandom generator we called CIPRNG(*p*,*q*) version 1 [BGW09, BGW10b] is obtained :* p *is* S *and the output of the generator is the subsequence* $\left(x^{\sigma(n)}\right)_{n \in \mathbb{N}}$*, where* $\sigma(0) = q^0$ *and* $\sigma(n+1) = \sigma(n) + q^n$*. Reason to be of the sequence* q *is that, between two iterates of unary chaotic iterations, at most 1 bit will change in the vector, and thus the sequence* (x^n) *cannot pass any statistical test: we must extract a subsequence* $(x^{\sigma(n)})$ *of* (x^n) *to produce the outputs. This generator is detailed in Section 4.3.1 of Chapter 4. The algorithm CIPRNG(*p*,*q*) version 2, for its part, extracts a subsequence from the strategy* $S = p$ *to prevent from negating several times a same position between two outputs (see Section A.2.1).*

Example 3. *If we consider the Boolean negation for f, then general chaotic iterations $(f,(x^0,S))$ of Definition 1 can be rewritten as: $x^{n+1}=x^n\oplus s^n$, where $s^n\in[\![0,2^{\mathsf{N}-1}]\!]$ is such that its $k-$th binary digit is 1 if and only if $k\in S^n$. Such a particular chaotic iterations is our generator called XOR CIPRNG, which is recalled in Chapter 4 too.*

Rem 1. *In most cases, chaotic iterations in this manuscript refers to unary chaotic iterations (the context always helps to determine it).*

A priori, there is no relation between these chaotic iterations and the mathematical theory of chaos evoked in the previous section. During our thesis [Guy10], we have regarded whether these chaotic iterations can behave chaotically, as it is defined for instance by Devaney, and if so, in which application context this behavior can be profitable. This questioning has led to a first necessary condition of non convergence [BCGG10].

Proposition 1. *Let $f:\mathbb{B}^{\mathsf{N}}\to\mathbb{B}^{\mathsf{N}}$ and $S\in[\![1,\mathsf{N}]\!]^{\mathbb{N}}$. If the chaotic iterations $\big(f,(x^0,S)\big)$ are not convergent, then:*

- *either f is not a contraction, meaning in the discrete mathematics context that there is no Boolean matrix M of size N satisfying $\forall x,y\in\mathbb{B}^{\mathsf{N}}$, $d(f(x),f(y))\leqslant Md(x,y)$, where d is here the "vectorial distance" $d(x,y)=\begin{pmatrix}\delta(x_1,y_1)\\ \vdots\\ \delta(x_{\mathsf{N}},y_{\mathsf{N}})\end{pmatrix}$, with δ is the discrete metric defined by $\delta(x,y)=\begin{cases}1 & \text{if } x\neq y,\\ 0 & \text{if } x=y\end{cases}$, and $\leqslant$ is the inequality term by term [Rob86].*
- *or S is not "pseudo-periodic": it is not constituted by an infinite succession of finite sequences, each having any element of $[\![1,\mathsf{N}]\!]$ at least once.*

The second alternative of the proposition above concerns the strategy, which should be provided by the outside world: in our thesis investigations regarding information security, it was typically a defective cryptographic tool we want to improve, on which such a pseudo-periodic property can be difficult to study. For instance, chaotic iterations can receive a PRNG S as input, and due to properties of disorder of f, generate a new pseudorandom sequence that presents better statistical properties than S. Having this approach in mind, we thus have searched vectorial Boolean iteration functions that are not contractions. The vectorial negation function $f_0:\mathbb{B}^{\mathsf{N}}\longrightarrow\mathbb{N}^{\mathsf{N}}$, $(x_1,\ldots,x_{\mathsf{N}})\longmapsto(\overline{x_1},\ldots,\overline{x_{\mathsf{N}}})$ is such a function, which explains why it is often used in our information security applications based on chaotic iterations.

The quantity of disorder generated by chaotic iterations, when satisfying the proposition above, has then been measured in [Guy10, BGW09]. To do so, chaotic iterations have firstly been rewritten as simple discrete dynamical systems, as follows.

2.1.3/ CHAOTIC ITERATIONS AS DYNAMICAL SYSTEMS

The problems raised by such a formalization can be summarized as follows. Chaotic iterations are defined in the discrete mathematics framework, considering $x^0\in\mathbb{B}^{\mathbb{N}}$, $S\in\mathbb{S}=[\![1,\mathsf{N}]\!]^{\mathbb{N}}$, and iterations having the form

$$x_i^{n+1}=\begin{cases}x_i^n & \text{if } i\neq S^n\\ f(x^n)_i & \text{if } i=S^n\end{cases}$$

where $f : \mathbb{B}^{\mathsf{N}} \to \mathbb{B}^{\mathsf{N}}$. However, the mathematical theory of chaos takes place into a topological space $(\mathcal{X}, \tau)$. It studies the iterations $x^0 \in \mathcal{X}$, $\forall n \in \mathbb{N}, x^{n+1} = f(x^n)$, where $f : \mathcal{X} \to \mathcal{X}$ is continuous for the topology τ.

To realize the junction between these two frameworks, the following material has been introduced [BGW09, Guy10]:

- the shift function: $\sigma : \mathbb{S} \longrightarrow \mathbb{S}, (S^n)_{n\in\mathbb{N}} \mapsto (S^{n+1})_{n\in\mathbb{N}}$,
- the initial function, defined by $i : \mathbb{S} \longrightarrow [\![1; \mathsf{N}]\!], (S^n)_{n\in\mathbb{N}} \mapsto S^0$,
- and $F_f : [\![1; \mathsf{N}]\!] \times \mathbb{B}^{\mathsf{N}} \longrightarrow \mathbb{B}^{\mathsf{N}}$,

$$(k, E) \longmapsto \left(E_j.\delta(k, j) + f(E)_k.\overline{\delta(k, j)}\right)_{j\in[\![1;\mathsf{N}]\!]}$$

where δ is the discrete metric. Let $\mathcal{X} = [\![1; \mathsf{N}]\!]^{\mathbb{N}} \times \mathbb{B}^{\mathsf{N}}$, and $G_f\,(S, E) = (\sigma(S), F_f(i(S), E))$. We have shown in [BGW09, Guy10] that chaotic iterations $\big(f, (S, x^0)\big)$ can be modeled by the discrete dynamical system:

$$\begin{cases} X^0 = (S, x^0) \in \mathcal{X}, \\ \forall k \in \mathbb{N}, X^{k+1} = G_f(X^k). \end{cases}$$

That is, at each iteration, we update the component whose index is the first term of the strategy, and we delete this first term in S. The topological disorder of chaotic iterations can then be studied. To do so, a relevant distance must be defined on $\mathcal{X}$, as follows [Guy10, BGW09]:

$$d((S, E); (\check{S}, \check{E})) = d_e(E, \check{E}) + d_s(S, \check{S}),$$

where $d_e(E, \check{E}) = \sum_{k=1}^{\mathsf{N}} \delta(E_k, \check{E}_k)$ and $d_s(S, \check{S}) = \frac{9}{\mathsf{N}} \sum_{k=1}^{\infty} \frac{|S^k - \check{S}^k|}{10^k}$.

This new distance has been introduced in [BGW09, BGW10b] to satisfy the following requirements.

- When the number of different cells between two systems is increasing, then their distance should increase too.
- In addition, if two systems present the same cells and their respective strategies start with the same terms, then the distance between these two points must be small because the evolution of the two systems will be the same for a while. Indeed, the two dynamical systems start with the same initial condition, use the same update function, and as strategies are the same for a while, then components that are updated are the same too.

The distance presented above follows these recommendations. Indeed, if the floor value $\lfloor d(X, Y) \rfloor$ is equal to n, then the systems $E, \check{E}$ differ in n cells. In addition, $d(X, Y) - \lfloor d(X, Y) \rfloor$ is a measure of the differences between strategies S and $\check{S}$. More precisely, this floating part is less than 10^{-k} if and only if the first k terms of the two strategies are equal. Moreover, if the k^{th} digit is nonzero, then the k^{th} terms of the two strategies are different.

We have then stated that,

Proposition 2. *$G_f : (\mathcal{X}, d) \to (\mathcal{X}, d)$ is a continuous function.*

With all this material, the study of chaotic iterations as a discrete dynamical system has been realized in our thesis. This study is summarized in the next section, for the sake of completeness of this manuscript.

2.1.4/ A TOPOLOGY FOR CHAOTIC ITERATIONS

The topological space on which chaotic iterations are defined has firstly been investigated, leading to the following result [GB12, Guy10]:

Proposition 3. *$\mathcal{X}$ is an infinitely countable metric space, being both compact, complete, and perfect (each point is an accumulation point).*

These properties proven in [Guy10] are required in some topological specific formalization of a chaotic dynamical system. Concerning G_{f_0}, it has been stated that.

Proposition 4. *G_{f_0} is surjective, but not injective, and so the dynamical system $(\mathcal{X}, G_{f_0})$, corresponding to particular unary chaotic iterations, is not reversible.*

Furthermore, if we denote by $Per_k(f)$ the set of periodic points of period k for f, we have $\forall k \in \mathbb{N}, Per_{2k+1}(G_{f_0}) = \varnothing, card\left(Per_{2k+2}(G_{f_0})\right) > 0$.

So G_{f_0} does not present the existence of points of any period referred as chaos in the article of Li and Yorke [LY75]. However [Guy10]:

- this kind of disorder can be stated for the general chaotic iterations (that is, on $\mathcal{X}^G = \mathcal{P}\left(\llbracket 1, \mathsf{N} \rrbracket\right)^{\mathbb{N}} \times \mathbb{B}^{\mathsf{N}}$),
- G_{f_0} possesses more than n^2 points of period $2n$.

Additionally, this existence of points of any period has been rejected by the community to the benefit of more recent notions of chaos, like those developed these last decades by Devaney [Dev89], Knudsen [Knu94a], etc. These approaches are recalled in the next section.

2.2/ THE MATHEMATICAL THEORY OF CHAOS

We will present in this section various understanding of a chaotic behavior for a discrete dynamical system. These properties will be stalked in complex systems found in computer science or bioinformatics, as detailed in upcoming chapters.

2.2.1/ APPROACHES SIMILAR TO DEVANEY

In these approaches, three ingredients are required for unpredictability. Firstly, the system must be intrinsically complicated, undecomposable: it cannot be simplified into two subsystems that do not interact, making any divide and conquer strategy applied to the

system inefficient. In particular, a lot of orbits must visit the whole space. Secondly, an element of regularity is added, to counteract the effects of the first ingredient, leading to the fact that closed points can behave in a completely different manner, and this behavior cannot be predicted. Finally, sensibility of the system is demanded as a third ingredient, making that closed points can finally become distant during iterations of the system. This last requirement is, indeed, often implied by the two first ingredients.

Having this understanding of an unpredictable dynamical system, Devaney has formalized in [Dev89] the following definition of chaos.

Definition 2. *A discrete dynamical system $x^0 \in \mathcal{X}, x^{n+1} = f(x^n)$ on a metric space $(\mathcal{X}, d)$ is said to be chaotic according to Devaney if it satisfies the three following properties:*

1. Transitivity: *For each couple of open sets $A, B \subset \mathcal{X}$, there exists $k \in \mathbb{N}$ such that $f^{(k)}(A) \cap B \neq \varnothing$.*

   Intuitively, a topologically transitive map has points that eventually move under iteration from one arbitrarily small neighborhood to any other. Consequently, the dynamical system cannot be decomposed into two disjoint open sets that are invariant under the map. Note that if a map possesses a dense orbit, then it is clearly topologically transitive.

2. Regularity: *Periodic points are dense in $\mathcal{X}$.*

3. Sensibility to the initial conditions: *There exists $\varepsilon > 0$ such that*

$$\forall x \in \mathcal{X}, \forall \delta > 0, \exists y \in \mathcal{X}, \exists n \in \mathbb{N}, d(x, y) < \delta \ and \ d(f^{(n)}(x), f^{(n)}(y)) \geqslant \varepsilon.$$

   Intuitively, a map possesses sensitive dependence on initial conditions if there exist points arbitrarily close to x that eventually separate from x by at least ε under iterations of f. Not all points near x need to eventually separate from x under iterations, but there must be at least one such point in every neighborhood of x. If a map possesses sensitive dependence on initial conditions, then for all practical purposes, the dynamics of the map defy numerical computation. Small errors in computation that are introduced by round-off may become magnified upon iteration. The results of numerical computation of an orbit, no matter how accurate, may bear no resemblance whatsoever with the real orbit.

When f is chaotic, then the system $(\mathcal{X}, f)$ is chaotic and quoting Devaney [Dev89]: "it is unpredictable because of the sensitive dependence on initial conditions. It cannot be broken down or decomposed into two subsystems which do not interact because of topological transitivity. And, in the midst of this random behavior, we nevertheless have an element of regularity." Fundamentally different behaviors are consequently possible and occur in an unpredictable way.

Instead of being transitive, the system can be intrinsically complicated for various other understanding of this wish that are not equivalent one another. Such understandings are:

- *Undecomposable*: the system is not the union of two nonempty closed subsets that are positively invariant ($f(A) \subset A$).
- *Total transitivity*: $\forall n \geqslant 1$, the function composition f^n is transitive.

- *Strong transitivity*: $\forall x, y \in \mathcal{X}$, $\forall r > 0$, $\exists z \in B(x, r)$, $\exists n \in \mathbb{N}$, $f^n(z) = y$.
- *Topological mixing*: for all pairs of disjoint open nonempty sets U and V, there exists $n_0 \in \mathbb{N}$ such that $\forall n \geqslant n_0, f^{(n)}(U) \cap V \neq \varnothing$.

Concerning the ingredient of sensibility, it can be reformulated as follows.

- $(\mathcal{X}, f)$ is *unstable* if all its points are unstable: $\forall x \in \mathcal{X}$, $\exists \varepsilon > 0$, $\forall \delta > 0$, $\exists y \in \mathcal{X}$, $\exists n \in \mathbb{N}$, $d(x, y) < \delta$ and $d(f^n(x), f^n(y)) \geqslant \varepsilon$.
- $(\mathcal{X}, f)$ is *expansive* if $\exists \varepsilon > 0$, $\forall x \neq y$, $\exists n \in \mathbb{N}$, $d(f^n(x), f^n(y)) \geqslant \varepsilon$: *all* the points in the neighborhood of any x will eventually separate from x during iterations.

This variety of definitions leads to various notions of chaos. For instance, a dynamical system is chaotic according to Wiggins if it is transitive and sensible to the initial conditions. It is said chaotic according to Knudsen [Knu94a, Knu94b] if it has a dense orbit while being sensible. Finally, we speak about expansive chaos when the properties of transitivity, regularity, and expansiveness are satisfied [For98, Rue01].

2.2.2/ Li-Yorke approach

The approach for chaos presented in the previous section, considering that a chaotic system is a system intrinsically complicated (undecomposable), with possibly an element of regularity and/or sensibility, has been completed by other understanding of chaos. Indeed, as "randomness" or "infiniteness", a single universal definition of chaos cannot be found. The kind of behaviors that are attempted to be described are too much complicated to enter into only one single definition. Instead, a large panel of mathematical descriptions have been proposed these last decades, being all theoretically justified. Each of these definitions give illustration to some particular aspects of a chaotic behavior.

The first of these parallel approaches can be found in the pioneer work of Li and Yorke [LY75]. In their well-known article entitled "Period three implies chaos", they rediscovered a weaker formulation of the Sarkovskii's theorem, meaning that when a discrete dynamical system $(f, [0, 1])$, with f continuous, has a 3-cycle, then it has too a $n-$cycle, $\forall n \geqslant 2$. The community has not adopted this definition of chaos, as several degenerated systems satisfy this property. However, on their article [LY75], Li and Yorke have studied too another interesting property, which has led to the notion of chaos "according to Li and Yorke" recalled below.

Let us firstly introduce the definition of Li-Yorke scrambled couple of points. This is points that never stop to oscillate.

Definition 3. *Let $(\mathcal{X}, d)$ a metric space and $f : \mathcal{X} \longrightarrow \mathcal{X}$ a continuous map. $(x, y) \in \mathcal{X}^2$ is a scrambled couple of points if $\liminf_{n\to\infty} d(f^n(x), f^n(y)) = 0$ and $\limsup_{n\to\infty} d(f^n(x), f^n(y)) > 0$.*

A scrambled set is a set in which any couple of points oscillates (are a scrambled couple).

Then,

Definition 4. *A Li-Yorke chaotic system is a system possessing an uncountable scrambled set.*

2.2.3/ TOPOLOGICAL ENTROPY APPROACH

The topological entropy of a topological dynamical system, firstly introduced in 1965 by Adler, Konheim, and McAndrew [AKM65], is a nonnegative real number that measures the complexity of the system. It represents the exponential growth rate of the number of distinguishable orbits of the iterates, for a system given by an iterated function. It can be formulated as follows.

Let $f : \mathcal{X} \longrightarrow \mathcal{X}$ be a continuous map on a compact metric space $(\mathcal{X}, d)$. For each natural number n, a new metric d_n is defined on $\mathcal{X}$ by

$$d_n(x, y) = \max\{d(f^i(x), f^i(y)) : 0 \leqslant i < n\}.$$

Given any $\varepsilon > 0$ and $n \geqslant 1$, two points of $\mathcal{X}$ are ε-close with respect to this metric if their first n iterates are ε-close. This metric allows one to distinguish in a neighborhood of an orbit the points that move away from each other during the iteration from the points that travel together.

A subset E of $\mathcal{X}$ is said to be (n, ε)-separated if each pair of distinct points of E is at least ε apart in the metric d_n. Denote by $N(n, \varepsilon)$ the maximum cardinality of a (n, ε)-separated set. $N(n, \varepsilon)$ represents the number of distinguishable orbit segments of length n, assuming that we cannot distinguish points within ε of one another.

Definition 5. *The topological entropy of the map f is defined by*

$$h(f) = \lim_{\epsilon \to 0} \left(\limsup_{n \to \infty} \frac{1}{n} \log N(n, \epsilon) \right).$$

The limit defining $h(f)$ may be interpreted as the measure of the average exponential growth of the number of distinguishable orbit segments. In this sense, it measures complexity of the topological dynamical system $(\mathcal{X}, f)$.

2.2.4/ THE LYAPUNOV EXPONENT

The last measure of chaos that has been regarded in our study of complex systems is the Lyapunov exponent. This quantity characterizes the rate of separation of infinitesimally close trajectories. Indeed, two trajectories in phase space with initial separation δ diverge at a rate approximately equal to $\delta e^{\lambda t}$, where λ is the Lyapunov exponent, which is defined below.

Definition 6. *Let $f : \mathbb{R} \longrightarrow \mathbb{R}$ be a differentiable function, and $x^0 \in \mathbb{R}$. The Lyapunov exponent is given by:*

$$\lambda(x^0) = \lim_{n \to +\infty} \frac{1}{n} \sum_{i=1}^{n} \ln \left| f' \left(x^{i-1} \right) \right|.$$

Obviously, this exponent must be positive to have a multiplication of initial errors, and thus chaos in this understanding.

Having all these definitions in mind, the topological behavior of chaotic iterations presented in Definition 1 have been studied in [BG10a, BG10d, GB12]. A good introduction to chaotic iterations and their topological properties can be found in [Guy10, Guy12], whereas [BG13] details other applications of these tools in computer science.

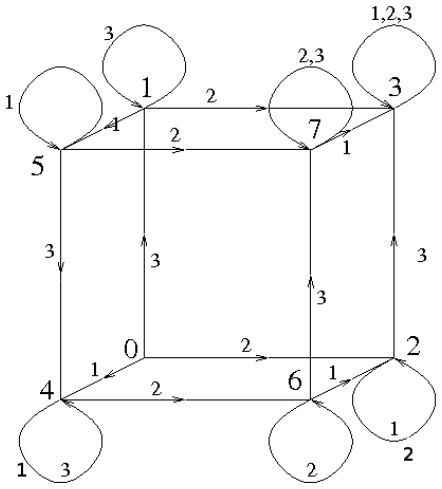


Figure 2.1: Example of an asynchronous iteration graph

2.3/ THE STUDY OF ITERATIVE SYSTEMS

2.3.1/ ON THE IMPORTANCE OF STRONGLY CONNECTED ASYNCHRONOUS ITERATION GRAPHS

It has firstly been stated that [BG10d, GB12]:

Theorem 1. *G_{f_0} is regular and transitive on $(\mathcal{X}, d)$, thus it is chaotic according to Devaney. Furthermore, its constant of sensibility is greater than* $\mathsf{N} - 1$.

Thus the set $\mathcal{C}$ of functions $f : \mathbb{B}^{\mathsf{N}} \longrightarrow \mathbb{B}^{\mathsf{N}}$ making the chaotic iterations of Definition 1 a case of chaos according to Devaney, is a nonempty set. To characterize functions of $\mathcal{C}$, it has been proven that transitivity implies regularity for these particular iterated systems [BCGR11]. To achieve characterization, the following graph has been introduced in [GB12, Guy10].

Let f be a map from $\mathbb{B}^{\mathsf{N}}$ to itself. The *asynchronous iteration graph* associated with f is the directed graph $\Gamma(f)$ defined by: the set of vertices is $\mathbb{B}^{\mathsf{N}}$; for all $x \in \mathbb{B}^{\mathsf{N}}$ and $i \in [\![1; \mathsf{N}]\!]$, the graph $\Gamma(f)$ contains an arc from x to $F_f(i, x)$. The relation between $\Gamma(f)$ and G_f is clear: there exists a path from x to x' in $\Gamma(f)$ if and only if there exists a strategy s such that the parallel iteration of G_f from the initial point (s, x) reaches the point x'. Figure 2.1 presents such an asynchronous iteration graph. It thus has been proven that [BCGR11].

Theorem 2. *G_f is transitive, and thus chaotic according to Devaney, if and only if $\Gamma(f)$ is strongly connected.*

This characterization makes it possible to quantify the number of functions in $\mathcal{C}$: it is equal to $\left(2^{\mathsf{N}}\right)^{2^{\mathsf{N}}}$. Then the study of the topological properties of disorder of these iterative systems has been further investigated, leading to the following results.

Theorem 3. *$\forall f \in \mathcal{C}$, $Per\,(G_f)$ is infinitely countable, G_f is strongly transitive and is chaotic according to Knudsen. It is thus undecomposable, unstable, and chaotic as defined by Wiggins.*

Theorem [4]. *$(\mathcal{X}, G_{f_0})$ is topologically mixing, expansive (with a constant equal to 1), chaotic as defined by Li and Yorke, and has a topological entropy and an exponent of Lyapunov both equal to $ln(\mathsf{N})$.*

2.3.2/ PRACTICAL RESOLUTION

Graphs whose strong connectivity is sought are constituted by 2^n vertices, each having n edges. For large values of n, to test such a property is a difficult task. We provided two solutions which are recalled below [BCGR11].

2.3.2.1/ ALGORITHMIC GENERATION OF STRONGLY CONNECTED GRAPHS

This section presents a first solution to compute a map f with a strongly connected graph of iterations $\Gamma(f)$. It is based on a generate and test approach.

We first consider the negation function f_0 whose iteration graph $\Gamma(f_0)$ is obviously strongly connected. Given a graph Γ, initialized with $\Gamma(f_0)$, the algorithm iteratively does the two following stages:

1. randomly select an edge of the current iteration graph Γ and
2. check whether the current iteration graph without that edge remains strongly connected (by a Tarjan algorithm for instance). In the positive case the edge is removed from Γ,

until a rate r of removed edges is greater than a threshold given by the user. If r is close to 0% (*i.e.*, few edges are removed), there should remain about $n \times 2^n$ edges. In the opposite case, if r is close to 100%, there are about 2^n edges left. In all cases, this step returns the last graph Γ that is strongly connected. It is then obvious to return the function f whose iteration graph is Γ.

Even if this algorithm always returns functions with strongly connected component (SCC) iteration graph, it suffers from iteratively verifying connectivity on the whole iteration graph, *i.e.*, on a graph with 2^n vertices. Next section tackles this problem: it presents sufficient conditions on a graph reduced to n elements that allow to obtain SCC iteration graph.

2.3.2.2/ SUFFICIENT CONDITIONS TO STRONGLY CONNECTED GRAPH

We are looking for maps f such that interactions between x_i and f_j make its iteration graph $\Gamma(f)$ strongly connected. We first need additional notations and definitions. For $x \in \mathbb{B}^n$ and $i \in [\![1; n]\!]$, we denote by $\overline{x}^i$ the configuration that we obtain be switching the $i-$th component of x, that is, $\overline{x}^i = (x_1, \ldots, \overline{x_i}, \ldots, x_n)$. Information interactions between the components of the system are obtained from the *discrete Jacobian matrix* f' of f, which is defined as being the map which associates to each configuration $x \in \mathbb{B}^n$, the $n \times n$ matrix

$$f'(x) = (f_{ij}(x)), \qquad f_{ij}(x) = \frac{f_i(\overline{x}^j) - f_i(x)}{\overline{x}^j_j - x_j} \qquad (i, j \in [\![1; n]\!]).$$

More precisely, interactions are represented under the form of a signed directed graph $G(f)$ defined by: the set of vertices is $[\![1;n]\!]$, and there exists an arc from j to i of sign $s \in \{-1,1\}$, denoted (j,s,i), if $f_{ij}(x) = s$ for at least one $x \in \mathbb{B}^n$. Note that the presence of both a positive and a negative arc from one vertex to another is allowed.

Let P be a sequence of arcs of $G(f)$ of the form

$$(i_1, s_1, i_2), (i_2, s_2, i_3), \ldots, (i_r, s_r, i_{r+1}).$$

Then, P is said to be a path of $G(f)$ of length r and of sign $\Pi_{i=1}^{r} s_i$, and i_{r+1} is said to be reachable from i_1. P is a *circuit* if $i_{r+1} = i_1$ and if the vertices $i_1,\ldots i_r$ are pairwise distinct. A vertex i of $G(f)$ has a positive (resp. negative) *loop*, if $G(f)$ has a positive (resp. negative) arc from i to itself.

Let $\alpha \in \mathbb{B}$. We denote by f^α the map from $\mathbb{B}^{n-1}$ to itself defined for any $x \in \mathbb{B}^{n-1}$ by

$$f^\alpha(x) = (f_1(x,\alpha), \ldots, f_{n-1}(x,\alpha)).$$

We denote by $\Gamma(f)^\alpha$ the subgraph of $\Gamma(f)$ induced by the subset $\mathbb{B}^{n-1} \times \{\alpha\}$ of $\mathbb{B}^n$.

Theorem [5]. *Let f be a map from $\mathbb{B}^n$ to itself such that:*

1. *$G(f)$ has no cycle of length at least two;*
2. *every vertex of $G(f)$ with a positive loop has also a negative loop;*
3. *every vertex of $G(f)$ is reachable from a vertex with a negative loop.*

Then, $\Gamma(f)$ is strongly connected.

At this stage, a new kind of chaotic, well-defined, and practically determinable iterative systems that only manipulates integers has been discovered, leading to the questioning of their computing for information security or numerical simulations of natural chaotic dynamics. In order to do so, the possibility of their computation without any loss of chaotic properties has first been investigated in [Guy10]. These chaotic machines, the last part of our thesis recall, are presented in the next section.

2.4/ From Theory to Practice

2.4.1/ How to cope with the problem of finite state machines

The two main problems raised by the common way to implement chaotic sequences on finite state machines are: (1) Chaotic sequences are usually defined in the real line whereas to define real numbers on computers is impossible. (2) All finite state machines always enter into a cycle when iterating, and this periodic behavior cannot really be stated as chaotic.

The first problem is disputable, as the shadow lemma proves that, when considering the sequence $x^{n+1} = trunc_k\left(f(x^n)\right)$, where $(f,[0,1])$ is a chaotic dynamical system and $trunc_k(x) = \dfrac{\lfloor 10^k x \rfloor}{10^k}$ is the truncated version of $x \in \mathbb{R}$ at its $k-$th digits, then the sequence (x^n) is as close as possible to a real chaotic orbit. Thus iterating a chaotic function on

floating point numbers does not deflate the chaotic behavior as much. However, even if this first claim is not really a problem, we have during our thesis researches prevent from any disputation by considering a tool (the chaotic iterations) that only manipulates integers bounded by N.

The second claim is surprisingly never considered as an issue when regarding cryptography or the generation of randomness on computers, whereas it is often reported when considering chaotic sequence generation. This issue can be solved by considering that the chaotic strategy, which is provided to the chaotic iterations that constitute our generator, *is not computed in our finite machine, but it is obtained at each iteration from the "outside world"*: encrypted video streams, microphones, or any other microscopic phenomenon that generates a low-level, statistically random "noise" signal, such as thermal noise, the photoelectric effect, or other quantum phenomenon. Indeed:

- If the chaotic strategy is a sequence (*e.g.*, provided by a pseudorandom number generator) computed into the finite machine, then it is periodic, and the output of chaotic iterations is periodic too (if the seed of the inputted PRNG is not regularly updated). Chaos, in that situation, is an abuse of language. However the $2^{4000000}$ possible states of common machines (4 Go of RAM memory) and the shadow lemma enable us, in a certain extend, to make such an abuse.

- If the chaotic strategy is a non periodic sequence provided at each iterate as input to the finite machine, then for well-chosen iteration functions, the outputs of chaotic iterations can be proven as truly chaotic (and non periodic) for the definitions recalled above. Roughly speaking, in that case, the stated problem can be solved in the following way. The computer must generate an output O computed from its current state E *and* the current value of the non periodic input S, which changes at each iteration (Fig. 2.2). Therefore, it is possible that the machine presents the same state twice, but with two future evolution completely different, depending on the values of the input. By doing so, we thus obtain a machine with a finite number of internal states, which can evolve in infinitely different ways, due to the new values provided by the input stream at each iteration. Thus such a machine can behave chaotically, as defined in the Devaney's formulation.

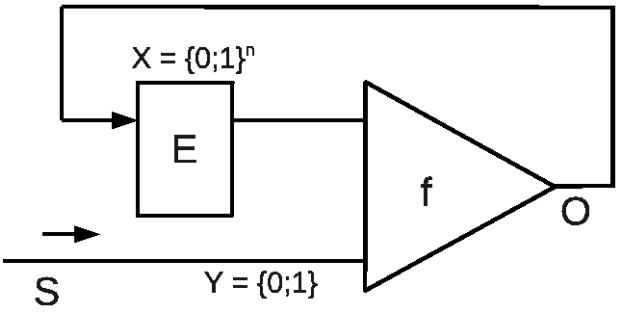


Figure 2.2: A chaotic finite-state machine. At each iteration, a new value is taken from the outside world (S). It is used by f as input together with the current state (E).

In the two situations, the first aims were initially related to information security. For instance, in pseudorandom number generators, the goal was to improve the statistics of the inputted (truly or pseudo) random sequence due to the topological properties of the

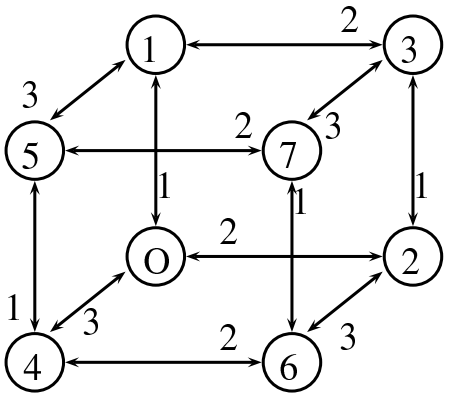


Figure 2.3: Asynchronous iteration graph $\Gamma(f_0)$ of the vectorial negation function.

iteration function, to add chaos properties when using truly random sequences, and finally to preserve the speed of the input[1]. For the pseudorandom input case, and in situation where this pseudorandom generator has proven properties (like some aspects of security), our hope was to preserve too such properties.

Example [4]. *A Moore machine whose directed graph is strongly connected is a chaotic finite-state machine in the most rigorous understanding of the term chaos. With such machines, the effects of an error or uncertainties on the inputs cannot be predicted. The ordered list of visited states and of machine's production can potentially be totally different after this error. For instance, Figure 2.3 depicts a chaotic Moore machine while it is not chaotic in Fig. 2.1. In the first case, if the initial state is 0 and the input word is 11111111, the visited states are 010101010, while it is 013232323 if the input word is 12111111: a slight change in the inputs lead to totally different visited states after this error.*

2.4.2/ EVALUATING CHAOS OF COMPUTER PROGRAMS

Conversely, any algorithm that uses at each iterate a new input taken from the outside world can potentially behaves chaotically, in the most rigorous definitions presented previously, and this behavior can be studied and proven theoretically. More precisely, let us consider a given algorithm. Because it must be computed one day, it is always possible to translate it as a Turing machine, and this last machine can be written as $x^{n+1} = f(x^n)$ in the following way. Let (w, i, q) be the current configuration of the Turing machine (Figure 2.4), where $w = \sharp^{-\omega} w(0) \dots w(k) \sharp^{\omega}$ is the paper tape, i is the position of the tape head, q is used for the state of the machine, and δ is its transition function (the notations used here are well-known and widely used). Following the Heam's proposal, we define f by:

- $f(w(0) \dots w(k), i, q) = (w(0) \dots w(i-1) a w(i+1) w(k), i+1, q')$, if $\delta(q, w(i)) = (q', a, \rightarrow)$,
- $f(w(0) \dots w(k), i, q) = (w(0) \dots w(i-1) a w(i+1) w(k), i-1, q')$, if $\delta(q, w(i)) = (q', a, \leftarrow)$.

[1] An interest of such an approach is that it is often very difficult to study mathematically the quality or security of a truly random generator based on physical noise. By the kind of post-treatment we propose, we can establish mathematical properties on the generator enriched by chaotic iterations.

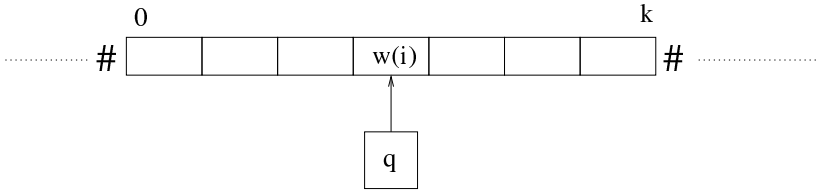


Figure 2.4: Turing Machine

Thus the Turing machine can be written as an iterate function $x^{n+1} = f(x^n)$ on a well-defined set $\mathcal{X}$, with x^0 as the initial configuration of the machine. Let τ be a well-chosen topology on $\mathcal{X}$. So the behavior of this dynamical system can be studied to know whether or not the algorithm is τ-chaotic.

Given a computer program, we wonder whether its complex behavior can be evaluated, understood, and/or used for applications. Such questionings, that have driven us from information security to bioinformatics, have firstly concerned the realization of chaotic machines more concrete than Turing or Moore ones. The results of such questioning are given in the next chapter.

3

Neural Networks and Chaos: Construction, Evaluation of Chaotic Networks, and Prediction of Chaos with MultiLayer Feedforward Network

Many research works deal with chaotic neural networks for various fields of application. Unfortunately, up to now these networks are usually claimed to be chaotic without any mathematical proof. The purpose of this chapter, which is a summary of our collaborations with Michel Salomon, Jean-François Couchot, and Jacques Bahi in this field [BGS11, BCGS12b], is to establish, based on a rigorous theoretical framework, an equivalence between chaotic iterations according to Devaney and a particular class of neural networks. On the one hand we show how to build such a network, on the other hand we provide a method to check if a neural network is a chaotic one. Finally, the ability of classical feedforward multilayer perceptrons to learn sets of data obtained from a dynamical system is regarded. Various Boolean functions are iterated on finite states. Iterations of some of them are proven to be chaotic as it is defined by Devaney. In that context, important differences occur in the training process, establishing with various neural networks that chaotic behaviors are far more difficult to learn.

3.1/ Introduction

Several research works have proposed or used chaotic neural networks these last years. The complex dynamics of such networks lead to various potential application areas: associative memories [CGH07] and digital security tools like hash functions [LDX10], digital watermarking [SJT+04, ZLW05], or cipher schemes [Lia09]. In the former case, the background idea is to control chaotic dynamics in order to store patterns, with the key advantage of offering a large storage capacity. For the latter case, the use of chaotic dynamics is motivated by their unpredictability and random-like behaviors. Indeed, investigating new concepts is crucial for the computer security field, because new threats are

constantly emerging.

Chaotic neural networks have been built with different approaches. In the context of associative memory, chaotic neurons like the nonlinear dynamic state neuron [CGH07] frequently constitute the nodes of the network. These neurons have an inherent chaotic behavior, which is usually assessed through the computation of the Lyapunov exponent. An alternative approach is to consider a well-known neural network architecture: the MultiLayer Perceptron (MLP). These networks are suitable to model nonlinear relationships between data, due to their universal approximator capacity [Cyb89, HSW89]. Thus, this kind of networks can be trained to model a physical phenomenon known to be chaotic such as Chua's circuit [DD10]. Sometimes a neural network, which is built by combining transfer functions and initial conditions that are both chaotic, is itself claimed to be chaotic [LDX10].

What all of these chaotic neural networks have in common is that they are claimed to be chaotic despite a lack of any rigorous mathematical proof. The first contribution since our thesis defense, regarding the chaotic machines, is to fill this gap using the mathematical theory of chaos. More precisely in this chapter, which summarizes [BGS11, BCGS12b], we establish the equivalence between chaotic iterations and a class of globally recurrent MLP. The second contribution is a study of the converse problem, indeed we investigate the ability of classical multilayer perceptrons to learn chaotic iterations, as defined previously. As such dynamical systems are chaotically iterated (as it is defined by Devaney) when the chosen function has a strongly connected iterations graph, we can thus experiment several MLPs and try to learn some iterations of this kind. We have shown that non-chaotic iterations can be learned, whereas it is far more difficult for chaotic ones. That is to say, we have discovered at least one family of problems with a reasonable size, such that artificial neural networks should not be applied due to their inability to learn chaotic behaviors in this context.

3.2/ A Chaotic Neural Network in the Sense of Devaney

Let us firstly define two functions f_0 and f_1 both in $\mathbb{B}^n \to \mathbb{B}^n$ that are used all along this chapter. The former is the already introduced vectorial negation, *i.e.*, $f_0(x_1, \ldots, x_n) = (\overline{x_1}, \ldots, \overline{x_n})$. The latter is $f_1(x_1, \ldots, x_n) = (\overline{x_1}, x_1, x_2, \ldots, x_{n-1})$. It is not hard to check that $\Gamma(f_0)$ and $\Gamma(f_1)$ are both strongly connected, then iterations of G_{f_0} and of G_{f_1} are chaotic according to Devaney.

With this material, we are now able to build a first chaotic neural network, as defined in the Devaney's formulation.

Let us build a multilayer perceptron neural network modeling $F_{f_0} : [\![1; n]\!] \times \mathbb{B}^n \to \mathbb{B}^n$ associated to the vectorial negation, where F_f has been defined in Chapter 2. More precisely, for all inputs $(s, x) \in [\![1; n]\!] \times \mathbb{B}^n$, the output layer will produce $F_{f_0}(s, x)$. It is then possible to link the output layer and the input one, in order to model the dependence between two successive iterations. As a result we obtain a global recurrent neural network that behaves as follows (see Fig. 3.1).

- The network is initialized with the input vector $\left(S^0, x^0\right) \in [\![1; n]\!] \times \mathbb{B}^n$ and computes the output vector $x^1 = F_{f_0}\left(S^0, x^0\right)$. This last vector is published as an output of the chaotic neural network and is sent back to the input layer through the feedback

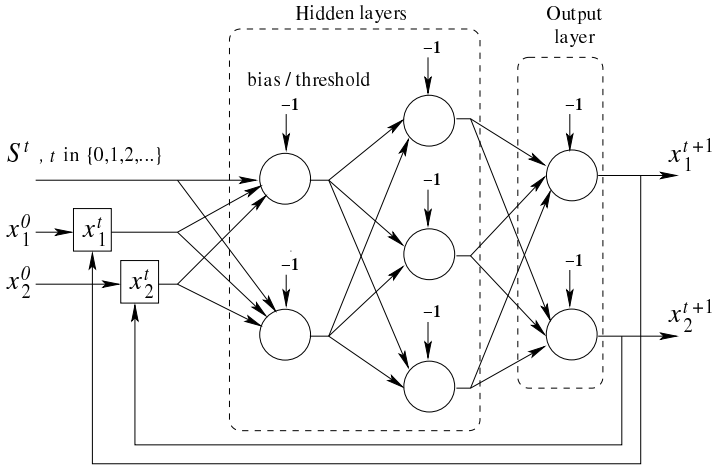


Figure 3.1: A perceptron equivalent to chaotic iterations

links.

- When the network is activated at the t^{th} iteration, the state of the system $x^t \in \mathbb{B}^n$ received from the output layer and the initial term of the sequence $(S^t)^{t\in\mathbb{N}}$ (*i.e.*, $S^0 \in [\![1;n]\!]$) are used to compute the new output vector. This new vector, which represents the new state of the dynamical system, satisfies:

$$x^{t+1} = F_{f_0}(S^0, x^t) \in \mathbb{B}^n \ .$$

The behavior of the neural network is such that when the initial state is $x^0 \in \mathbb{B}^n$ and a sequence $(S^t)^{t\in\mathbb{N}}$ is given as outside input, then the sequence of successive published output vectors $\left(x^t\right)^{t\in\mathbb{N}^*}$ is exactly the one produced by the chaotic iterations formally described in the equation above. It means that mathematically if we use similar input vectors they both generate the same successive outputs $\left(x^t\right)^{t\in\mathbb{N}^*}$, and therefore that they are equivalent reformulations of the iterations of G_{f_0} in $\mathcal{X}$. Finally, since the proposed neural network is built to model the behavior of G_{f_0}, whose iterations are chaotic according to the Devaney's definition of chaos, we can conclude that the network is also chaotic in this sense.

The previous construction scheme is not restricted to function f_0. It can be extended to any function f such that G_f is a chaotic map by training the network to model $F_f : [\![1;n]\!] \times \mathbb{B}^n \to \mathbb{B}^n$. Due to Theorem 2, we can find alternative functions f for f_0 through a simple check of their graph of iterations $\Gamma(f)$ (or using a more efficient check as recalled in previous chapter). For example, we can build another chaotic neural network by using f_1 instead of f_0.

3.3/ Checking Whether a Neural Network is Chaotic or Not

We focus now on the case where a neural network is already available, and for which we want to know if it is chaotic. Typically, in many research papers neural network are usually claimed to be chaotic without any convincing mathematical proof. We propose an

approach to overcome this drawback for a particular category of multilayer perceptrons defined below, and for the Devaney's formulation of chaos. In spite of this restriction, we think that this approach can be extended to a large variety of neural networks.

We consider a multilayer perceptron of the following form: inputs are n binary digits and one integer value, while outputs are n bits. Moreover, each binary output is connected with a feedback connection to an input one.

- During initialization, the network is seeded with n bits denoted $\left(x_1^0, \ldots, x_n^0\right)$ and an integer value S^0 that belongs to $[\![1; n]\!]$.
- At iteration t, the last output vector $\left(x_1^t, \ldots, x_n^t\right)$ defines the n bits used to compute the new output one $\left(x_1^{t+1}, \ldots, x_n^{t+1}\right)$. While the remaining input receives a new integer value $S^t \in [\![1; n]\!]$, which is provided by the outside world.

The topological behavior of these particular neural networks can be proven to be chaotic through the following process. Firstly, we denote by $F : [\![1; n]\!] \times \mathbb{B}^n \to \mathbb{B}^n$ the function that maps the value $(s, (x_1, \ldots, x_n)) \in [\![1; n]\!] \times \mathbb{B}^n$ into the value $(y_1, \ldots, y_n) \in \mathbb{B}^n$, where $(y_1, \ldots, y_n)$ is the response of the neural network after the initialization of its input layer with $(s, (x_1, \ldots, x_n))$. Secondly, we define $f : \mathbb{B}^n \to \mathbb{B}^n$ such that $f(x_1, x_2, \ldots, x_n)$ is equal to

$$\left(F\left(1, (x_1, x_2, \ldots, x_n)\right), \ldots, F\left(n, (x_1, x_2, \ldots, x_n)\right)\right) \ .$$

Thus, for any j, $1 \leqslant j \leqslant n$, we have $f(x_1, x_2, \ldots, x_n)_j = F\left(j, (x_1, x_2, \ldots, x_n)\right)$. If this recurrent neural network is seeded with $\left(x_1^0, \ldots, x_n^0\right)$ and $S \in [\![1; n]\!]^{\mathbb{N}}$, it produces exactly the same output vectors than the chaotic iterations of F_f with initial condition $\left(S, (x_1^0, \ldots, x_n^0)\right) \in [\![1; n]\!]^{\mathbb{N}} \times \mathbb{B}^n$. In other words, the output vectors of the MLP correspond to the sequence of configurations given by the equation above. Theoretically speaking, such iterations of F_f are thus a formal model of these kind of recurrent neural networks. In the remainder of this chapter, we will call such multilayer perceptrons "CI-MLP(f)", which stands for "Chaotic Iterations based MultiLayer Perceptron".

Checking if CI-MLP(f) behaves chaotically according to Devaney's definition of chaos is simple: we need just to verify if the associated graph of iterations $\Gamma(f)$ is strongly connected or not. As an incidental consequence, we finally obtain an equivalence between chaotic iterations and CI-MLP(f). Therefore, we can obviously study such multilayer perceptrons with mathematical tools like topology to establish, for example, their convergence or, contrarily, their unpredictable behavior. An example of such a study is given in the next section.

3.4/ Topological Properties of Chaotic Neural Networks

As recalled previously, it has been proven in our thesis that chaotic iterations are expansive and topologically mixing when f is the vectorial negation f_0. Consequently, these properties are inherited by the CI-MLP(f_0) recurrent neural network previously presented, which induces a greater unpredictability. Any difference on the initial value of the input layer is in particular magnified up to be equal to the expansiveness constant.

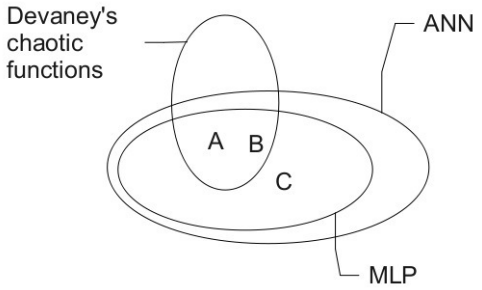


Figure 3.2: Summary of addressed neural networks and chaos problems

Let us then focus on the consequences for a neural network to be chaotic according to Devaney's definition. Topological transitivity property implies indecomposability. Hence, reducing the set of outputs generated by CI-MLP(f), in order to simplify its complexity, is impossible if $\Gamma(f)$ is strongly connected. Moreover, under this hypothesis CI-MLPs(f) are strongly transitive. Thus, for all pairs of points (x, y) in the phase space, a point z can be found in the neighborhood of x such that one of its iterates $f^n(z)$ is y. Among other things, the strong transitivity leads to the fact that without the knowledge of the initial input layer, all outputs are possible. Additionally, no point of the output space can be discarded when studying CI-MLPs: this space is intrinsically complicated and it cannot be decomposed or simplified.

Furthermore, these recurrent neural networks exhibit the instability property. This property, which is implied by the sensitive point dependence on initial conditions, leads to the fact that in all neighborhoods of any point x, there are points that can be apart by ε in the future through iterations of the CI-MLP(f). Thus, we can claim that the behavior of these MLPs is unstable when $\Gamma(f)$ is strongly connected.

Figure 3.2 is a summary of addressed neural networks and chaos problems. In Section 3.2 we have explained how to construct a truly chaotic neural networks, A for instance. Section 3.3 has shown how to check whether a given MLP A or C is chaotic or not in the sense of Devaney, and how to study its topological behavior. Another relevant point to investigate, when studying the links between neural networks and Devaney's chaos, is to determine whether a multilayer perceptron C is able to learn or predict some chaotic behaviors of B. This statement is studied in the next section.

3.5/ Suitability of Feedforward Neural Networks for Predicting Chaotic and Non-chaotic Behaviors

In the context of computer science different topic areas have an interest in chaos, such as steganographic techniques that are detailed in Chapter 5. Steganography consists in embedding a secret message within an ordinary one, while the secret extraction takes place once at destination [SJT+04, ZLW05]. The reverse (*i.e.*, automatically detecting the presence of hidden messages inside media) is called steganalysis. Among the deployed strategies inside detectors, there are support vectors machines [QSL09], neural networks [SHW03,HOZS10], and Markov chains [SMCM06]. Most of these detectors give quite good results and are rather competitive when facing steganographic tools. However, to the best of our knowledge none of the information hiding schemes that have been ste-

ganalyzed fulfills the Devaney definition of chaos [Dev89]. As we have proposed in our thesis a chaotic stego-system, one can wonder whether detectors continue to give good results when facing truly chaotic schemes. More generally, there remains the open problem of deciding whether artificial intelligence is suitable for predicting topological chaotic behaviors.

3.5.1/ REPRESENTING CHAOTIC ITERATIONS FOR NEURAL NETWORKS

The problem of deciding whether classical feedforward ANNs are suitable to approximate topological chaotic iterations may then be reduced to evaluate such neural networks on iterations of functions with strongly connected graph of iterations. To compare with non-chaotic iterations, the experiments detailed in the following sections are carried out using both kinds of function (chaotic and non-chaotic). Let us emphasize on the difference between this kind of neural networks and the chaotic iterations based multilayer perceptron.

We are then left to compute two disjoint function sets that contain either functions with topological chaos properties or not, depending on the strong connectivity of their iterations graph. As stated in the previous chapter, this can be achieved for instance by removing a set of edges from the iteration graph $\Gamma(f_0)$ of the vectorial negation function f_0. One can deduce whether a function verifies the topological chaos property or not by checking the strong connectivity of the resulting graph of iterations.

For instance let us consider the functions f and g from $\mathbb{B}^4$ to $\mathbb{B}^4$ respectively defined by the following lists:

$$[0, 0, 2, 3, 13, 13, 6, 3, 8, 9, 10, 11, 8, 13, 14, 15]$$

$$\text{and } [11, 14, 13, 14, 11, 10, 1, 8, 7, 6, 5, 4, 3, 2, 1, 0] \ .$$

In other words, the image of 0011 by g is 1110: it is obtained as the binary value of the fourth element in the second list (namely 14). It is not hard to verify that $\Gamma(f)$ is not strongly connected (*e.g.*, $f(1111)$ is 1111) whereas $\Gamma(g)$ is. The remaining of this section shows how to translate iterations of such functions into a model amenable to be learned by an ANN. Formally, input and output vectors are pairs $((S^t)^{t\in\mathbb{N}}, x)$ and $\big(\sigma((S^t)^{t\in\mathbb{N}}), F_f(S^0, x)\big)$ as defined previously.

Firstly, let us focus on how to memorize configurations. Two distinct translations are proposed. In the first case, we take one input in $\mathbb{B}$ per component; in the second case, configurations are memorized as natural numbers. A coarse attempt to memorize configuration as natural number could consist in labeling each configuration with its translation into decimal numeral system. However, such a representation induces too many changes between a configuration labeled by a power of two and its direct previous configuration: for instance, 16 (10000) and 15 (01111) are close in a decimal ordering, but their Hamming distance is 5. This is why Gray codes [Gra53] have been preferred.

Secondly, let us detail how to deal with strategies. Obviously, it is not possible to translate in a finite way an infinite strategy, even if both $(S^t)^{t\in\mathbb{N}}$ and $\sigma((S^t)^{t\in\mathbb{N}})$ belong to $\{1, \ldots, n\}^{\mathbb{N}}$. Input strategies are then reduced to have a length of size $l \in [\![2, k]\!]$, where k is a parameter of the evaluation. Notice that l is greater than or equal to 2 since we do not want the shift σ function to return an empty strategy. Strategies are memorized as natural numbers expressed in base $n + 1$. At each iteration, either none or one component is modified (among the n components) leading to a radix with $n + 1$ entries. Finally, we give

another input, namely $m \in [\![1, l-1]\!]$, which is the number of successive iterations that are applied starting from x. Outputs are translated with the same rules.

To address the complexity issue of the problem, we have computed the size of the data set an ANN has to deal with in [BCGS12b]. We have obtained that the number of input-output pairs for our ANNs is

$$2^n \times \left(\frac{(k-1) \times n^{k+1}}{n-1} - \frac{n^{k+1} - n^2}{(n-1)^2} \right) .$$

For instance, for 4 binary components and a strategy of at most 3 terms we obtain 2304 input-output pairs.

3.5.2/ EXPERIMENTS

To study if chaotic iterations can be predicted, we have chosen in [BCGS12b] to train the multilayer perceptron. As stated before, this kind of network is in particular well known for its universal approximation property [Cyb89, HSW89]. Furthermore, MLPs have been already considered for chaotic time series prediction. In [DD10] for instance, the authors have shown that a feedforward MLP with two hidden layers, and trained with Bayesian Regulation back-propagation, can learn successfully the dynamics of Chua's circuit.

In these experiments we consider MLPs having one hidden layer of sigmoidal neurons and output neurons with a linear activation function. They are trained using the Limited-memory Broyden-Fletcher-Goldfarb-Shanno quasi-newton algorithm in combination with the Wolfe linear search. The training process is performed until a maximum number of epochs is reached. To prevent overfitting and to estimate the generalization performance we use holdout validation by splitting the data set into learning, validation, and test subsets. These subsets are obtained through random selection such that their respective size represents 65%, 10%, and 25% of the whole data set.

Several neural networks are trained for both iterations coding schemes. In both cases iterations have the following layout: configurations of four components and strategies with at most three terms. Thus, for the first coding scheme a data set pair is composed of 6 inputs and 5 outputs, while for the second one it is respectively 3 inputs and 2 outputs. As noticed at the end of the previous section, this leads to data sets that consist of 2304 pairs. The networks differ in the size of the hidden layer and the maximum number of training epochs. We remember that to evaluate the ability of neural networks to predict a chaotic behavior for each coding scheme, the training of two data sets, one of them describing chaotic iterations with strongly connected graph, are compared.

Thereafter we give, for the different learning setups and data sets, the mean prediction success rate obtained for each output. Such a rate represents the percentage of input-output pairs belonging to the test subset for which the corresponding output value was correctly predicted. These values are computed considering 10 training with random subsets construction, weights and biases initialization. Firstly, neural networks having 10 and 25 hidden neurons are trained, with a maximum number of epochs that takes its value in $\{125, 250, 500\}$ (see Tables 3.1 and 3.2). Secondly, we refine the second coding scheme by splitting the output vector such that each output is learned by a specific neural network (Table 3.3). In this last case, we increase the size of the hidden layer up to 40 neurons and we consider larger number of epochs.

Table 3.1: Prediction success rates for configurations expressed as Boolean vectors.

Networks topology: 6 inputs, 5 outputs, and one hidden layer				
Hidden neurons		10 neurons		
	Epochs	125	250	500
Chaotic	Output (1)	90.92%	91.75%	91.82%
	Output (2)	69.32%	78.46%	82.15%
	Output (3)	68.47%	78.49%	82.22%
	Output (4)	91.53%	92.37%	93.4%
	Config.	36.10%	51.35%	56.85%
	Strategy (5)	1.91%	3.38%	2.43%
Non-chaotic	Output (1)	97.64%	98.10%	98.20%
	Output (2)	95.15%	95.39%	95.46%
	Output (3)	100%	100%	100%
	Output (4)	97.47%	97.90%	97.99%
	Config.	90.52%	91.59%	91.73%
	Strategy (5)	3.41%	3.40%	3.47%
Hidden neurons		25 neurons		
	Epochs	125	250	500
Chaotic	Output (1)	91.65%	92.69%	93.93%
	Output (2)	72.06%	88.46%	90.5%
	Output (3)	79.19%	89.83%	91.59%
	Output (4)	91.61%	92.34%	93.47%
	Config.	48.82%	67.80%	70.97%
	Strategy (5)	2.62%	3.43%	3.78%
Non-chaotic	Output (1)	97.87%	97.99%	98.03%
	Output (2)	95.46%	95.84%	96.75%
	Output (3)	100%	100%	100%
	Output (4)	97.77%	97.82%	98.06%
	Config.	91.36%	91.99%	93.03%
	Strategy (5)	3.37%	3.44%	3.29%

Table 3.1 formerly published in [BCGS12b] presents the rates obtained for the first coding scheme. For the chaotic data, it can be seen that as expected configuration prediction becomes better when the number of hidden neurons and maximum epochs increases: an improvement by a factor two is observed (from 36.10% for 10 neurons and 125 epochs to 70.97% for 25 neurons and 500 epochs). We also notice that the learning of outputs (2) and (3) is more difficult. Conversely, for the non-chaotic case the simplest training setup is enough to predict configurations. For all these feedforward network topologies and all outputs the obtained results for the non-chaotic case outperform the chaotic ones. Finally, the rates for the strategies show that the different feedforward networks are unable to learn them.

For the second coding scheme (*i.e.*, with Gray Codes) Table 3.2 shows that any network learns about five times more non-chaotic configurations than chaotic ones. As in the previous scheme, the strategies cannot be predicted. Figures 3.3 and 3.4 present the predictions given by two feedforward multilayer perceptrons that were respectively trained to learn chaotic and non-chaotic data using the second coding scheme. Each figure shows for each sample of the test subset (577 samples, representing 25% of the 2304 samples) the configuration that should have been predicted and the one given by the multilayer perceptron. It can be seen that for the chaotic data the predictions are far away from the expected configurations. Obviously, the better predictions obtained for the

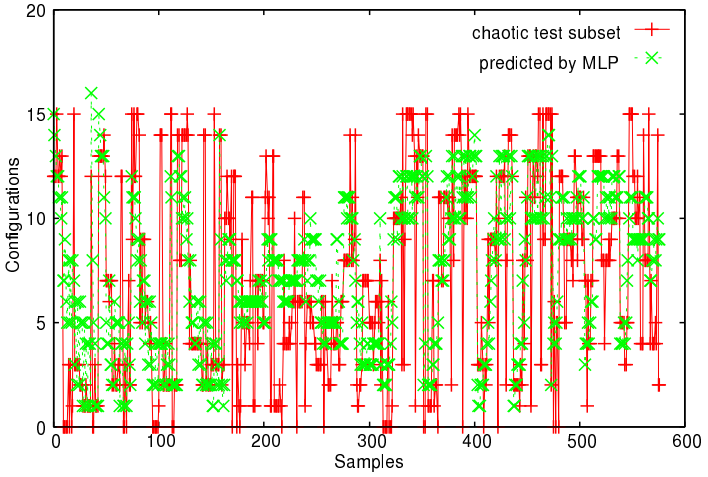


Figure 3.3: Second coding scheme - Predictions obtained for a chaotic test subset.

non-chaotic data reflect their regularity.

Let us now compare the two coding schemes. Firstly, the second scheme disturbs the learning process. In fact in this scheme the configuration is always expressed as a natural number, whereas in the first one the number of inputs follows the increase of the Boolean vectors coding configurations. In this latter case, the coding gives a finer information on configuration evolution.

Table 3.2: Prediction success rates for configurations expressed with Gray code

Networks topology: 3 inputs, 2 outputs, and one hidden layer				
	Hidden neurons	10 neurons		
	Epochs	125	250	500
Chaotic	Config. (1)	13.29%	13.55%	13.08%
	Strategy (2)	0.50%	0.52%	1.32%
Non-Chaotic	Config. (1)	77.12%	74.00%	72.60%
	Strategy (2)	0.42%	0.80%	1.16%
	Hidden neurons	25 neurons		
	Epochs	125	250	500
Chaotic	Config. (1)	12.27%	13.15%	13.05%
	Strategy (2)	0.71%	0.66%	0.88%
Non-Chaotic	Config. (1)	73.60%	74.70%	75.89%
	Strategy (2)	0.64%	0.97%	1.23%

Unfortunately, in practical applications the number of components is usually unknown. Hence, the first coding scheme cannot be used systematically. Therefore, we provide a refinement of the second scheme: each output is learned by a different ANN. Table 3.3 presents the results for this approach. In any case, whatever the considered feedforward network topologies, the maximum epoch number, and the kind of iterations, the configuration success rate is slightly improved. Moreover, the strategies predictions rates reach almost 12%, whereas in Table 3.2 they never exceed 1.5%. Despite of this improvement, a long term prediction of chaotic iterations still appear to be an open issue.

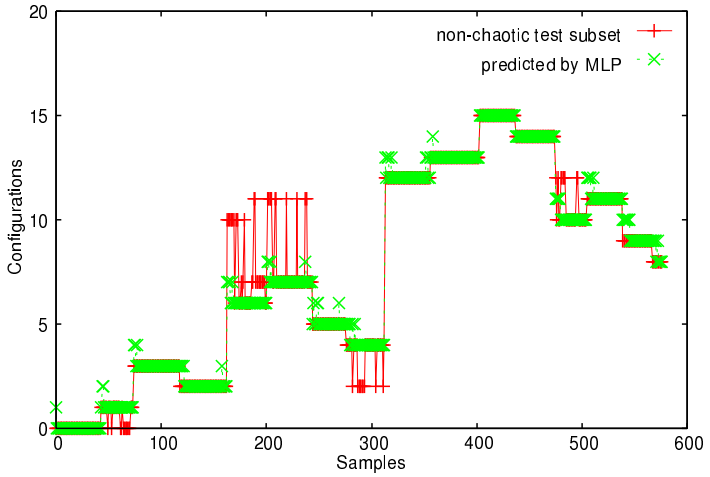


Figure 3.4: Second coding scheme - Predictions obtained for a non-chaotic test subset.

Table 3.3: Prediction success rates for split outputs.

Networks topology: 3 inputs, 1 output, and one hidden layer			
Epochs	125	250	500
Chaotic	Output = Configuration		
10 neurons	12.39%	14.06%	14.32%
25 neurons	13.00%	14.28%	14.58%
40 neurons	11.58%	13.47%	14.23%
Non chaotic	Output = Configuration		
10 neurons	76.01%	74.04%	78.16%
25 neurons	76.60%	72.13%	75.96%
40 neurons	76.34%	75.63%	77.50%
Chaotic/non chaotic	Output = Strategy		
10 neurons	0.76%	0.97%	1.21%
25 neurons	1.09%	0.73%	1.79%
40 neurons	0.90%	1.02%	2.15%

Epochs	1000	2500	5000
Chaotic	Output = Configuration		
10 neurons	14.51%	15.22%	15.22%
25 neurons	16.95%	17.57%	18.46%
40 neurons	17.73%	20.75%	22.62%
Non chaotic	Output = Configuration		
10 neurons	78.98%	80.02%	79.97%
25 neurons	79.19%	81.59%	81.53%
40 neurons	79.64%	81.37%	81.37%
Chaotic/non chaotic	Output = Strategy		
10 neurons	3.47%	9.98%	11.66%
25 neurons	3.92%	8.63%	10.09%
40 neurons	3.29%	7.19%	7.18%

3.6/ CONCLUSION

In this chapter, we have recalled our works published in [BGS11, BCGS12b], in which we have established an equivalence between chaotic iterations and a class of multilayer perceptron neural networks. Firstly, we have described how to build a neural network that can be trained to learn a given chaotic map function. Secondly, we found a condition that allow to check whether the iterations induced by a function are chaotic or not, and thus if a chaotic map is obtained. Thanks to this condition our approach is not limited to a particular function. In the dual case, we show that checking if a neural network is chaotic consists in verifying a property on an associated graph, called the graph of iterations. These results are valid for recurrent neural networks with a particular architecture. However, we believe that a similar work can be done for other neural network architectures. Finally, we have discovered at least one family of problems with a reasonable size, such that artificial neural networks should not be applied in the presence of chaos, due to their inability to learn chaotic behaviors in this context. Such a consideration is not reduced to a theoretical detail: we have concretely implemented this family of discrete iterations in a steganographic method in [BG10b]. As steganographic detectors embed tools like neural networks to distinguish between original and stego contents, our studies tend to prove that such detectors might be unable to tackle with chaos-based information hiding schemes (this field of information security will be further investigated in Chapter 5).

These considerations end our post-thesis questionings regarding the theory of chaotic finite state machines. They have been widely used these last three years to propose concrete solutions in the field of information security. These solutions are summarized in the next part of this manuscript.

II

Applications in the Information Security Field

4

Application to Pseudorandom Numbers Generation

Arguments presented in the previous chapters show that it is both possible to construct a chaotic program that runs in a finite state machine open to the outside world, and that any program of this kind has a behavior that can be studied in the mathematical theory of chaos framework. We have shown that Moore or Turing machines, and even neural networks, can be used to construct concretely such finite state machines that behave chaotically according to Devaney and consort. A first natural use of a chaotic program is to generate randomness for a large variety of applications, from information security to numerical simulations: even if chaos is not randomness, these two behaviors are enough close to hope that a chaotic dynamical system will present a good statistical profile. This is why we have considered in our researches, new when compared to our thesis topics, that chaotic iterations can be used for (pseudo)random number generation, leading to a collection of so-called “CIPRNGs” chaos-based generators reviewed here.

4.1/ Introduction

Two kinds of random generators are necessary in finite state machines, namely truly random number generators (TRNGs) and pseudorandom number generators (PRNGs). In a TRNG, numbers are generated using a non-reproducible source of entropy using a physical noise to generate numbers. This noise can be obtained in different ways, using for instance a mixture of temperature sensors, microphones, memory residuals, and user manipulations. Such TRNG are used for instance to generate cryptographic keys or nonces, in which non reproducibility is an important security requirement. An issue with such generators is that, to prove or to test the randomness unbiased characteristics of the generated numbers is a very hard task, most of the time impossible to achieve.

Even though it is sometimes possible to press ahead theoretical physics results to deduce a random-like behavior of the considered device, such results only hold for ideal devices, and unpredictable biases are often introduced during the concrete realization of such devices. In a large scale of applications, like Monte Carlo based numerical simulations of critical platforms (nuclear plants, etc.), the existence of possible uncontrollable biases in the input random stream is unacceptable. Furthermore, in various concrete applications that require a random stream, reproducibility is not a problem; indeed, it is often required. In numerical simulations again, to be able to reproduce the same randomness to equi-

tably test various platforms is a fair necessity. Additionally, to be dependent on hardware devices (like sensors) may be problematic, and these latter may be more or less defective given two computers, while it is often a necessity to have the same behavior of a given program on two different finite machines. These reasons explain why both TRNGs and PRNGs are useful in computer science, and more generally in any field that requires a random source.

PRNGs being reproducible, they are most of the time constituted by a recurrent sequence in which the first term (and sometimes the parameters of the sequence) play the role of a seed: using the same seed and the same parameters twice leads to the same generated pseudorandom sequence. The random-like characteristics then is verified using statistical batteries of tests, like the NIST [BR10], DieHARD [Mar96], or TestU01 [SM07] reputed ones. The adequacy of a large quantity of generated digits to the uniform distribution on $\{0, 1\}$ is tested for a large variety of properties, like the intended frequency of all finite patterns (poker test).

A good rule of thumb, developed since our thesis, consists in selecting *a priori* good recurrent sequences, that is, discrete dynamical systems that have a large amount of provable properties of disorder, before checking *a posteriori* its random-like behavior using the aforementioned batteries of tests. An interesting and promising category of such discrete dynamical systems is constituted by chaotic sequences, whose disordered and unpredictable behaviors are defined and proven in the mathematical topology framework recalled previously.

After having proven that such chaotic finite machines exist and are concretely buildable in previous chapters, we now investigate the interest of such chaotic finite machines for random numbers generation. Our technique, extensively developed since our thesis, can be considered as a post-treatment on TRNGs or PRNGs: at each iterate, a new value is taken either from the truly random stream or from the pseudorandom one, and the new output is computed using chaotic iterations on this input value and the internal state. Thus our discrete dynamical system operates on the Cartesian product of the memory states $\mathbb{B}^N$ of the finite state machine *and* of an infinite set of sequences. The sole difference from TRNG to PRNG is that this set of sequences is restricted to the ones being eventually periodic, which is an infinite countable set. In the both cases, the finite state machine operates on an infinite Cartesian product: the hazard of iterating on a finite state on which chaos is at best degenerated is avoided, thanks to this approach already presented in Chapter 2. For TRNGs, the main interest is that, when it is used as input of the chaotic Moore machine, the output stream corresponds to a truly chaotic and random generator. Here, chaos is mathematically proven, while random comes from theoretical physics considerations. For PRNGs, we have finally regarded the interest to add chaos properties on a pseudorandom sequences. To do so, we have investigated these last years a large category of defective PRNGs, and have shown that to mix these generators using chaotic iterations leads systematically to an improvement of their results to each of the statistical battery of tests previously mentioned: iterating a chaotic post-treatment on these defective generators improve their statistical behaviors. A strong correlation appeared between chaos and random as it is understood in statistical tests, which is not surprising as, when some chaos properties are not satisfied, then some flaws appear in the recurrent sequence (for instance, not all the states are visited), which must be signaled by a statistical test if the battery is well designed.

The aim of this HDR chapter is to recall and summarize all the researches we have

performed these last years in the chaos-based pseudorandom number generation field.

4.2/ Qualitative Relations Between Topological Properties and Statistical Tests

Let us firstly explain why we have reasonable ground to believe that chaos can improve the statistical properties of inputs. We will show in this section that chaotic properties as defined in the mathematical theory of chaos are related to some statistical tests that can be found in the NIST battery of tests [BR10]. We will verify later in this chapter that, when mixing defective PRNGs with chaotic iterations, the new generator presents better statistical properties (this section summarizes and extends the work of [BFG12a]).

There are various relations between topological properties that describe an unpredictable behavior for a discrete dynamical system on the one hand, and statistical tests to check the randomness of a numerical sequence on the other hand. These two mathematical disciplines follow a similar objective in case of a recurrent sequence (to characterize an intrinsically complicated behavior), with two different but complementary approaches. It is true that the following illustrative links give only qualitative arguments, and proofs should be provided to make such arguments irrefutable. However they give a first understanding of the reason why chaotic properties tend to improve the statistical quality of PRNGs, which is experimentally verified as shown in the end of this chapter. Let us now list some of these relations between topological properties defined in the mathematical theory of chaos and tests embedded into the NIST battery.

- **Regularity**. As recalled in Chapter 2, a chaotic dynamical system must have an element of regularity. Depending on the chosen definition of chaos, this element can be the existence of a dense orbit, the density of periodic points, etc. The key idea is that a dynamical system with no periodicity is not as chaotic as a system having periodic orbits: in the first situation, we can predict something and gain a knowledge about the behavior of the system, that is, it never enters into a loop. A similar importance for periodicity is emphasized in the two following NIST tests [BR10]:
  - **Non-overlapping Template Matching Test**. Detect the production of too many occurrences of a given non-periodic (aperiodic) pattern.
  - **Discrete Fourier Transform (Spectral) Test**. Detect periodic features (i.e., repetitive patterns that are close one to another) in the tested sequence that would indicate a deviation from the assumption of randomness.
- **Transitivity**. This topological property previously introduced states that the dynamical system is intrinsically complicated: it cannot be simplified into two subsystems that do not interact, as we can find in any neighborhood of any point another point whose orbit visits the whole phase space. This focus on the places visited by the orbits of the dynamical system takes various nonequivalent formulations in the mathematical theory of chaos, namely: transitivity, strong transitivity, total transitivity, topological mixing, and so on [BG10a]. A similar attention is brought on the states visited during a random walk in the two tests below [BR10]:
  - **Random Excursions Variant Test**. Detect deviations from the expected number of visits to various states in the random walk.

  - **Random Excursions Test**. Determine if the number of visits to a particular state within a cycle deviates from what one would expect for a random sequence.

- **Chaos according to Li and Yorke**. We recalled that two points of the phase space (x, y) define a couple of Li-Yorke when $\limsup_{n\to+\infty} d(f^{(n)}(x), f^{(n)}(y)) > 0$ and $\liminf_{n\to+\infty} d(f^{(n)}(x), f^{(n)}(y)) = 0$, meaning that their orbits always oscillate as the iterations pass. When a system is compact and contains an uncountable set of such points, it is claimed as chaotic according to Li-Yorke [LY75, Rue01]. A similar property is regarded in the following NIST test [BR10].

  - **Runs Test**. To determine whether the number of runs of ones and zeros of various lengths is as expected for a random sequence. In particular, this test determines whether the oscillation between such zeros and ones is too fast or too slow.

- **Topological entropy**. The desire to formulate an equivalency of the thermodynamics entropy has emerged both in the topological and statistical fields. Once again, a similar objective has led to two different rewriting of an entropy based disorder: the famous Shannon definition is approximated in the statistical approach, whereas topological entropy has been defined previously. This value measures the average exponential growth of the number of distinguishable orbit segments. In this sense, it measures the complexity of the topological dynamical system, whereas the Shannon approach comes to mind when defining the following test [BR10]:

  - **Approximate Entropy Test**. Compare the frequency of the overlapping blocks of two consecutive/adjacent lengths (m and $m + 1$) against the expected result for a random sequence.

- **Non-linearity, complexity**. Finally, let us remark that non-linearity and complexity are not only sought in general to obtain chaos, but they are also required for randomness, as illustrated by the two tests below [BR10].

  - **Binary Matrix Rank Test**. Check for linear dependence among fixed length substrings of the original sequence.
  - **Linear Complexity Test**. Determine whether or not the sequence is complex enough to be considered random.

We have recalled in Chapter 2 that chaotic iterations are, among other things, strongly transitive, topologically mixing, chaotic as defined by Li and Yorke, and that they have a topological entropy and an exponent of Lyapunov both equal to $ln(\mathsf{N})$, where N is the size of the iterated vector, see theorems of Section 2.3. Due to these topological properties, we are ground to believe that a generator based on chaotic iterations could probably be able to pass batteries for pseudorandomness like the NIST one. The following sections, recalling all our work in this field since our thesis, show that defective generators have their statistical properties improved by chaotic iterations.

4.3/ THE CIPRNGS: CHAOTIC ITERATION BASED PRNGS

This section focus on the presentation of various realizations of pseudorandom number generators based on chaotic iterations.

4.3.1/ CIPRNG, VERSION 1

Let $\mathsf{N} \in \mathbb{N}^*, \mathsf{N} \geqslant 2$, and $\mathcal{M}$ be a finite subset of $\mathbb{N}^*$. Consider two possibly defective generators called PRNG1 and PRNG2 we want to improve (like the generators detailed in the Appendix A.1), the first one having his terms into $[\![1, \mathsf{N}]\!]$ whereas the second ones return integers in $\mathcal{M}$, which is always possible. The first version of a generator resulting on a post-treatment on these defective PRNGs using chaotic iterations has been denoted by CIPRNG(PRNG1,PRNG2) version 1. This (inefficient) proof of concept is designed by the following process [BGW09, BGW10b]:

1. Some chaotic iterations are fulfilled, with the vectorial negation and PRNG1 as strategy, to generate a sequence $(x^n)_{n\in\mathbb{N}} \in \left(\mathbb{B}^{\mathsf{N}}\right)^{\mathbb{N}}$ of Boolean vectors: the successive internal states of the iterated system.
2. Some of these vectors are randomly extracted with PRNG2 and their components constitute our pseudorandom bit flow. Algorithm 1 provides the way to produce one output.

Algorithm 1: An arbitrary round of CIPRNG(PRNG1,PRNG2) version 1

Input: The internal state x (an array of N 1-bit words)
Output: An array of N 1-bit words

1: **for** $i = 0, \ldots, PRNG1()$ **do**
2: $\quad S \leftarrow PRNG2()$;
3: $\quad x_S \leftarrow \overline{x_S}$;
4: return x;

In other words, chaotic iterations are realized as follows. Initial state $x^0 \in \mathbb{B}^{\mathsf{N}}$ is a Boolean vector taken as a seed and strategy $(S^n)_{n\in\mathbb{N}} \in [\![1, \mathsf{N}]\!]^{\mathbb{N}}$ is a sequence produced by PRNG2. Lastly, iteration function f is the vectorial Boolean negation. So, at each iteration, only the S^i-th component of state x^n is updated, as follows

$$x_i^n = \begin{cases} x_i^{n-1} & \text{if } i \neq S^i, \\ \overline{x_i^{n-1}} & \text{if } i = S^i. \end{cases} \tag{4.1}$$

Finally, some x^n are selected by a sequence m^n as the pseudorandom bit sequence of our generator, where $(m^n)_{n\in\mathbb{N}} \in \mathcal{M}^{\mathbb{N}}$ is obtained using PRNG2. That is, the generator returns the following values: the components of x^{m^0}, followed by the components of $x^{m^0+m^1}$, followed by the components of $x^{m^0+m^1+m^2}$, *etc.*

Generators investigated in the first set of experiments are the Logistic map, XORshift, and ISAAC (these generators are defined in the Appendix A.1), while the reputed NIST [BR10], DieHARD [Mar96], and TestU01 [SM07] test suites have been considered for statistical evaluation. Table 4.1 contains the statistical results (number of

Table 4.1: Statistical results of well-known PRNGs

	BBS	Logistic	XORshift	ISAAC
NIST SP 800-22 (15 tests)	2	14	14	15
DieHARD (18 tests)	2	16	15	18
TestU01 (516 tests)	212	250	370	516

Table 4.2: Statistical results for the CIPRNG version 1

Test name	CIPRNG Version 1			
	Logistic + Logistic	XORshift + XORshift	ISAAC + XORshift	ISAAC + ISAAC
NIST (15)	15	15	15	15
DieHARD (18)	18	18	18	18
TestU01 (516)	378	507	516	516

tests successfully passed) obtained by the considered inputted generators, while Table 4.2 shows the results with the first version of our CIPRNGs: improvements, published in [BGW09, BGW10b], are obvious.

We have enhanced this CIPRNG several times, and tested these generators deeply during our cosupervision of Qianxue Wang and Xiaole Fang theses. These improvements and studies are detailed in Appendix A.2. We only explain in this main chapter the XOR CIPRNG version, whose study has been realized outside these theses context.

4.3.2/ XOR CIPRNG

Instead of updating only one cell at each iteration as the previous versions of our CIPRNGs, we can try to choose a subset of components and to update them together. Such an attempt leads to a kind of merger of the two random sequences. When the updating function is the vectorial negation, this algorithm can be rewritten as follows [BCGH11]:

$$\begin{cases} x^0 \in [\![0, 2^{\mathsf{N}} - 1]\!], S \in [\![0, 2^{\mathsf{N}} - 1]\!]^{\mathbb{N}} \\ \forall n \in \mathbb{N}^*, x^n = x^{n-1} \oplus S^{n-1}, \end{cases} \tag{4.2}$$

and this rewriting can be understood as follows. The n^{th} term S^n of the sequence S, which is an integer of N binary digits, whose list of digits in binary decomposition is the list of cells to update in the state x^n of the system (represented as an integer having N bits too). More precisely, the k^{th} component of this state (a binary digit) changes if and only if the k^{th} digit in the binary decomposition of S^n is 1. This generator has been called XOR CIPRNG, it has been introduced, theoretically studied, and tested in [BCGH11, BFG12a]. It uses a very classical pseudorandom generation approach, the unique contribution is its relation with chaotic iterations: the single basic component presented in the previous equation is of ordinary use as a good elementary brick in various PRNGs. It corresponds

to the discrete dynamical system in chaotic iterations.

4.4/ PRESERVING SECURITY

This section is dedicated to the security analysis of the proposed PRNGs, both from a theoretical and from a practical point of view.

4.4.1/ THEORETICAL PROOF OF SECURITY

The standard definition of *indistinguishability* used is the classical one as defined for instance in [Gol07, chapter 3]. This property shows that predicting the future results of the PRNG cannot be done in a reasonable time compared to the generation time. It is important to emphasize that this is a relative notion between breaking time and the sizes of the keys/seeds. Of course, if small keys or seeds are chosen, the system can be broken in practice. But it also means that if the keys/seeds are large enough, the system is secured. As a complement, an example of a concrete practical evaluation of security is outlined in the next subsection.

In a cryptographic context, a pseudorandom generator is a deterministic algorithm G transforming strings into strings and such that, for any seed s of length m, $G(s)$ (the output of G on the input s) has size $\ell_G(m)$ with $\ell_G(m) > m$. The notion of *secure* PRNGs can now be defined as follows.

Definition 7. *A cryptographic PRNG G is secure if for any probabilistic polynomial time algorithm D, for any positive polynomial p, and for all sufficiently large m's,*

$$|\Pr[D(G(U_m)) = 1] - Pr[D(U_{\ell_G(m)}) = 1]| < \frac{1}{p(m)},$$

where U_r is the uniform distribution over $\{0,1\}^r$ and the probabilities are taken over U_m, $U_{\ell_G(m)}$ as well as over the internal coin tosses of D.

Intuitively, it means that there is no polynomial time algorithm that can distinguish a perfect uniform random generator from G with a non negligible probability. An equivalent formulation of this well-known security property means that it is possible *in practice* to predict the next bit of the generator, knowing all the previously produced ones. The interested reader is referred to [Gol07, chapter 3] for more information. Note that it is quite easily possible to change the function ℓ into any polynomial function ℓ' satisfying $\ell'(m) > m$) [Gol07, Chapter 3.3].

The generation schema developed in the XOR CIPRNG is based on a pseudorandom generator. Let H be a cryptographic PRNG. Let $S_1, \ldots, S_k$ be the strings of length N such that $H(S_0) = S_1 \ldots S_k$ ($H(S_0)$ is the concatenation of the S_i's). The XOR CIPRNG X defined previously is the algorithm mapping any string of length $2N$ $x_0 S_0$ into the string $(x_0 \oplus S_0 \oplus S_1)(x_0 \oplus S_0 \oplus S_1 \oplus S_2) \ldots (x_o \bigoplus_{i=0}^{i=k} S_i)$. We have proven in [BCGH11] that,

Theorem 6. *If H is a secure cryptographic PRNG, then the XOR CIPRNG X is a secure cryptographic PRNG too.*

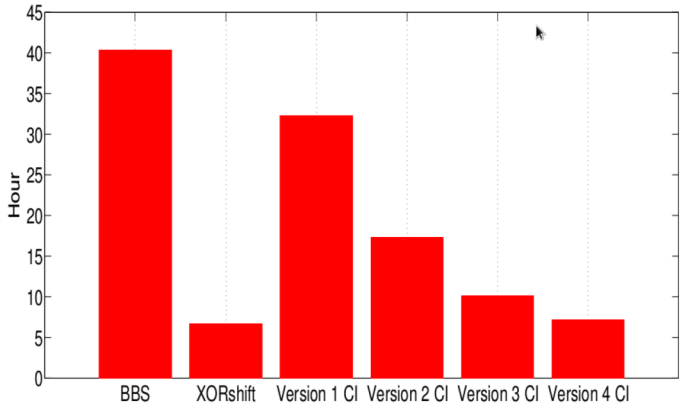


Figure 4.1: Speed comparison between BBS, XORshift, and CIPRNGs version 1-4.

4.4.2/ PRACTICAL SECURITY EVALUATION

Given a key size, it is possible to measure in practice the minimum duration needed for an attacker to break a cryptographically secure PRNG, if we know the power of his/her machines. Such a concrete security evaluation is related to the $(T, \varepsilon)-$security notion, which has been evaluated for various CIPRNGs in [BCGH11] and in submitted papers. A short example of such study for the XOR CIPRNG is provided as an illustrative example in what follows.

Let us firstly recall that,

Definition [8]. *Let $\mathcal{D} : \mathbb{B}^M \longrightarrow \mathbb{B}$ be a probabilistic algorithm that runs in time T. Let $\varepsilon > 0$. $\mathcal{D}$ is called a $(T, \varepsilon)-$distinguishing attack on pseudorandom generator G if*

$$\left| Pr[\mathcal{D}(G(k)) = 1 \mid k \in_R \{0,1\}^\ell] - Pr[\mathcal{D}(s) = 1 \mid s \in_R \mathbb{B}^M] \right| \geqslant \varepsilon,$$

where the probability is taken over the internal coin flips of $\mathcal{D}$, and the notation "$\in_R$" indicates the process of selecting an element at random and uniformly over the corresponding set.

Let us recall that the running time of a probabilistic algorithm is defined to be the maximum of the expected number of steps needed to produce an output, maximized over all inputs; the expected number is averaged over all coin flips made by the algorithm [Knu97]. We are now able to define the notion of cryptographically secure PRNGs:

Definition [9]. *A pseudorandom generator is $(T, \varepsilon)-$secure if there exists no $(T, \varepsilon)-$distinguishing attack on this pseudorandom generator.*

We have proven in [BCGH11] that,

Proposition [5]. *If the inputted PRNG is (T, ε)-secure, then this is the case too for the XOR CIPRNG.*

Suppose for instance that the XOR CIPRNG with the cryptographically secure BBS as input will work during $M = 100$ time units, and that during this period, an attacker can realize 10^{12} clock cycles. We thus wonder whether, during the PRNG's lifetime, the attacker can distinguish this sequence from a truly random one, with a probability greater

than $\varepsilon = 0.2$. We consider that the modulus of BBS N has 900 bits, that is, contrarily to previous sections, we use here the BBS generator with relevant security parameters.

Predicting the next generated bit knowing all the previously released ones by the XOR CIPRNG is obviously equivalent to predicting the next bit in the BBS generator, which is cryptographically secure. More precisely, it is $(T, \varepsilon)-$secure: no $(T, \varepsilon)-$distinguishing attack can be successfully realized on this PRNG, if [FS97]

$$T \leqslant \frac{L(N)}{6N(log_2(N))\varepsilon^{-2}M^2} - 2^7 N\varepsilon^{-2}M^2 log_2(8N\varepsilon^{-1}M) \tag{4.3}$$

where M is the length of the output ($M = 100$ in our example), and $L(N)$ is equal to

$$2.8 \times 10^{-3} exp\left(1.9229 \times (N\ ln\ 2)^{\frac{1}{3}} \times (ln(N\ ln\ 2))^{\frac{2}{3}}\right)$$

is the number of clock cycles to factor a $N-$bit integer.

A direct numerical application shows that this attacker cannot achieve his $(10^{12}, 0.2)$ distinguishing attack in that context.

4.5/ The CIPRNG Family: Further Proposals

An approach to find update functions such that the associated generator presents a random-like and chaotic behavior has been proposed in [BFGW11]. To do so, the vectorial Boolean negation has been used as a prototype. It is then explained how to modify this iteration function without deflating the good properties of the associated generator, leading to eight $CI_fPRNG(PRNG1, PRNG2)$ versions of the CIPRNG version 1. They all can pass the DieHARD battery of tests.

Similarly, in [BCGW11], a method using graph with strongly connected components is proposed as a selection criterion for chaotic iterate function. By using the algorithm proposed in [BCGR11] and recalled in Section 2.3.2.1, ten new functions are proposed for replacing the vectorial negation. These chaotic PRNGs are then subjected to the NIST statistical battery of tests.

Finally, in [BCGH11, CG13] a new pseudorandom number generator based on the XOR CIPRNG is proven to be chaotic according to the Devaney's formulation. We thus proposed an efficient implementation for GPU that successfully passes the BigCrush tests, deemed to be the hardest battery of tests in TestU01. Experiments show that this PRNG can generate about 20 billion of random numbers per second on Tesla C1060 and NVidia GTX280 cards. It is then established that, under reasonable assumptions, the proposed PRNG can be cryptographically secure.

4.6/ Conclusion

In this chapter, the researches we have published these last years on chaotic iterations based pseudorandom number generators have been reviewed and summarized. The effects of mixing defective PRNGs using chaotic iterations has been largely regarded, by recalling our previously obtained improvements of their statistics, for various ways to operate the chaotic iterations based post-treatment we called CIPRNG. These researches

allow to construct both machines and computer programs having unpredictable behaviors. Furthermore, new topological properties can be added to existing tools with the proposed post-treatment, without loss of security.

Motivations to use chaotic programs like chaotic PRNGs are manifold: to place itself in good conditions when designing new algorithms, to create new kind of attacks like chaotic viruses, to numerically simulate chaotic processes, to reinforce the security of schemes already proven as cryptographically secure (for instance, a chaotic version of the Blum-Goldwasser asymmetric key encryption scheme has been proposed in [BCGH11]). Or, when regarding the information hiding field of researches studied in the next chapter, to struggle with artificial intelligence that are used for instance in steganalyzers: we have shown in Chapter 3 that neural networks fail in learning some chaotic iterations behaviors.

Further investigations in chaos-based PRNGs encompass the study of the choice of the topology when comparing the quality of two different programs, or to have an absolute scale to evaluate an algorithm. Situations where the inputted generator is a TRNG must be deepened too, by investigating more largely the analog/numerical mixture that has been initiated in submitted papers and in [Fan13], in which our CIPRNG receives the output of a chaotic opto-electronic laser. Finally, the correlations between some statistical tests and some topological properties must be systematically investigated.

5

Application to Information Hiding

5.1/ Introduction

We have explained at the beginning of this manuscript that any algorithm can be rewritten as an iterative process, leading to the possibility to study its topological behavior. As a concrete example, we have shown during our thesis that the security level of some information hiding algorithms (of the spread-spectrum kind) can be studied into a novel framework based on unpredictability, as it is understood in the theory of chaos. The key idea motivating our research works is that: *if artificial intelligence tools seem to have difficulties to deal with chaos, then steganalyzers may be proven defective against chaotic information hiding schemes*. Our work has thus constituted in showing theoretically that such chaotic schemes can be constructed. We are not looking to struggle with best available information hiding techniques and we do not focus on effective and operational aspects, as our questioning are more locating in a conceptual domain. Among other things, we do not specify how to chose embedding coefficients, but the way to insert the hidden message in a selection of these "least significant coefficient". To say this another way, our intention is not to realize an hidden channel that does not appear as sleazy to a steganalyzer, but to construct an information hiding scheme whose behavior cannot be predicted: supposing that the adversary has anything (algorithm, possible embedding coefficient, etc.) but the secret key, we want to determine if he can predict which coefficients will be finally used, and in which order.

To do so and in the continuation of our thesis, a new class of security has been introduced in [BG10c], namely the topological security. This new class can be used to study some categories of attacks that are difficult to investigate in the existing security approach. It also enriches the variety of qualitative and quantitative tools that evaluate how strong the security is, thus reinforcing the confidence that can be added in a given scheme.

In addition of being stego-secure, we have proven during our thesis and published later in [GFB10] that Natural Watermarking (NW) technique is topologically secure. Moreover, this technique possesses additional properties of unpredictability, namely, strong transitivity, topological mixing, and a constant of sensitivity equal to $\frac{N}{2}$: all these results are currently submitted. However NW are not expansive, which is in our opinion problematic in the Constant-Message Attack (CMA) and Known Message Attack (KMA) setups, when we consider that the attacker has all but the embedding key [Guy10]. Since our thesis, our research works in that information hiding field have thus consisted in searching more secure schemes than NW, regarding the concerns presented in the first paragraph of this introduction.

5.2/ THE $\mathcal{CIW}_1$ CHAOTIC ITERATION BASED WATERMARKING PROCESS

5.2.1/ USING CHAOTIC ITERATIONS AS INFORMATION HIDING SCHEMES

5.2.1.1/ PRESENTATION OF THE DHCI PROCESS

During our thesis, we have proposed a data hiding protocol based on chaotic iterations. The process, referred as dhCI, consisted in iterating CIs on least significant coefficients of a cover medium. The same original image was supposed to be shared by the sender and the receiver, the sender either iterates or not CIs on these coefficients, depending on whether the binary information to transfer was 0 or 1, while the receiver computed the differences between its stored image and the received one. Again, we do not focus on the operational domain, really interesting and important but largely studied by plus competent researchers: we take place on a conceptual level regarding the possibility to write chaotic information hiding algorithms.

The first deepened study of such a dhCI algorithm was published in Secrypt'10 [BG10c]. The aims were to prove that a particular instance of the dhCI algorithm, called the $\mathcal{CIW}_1$ process, is both stego-secure and topologically secure, to study its qualitative and quantitative properties of unpredictability, and then to compare it with Natural Watermarking (the topological study was realized at the end of our thesis while the stego-security has been proven later in [GFB10]). To be able to recall the $\mathcal{CIW}_1$ scheme, we must firstly define the significance of a given coefficient.

This definition, given in our thesis, has been published later (in Secrypt'2011 [FGB11]).

5.2.1.2/ MOST AND LEAST SIGNIFICANT COEFFICIENTS

We first notice that terms of the original content x that may be replaced by terms issued from the watermark y are less important than other: they could be changed without be perceived as such. More generally, a *signification function* attaches a weight to each term defining a digital media, depending on its position t.

Definition 10. *A* signification function *is a real sequence* $(u^k)^{k\in\mathbb{N}}$.

Example 5. *Let us consider a set of grayscale images stored into portable graymap format (P3-PGM): each pixel ranges between 256 gray levels, i.e., is memorized with eight bits. In that context, we consider* $u^k = 8 - (k \mod 8)$ *to be the* k*-th term of a signification function* $(u^k)^{k\in\mathbb{N}}$. *Intuitively, in each group of eight bits (i.e., for each pixel) the first bit has an importance equal to 8, whereas the last bit has an importance equal to 1. This is compliant with the idea that changing the first bit affects more the image than changing the last one.*

Definition 11. *Let* $(u^k)^{k\in\mathbb{N}}$ *be a signification function,* m *and* M *be two reals s.t.* $m < M$.

- *The* most significant coefficients (MSCs) *of* x *is the finite vector*

$$u_M = \left(k \mid k \in \mathbb{N} \ and\ u^k \geqslant M \ and\ k \leqslant \mid x \mid\right);$$

- *The* least significant coefficients (LSCs) *of* x *is the finite vector*

$$u_m = \left(k \mid k \in \mathbb{N} \ and \ u^k \leqslant m \ and \ k \leqslant \mid x \mid\right);$$

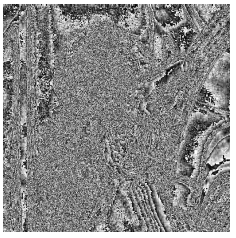
- *The* passive coefficients *of* x *is the finite vector*

$$u_p = \left(k \mid k \in \mathbb{N} \ and \ u^k \in]m; M[\ and \ k \leqslant \mid x \mid\right).$$

For a given host content x, MSCs are then ranks of x that describe the relevant part of the image, whereas LSCs translate its less significant parts. These two definitions are illustrated on Figure 5.1, where the significance function (u^k) is defined as in Example 5, $M = 5$, and $m = 6$.

(a) Lena.

(b) MSCs of Lena.



(c) LSCs of Lena ($\times 17$).

Figure 5.1: Most and least significant coefficients of Lena.

Rem [2]. *The way to define these MSCs and LSCs coefficients in a primordial question in concrete applications, but we currently have not deeply investigated it. The best situation, and perhaps the only one that makes possible a secure hiding process, is when the LSCs are* random and independent from the MSCs*. We do not know if, given a collection of images, such coefficients can be found in practice (perhaps by using statistical batteries of tests ?). Even, we do not know if having random LSCs implies that they are independent from the MSCs. We just signal that such images can be easily constructed, so it all depends on the chosen game: must we necessarily use natural images ? How a steganalyzer can separate natural from stego contents when all images are artificial ?* etc. *(These fundamental questions will be a little investigated at the end of this chapter.)*

5.2.1.3/ PRESENTATION OF THE $\mathcal{CIW}_1$ DHCI SCHEME

We have proposed in Secrypt'10 [BG10c] to study a particular instance of the dhCI class, introduced in our thesis and further investigated at the end of this chapter, which considers the negation function as iteration mode. The resulting chaotic iterations watermarking process has been denoted by $\mathcal{CIW}_1$ in this publication. It operates as follows. Let:

- $(K, N) \in [0; 1] \times \mathbb{N}$ be an embedding key,
- $X \in \mathbb{B}^{\mathsf{N}}$ be the N least significant coefficients (LSCs) of a given cover media C,
- $(S^n)_{n\in\mathbb{N}} \in [\![1, \mathsf{N}]\!]^{\mathbb{N}}$ be a strategy, which depends on the message to hide $M \in [0; 1]$ and K,
- $f_0 : \mathbb{B}^{\mathsf{N}} \to \mathbb{B}^{\mathsf{N}}$ be the vectorial logical negation.

(a) Original Lena. (b) Watermark. (c) Watermarked Lena.

Figure 5.2: Data hiding with chaotic iterations

So the watermarked media is C whose LSCs are replaced by $Y_K = X^N$, where:

$$\begin{cases} X^0 = X \\ \forall n < N, X^{n+1} = G_{f_0}\left(X^n\right). \end{cases}$$

In the following section, two ways to generate $(S^n)_{n\in\mathbb{N}}$ are given, namely Chaotic Iterations with Independent Strategy (CIIS) and Chaotic Iterations with Dependent Strategy (CIDS). In CIIS, the strategy is independent from the cover media X, whereas in CIDS the strategy will be dependent on X. These strategies have been introduced in [GFB10]. Their stego-security are studied in Section 5.2.2 and their topological security in Section 5.2.3.2.

5.2.1.4/ EXAMPLES OF STRATEGIES

CIIS strategy Let us first introduce the Piecewise Linear Chaotic Map (PLCM, see [SQW+01]), defined by:

Definition 12 (PLCM)**.**

$$F(x, p) = \begin{cases} x/p & \textit{if} \quad x \in [0; p] \\ (x - p)/(\frac{1}{2} - p) & \textit{if} \quad x \in \left[p; \frac{1}{2}\right] \\ F(1 - x, p) & \textit{else.} \end{cases}$$

where $p \in \left]0; \frac{1}{2}\right[$ is a "control parameter". Contrary to well-known chaotic maps like the logistic map, this PLCM is unbiased and does not present obvious security flaws [SQW+01].

We define the general term of the strategy $(S^n)_n$ in CIIS setup by the following expression: $S^n = \lfloor \mathsf{N} \times K^n \rfloor + 1$, where:

$$\begin{cases} p \in \left[0; \frac{1}{2}\right] \\ K^0 = M \otimes K \\ K^{n+1} = F(K^n, p), \forall n \leqslant N_0 \end{cases}$$

in which $\otimes$ denotes the bitwise exclusive or (XOR) between two floating part numbers (*i.e.*, between their binary digits representation). Lastly, to be certain to enter into the chaotic regime of PLCM [SQW+01], the strategy can be preferably defined by: $S^n = \left\lfloor \mathsf{N} \times K^{n+D} \right\rfloor + 1$, where $D \in \mathbb{N}$ is large enough.

CIDS strategy The same notations as above are used. We define CIDS strategy as in [GFB10]: $\forall k \leqslant N$,

- if $k \leqslant \mathsf{N}$ and $X^k = 1$, then $S^k = k$,
- else $S^k = 1$.

In this situation, if $N \geqslant \mathsf{N}$, then only two watermarked contents are possible with the scheme proposed in Section 5.2.1, namely: $Y_K = (0, 0, \cdots, 0)$ and $Y_K = (1, 0, \cdots, 0)$.

Before being able to present the security study we performed after our thesis, we must firstly recall the notion of security we have regarded and its difference with robustness.

5.2.2/ SECURITY VERSUS ROBUSTNESS

5.2.2.1/ PRESENTATION

In [PFCTPPG06], it is claimed that: "security and robustness are neighboring concepts without clearly established mathematical definitions". However, robustness is often considered to be mostly concerned with blind elementary attacks, whereas security is not limited to certain specific attacks. Indeed, it is said in [Kal01, CPFPG05] that security encompasses robustness and intentional attacks. Following Kalker [Kal01], we will consider in this manuscript the two following definitions[1]:

[1] Remark that when robustness is required in information hiding, the community tends to speak about digital watermarking, while researchers prefer to say steganography when security is regarded. Similarly, some authors speak about watermarking when the hidden information is only one bit, while steganography is for larger messages, even if such use of the terminologies is less common. Such ambiguities come from the fact that, to the best of our knowledge, no clear common mathematical definitions of steganography and digital watermarking have been accepted by the whole community, and that this community is indeed split in various closed subcommunities that do not communicate together, who have never seen or heard of one another (IH/IWDW vs IIHMSP vs Crypto/Secrypt vs IEEE trans vs...). So, as we must realize a choice in this manuscript, security will always refer to mathematical proofs (detection, separation, or extraction) while robustness will be related to brute-force attacks (destruction of hidden message). Steganography, watermarking, as well as steganalysis will however be used with various significations that can be deduced from the context.

Definition 13 (Security [Kal01])**.** *Security refers to the inability by unauthorized users to have access to the raw watermarking channel [...] to remove, detect and estimate, write or modify the raw watermarking bits.*

Remark that this historical security notion is not restricted to the question of determining whether a channel of "natural" images contains or not stego-contents.

Definition 14 (Robustness [Kal01])**.** *Robust watermarking is a mechanism to create a communication channel that is multiplexed into original content [...] It is required that, firstly, the perceptual degradation of the marked content [...] is minimal and, secondly, that the capacity of the watermark channel degrades as a smooth function of the degradation of the marked content.*

5.2.2.2/ CLASSIFICATION OF ATTACKS

We firstly decided to take place in the framework developed in [CB08a], which defines the following attack contexts. A few other formal security framework exist, like the one developed by Barbier and Filiol [BM08b], but they look further away than what we intended to study.

Definition 15. *The following classes of attacks can be defined:*

- ***Watermark-Only Attack (WOA):*** *occurs when an attacker has only access to several watermarked contents.*
- ***Known-Message Attack (KMA):*** *occurs when an attacker has access to several pairs of watermarked contents and corresponding hidden messages.*
- ***Known-Original Attack (KOA):*** *is when an attacker has access to several pairs of watermarked contents and their corresponding original versions.*
- ***Constant-Message Attack (CMA):*** *occurs when the attacker observes several watermarked contents and only knows that the unknown hidden message is the same in all contents.*

5.2.2.3/ DEFINITION OF STEGO-SECURITY

In the Simmons' prisoner problem [Sim84], Alice and Bob are in jail and they want to, possibly, devise an escape plan by exchanging hidden messages in innocent-looking cover contents. These messages are to be conveyed to one another by a common warden named Eve, who eavesdrops all contents and can choose to interrupt the communication if they appear to be stego-contents. Stego-security, defined in this context, is the highest security class in Watermark-Only Attack setup, which occurs when Eve has only access to several marked contents [CB08a].

Let $\mathbb{K}$ be the set of embedding keys, $p(X)$ the probabilistic model of N_0 initial host contents, and $p(Y|K)$ the probabilistic model of N_0 marked contents s.t. each host content has been marked with the same key K and the same embedding function.

Definition 16 (Stego-Security [CB08a])**.** *The embedding function is* stego-secure *if* $\forall K \in \mathbb{K}, p(Y|K) = p(X)$ *is established.*

Stego-security states that the knowledge of K does not help to make the difference between $p(X)$ and $p(Y)$. This definition implies the following property:

$$p(Y|K_1) = \cdots = p(Y|K_{N_k}) = p(Y) = p(X)$$

This property is equivalent to a zero Kullback-Leibler divergence, which is often referred as the best definition of the "perfect secrecy" in steganography [Cac98].

5.2.3/ SECURITY EVALUATION

5.2.3.1/ EVALUATION OF THE STEGO-SECURITY

We have proven in [GFB10] the following proposition.

Proposition 6. *CIIS is stego-secure, while CIDS does not satisfy this security property.*

5.2.3.2/ EVALUATION OF THE TOPOLOGICAL SECURITY

To check whether an information hiding scheme S is topologically secure or not, we have proposed in our thesis, and published later in [GFB10], to write S as an iterate process $x^{n+1} = f(x^n)$ on a metric space $(\mathcal{X}, d)$. As recalled in the first chapter of this manuscript, this formulation is always possible. So,

Definition 17. *An information hiding scheme S is said to be topologically secure on $(\mathcal{X}, d)$ if its iterative process has a chaotic behavior according to Devaney.*

Once again, this topological notion of security has been introduced in Kerchhoffs' based situations (all is known but the secrete key), and our goal is that the attacker cannot determine which least significant coefficients will or have been altered, and in which order. He or she must not predict the behavior of the algorithm without knowing the secret key.

Due to the chaos properties of the so-called chaotic iterations, we have then deduced in [GFB10] that,

Proposition 7. *CIIS and CIDS are topologically secure.*

We have then deduced qualitative and quantitative properties of topological security for this information hiding scheme in [GFB10]: it is expansive (with a constant of expansiveness equal to 1), topologically mixing, etc. These properties can measure the disorder generated by our scheme, giving by doing so an important information about the unpredictability level of such a process, which helps to compare it to other data hiding methods. Such a comparison is outlined in the next section [GFB10].

5.2.4/ COMPARISON BETWEEN SPREAD-SPECTRUM AND CHAOTIC ITERATIONS

The consequences of topological mixing for data hiding are multiple. Firstly, security can be largely improved by considering the number of iterations as a secret key. An attacker will reach all of the possible media when iterating without this key. Additionally, he cannot benefit from a KOA setup, by studying media in the neighborhood of the original cover.

Moreover, as in a topological mixing situation, it is possible that any hidden message (the initial condition), is sent to the same fixed watermarked content (with different numbers of iterations), the interest to be in a KMA setup is drastically reduced. Lastly, as all of the watermarked contents are possible for a given hidden message, depending on the number of iterations, CMA attacks will fail.

The property of expansiveness reinforces drastically the sensitivity in the aims of reducing the benefits that Eve can obtain from an attack in KMA or KOA setup. For example, it is impossible to have an estimation of the watermark by moving the message (or the cover) as a cursor in situation of expansiveness: this cursor will be too much sensitive and the changes will be too important to be useful. On the contrary, a very large constant of expansiveness ε is unsuitable: the cover media will be strongly altered whereas the watermark would be undetectable. Finally, spread-spectrum is relevant when a discrete and secure data hiding technique is required in WOA setup. However, this technique should not be used in KOA and KMA setup, due to its lack of expansiveness.

5.2.5/ LYAPUNOV EXPONENT EVALUATION

The Lyapunov exponent of the $\mathcal{CIW}_1$ algorithm has been computed in [BFG12b], to improve our knowledge of its topological security. It is equal to $\ln \mathrm{N}$, where N stands for the number of LSCs chosen in the implementation of the algorithm.

To evaluate this Lyapunov exponent, chaotic iterations must be described by a differentiable function on $\mathbb{R}$. To do so, a topological semiconjugacy between the phase space $\mathcal{X}$ and $\mathbb{R}$ has been written. As this proof is simply a rewriting in the digital watermarking field of an unpublished result on chaotic iterations obtained during our thesis, and as Section B.1.7 provides a Lyapunov exponent evaluation for a completely different algorithm, we will not say any more about this publication.

5.2.6/ THE $\mathcal{CIS}_2$ AND $\mathcal{DI}_3$ IMPROVEMENTS

This first proposal extending our thesis research works has been further investigated at the occasion of my cosupervision of Nicolas Friot's thesis. The new methods called $\mathcal{CIS}_2$ and $\mathcal{DI}_3$ are detailed in Appendix B.

5.3/ FURTHER INVESTIGATIONS OF THE DHCI CLASS

We have recalled at the beginning of this chapter that chaotic iterations can be applied on the least significant coefficients of a medium, either in spatial or in frequency domain, in order to watermark it. The general process has been denoted by dhCI in our thesis, while its particular instantiation with the negation function has been later called $\mathcal{CIW}_1$ (remark that $\mathcal{CIS}_2$ and $\mathcal{DI}_3$ processes do not belong, stricto sensu, to the dhCI class). Since our thesis defense, the dhCI class has been investigated more largely, by discovering new iteration functions and evaluating both its security and robustness. Results of such questioning are summarized thereafter.

5.3.1/ INTRODUCTION

The study of the dhCI class has been deepened in [BCG11c] for its theoretical aspects and in [BCG12b] for practical ones.

As for the $\mathcal{CIW}_1$ scheme, the work around the dhCI class focuses on non-blind binary information hiding scheme: the original host is required to extract the binary hidden information. This context is indeed not as restrictive as it could primarily appear. Firstly, it allows to prove the authenticity of a document sent through the Internet (the original document is stored whereas the stego content is sent). Secondly, Alice and Bob can establish an hidden channel into a streaming video (Alice and Bob both have the same movie, and Alice hide information into the frame number k iff the binary digit number k of its hidden message is 1). Thirdly, based on a similar idea, a same given image can be marked several times by using various secret parameters owned both by Alice and Bob. Thus more than one bit can be embedded into a given image by using dhCI dissimulation. Lastly, non-blind watermarking is useful in network's anonymity and intrusion detection [HKB09], and to protect digital data sending through the Internet [ETOSD05].

Before [BCG11c], stego-security [CB08a] and topological security were only proven on the spread spectrum watermarking [CMK+97, HF10], and on the $\mathcal{CIW}_1$ algorithm, which is notably an instance of the dhCI method, but which restricts itself to the negation mode (security proofs of $\mathcal{CIS}_2$ and $\mathcal{DI}_3$ have occurred later). We argued in [BCG11c] that dhCI with other functions can provide algorithms as secure as the $\mathcal{CIW}_1$ one. This work has then generalized the algorithm recalled in Section 5.2 and formalized all its stages, independently from the iteration mode. Due to this formalization, it has then been possible to address the proofs of the two security properties for a larger class of iteration modes in [BCG11c].

Then, in The Computer Journal [BCG12b], a review of the researches on the dhCI class has been presented. Additionally, this article has investigated robustness aspects of the process: applications in frequency domains (namely DWT and DCT embedding) have been formalized and corresponding experiments have been given [BCG12b]. Such a study shows the applicability of the whole approach.

5.3.2/ FORMALIZATION OF STEGANOGRAPHIC METHODS

The data hiding scheme presented in previous works does not constrain media to have a constant size. It is indeed sufficient to provide a function and a strategy that may be parametrized with the size of the elements to modify. Parametrized strategies have already been introduced in a previous section, leading to the notion of *strategy-adapter*. The *mode* notion defined below achieves the same goal but for the iteration function [BCG11c].

Definition 18 (Mode)**.** *A map f, which associates to any $n \in \mathbb{N}$ an application $f_n : \mathbb{B}^n \to \mathbb{B}^n$, is called a* mode*.*

For instance, the *negation mode* is defined by the map that assigns to every integer $n \in \mathbb{N}^*$ the function $\neg_n : \mathbb{B}^n \to \mathbb{B}^n, \neg_n(x_1, \ldots, x_n) \mapsto (\overline{x_1}, \ldots, \overline{x_n})$.

We now use the previously introduced *signification function* to attach a weight to each term defining a digital media, w.r.t. its position t, leading to the following notion of a decomposition function [BCG11c].

Definition [19] (Decomposition function)**.** *Let* $(u^k)^{k\in\mathbb{N}}$ *be a signification function,* $\mathfrak{B}$ *the set of finite binary sequences,* $\mathfrak{N}$ *the set of finite integer sequences,* m *and* M *be two reals s.t.* $m < M$*. Any host* x *may be decomposed into*

$$(u_M, u_m, u_p, \phi_M, \phi_m, \phi_p) \in \mathfrak{N} \times \mathfrak{N} \times \mathfrak{N} \times \mathfrak{B} \times \mathfrak{B} \times \mathfrak{B}$$

where

- u_M*,* u_m*, and* u_p *are coefficients defined in Definition 11;*
- $\phi_M = \left(x^{u_M^1}, x^{u_M^2}, \ldots, x^{u_M^{|u_M|}}\right)$*;*
- $\phi_m = \left(x^{u_m^1}, x^{u_m^2}, \ldots, x^{u_m^{|u_m|}}\right)$*;*
- $\phi_p = \left(x^{u_p^1}, x^{u_p^2}, \ldots, x^{u_p^{|u_p|}}\right)$*.*

The function that associates the decomposed host to any digital host is the decomposition function*. It is further referred as dec*(u, m, M) *since it is parametrized by* u*,* m *and* M*. Notice that* u *is a shortcut for* $(u^k)^{k\in\mathbb{N}}$*.*

Definition [20] (Recomposition)**.** *Let* $(u_M, u_m, u_p, \phi_M, \phi_m, \phi_p) \in \mathfrak{N} \times \mathfrak{N} \times \mathfrak{N} \times \mathfrak{B} \times \mathfrak{B} \times \mathfrak{B}$ *s.t.*

- *the sets of elements in* u_M*, elements in* u_m*, and elements in* u_p *are a partition of* $[\![1, n]\!]$*;*
- $|u_M| = |\varphi_M|$*,* $|u_m| = |\varphi_m|$*, and* $|u_p| = |\varphi_p|$*.*

One may associate the vector

$$x = \sum_{i=1}^{|u_M|} \varphi_M^i.e_{u_M^i} + \sum_{i=1}^{|u_m|} \varphi_m^i.e_{u_m^i} + \sum_{i=1}^{|u_p|} \varphi_p^i.e_{u_p^i}$$

where $(e_i)_{i\in\mathbb{N}}$ *is the usual basis of the* $\mathbb{R}-$*vectorial space* $\left(\mathbb{R}^{\mathbb{N}}, +, .\right)$*. The function that associates* x *to any* $(u_M, u_m, u_p, \phi_M, \phi_m, \phi_p)$ *following the above constraints is called the* recomposition function*.*

The embedding consists in the replacement of the values of ϕ_m of x's LSCs by y. It then composes the two decomposition and recomposition functions seen previously. More formally [BCG11c]:

Definition [21] (Embedding media)**.** *Let dec*(u, m, M) *be a decomposition function,* x *be a host content,* $(u_M, u_m, u_p, \phi_M, \phi_m, \phi_p)$ *be its image by dec*(u, m, M)*, and* y *be a digital media of size* $|u_m|$*. The digital media* z *resulting on the embedding of* y *into* x *is the image of* $(u_M, u_m, u_p, \phi_M, y, \phi_p)$ *by the recomposition function rec.*

We have thus been able in [BCG11c] to reformulate the *dhCI* information hiding scheme, as follows:

Definition 22 (Data hiding dhCI)**.** *Let dec*(u, m, M) *be a decomposition function,* f *be a mode,* $\mathcal{S}$ *be a strategy adapter,* x *be an host content,* $(u_M, u_m, u_p, \phi_M, \phi_m, \phi_p)$ *be its image by dec*(u, m, M)*,* q *be a positive natural number, and* y *be a digital media of size* $l = |u_m|$*.*

The dhCI dissimulation *maps any* (x, y) *to the digital media* z *resulting on the embedding of* $\hat{y}$ *into* x*, s.t.*

- *We instantiate the mode* f *with parameter* $l = |u_m|$*, leading to the function* $f_l : \mathbb{B}^l \to \mathbb{B}^l$*.*
- *We instantiate the strategy adapter* $\mathcal{S}$ *with parameter* y *(and some other ones eventually). This instantiation leads to the strategy* $S_y \in [\![1; l]\!]^{\mathbb{N}}$*.*
- *We iterate* G_{f_l} *with initial configuration* (S_y, ϕ_m)*.*
- $\hat{y}$ *is the* q*-th term.*

To summarize, iterations are realized on the LSCs of the host content (the mode gives the iterate function, the strategy-adapter gives its strategy), and the last computed configuration is re-injected into the host content, in place of the former LSCs.

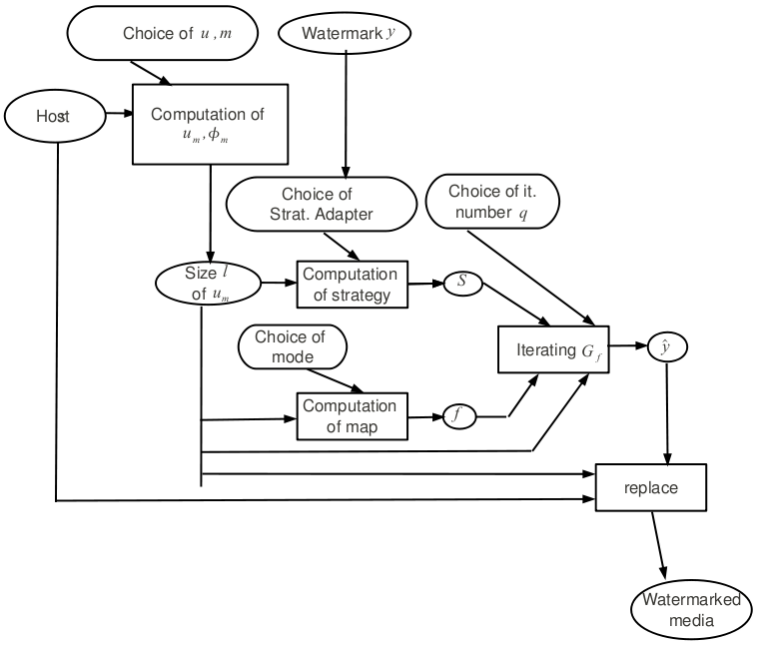


Figure 5.3: The dhCI dissimulation scheme

We are then left to show how to formally check whether a given digital media z results from the dissimulation of y into the digital media x [BCG11c].

Definition 23 (Marked content)**.** *Let dec*(u, m, M) *be a decomposition function,* f *be a mode,* $\mathcal{S}$ *be a strategy adapter,* q *be a positive natural number, and* y *be a digital media,* $(u_M, u_m, u_p, \phi_M, \phi_m, \phi_p)$ *be the image by dec*(u, m, M) *of a digital media* x*.*

Then z *is* marked *with* y *if the image by dec*(u, m, M) *of* z *is* $(u_M, u_m, u_p, \phi_M, \hat{y}, \phi_p)$ *where* $\hat{y}$ *is the right member of* $G^q_{f_l}(S_y, \phi_m)$*.*

Various decision strategies are obviously possible to determine whether a given image z is marked or not, depending on the eventuality that the considered image may have been attacked. For example, a similarity percentage between x and z can be computed, and the result can be compared to a given threshold. Other possibilities are the use of ROC curves or the definition of a null hypothesis problem. These strategies have already been discussed in a previous section, they can be adapted, mutatis mutandis, to the generalized dhCI algorithm detailed above.

The next section, always extracted from [BCG11c], recalls some security properties and shows how the *dhCI dissimulation* algorithm verifies them.

5.3.3/ SECURITY ANALYSIS

We have proven in [BCG11c], using the stochastic matrix theorem, that,

Theorem 7. *Let ϵ be positive, l be any size of LSCs, $X \sim \mathcal{U}\left(\mathbb{B}^l\right)$, f_l be an image mode s.t. $\Gamma(f_l)$ is strongly connected and the Markov matrix associated to f_l is doubly stochastic.*

In the instantiated dhCI dissimulation *algorithm with any uniformly distributed (u.d.) strategy-adapter which is independent from X, there exists some positive natural number q s.t. $|p(X^q) - p(X)| < \epsilon$.*

Proof 1. *See [BCG11c].*

Since $p(Y|K)$ is $p(X^q)$ the method is then stego-secure. We have then focused on topological security properties, and have deduced from the characterization recalled in Theorem 2 that,

Proposition 8. *Functions $f : \mathbb{B}^n \to \mathbb{B}^n$ such that the graph $\Gamma(f)$ is strongly connected lead to topologically secure* dhCI dissimulation *algorithms.*

Theorem 7 relies on a u.d. strategy-adapter that is independent from the cover, and on an image mode f_l whose iteration graph $\Gamma(f_l)$ is strongly connected and whose Markov matrix is doubly stochastic. We have shown in [BCG11c] that the CIIS strategy adapter [GFB10] has the required properties and have mentioned that [QBCG11] has presented an iterative approach (which has been recalled in Section 2.3.2) to generate image modes f_l such that $\Gamma(f_l)$ is strongly connected. Among these maps, it is obvious to check which verifies or not the doubly stochastic constrain.

5.3.4/ DISCOVERING ANOTHER RELEVANT MODE

We can conclude from the previously summarized article that we are left to provide:

- an u.d. strategy-adapter that is independent from the cover,
- an image mode f_l whose iteration graph $\Gamma(f_l)$ is strongly connected and whose Markov matrix is doubly stochastic.

We have recalled in the previous section that the *CIIS*(K, y, α, l) strategy adapter has the required properties. In all the experiments provided in [BCG12b], parameters K and α

are randomly chosen in $]0,1[$ and $]0,0.5[$ respectively, while the number of iteration is set to $4 \times lm$, where lm is the number of LSCs that depends on the domain.

[BCG12b] has then used the iterative approach of Section 2.3.2 to generate image modes f_l such that $\Gamma(f_l)$ is strongly connected, which has been proposed in [BCGR11] and recalled in the first part of this manuscript. Among these maps, it is obvious to check which verifies or not the doubly stochastic constrain. We have already stated that the negation mode matches these hypotheses, so it is relevant in that context. As a second example, we have considered in [BCG12b] the mode $f_l : \mathbb{B}^l \to \mathbb{B}^l$ s.t. its i-th component is defined by

$$f_l(x)_i = \begin{cases} \overline{x_i} \text{ if } i \text{ is odd} \\ x_i \oplus x_{i-1} \text{ if } i \text{ is even.} \end{cases} \tag{5.1}$$

Thanks to Theorem 7, we have deduced in [BCG12b] that its iteration graph $\Gamma(f_l)$ is strongly connected, and have finally proven that its Markov chain is doubly stochastic by induction on the length l.

5.3.5/ DHCI IN FREQUENCY DOMAINS

Even though, as stated previously, our aim is not to focus on operational realizations of our proposal, this aspect must be regarded a few, at least to show the possibility to construct in practice a chaotic information hiding scheme which is reasonable regarding commonly admitted requirements. We recall in this section the experimental protocol applied in [BCG12b].

5.3.5.1/ DWT EMBEDDING

We have firstly explained in [BCG12b] how the dhCI dissimulation can be applied in the discrete wavelets transform domain (DWT). The Daubechies family of wavelets has been chosen: each DWT decomposition depends on a decomposition level and a coefficient matrix (Figure 5.4): *LL* means approximation coefficient, when *HH*, *LH*, *HL* denote respectively diagonal, vertical, and horizontal detail coefficients. For example, the DWT coefficient *HH*2 is the matrix equal to the diagonal detail coefficient of the second level of decomposition of the image.

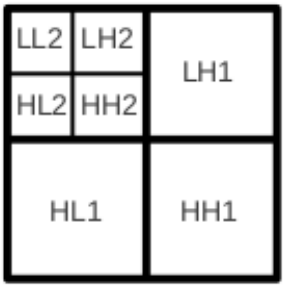


Figure 5.4: Wavelets coefficients.

The choice of the detail level is motivated by finding a good compromise between robustness and invisibility. Choosing low or high frequencies in DWT domain leads either to a

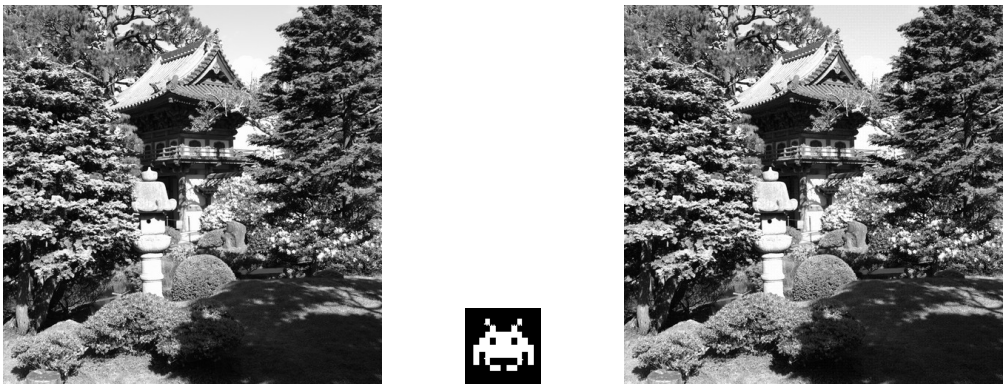
very fragile watermarking without robustness (especially when facing a JPEG 2000 compression attack) or to a large degradation of the host content. In order to have a robust but discrete DWT embedding, the second detail level (*i.e.*, $LH2, HL2, HH2$) that corresponds to the middle frequencies, has been retained in [BCG12b].

Let us consider the Daubechies wavelet coefficients of a third level decomposition as represented in Figure 5.4. We then have translated these float coefficients into their 32-bits values, and have defines in [BCG12b] the significance function u that associates to any index k in this sequence of bits the following numbers:

- $u^k = -1$ if k is one of the three last bits of any index of coefficients in $LH2$, $HL2$, or in $HH2$;
- $u^k = 0$ if k is an index of a coefficient in $LH1$, $HL1$, or in $HH1$;
- $u^k = 1$ otherwise.

According to the definition of significance of coefficients (Def. 11), if (m, M) is $(-0.5, 0.5)$, LSCs are the last three bits of coefficients in $HL2$, $HH2$, and $LH2$. Thus, decomposition and recomposition functions are fully defined and dhCI dissimulation scheme can now be applied.

Figure 5.5 shows the result of a dhCI dissimulation embedding into DWT domain. The original is the image 5007 of the BOSS contest [PFB10b]. Watermark y is given in Fig. 5.5(b). From a random selection of 50 images into the database from the BOSS contest [PFB10b], we have applied in [BCG12b] the dhCI algorithm with mode f_l defined in the previous section and with the negation mode.



(a) Original Image. (b) Watermark y. (c) Watermarked Image.

Figure 5.5: Data hiding in DWT domain

5.3.5.2/ DCT EMBEDDING

We have then explored the discrete cosinus transform (DCT) frequency domain embedding in [BCG12b], by following the protocol detailed below.

Let us denote by x the original image of size $H \times L$, and by y the hidden message, supposed here to be a binary image of size $H' \times L'$. The image x is transformed from the spatial domain to DCT domain frequency bands, in order to embed y inside it. To do so, the host image is firstly divided into 8×8 image blocks as given below:

$$x = \bigcup_{k=1}^{H/8} \bigcup_{k'=1}^{L/8} x(k, k').$$

Thus, for each image block, a DCT is performed and the coefficients in the frequency bands are obtained as follows: $x_{DCT}(m; n) = DCT(x(m; n))$.

To define a discrete but robust scheme, only the three following coefficients of each 8×8 block in position (m, n) has been possibly modified in [BCG12b]: $x_{DCT}(m; n)_{(3,1)}$, $x_{DCT}(m; n)_{(2,2)}$, or $x_{DCT}(m; n)_{(1,3)}$. This choice can be reformulated as follows. Coefficients of each DCT matrix are re-indexed by using a southwest/northeast diagonal, such that $i_{DCT}(m, n)_1 = x_{DCT}(m; n)_{(1,1)}$, $i_{DCT}(m, n)_2 = x_{DCT}(m; n)_{(2,1)}$, $i_{DCT}(m, n)_3 = x_{DCT}(m; n)_{(1,2)}$, $i_{DCT}(m, n)_4 = x_{DCT}(m; n)_{(3,1)}$, ..., and $i_{DCT}(m, n)_{64} = x_{DCT}(m; n)_{(8,8)}$. So the signification function can be defined in this context by:

- if k mod $64 \in \{1, 2, 3\}$ and $k \leqslant H \times L$, then $u^k = 1$;
- else if k mod $64 \in \{4, 5, 6\}$ and $k \leqslant H \times L$, then $u^k = -1$;
- else $u^k = 0$.

The significance of coefficients are obtained for instance with $(m, M) = (-0.5, 0.5)$ leading to the definitions of MSCs, LSCs, and passive coefficients. Thus, decomposition and recomposition functions are fully defined and dhCI dissimulation scheme has then been applied in [BCG12b].

5.3.6/ IMAGE QUALITY

This section focuses on measuring visual quality of our steganographic method. Traditionally, this is achieved by quantifying the similarity between the modified image and its reference image. The Mean Squared Error (MSE) and the Peak Signal to Noise Ratio (PSNR) are the most widely known tools that provide such a metric. However, both of them do not take into account Human Visual System (HVS) properties. Recent works [EAP+06,SB06,PSE+07,MB10] have tackled this problem by creating new metrics. Among them, what follows focuses on PSNR-HVS-M [PSE+07] and BIQI [MB10], considered as advanced visual quality metrics. The former efficiently combines PSNR and visual between-coefficient contrast masking of DCT basis functions based on HVS. This metric has been computed in [BCG12b] by using the implementation given at [psn11]. The latter allows to get a blind image quality assessment measure, *i.e.*, without any knowledge of the source distortion. Its implementation is available at [biq11].

Results of the image quality metrics obtained in [BCG12b] are summarized in Table 5.1. In wavelet domain, the PSNR values obtained in [BCG12b] are comparable to other approaches (for instance, PSNR are 44.2 in [TCL05] and 46.5 in [VDB10]), but a real improvement for the discrete cosine embedding is obtained (PSNR is 45.17 for [CFS08], it is always lower than 48 for [MB08], and always lower than 39 for [MK08]). Among steganography approaches that evaluate PSNR-HVS-M, results of our approach are convincing.

Embedding	DWT		DCT	
Mode	f_l	neg.	f_l	neg.
PSNR	42.74	42.76	52.68	52.41
PSNR-HVS-M	44.28	43.97	45.30	44.93
BIQI	35.35	32.78	41.59	47.47

Table 5.1: Quality measures of our steganography approach [BCG12b]

Firstly, optimized method developed along [Ran11] has a PSNR-HVS-M equal to 44.5 whereas our approach, with a similar PSNR-HVS-M, should be easily improved by considering optimized mode. Next, another approach [MCBE10] have higher PSNR-HVS-M, certainly, but this work does not address robustness evaluation whereas the study presented in [BCG12b] is complete. Finally, as far as we know, [BCG12b] is the first one that has evaluated the BIQI metric in a steganographic context.

With all this material, we have then evaluated the robustness of our approach in [BCG12b].

5.3.7/ Robustness

Previous sections have formalized frequency domains embedding and has focused on the negation and f_l modes. In the robustness given in this continuation, dwt(neg), dwt(fl), dct(neg), and dct(fl) respectively stand for the DWT and DCT embedding with the negation mode and with this instantiated mode.

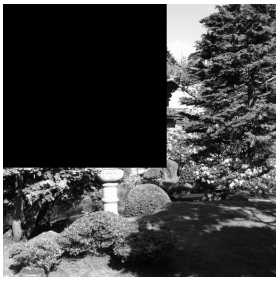
For each experiment presented in [BCG12b], a set of 50 images is randomly extracted from the database taken from the BOSS contest [PFB10b]. Each cover is a 512×512 grayscale digital image and the watermark y is given in Figure 5.5(b). Testing the robustness of the approach is achieved in [BCG12b] by successively applying on watermarked images attacks like cropping, compression, and geometric transformations. Differences between $\hat{y}$ and $\varphi_m(z)$ have then been computed. Behind a given threshold rate, the image is said to be watermarked. Finally, discussion on metric quality of the approach given in [BCG12b] is recalled in Section B.1.6.

5.3.7.1/ Robustness against cropping

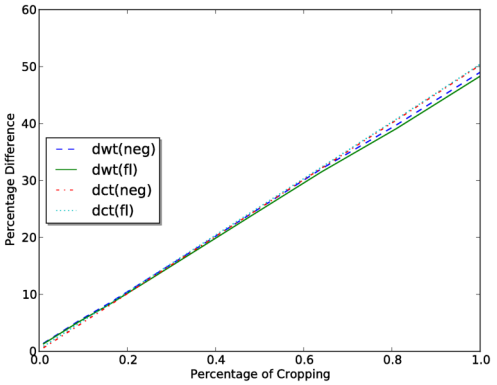
Robustness of the approach is evaluated by applying different percentage of cropping: from 1% to 81%. Results obtained in [BCG12b] are recalled in Figure B.1. Figure 5.6(a) gives the cropped image where 36% of the image is removed, while Figure B.1.5 presents effects of such an attack. From this experiment, we have concluded in [BCG12b] that all embedding has similar behavior. All the percentage differences are so far less than 50% (which is the mean random error) and thus robustness is established.

5.3.7.2/ Robustness against compression

Robustness against compression is addressed by studying both JPEG and JPEG 2000 image compression. Results obtained in [BCG12b] are respectively presented in



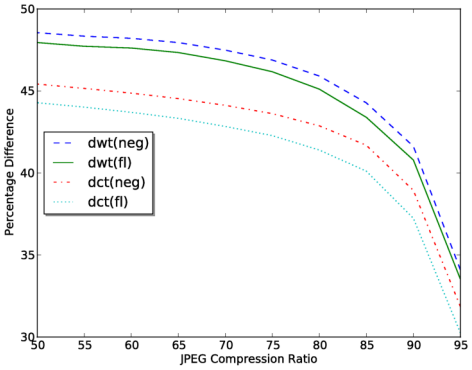
(a) Cropped image



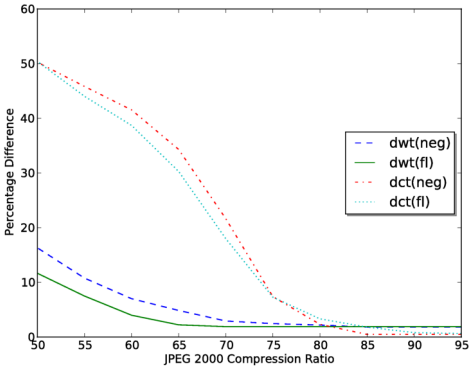
(b) Cropping effects

Figure 5.6: Cropping results

Fig. B.2(a) and Fig. B.2(b). Without surprise, DCT embedding which is based on DCT (as JPEG compression algorithm is) is more adapted to JPEG compression than DWT embedding. Furthermore, we have a similar behavior for the JPEG 2000 compression algorithm, which is based on wavelet encoding: DWT embedding naturally outperforms DCT one in that case.



(a) JPEG effects

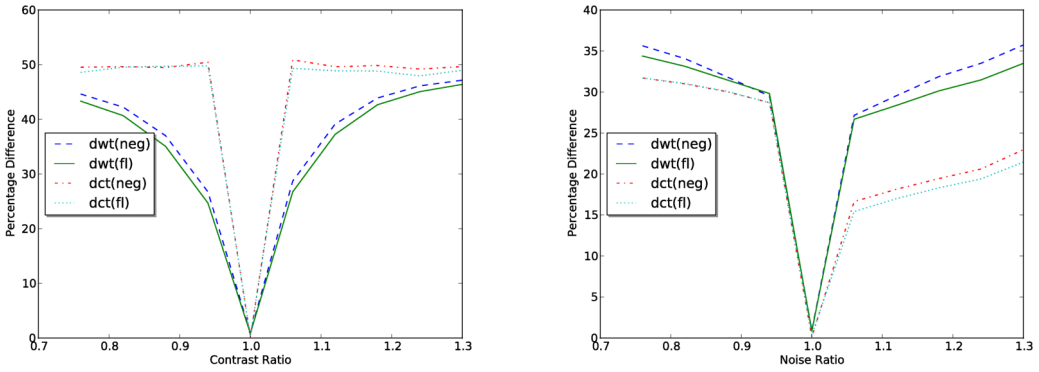


(b) JPEG 2000 effects

Figure 5.7: Compression results

5.3.7.3/ Robustness against contrast and sharpness

Contrast and Sharpness adjustments belong to the classical set of filtering image attacks. Results of such attacks are presented in Fig. 5.8 where Fig. 5.8(a) and Fig. 5.8(b) summarize effects of contrast and sharpness adjustment respectively [BCG12b].



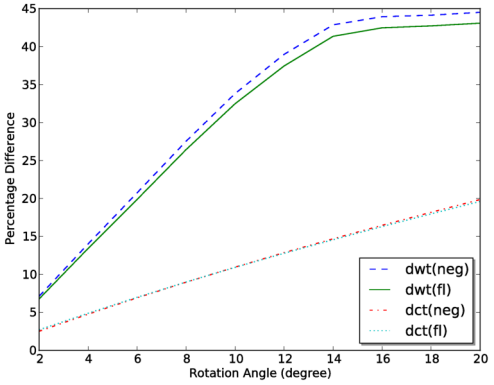
(a) Contrast effects (b) Sharpness effects

Figure 5.8: Filtering results

5.3.7.4/ Robustness against geometric transformations

Among geometric transformations, we have focused in [BCG12b] on rotations, *i.e.*, when two opposite rotations of angle θ are successively applied around the center of the image. In these geometric transformations, angles range from 2 to 20 degrees. Results obtained in [BCG12b] are summed up in Figure B.3: Fig. 5.9(a) gives the image of a rotation of 20 degrees whereas Fig. B.1.5 presents effects of such an attack. It is not a surprise that results are better for DCT embedding: this approach is based on cosine as rotation is.

(a) 20 degrees rotation image



(b) Rotation effects

Figure 5.9: Rotation attack results

5.3.8/ Evaluation of the Embedding

We are then left to set a convenient threshold that is accurate to determine whether an image is watermarked or not. Starting from a set of 100 images selected among the Boss image panel, we have computed in [BCG12b] the following three sets: the one with all the watermarked images W, the one with all successively watermarked and attacked images

WA, and the one with only the attacked images A. Notice that the 100 attacks for each image are selected among these detailed previously.

For each threshold $t \in [\![0, 55]\!]$ and a given image $x \in WA \cup A$, differences on DCT have been computed in [BCG12b]. The image has been claimed as watermarked if these differences are less than the threshold.

- In the positive case and if x really belongs to *WA* it is a True Positive (TP) case.
- In the negative case but if x belongs to *WA*, it is a False Negative (FN) case.
- In the positive case but if x belongs to *A*, it is a False Positive (FP) case.
- Finally, in the negative case and if x belongs to *A*, it is a True Negative (TN).

The True (resp. False) Positive Rate (TPR) (resp. FPR) has thus been computed by dividing the number of TP (resp. FP) by 100.

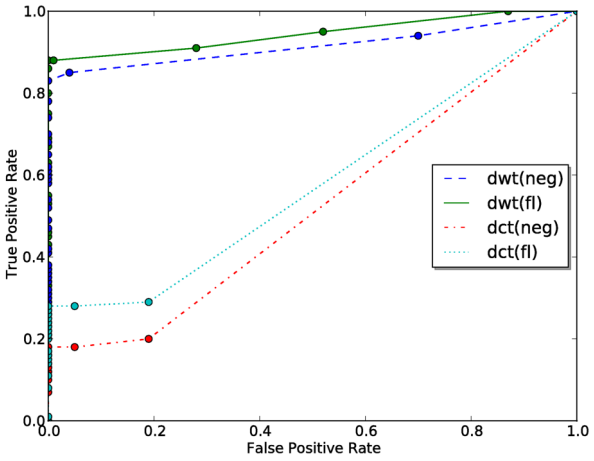


Figure 5.10: ROC curves for DWT or DCT embeddings

Figure B.5 recalled the obtained Receiver Operating Characteristic (ROC) curve. For the DWT, it shows that best results are obtained when the threshold is 45% for the dedicated function (corresponding to the point (0.01, 0.88)) and 46% for the negation function (corresponding to (0.04, 0.85)). It allows to conclude that each time LSCs differences between a watermarked image and another given image i' are less than 45%, we can claim that i' is an attacked version of the original watermarked content. For the two DCT embedding, best results have been obtained when the threshold is 44% (corresponding to the points (0.05, 0.18) and (0.05, 0.28)).

We have thus conclude some confidence intervals for all the evaluated attacks in [BCG12b]. The approach is resistant to:

- all the cropping where percentage is less than 85;
- compression where quality ratio is greater than 82 with DWT embedding and where quality ratio is greater than 67 with DCT one;

- contrast when strengthening belongs to $[0.76, 1.2]$ (resp. $[0.96, 1.05]$) in DWT (resp. in DCT) embedding;
- all the rotation attacks with DCT embedding and a rotation where angle is less than 13 degrees with DWT one.

5.4/ A Cryptographic Approach for Steganography

Our last reflections in the field of information hiding are theoretical ones, discussing the relevance of the stego-security notion in this field. Results of these thoughts, published in a second accepted paper at IIHMSP'13 (9th Int. Conf. on Intelligent Information Hiding and Multimedia Signal Processing, Beijing, China) are given thereafter [BGH13].

5.4.1/ Drawbacks of the stego-security notion

Theoretically speaking, the stego-security notion matches well with the idea of a perfect secrecy in the WOA category of attacks. However, its concrete verification raises several technical problems difficult to get around. These difficulties impact drastically the effective security of the scheme, as explained in [BGH13].

For instance, in a stego-secure scheme, the distribution of the set of watermarked images must be the same than the one of the original contents, no matter the chosen keys. But *how to determine practically the distribution of the original contents?* Furthermore, claiming that Alice can constitutes her own subset of well-chosen images having the same "good" distribution is quite unreasonable in several contexts of steganography: Alice has not *always* the choice of the supports. Moreover, it introduces a kind of bias, as the warden can find such similarities surprising. Suppose however that Alice is in the best situation for her, that is, she has the possibility to constitute herself the set of original contents. How can she proceed practically to be certain that all media into the set follow a same distribution $p(X)$? According to the authors opinion, Alice has two possible choices:

1. Either she constitutes the set by testing, for each new content, whether this media has a same distribution than the ones that have been already selected.
2. Or she forges directly new images by using existing ones. For instance, she can replace all the least significant bits of the original contents by using a good pseudo-random number generator.

In the first situation, Alice will realize a χ^2 test, or other statistical tests of this kind, to determine if the considered image (its least significant bits, or its low frequency coefficients, etc.) has a same distribution than images already selected. In that situation, Alice does not have the liberty to choose the distribution, and it seems impossible to find a scheme being able to preserve any kind of distribution, for all secret keys and all hidden messages. Furthermore, such statistical hypothesis testings are not ideal ones, as they only regard if a result is unlikely to have occurred by chance alone *according to a pre-determined threshold probability* (the significance level). Errors of the first (false positive) and second kind (false negative) occur necessarily, with a certain probability. In other

words, with such an approach, Alice cannot design a perfect set of cover contents having all the same probability $p(X)$. This process leads to a set of media that follows a distribution Alice does not have access to.

The second situation seems more realistic, it will thus be further investigated in the next section [BGH13].

5.4.2/ TOWARD A CRYPTOGRAPHICALLY SECURE HIDING

We recall in this section the theoretical framework for information hiding security we have proposed in [BGH13], which is more closely resembling that of usual approaches in cryptography, as those presented for PRNGs in Definitions 7 and 8. It allows to define the notion of steganalyzers, it is compatible with the new original scenarios of information hiding that have been dressed above, and it does not have the drawbacks of the stego-security definition. A similar but not equivalent approach can be found in the works of Barbier and Filiol (see [BM08b] for instance).

5.4.2.1/ DEFINITION OF A STEGOSYSTEM

Definition 24 (Stegosystem)**.** *Let $\mathcal{S}, \mathcal{M},$ and $\mathcal{K} = \mathbb{B}^\ell$ three sets of words on $\mathbb{B}$ called respectively the sets of supports, of messages, and of keys (of size ℓ).*

A stegosystem *on $(\mathcal{S}, \mathcal{M}, \mathcal{K})$ is a tuple $(\mathcal{I}, \mathcal{E}, inv)$ such that:*

- *$\mathcal{I}$ is a function from $\mathcal{S} \times \mathcal{M} \times \mathcal{K}$ to $\mathcal{S}$, $(s, m, k) \longmapsto \mathcal{I}(s, m, k) = s'$,*
- *$\mathcal{E}$ is a function from $\mathcal{S} \times \mathcal{K}$ to $\mathcal{M}$, $(s, k) \longmapsto \mathcal{E}(s, k) = m'$.*
- *inv is a function from $\mathcal{K}$ to $\mathcal{K}$, s.t. $\forall k \in \mathcal{K}, \forall (s, m) \in \mathcal{S} \times \mathcal{M}$, $\mathcal{E}(\mathcal{I}(s, m, k), inv(k)) = m$.*
- *$\mathcal{I}(s, m, k)$ and $\mathcal{E}(c, k')$ can be computed in polynomial time.*

$\mathcal{I}$ is called the insertion or embedding function, $\mathcal{E}$ the extraction function, s the host content, m the hidden message, k the embedding key, $k' = inv(k)$ the extraction key, and s' is the stego-content. If $\forall k \in \mathcal{K}, k = inv(k)$, the stegosystem is symmetric (private-key), otherwise it is asymmetric (public-key).

5.4.2.2/ HEADING NOTIONS

Definition 25 ($(T, \varepsilon)-$distinguishing attack)**.** *Let $S = (\mathcal{I}, \mathcal{E}, inv)$ a stegosystem on $(\mathcal{A}, \mathcal{M}, \mathcal{K})$, with $\mathcal{A} \subset \mathbb{B}^M$. A $(T, \varepsilon)-$distinguishing attack on the stegosystem S is a probabilistic algorithm $\mathcal{D} : \mathcal{A} \longrightarrow \{0, 1\}$ in running time T, such that there exists $m \in \mathcal{M}$,*

$$|Pr\left[\mathcal{D}\left(\mathcal{I}(s, m, k)\right) = 1 \mid k \in_R \mathcal{K}, s \in_R \mathcal{A}\right] - Pr\left[\mathcal{D}\left(x\right) = 1 \mid x \in_R \mathcal{A}\right]| \geqslant \varepsilon,$$

where the probability is also taken over the internal coin flips of $\mathcal{D}$, and the notation $\in_R$ indicates the process of selecting an element at random and uniformly over the corresponding set.

Definition [26]. *A stegosystem is $(T,\varepsilon)-$undistinguishable if there exists no $(T,\varepsilon)-$distinguishing attack on this stegosystem.*

Intuitively, it means that there is no polynomial-time probabilistic algorithm being able to distinguish the host contents from the stego-contents [BGH13].

5.4.2.3/ A CRYPTOGRAPHICALLY SECURE INFORMATION HIDING SCHEME

To show the effectiveness of the approach, we have provided and proven in [BGH13] a first cryptographically secure information hiding scheme, according to the definition above.

Theorem [8]. *Let*

$$\mathcal{S} = \left\{s_1^1, s_2^1, \ldots, s_{2^N}^1, s_1^2, s_2^2, \ldots, s_{2^N}^2, \ldots, s_1^r, s_2^r, \ldots, s_{2^N}^r\right\}$$

a subset of $\mathbb{B}^M = \mathcal{A}$. Consider $G : \mathbb{B}^L \longrightarrow \mathbb{B}^N$ a $(T,\varepsilon)-$secure pseudorandom number generator, and $\mathcal{I}(s_j^i, m, k) = s^i_{m\oplus G(k)}$. Assuming that r is a constant, and that from i, j one can compute the image s_j^i in time T_1, the stegosystem is $(T - T_1 - N - 1, \varepsilon)$-secure.

Intuitively, $\mathcal{S}$ is built from r images containing N bits of low information. The image s_j^i corresponds to the i-th image where the N bits are set to j.

The last application in information security, namely the chaotic hash functions, are detailed in Appendix C.

III

Applications to Wireless Sensor Networks' Security

6

Low Cost Monitoring and Intruders Detection using Wireless Video Sensor Networks

In [BGMP11, BGMP12], we have presented a solution to the joint scheduling problem in surveillance applications using a wireless video sensor network (WVSN). We have provided a chaotic sleeping scheme and conducted a theoretical and simulation analysis of both performances and security. To our knowledge, only random approaches have been extensively studied in the literature to turn off video nodes without degrading the surveillance quality. Even if such methods present good scores in detecting random intrusions while preserving the lifetime of the network, they do not encompass the situation of a malicious attacker. That is to say, the intruder is not supposed to know something about the surveillance scheme, he cannot observe the behavior of WVSN for a while, or he is not authorized to deduce anything from his possible knowledge. In this chapter, we recall our proposal to tackle the situation where the attacker is not supposed passive: he is smart and does not necessarily choose a random way to achieve his intrusion. In addition of preserving the network lifetime and being able to face random attacks, we have shown in [BGMP11, BGMP12] that our scheme is also capable to withstand attacks of a malicious adversary due to its unpredictable behavior.

6.1/ Smart Threats

6.1.1/ Introduction

Let us suppose that an adversary tries to reach a location X into the area without being detected. We have considered in [BGMP11] that this situation leads to two categories of attacks against WVSN surveillance.

On the one hand, the attacker only knows that the area is under surveillance. He tries to take its chance, for example by following the shortest way or by trying a random path. In this first category of attacks that we called “blind elementary attacks” in [BGMP11], the intruder does not know how the surveillance is achieved as he does not observe the WVSN.

On the other hand, in the second category of attacks, called “malicious attacks” in [BGMP11], the intruder is supposed to be intelligent. He can try to take benefits from his observations to understand the behavior of the WVSN. After having recorded the dynamic of the WVSN for a given time, the malicious intruder can try to determine when video nodes are turned on. This prediction can help the intruder to find a way to reach X without being detected.

In our opinion, the most reasonable way to evaluate the consequences of a malicious attack is to suppose, following [BGMP11], that the intruder has access to the surveillance scheme. With this supposition, our security model proposed in [BGMP11] encompasses the case where an attacker can have a physical access to a given node, thus determining the embedded mechanism used for video surveillance. In this Kerckhoffs-based principle, the attacker knows all but the initial parameters of the nodes. Moreover, he can observe the WVSN for a while. To achieve his intrusion, he can use all of the acquired knowledge – the sole difficulty is his lack of a secret parameter (the secret key) used to initialize the surveillance process.

The context of blind elementary attacks is well-known and understood: it has been studied a lot in the last decade, and various solutions have already been proposed, see [BGMP11] for related works. However, to the best of our knowledge, the case of an intelligent intruder (smart threat) has not yet really been treated. In [BGMP11], we have proposed a scheme able to withstand attacks encompassing these malicious intrusions, and thus to offer a first solution to the problem raised by the smart threats existence hypothesis.

Technically speaking, the approach proposed in [BGMP11] offers several benefits. Firstly, the node scheduling algorithm does not need location information. Therefore, the energy consumption is reduced because there is no need to locate the node itself and its neighbors. Secondly, we have shown that it performs as well as a random scheduling, in terms of lifetime and intrusion detection against blind elementary attacks (see Section 6.4). Lastly, due to its chaotic properties, its coverage is unpredictable, and thus a malicious adversary has no solution to attack the network (Section 6.3).

6.1.2/ CLASSIFICATION OF MALICIOUS ATTACKS

We have initiated during our thesis a link between security notions in wireless sensor networks (WSNs) and digital watermarking, by proposing in [BGM10b, BGM14] to use digital watermarking techniques for data aggregation through WSNs. We then have naturally proposed in [BGMP12], which is an extension of [BGMP11], to translate notions from the information hiding security field to describe malicious attacks in wireless video sensor networks, due to numerous relations between these two disciplines. This proposal is recalled in what follows.

When a malicious adversary attacks a WVSN, he can concentrate his efforts either on the global network or on some specific nodes. Depending on the considered situation, he can perform either an active attack, modifying the network architecture or a node, or a passive attack based only on observations. He can have access to several WVSN using the same algorithm. Furthermore, he can build its own network to make some experiments. His objective is to find the secret key used in the targeted network: with this knowledge, the attacker will be able to predict the behavior of the video sensor nodes.

Active attacks have been already investigated several times in the literature. These stud-

ies encompass the cases where nodes can be added, moved, modified, or removed, where communications between nodes can be observed or changed, and where the global architecture of the network is attacked. However, some WVSN are such that any modification of the network is signaled, leading to the impossibility of such active attacks. On the contrary, passive observations and deductions of a malicious attacker are always possible. To the best of our knowledge, these threats have not been investigated before [BGMP12].

The passive malicious attacks have been classified in [BGMP12] as follows.

- In the **Target Only Attack (TOA)**, the adversary can only observe targeted networks.
- In the **Constant Key Attack (CKA)**, the adversary has access to several WVSNs using the same secret key. The areas under surveillance and the network architecture change from one WVSN to another, but the attacker knows that all these networks use the same algorithm with the same secret key.
- In the **Known Original Attack (KOA)**, the attacker had previously accessed to the WVSN and its area. He had the opportunity to test various keys in a previous access. He hopes that this knowledge will help him to determine a way to realize his intrusion when the WVSN is really launched.
- In the **Chosen Key Attack (CKA)**, the adversary has access to an exact copy of the network and area under surveillance than the one he want to attack. He has realized for instance a miniature model or a computer simulator having exactly the same behavior than the targeted network and its area. He can thus try several secret keys, and if he achieves to reproduce exactly the same behavior for the network, then he can reasonably suppose that the true secret key has been discovered.
- Finally, in the **Estimated Original Attack (EOA)**, the attacker has only an estimation/approximation of the network and its area.

In each of these categories, the sole objective of the attacker is to obtain the value of the secret key. With this knowledge, he will be able to determine the WVSN behavior, finding by doing so a way to achieve his intrusion.

6.1.3/ SECURITY LEVELS IN CKA

We now take place in the Chosen Key Attack problem, and following [BGMP12], we recall here how to map the stego-security notion in this field. Let k_0 be the secret key used to initiate the video-surveillance. Denote by Y_k the probabilistic model that the attacker can build with his observations, and by $\mathbb{K}$ the set of all possible keys.

Definition 27 (Insecurity)**.** *The WVSN is insecure against the Target Only Attack if and only if* $\exists k_1 \in \mathbb{K}, p(Y_{k_1}) = p(Y_{k_0})$ *and* $\forall k_2 \in \mathbb{K}, p(Y_{k_2}) \neq p(Y_{k_0})$

On the contrary,

Definition 28 (Security)**.** *The WVSN is secure against the Target Only Attack if and only if* $\forall k \in \mathbb{K}, p(Y_k) = p(Y_{k_0})$

In that situation, we can easily translate the fact that the mutual information $\mathcal{I}(k_0, Y_{k_0})$ is equal to 0, which means a *perfect secrecy* [BGMP12].

6.2/ CHAOS-BASED SCHEDULING

We then have proposed in [BGMP11] a scheduling process making possible to withstand attacks of a malicious intruder. This scheme is summarized thereafter.

6.2.1/ NETWORK CAPABILITIES

The WVSN is supposed to be constituted by 2^{N} nodes $V_i, i \in [\![0, 2^{\mathsf{N}} - 1]\!]$. Each V_i is able to wake up on a specific signal, to survey a given area (and to detect intrusions), to send a wake up signal to another node V_j, and to go to sleep when it is required. Furthermore, it is supposed that V_i embeds:

- The mechanisms required by the intrusion detection: a sensing function $c_i(t)$, such as a camera, which returns some digital data at each listening time, and a decision function $d_i(c)$ which returns if an intrusion is detected in this sensing values ($c_i(t)$) or not.
- An internal clock having the time $T_i = r_i T_0$ as a reference.
- A vector of N binary digits, called *the state of the system* V_i, and the capability to swap each bit of this vector ($0 \leftrightarrow 1$).
- An integer e_i, called *listening time*, initialized to 0.

In other words, each node V_i can achieve chaotic iterations, as they have been presented at the beginning of this manuscript. Thus, each node can compute, easily and by using a few resources, a hash value and some pseudorandom numbers following methods that have been recalled in Chapters C and 4 respectively. We denote by g_i the seed of the PRNG used in node V_i, which is equal to a secret parameter p_i at time $t = 0$. This secret parameter with N bits has been generated by a cryptographically secure PRNG, and thus it is uniformally distributed into $[\![0; 2^{\mathsf{N}} - 1]\!]$. The internal state of node V_i is initialized to the binary decomposition of g_i.

6.2.2/ DEPLOYING THE NETWORK

The deployment of video sensor nodes in the physical environment is the first operation (step) in the network lifecycle proposed in [BGMP11]. It may take several forms. Sensor nodes may be randomly deployed dropping them from a plane, and placed one by one by a human or a robot. Deployment may be a one time activity or a continuous process. These methods have been extensively studied in the literature. In the method proposed in [BGMP11], the sole requirement to satisfy is to guarantee the uniform distribution into the region of interest.

6.2.3/ INITIALIZATION OF THE WVSN

At time $t = 0$, a subset $\mathcal{I} \subset [\![0, 2^{\mathsf{N}} - 1]\!]$ of nodes are in the wake-up state and $\forall i \in \mathcal{I}, e_i^{t_0} = T_i$.

6.2.4/ SURVEILLANCE

The principle of surveillance application is defined as follows. At each time $t_j = j \times T_0, j = 1, 2, \ldots$:

1. If a sleeping node V_i has received $n_i^{t_j - 1} \geqslant 1$ wake up orders during the time interval $[t_j - 1, t_j]$, then it goes into active mode and sets its listening time $e_i^{t_j}$ to $n_i^{t_j - 1} T_i$.
2. If an active node V_i has received $n_i^{t_j - 1} \geqslant 1$ orders to wake up during the time interval $[t_{j-1}, t_j]$, then it increments its listening time: $e_i^{t_j} = e_i^{t_j - 1} + n_i^{t_j - 1} T_i$.
3. For each node V_i having a listening time $e_i^{t_j} \neq 0$:
   - V_i ensures the surveillance of its area during T_0,
   - If, during this time interval, an intrusion is detected, then the WVSN is under alert.
   - If t_j is the first listening time of V_i after having activated, then:
     - The hash value $h_i^{t_j}$ of the sensed value $c_i(t_j)$ is computed.
     - The seed g_i of the PRNG of V_i is set to $h_i^{t_j} + t_j$, where $+$ is the concatenation of the digits of $h_i^{t_j}$ and t_j (thus even if $h_i^{t_j} = h_i^{t_k}, k < j$, we have $g_i^{t_j} \neq g_i^{t_k}$).
     - The N bits of the state of the system V_i are set to $E_i^{t_j}$, where $E_i^{t_j}$ is the binary decomposition of i shown as a binary vector of length N.
4. N bits are computed with the PRNG of V_i. These bits define an integer $S_i^{t_j} \in [\![0, 2^{\mathsf{N}} - 1]\!]$. Then the bit of $E_i^{t_j}$ in position $S_i^{t_j}$ is switched, which leads to a new state $E_i^{t_j + 1}$. By doing so, chaotic iterations (CIs) are realized.
5. Each active node V_i decreases its listening time: $e_i^{t_j} = e_i^{t_j} - 1$.
6. For each active node having its listening time $e_i^{t_j} = 0$:
   - V_i sends the wake up order to node V_k, where $k \in [\![0, 2^{\mathsf{N}} - 1]\!]$ is the integer whose binary decomposition is the last state of the system V_i ($E_i^{t_j + 1}$).
   - V_i goes to sleep.

6.3/ THEORETICAL STUDY

6.3.1/ SCHEDULING AS CHAOTIC ITERATIONS

The scheduling scheme presented in [BGMP11] can be described as CIs. The global state E^t of the whole system is constituted by the reunion of each internal state E_i^t of each

node i. This is an element of $\mathbb{B}^{\mathsf{N}\times 2^{\mathsf{N}}}$. The strategy at time t is the subset of $[\![0;\mathsf{N}\times 2^{\mathsf{N}}]\!]$ constituted by all of the strategies that are computed into the awaken nodes at time t. More precisely, if the node V_k has computed the strategy S_k^t at time t, then the global strategy S^t will contain the value $S_k^t + k \times \mathsf{N}$. Lastly, the iteration function is the vectorial negation defined : $\mathbb{B}^{\mathsf{N}\times 2^{\mathsf{N}}} \rightarrow \mathbb{B}^{\mathsf{N}\times 2^{\mathsf{N}}}$. A subsequence E^{m^t} is extracted from E^t, which determines the changes that occur in the network: nodes whose binary id is into E^{m^t} are nodes that achieve the surveillance at the considered time. Let us remark that S^k and m^k depend both on the outside world, due to the fact that S_i^t are regularly seeded with the digest of some sensed values.

6.3.2/ COMPLEXITY

Even if the hash function and the PRNG presented in previous chapters can be replaced by any cryptographically secure hash function and PRNG, we do not recommend their substitution. Indeed, all of the operations used by our scheme can be achieved by CIs. Each iteration of CIs is only constituted by the negation of a few binary digits. Obviously, such an operation is fast and does not consume a lot of energy. By doing so, we thus obtain an efficient video surveillance scheduling scheme compliant with WVSN requirements. Section 6.4 will detail more quantitatively this fact.

6.3.3/ COVERAGE

The coverage of the whole area has been guaranteed in [BGMP11] due to the following reasons.

Firstly, the scheduling process corresponds to CIs. These iterations are chaotic according to Devaney, thus they are transitive. This transitivity property is the formulation of an uniform distribution in terms of topology. It claims that the system will never stop to visit any sub-region of the whole area, regardless of how tiny the region is.

Secondly, as the choice of the nodes to wake up at each time are done by using CIs, this selection corresponds to the returned value of our PRNG recalled in Chapter 4. This CIPRNG(X,Y) version 1 takes two PRNGs X,Y as input sequences, realizes CIs with X as strategy, the vectorial negation as update function, and selects the states to publish as outputs by using the second PRNG Y. By such a combination, we improve the statistical properties of the inputted PRNG used as strategy, and we add chaotic properties. Indeed, the scheduling process corresponds to the CIPRNG(X,Y) generator, with X=m and Y=S. As Y is statistically perfect in [BGMP11] (Y is CIPRNG(ISAAC,ISAAC) version 1, which can pass the whole NIST, DieHARD, and TestU01 batteries of tests, see Chapter 4), the random distribution of the states is then guaranteed.

Finally, experiments of Section 6.4 will recall that this intended uniform coverage is well obtained in practice.

6.3.4/ SECURITY STUDY

6.3.4.1/ QUALITATIVE APPROACH

Let us suppose that Oscar, an intruder, knows that the scheduling process is based on CIs, i.e. he knows the whole algorithm, except the seeds that have been used to initiate the PRNGs of each node. By doing so, we respect the Kerckhoffs' principle: the adversary has all except the secret key. Oscar's desire is to reach a particular location X of the area without being detected. To achieve his goal, he can choose two strategies. On the one hand, he can try a blind elementary attack, either by following a random way from its position to X, of by choosing the shortest path. The next subsection and the experiments recall the arguments of [BGMP11], which indicate that such an attack cannot work. On the other hand, Oscar can try to take benefits both from his knowledge and his observations. However, if he can determine the nodes that are awaken at time t, he cannot predict the awaken nodes at time $t+1, t+2, ...$ To do so, he should be able to obtain $S^{t+1}, S^{t+2}, ...$, which are computed from the digests of some values that will be sensed in the future. As our hash function satisfy the avalanche effect, due to its chaotic properties, any error on the sensed value lead to a completely different digest [BGMP11].

As Oscar cannot determine the sensed values of each node, at each time and with an infinite precision, he does not have the knowledge of the current state of the global system. He has only access to an approximation of this state. As the global scheduling process is chaotic, this error on the initial condition is magnified at each iteration, leading to the impossibility for Oscar to predict the scheduling process. This qualitative approach for security is recalled in the section below.

6.3.4.2/ CHAOTIC PROPERTIES

We have investigated in [BGMP12] the topological properties presented by the proposed video-surveillance scheme. As proven in [GFB10] and recalled at the beginning of this manuscript, chaotic iterations are expansive and topologically mixing when f is the vectorial negation f_0. Consequently, these properties are inherited by the WVSN presented previously, which induce a greater unpredictability. Any difference on the initial parameter of the WVSN is in particular magnified up to be equal to the expansivity constant.

The topological transitivity property, for its part, implies indecomposability. Hence, reducing the observed area in order to simplify its complexity, is impossible if $\Gamma(f)$ is strongly connected. Moreover, under this hypothesis the surveillance scheme is strongly transitive. Among other things, the strong transitivity leads to the fact that without the knowledge of the initial awaken nodes, all scheduling are possible. Additionally, no nodes of the output space can be discarded when studying the video-surveillance scheme: this space is intrinsically complicated and it cannot be decomposed or simplified [BGMP12].

Finally, these WVSNs possess the instability property. This property, which is implied by sensitive point dependence on initial conditions, leads to the fact that in all neighborhoods of any point x there are points that can be apart by ε in the future through iterations of the WVSN. Thus, we can claim that the behavior of these networks is unstable when $\Gamma(f)$ is strongly connected.

6.3.4.3/ CRYPTANALYSIS IN CKA FRAMEWORK

As recalled in Section 6.3.1, the proposed videosurveillance scheme can be rewritten as:

$$\begin{cases} X^0 \in \mathcal{X} \\ X^{k+1} = G_{f_0}(X^k), \end{cases} \tag{6.1}$$

where the phase space is $\mathcal{X} = [\![1; \mathsf{N} \times 2^{\mathsf{N}}]\!]^{\mathbb{N}} \times \mathbb{B}^{\mathsf{N} \times 2^{\mathsf{N}}}$, X^0 depends on a secret parameter $p = (p_1, \ldots, p_{\mathsf{N}}) \in \left(\mathbb{B}^{\mathsf{N}}\right)^{\mathsf{N}}$ whose binary digits are uniformy distributed, and f_0 stands for the vectorial negation on $\mathbb{B}^{\mathsf{N} \times 2^{\mathsf{N}}}$.

We then have proven in [BGMP12] that,

Proposition 9. *The videosurveillance scheme proposed in this chapter is secure when facing a chosen key attack.*

6.4/ SIMULATION RESULTS

This section recalls simulation results, proposed in [BGMP11] and extended in [BGMP12], on comparing our chaotic approach to the standard C++ `rand()`-based approach with random intrusions. We have used the OMNET++ simulation environment and the next node selection will either use chaotic iterations or the C++ `rand()` function (`rand()` % 2^n) to produce a random number between 0 and 2^n.

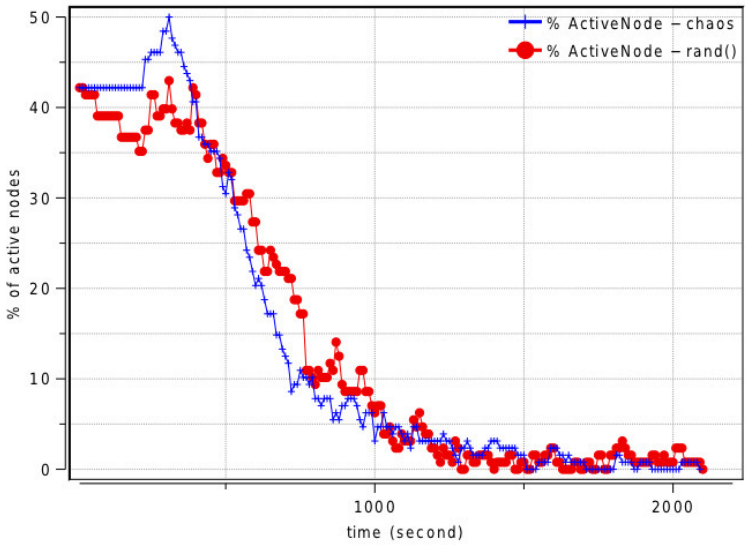


Figure 6.1: Percentage of active nodes.

For these set of simulations, 128 sensor nodes (therefore $n = 7$) are randomly deployed in a $75 \times 75m^2$ area. Unless specified, sensors have a 36^o AoV and sensor nodes capture at the rate of 0.2fps. Each node starts with a battery level of 100 units and taking 1 picture consummes 1 unit of battery. When a node V_i is selected to wake up, it will be awake for T_i seconds. We set all $T_i = T = 20s$ in [BGMP11]. According to the behavior recalled in Section 6.2, before going to sleep after an activity period of e_iT, V_i will determine the next node to be waked up. It can potentially elect itself in which case V_i stays active for at

least another T period. The elected node can be already active, in which case it simply increases its e_i counter. We set about 50% of the sensor nodes to be active initially (each sensor draws a random value between 0 and 1 and if the value is greater than 0.5, it will be active). This initial threshold is tunable but we did not try to vary this parameter in [BGMP11]. The obtained results have been averaged over 10 simulation runs with different initial seeds. Figure 6.1 recalls the percentage of active nodes. Both the chaotic and the standard `rand()` function have similar behavior: the percentage of active nodes progressively decreases due to battery shortage.

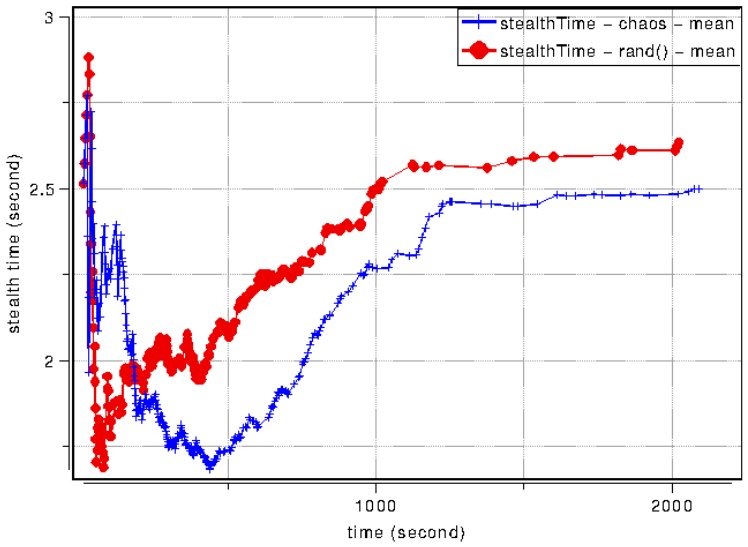


Figure 6.2: Stealth time.

To compare both approaches in term of surveillance quality, we recorded to stealth time when intrusions are introduced in the area of interest [BGMP11]. The stealth time is the time during which an intruder can travel in the field without being seen. The first intrusion starts at time 10s at a random position in the field. The scan line mobility model is then used with a constant velocity of 5m/s to make the intruder moving to the right part of the field. When the intruder is seen for the first time by a sensor, the stealth time is recorded and the mean stealth time computed. Then a new intrusion appears at another random position. This process is repeated until the simulation ends (*i.e.*, no more sensor nodes with energy). Figure 6.2 recalls the obtained mean stealth time over the whole simulation duration. Figure 6.3, for its part, shows the same results but with a sliding window averaging filter of 20 values. As the nodes are uniformly distributed in the area of interest, we found in [BGMP11] a strong correlation between the percentage of active nodes and the stealth time, as it can be expected. The result we wanted to highlight in [BGMP11] is that our chaotic node selection approach has a similar level of performance in the presence of random intrusions than standard `rand()` function while providing a formal proof of non-prediction by malicious intruders.

The last result shown in [BGMP11] is the energy consumption distribution. We recorded every 10s the energy level of each sensor node in the field and computed the mean and the standard deviation. Figure 6.4 recalls the evolution of the standard deviation during the network lifetime. We can see that the chaotic node selection provides a slightly better distribution of activity than the standard `rand()` function [BGMP11].

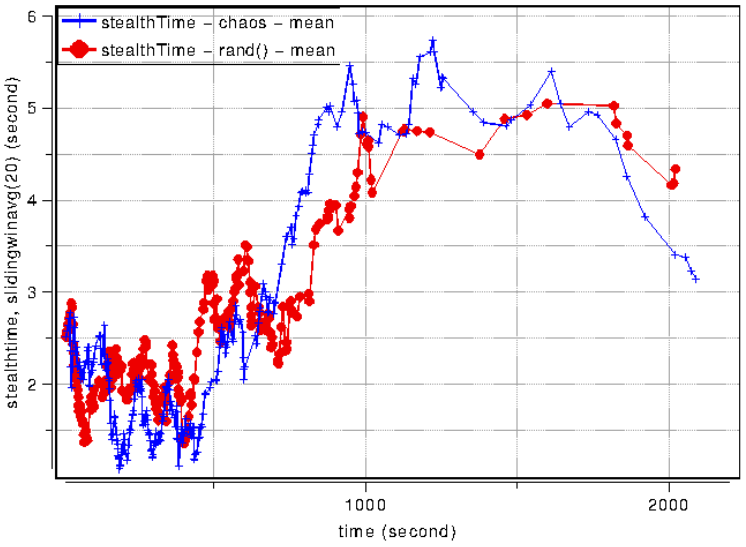


Figure 6.3: Stealth time.

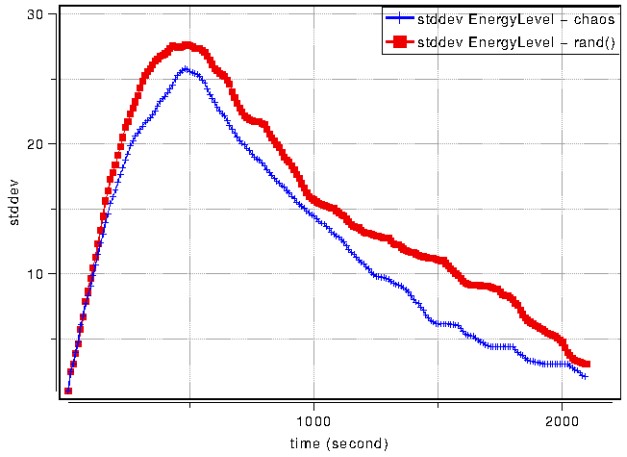


Figure 6.4: Evolution of the energy consumption's standard deviation.

7

TOWARD A SECURITY FRAMEWORK FOR WIRELESS SENSOR NETWORKS

Previous chapter that focuses on WSN based videosurveillance has emphasizes, in our opinion, the importance to have a mathematical approach for dealing with security when facing malicious attackers in wireless sensor networks. These first investigations, inspired by information hiding security whose issues have been signaled previously in this manuscript, have then been more systematically regarded, with the need to bring a rigorous framework for security questions in WSNs. This is why a complete security framework for wireless sensor networks, investigating all the aspects of security related to WSNs, is proposed in [BGM] (currently submitted). This rigorous theoretical framework encompasses: (1) secure communication (communication systems, indistinguability, nonmalleability, message detection), (2) cryptographically secure scheduling, (3) secure routing, and (4) secure aggregation of data. We tried to constitute in [BGM] a formalism as rigorous as possible, inspired by equivalent formulations in cryptography, but compliant with all the constraints of a WSN. This contribution is summarized thereafter.

7.1/ SECURITY IN WSN: GENERAL PRESENTATION

Wireless nature of communication, lack of infrastructure and uncontrolled environment improve capabilities of adversaries in WSN. Stationary adversaries equipped with powerful computers and communication devices may access whole WSN from a remote location. They can gain mobility by using powerful laptops, batteries and antennas, and move around or within the WSN. In this section, we consider a WSN where nodes communicate together by sending data publicly. These transmitted data contain a message whose confidentiality must be preserved. For instance, transmitted data is the cryptogram of a message, modulated in an electromagnetic radiation, or the message is dissimulated into the electromagnetic radiation by using a spread spectrum information hiding technique.

Wireless communication helps adversaries to perform variety of attacks. A secure communication can be used to provide the following general security goals:

- **One-wayness (OW):** The adversary who sees transmitted data is not able to compute the corresponding message.
- **Indistinguability (IND):** Observing transmitted data, the adversary learns nothing about the contained message.

- **Non-malleability (NM):** The adversary, observing data for a message m, cannot derive another data for a meaningful message m' related to m.

The OW and IND goals relate to the confidentiality of messages through the WDN. The IND goal is, however, much more difficult to achieve than the one-wayness. Non-malleability guarantees that any attempt to manipulate the observed data in order to obtain a valid data will be unsuccessful (with a high probability).

The power of a polynomial attacker (with polynomial computing resources) very much depends on his/her knowledge about the system used to transform *information* in *data*. The weakest attacker is an outsider who knows the public embedding algorithm together with other public information about the setup of the system. The strongest attacker seems to be an insider (he/she is inside the network) who can access the extraction device (recovering information from data) in regular interval. The access to the extraction key is not possible as the extraction device is assumed to be tamperproof.

An *extraction oracle* is a formalism that mimics an attacker's access to the extraction device. The attacker can experiment with it providing *data* and collecting corresponding *information* from the oracle (the attacker cannot access to the decryption key). In general, the public-key WSN may be subject to the following attacks (ordered in increasing strength):

- **Chosen information attack (CIA):** The attacker knows the embedding algorithm and the public elements including the public key (the embedding oracle is publicly accessible).
- **Nonadaptative chosen data attack (CDA1):** The attacker has access to the extraction oracle before he sees a data that he wishes to manipulate.
- **Adaptative chosen data attack (CDA2):** The attacker has access to the extraction oracle before and after he observes a data s that he wishes to manipulate (assuming that he is not allowed to query the oracle about the data s).

The security level that a public-key WSN achieves can be specified by the pair (goal, attack), where the goal can be either OW, IND, or NM, and the attack can be either CIA, CDA1, or CDA2. For example, the level (NM,CIA) assigned to a public-key network says that the system is nonmalleable under the chosen message attack. There are two sequences of trivial implications

- (NM,CDA2) $\Rightarrow$ (NM, CDA1) $\Rightarrow$ (NM,CIA),
- (IND,CDA2) $\Rightarrow$ (IND, CDA1) $\Rightarrow$ (IND,CIA),

which are true because the amount of information available to the attacker in CIA, CDA1, and CDA2 grows. Figure 7.1 shows the relation among different security notions. Consequently, we can identify the hierarchy of security levels. The top level is occupied by $(NM, CDA2)$ and $(IND, CDA2)$. The bottom level contains (IND, CIA) only as the weakest level of security. If we are after the strongest security level, it is enough to prove that our network attains the $(IND, CDA2)$ level of security.

$$\begin{array}{ccccc} (NM, CDA2) & \longrightarrow & (NM, CDA1) & \longrightarrow & (NM, CIA) \\ \updownarrow & & \downarrow & & \downarrow \\ (IND, CDA2) & \longrightarrow & (IND, CDA1) & \longrightarrow & (IND, CIA) \end{array}$$

Figure 7.1: Relations among security notions

7.2/ Rigorous Formalism for Secure Communications in WSNs

In this section, we explain the new principles formalism for secure communication in wireless sensor networks proposed in [BGM].

7.2.1/ Communication system in a WSN

Definition [29] (Communication system)**.** *Let $\mathcal{S}, \mathcal{M}$, and $\mathcal{K} = \{0,1\}^\ell$ be three sets of words on $\{0,1\}$ called respectively the sets of transmission supports, of messages, and of keys (of size ℓ).*

A communication system *on $(\mathcal{S}, \mathcal{M}, \mathcal{K})$ is a tuple $(\mathcal{I}, \mathcal{E}, inv)$ such that:*

- $\mathcal{I} : \mathcal{S} \times \mathcal{M} \times \mathcal{K} \longrightarrow \mathcal{S}$, $(s, m, k) \longmapsto \mathcal{I}(s, m, k) = s'$, *is the* insertion function*, which put the message m into the support of transmission s according to the key k, leading to the transmitted data s'.*
- $\mathcal{E} : \mathcal{S} \times \mathcal{K} \longrightarrow \mathcal{M}$, $(s, k) \longmapsto \mathcal{E}(s, k) = m'$, *defined as the* extraction function*, which extracts a message m' from a transmitted data s, depending on a key k.*
- $inv : \mathcal{K} \longrightarrow \mathcal{K}$, **s.t.** $\forall k \in \mathcal{K}, \forall (s, m) \in \mathcal{S} \times \mathcal{M}, \mathcal{E}(\mathcal{I}(s, m, k), inv(k)) = m$, *which is the function that can "invert" the effects of the key k, producing the message m that has been embedded into s using k.*
- *$\mathcal{I}$ and $\mathcal{E}$ can be computed in polynomial time, and $\mathcal{I}$ is a probabilistic algorithm (the same values inputted twice produce two different transmitted data).*

k is called the embedding key and $k' = inv(k)$ the extraction key. If $\forall k \in \mathcal{K}, k = inv(k)$, the communication system through the WSN is said symmetric *(private-key), otherwise it is* asymmetric *(public-key).*

7.2.2/ Indistinguability

Suppose that the adversary has two messages m_1, m_2 and a transmitted data s in his/her possession. He/she knows that s contains either m_1 or m_2. Our intention is to define the fact that, having all these materials, the key, and the insertion function (we take place into the (IND,CIA) context), he cannot determine with a non negligible probability the message that has been embedded into s.

The difficulty of the challenge comes, for a large extend, from the fact that the insertion algorithm $\mathcal{I}$ is a probabilistic one, which is a common sense assumption usually required in cryptography.

Definition [30]. *An Indistinguability I-adversary is a couple* $(\mathsf{A}_1, \mathsf{A}_2)$ *of nonuniform algorithms, each with access to an oracle* $\mathcal{O}$*.*

Definition [31] (Indistinguability)**.** *For a public communication system in WSN* $(\mathcal{I}, \mathcal{E}, inv)$ *on* $(\mathcal{S}, \mathcal{M}, \{0,1\}^\ell)$*, we define the advantage of an I-adversary* A *by*

$$Adv_{\mathsf{A}}^{I-\mathcal{O}} = Pr\left[\begin{array}{c} k \xleftarrow{\$} \{0,1\}^\ell \\ (m_0, m_1, s) \leftarrow \mathsf{A}_1(k) \\ b \leftarrow \{0,1\} \\ \alpha = \mathcal{I}(s, m_b, k) \end{array} : \mathsf{A}_2(k, s, m_1, m_2, \alpha) = b\right]$$

We also define the insecurity of $S = (\mathcal{I}, \mathcal{E}, inv)$ with respect to the Indistinguability as

$$InSec_S^{I-\mathcal{O}}(t) = \max_{\mathsf{A}} \left\{ Adv_{\mathsf{A}}^{I-\mathcal{O}} \right\}$$

where the maximum is taken over all adversaries A with total running time t.

We distinguish three kinds of oracles:

- The Non-adaptative oracle, denoted $\mathcal{NA}$, where A_1 and A_2 can only access to the elements of the communication system.
- The Adaptative oracle, denoted $\mathcal{AD}1$, where A_1 has access to the communication system and to an oracle that can in a constant time provide a message m' from any transmitted data $\mathcal{I}(M', m', k')$, without knowing neither M' nor k' nor $inv(k')$. In this context, A_2 has no access to this oracle.
- The Strong adaptative oracle, denoted $\mathcal{AD}2$, where A_1 has access to the communication system and to an oracle that can in a constant time provide a message m' from any transmitted data $\mathcal{I}(M', m', k')$, without knowing neither M nor k' nor $inv(k')$. In this context, A_2 has also access to this oracle but for the message $\mathcal{I}(M, m_b, k)$.

7.2.3/ RELATION BASED NON-MALLEABILITY

In some scenarios malicious nodes can integrate the WSN, hoping by doing so to communicate false information to the other nodes. We naturally suppose that communications are secured. The problem can be formulated as follows: is it possible for the attacker to take benefits from his/her observations, in order to forge transmitted data either by embedding erroneous messages, or sending data that appear to be similar with what a node is supposed to produce?

As wireless sensor networks have usually a dynamical architecture, the (dis)appearance of nodes is not necessarily suspect. Authentication protocols can be deployed into the WSN, but in some cases such authentication is irrelevant, because of its energy consumption, communication cost, or rigidity. We focus, in this section, on the possibility to propose a secured communication scheme in WSN that prevents an attacker to forge such malicious transmitted data. Such non-malleability property can be formulated as follows.

Definition [32]. *A Relation Based NM-adversary is a nonuniform algorithm* A *having access to an oracle* $\mathcal{O}$*.*

Definition 33 (Relation Based Non-malleability)**.** *For a public communication system* $(\mathcal{I}, \mathcal{E}, inv)$ *on* $(\mathcal{S}, \mathcal{M}, \{0,1\}^\ell)$*, define the advantage of a NM-adversary* A *by*

$$Adv_{\mathsf{A}}^{NM-\mathcal{O}}(m) = Pr\left[\begin{array}{c} s \leftarrow \mathcal{S} \\ k \xleftarrow{\$} \{0,1\}^\ell \\ s' \leftarrow \mathsf{A}(\mathcal{I}(s,m,k)) \\ m' \leftarrow \mathcal{E}(s',k) \end{array} : \; m' \in R(m)\right]$$

where $R : \mathcal{M} \longrightarrow \mathcal{P}(\mathcal{M})$ is a function that map any message m to a subset of $\mathcal{M}$ containing messages related to m (for a given property). For instance, if we suppose that an attacker has inserted or corrupted some nodes in a network that measures temperature, he can make these nodes send wrong temperatures values fixed *a priori*.

We can now define the insecurity of $S = (\mathcal{I}, \mathcal{E}, inv)$ with respect to the Relation Based Non-malleability as

$$InSec_S^{NM-\mathcal{O}}(t) = \max_{\mathsf{A}}\left\{\max_{m \in \mathcal{M}}\left\{Adv_{\mathsf{A}}^{NM-\mathcal{O}}(m)\right\}\right\}$$

where the maximum is taken over all adversaries A with total running time t. Similar kinds of oracles than previously can be defined in this context.

7.2.4/ MESSAGE DETECTION RESILIENCY

We now address the particular case where transmitted data can contain or not an embedded message. For security reasons, it is sometimes required that an attacker cannot determine when information are transmitted through the network. For instance, in a video surveillance context, suppose that an attacker can determine when an intrusion is detected, or when something considered as suspicious is forwarded through the nodes to the sink. Then he/she can use this knowledge to deduce what kind of behavior is suspicious for the network, adapting so his/her attacks. Decoys are often proposed to make such attacks impossible: transmitted data do not always contain information, some of the communications are only realized to mislead the attacker. The quantity and frequency of these decoys must naturally take into account the energy consumption constraint, and a trade-off must be found on the message/decoy rate to face such attacks while preserving the WSN lifetime. However, such an approach supposes that the attacker is unable to make the distinction between decoys and meaningful communications. Such a supposition leads to the following definition.

Definition 34. *A Detection Resistance DR-adversary is a couple* $(\mathsf{A}_1, \mathsf{A}_2)$ *of nonuniform algorithms, each with access to an oracle* $\mathcal{O}$*.*

Definition 35 (Message Detection Resistance)**.** *For a public communication system* $(\mathcal{I}, \mathcal{E}, inv)$ *on* $(\mathcal{S}, \mathcal{M}, \{0,1\}^\ell)$*, define the advantage of a DR-adversary* A *by*

$$Adv_{\mathsf{A}}^{DR-\mathcal{O}} = Pr\left[\begin{array}{c} M_0, M_1 \leftarrow \mathcal{S} \\ k \xleftarrow{\$} \{0,1\}^\ell \\ m \leftarrow \mathsf{A}_1(k) \\ b \leftarrow \{0,1\} \\ \alpha = \{M_b, \mathcal{I}(M_{\bar{b}}, m, k))\} \end{array} : \; \mathsf{A}_2(m,k,\alpha) = M_b\right]$$

where the set defining α is a non-ordered one.

We define the insecurity of $S = (\mathcal{I}, \mathcal{E}, inv)$ with respect to the Message Detection Resistance as

$$InSec_S^{DR-\mathcal{O}}(t) = \max_{\mathsf{A}} \left\{ Adv_{\mathsf{A}}^{DR-\mathcal{O}} \right\}$$

where the maximum is taken over all adversaries A with total running time t. Similar kinds of oracles than previously can be defined in that context.

7.3/ SECURE SCHEDULING

7.3.1/ MOTIVATIONS

As stated in previous chapter, a common way to enlarge lifetime of a wireless sensor network is to consider that not all of the nodes have to be awakened: a subset of well-chosen nodes participates temporarily to the task devoted to the network [PMS11, MP09] (video surveillance of an area of interest, sensing environmental values, ...), whereas the other nodes sleep in order to preserve their batteries. Obviously, the scheduling process determining the nodes that have to be awakened at each time step must be defined accurately, both for guaranteeing a certain level of quality in the assigned task and to preserve the network capability over time. Problems that are of importance in that approach are often related to coverage, ratio of working vs sleeping nodes, efficient transmission of wake up orders, and capability for the subset of network nodes to satisfy, with a sufficient quality, the objectives it has been designed for.

In case of hostile environments, security plays an important role in the fulfillment of the scheduling program. Indeed an attacker, observing the manner nodes are waken up, should not be able to determine the scheduling process. For instance, in a video surveillance context, if the attacker is able to determine at some time the list of the sleeping nodes, then he can possibly achieve an intrusion without being detected (see [BGMP11] or previous chapter).

Obviously, a random scheduling can solve the issues raised above, by guaranteeing a uniform coverage while preventing attackers to predict the list of awaken nodes. However, this approach needs random generators on each node, which cannot be obtained by deterministic algorithms embedded into the network. Even if truly random generators (TRNG) can be approximated by physical devices, they need a certain quantity of resources, suppose that the environment under observation has a sufficient variability of a given set of physical properties (to produce the physical noise source required in that TRNG), and are less flexible or adaptable on demand than pseudorandom number generators (PRNGs). Furthermore, as recalled in Chapter 4, neither their randomness nor their security can be mathematically proven: these generators can be biased or wrongly designed.

Being able to guarantee a certain level of security in scheduling leads to the notion of *secure scheduling* proposed in [BGM] and presented below.

7.3.2/ SECURE SCHEDULING IN WIRELESS SENSOR NETWORKS

Two kinds of scheduling processes can be defined: each node can embed its own program, determining when it has to sleep (local approach), or the sink or some specific nodes can be responsible of the scheduling process, sending sleep or wake up orders to the nodes that have to change their states (global approach).

We consider that a deterministic scheduling algorithm is a function $S : \{0,1\}^n \rightarrow \{0,1\}^M$, where $M > n$. This definition can be understood as follows:

- The value inputted in S is the secret key launching the scheduling process. It can be shown as the seed of a PRNG.
- In case of a local approach, the binary sequence produced by this function corresponds to the moments where the node must be awaken: if the k-th term of this sequence is 0, then the node can go to sleep mode between t_k and t_{k+1}.
- In case of a global approach, the binary sequence returned by S can be divided into blocs, such that each bloc contains the *id* of the node to which an order of state change will be send.

Loosely speaking, S is called a secure scheduling if it maps uniformly distributed input (the secret key or seed of the scheduling process) into an output which is computationally indistinguishable from uniform. The precise definition is given below.

Definition [36]. *A T-time algorithm $\mathcal{D} : \{0,1\}^M \longrightarrow 0,1$ is said to be a (T,ε)-distinguisher for S if*

$$\left|Pr[\mathcal{D}(S(\mathfrak{U}_2^n)) = 1] - Pr[\mathcal{D}(\mathfrak{U}_2^M) = 1]\right| \geqslant \varepsilon.$$

where $\mathfrak{U}_2$ is the uniform distribution on $\{0,1\}$.

Definition [37] (Secure scheduling)**.** *Algorithm S is called a (T,ε)-secure scheduling if no (T,ε)-distinguisher exists for S.*

Adapting the proofs of [Yao82, GGM86], it is possible to show that a $(T,\varepsilon)-$distinguisher exists if and only if a T-time algorithm can, knowing the first l bits of a scheduling s, predict the $(l+1)-$st bit of s with probability significantly greater than 0.5. This comes from the fact that a PRNG passes the next-bit test if and only if it passes all polynomial-time statistical tests [Yao82, GGM86].

An important question is what level of security (T,ε) suffices for realistic applications in scheduled wireless sensor networks. Unfortunately, the level of security is often chosen arbitrarily. It is reasonable to require that a scheduling process is secure for all pairs (T,ε) such that the time-success ratio T/ε is bounded. In the next section we present an illustration of this notion.

7.3.3/ PRACTICAL STUDY

Suppose that a wireless sensor node has been scheduled by a Blum-Blum-Shub BBS pseudorandom generator. This generator produces bits $y_0, y_1, \ldots$, and the node is awaken during the time interval $[t_i; t_{i+1}[$ if and only if $y_i = 1$.

Let us recall that the Blum Blum Shum generator [BG85] (usually denoted by BBS) is defined by the following process:

1. Generate two large secret random and distinct primes p and q, each congruent to 3 modulo 4, and compute $N = pq$.
2. Select a random and secret seed $s \in [\![1, N-1]\!]$ such that $gcd(s, N) = 1$, and compute $x_0 = s^2 (\mathrm{mod}\ N)$.
3. For $1 \leqslant i \leqslant l$ do the following:
   (a) $x_i = x_{i-1}^2 (\mathrm{mod}\ N)$.
   (b) $y_i =$ the least significant bit of x_i.
4. The output sequence is $y_1, y_2, \ldots, y_l$.

Suppose now that the network will work during $M = 100$ time units, and that during this period, an attacker can realize 10^{12} clock cycles. We thus wonder whether, during the network's lifetime, the attacker can distinguish this sequence from truly random one, with a probability greater than $\varepsilon = 0.2$. We consider that N has 900 bits.

The scheduling process is the BBS generator, which is cryptographically secure. More precisely, it is $(T, \varepsilon)-$secure: no $(T, \varepsilon)-$distinguishing attack can be successfully realized on this PRNG, if [FS97]

$$T \leqslant \frac{L(N)}{6N(log_2(N))\varepsilon^{-2}M^2} - 2^7 N\varepsilon^{-2}M^2 log_2(8N\varepsilon^{-1}M)$$

where M is the length of the output ($M = 100$ in our example), and

$$L(N) = 2.8 \times 10^{-3} exp\left(1.9229 \times (N\ ln(2)^{\frac{1}{3}}) \times ln(N\ ln2)^{\frac{2}{3}}\right)$$

is the number of clock cycles to factor a $N-$bit integer.

A direct numerical application shows that this attacker cannot achieve its $(10^{12}, 0.2)$ distinguishing attack in that context.

7.4/ SECURE ROUTING

For easy understanding, let us consider a wireless sensor network in which each node is positioned on a square lattice $\mathcal{P} = \mathbb{N}^2$, as depicted in Fig. 7.2. The sink can be considered as the origin of axes, leading to a system of integer coordinates for each node. When two sensors are not aligned, there are at least two different paths of minimum length between these two points. For instance, sensor in position $(3, -2)$ can send messages to node $(2, 1)$ by choosing one of the 4 following routes of same length:

- $(3, -2) \rightarrow (2, -2) \rightarrow (2, -1) \rightarrow (2, 0) \rightarrow (2, 1)$,
- $(3, -2) \rightarrow (3, -1) \rightarrow (2, -1) \rightarrow (2, 0) \rightarrow (2, 1)$,
- $(3, -2) \rightarrow (3, -1) \rightarrow (3, 0) \rightarrow (2, 0) \rightarrow (2, 1)$,

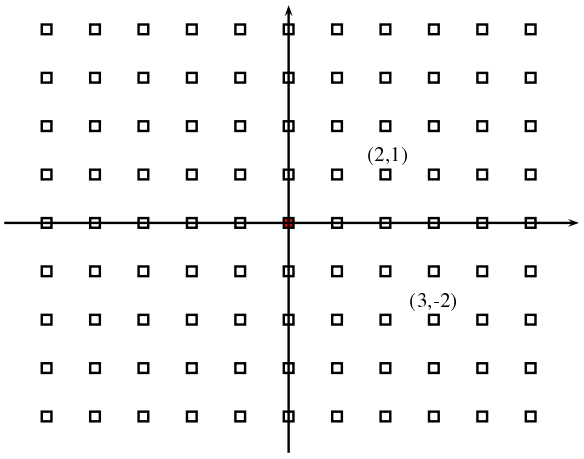


Figure 7.2: Square lattice network

- $(3,-2) \rightarrow (3,-1) \rightarrow (3,0) \rightarrow (3,1) \rightarrow (2,1)$.

The routing of packages through the network will be claimed as secure if the choice of the route always appears random for each external observer having polynomial time capabilities of traffic analysis. This objective can be formalized as follows.

Definition 38. *We define a distance d between two points of the square lattice $\mathcal{P}$ as $d((x_1,y_1),(x_2,y_2)) = |x_2 - x_1| + |y_2 - y_1|$.*

For instance, $d((3,-2),(2,1)) = 1+3 = 4$. We can now define the set of routes between two sensors of the network as the set of shortest paths on $\mathcal{P}$ that join these two points.

Definition 39. *The set of routes between P and Q in $\mathcal{P}$ is defined by:*

$$\mathcal{R}(P,Q) = \left\{(s_0,\ldots,s_{d(P,Q)}) \in \mathcal{P}^{d(P,Q)+1} \mid s_0 = P, s_{d(P,Q)} = Q, \text{ and } \forall i \in [\![0, d(P,Q)-1]\!], d(s_i,s_{i+1}) = 1\right\}.$$

We can now define a routing algorithm as follows [BGM]:

Definition 40. *Let $\mathcal{S}$ be the set of all finite sequences of $\mathcal{P}$ and $\mathcal{K} = \{0,1\}^M$ be the set of keys ($M \in \mathbb{N}^*$ is the security parameter). A deterministic routing algorithm on $\mathcal{P}$ is a function $f : \mathcal{P}^2 \times \mathcal{K} \longrightarrow \mathcal{S}$ such that $\forall (P,Q) \in \mathcal{P}^2$, $\forall k \in \mathcal{K}$, $f((P,Q),k) \in \mathcal{R}(P,Q)$.*

Definition 41. *Let f be a routing algorithm. A T-time algorithm $\mathcal{D} : \mathcal{S} \longrightarrow \{0,1\}$ is said to be a (T,ε)-distinguisher for f if*

$$\exists (P,Q) \in \mathcal{P}^2, |Pr[\mathcal{D}(f((P,Q),\mathfrak{U}(\mathcal{K}))) = 1] - Pr[\mathcal{D}(\mathfrak{U}(\mathcal{R}(P,Q))) = 1]| \geqslant \varepsilon.$$

where $\mathfrak{U}(X)$ is the uniform distribution on the set X.

Definition 42 (Secure routing)**.** *The routing algorithm f is called (T,ε)-secure routing if no (T,ε)-distinguisher exists for f.*

7.5/ Cryptographically Secure Data Aggregation

To finalize the definition of a cryptographically secure wireless sensor network, we need to introduce rigorously the notion of secure data aggregation in such networks.

Definition [43] (Aggregator)**.** *Let* $S = (\mathcal{I}, \mathcal{E}, inv)$ *a public communication system on* $(\mathcal{S}, \mathcal{M}, \{0,1\}^{\ell})$*, and* $c : \{0,1\}^n \longrightarrow \{0,1\}^m$*,* $m < n$*, a compression function.*

A $c-$*aggregator function on* S *is a couple of algorithms* $Agg : \mathcal{S}^p \longrightarrow \mathcal{S}$*,* $Inv : \mathcal{K}^p \longrightarrow \mathcal{K}$*, with* $p > 1$*, such that* Agg *is a probabilistic algorithm,* Agg *and* Inv *can be computed polynomially, and* $\forall s_1, \ldots, s_p \in \mathcal{S}$*,* $\forall m_1, \ldots, m_p \in \mathcal{M}$*,* $\forall k_1, \ldots, k_p \in \mathcal{K}$*,* $Agg(\mathcal{I}(s_1, m_1, k_1), ..., \mathcal{I}(s_p, m_p, k_p)) = s'$ *satisfies* $\mathcal{E}(s', Inv(k_1, ..., k_p)) = c(m_1 ... m_p)$*.*

The idea is that, as for hash functions, the security of the aggregator is based on the security of the compression function. We have recently published examples of such secured aggregation in [BGM14, BMG10, BGM10a, BGM10b]. However, as these aggregation methods are based on investigations in our thesis, we will not detail them in this manuscript.

Our last application in the field of WSNs security is about epidemiological approaches for data survivability in unattended wireless sensor networks. We do not detail it in this part of our manuscript, as this work has not yet been accepted. See Appendix D for further information.

IV

Applications in Bioinformatics

8

The Complex Dynamics of Protein Folding

We recall in this chapter our investigations in the field of protein folding, either already published (in [BCG11a, BCGS12a, GCBB]) or currently submitted (in [BGNP13, BGG13, BGMP13]).

8.1/ Protein Folding in the 2D Hydrophobic-Hydrophilic (HP) Square Lattice Model is Chaotic

8.1.1/ Introduction

Proteins, polymers formed by different kinds of amino acids, fold to form a specific tridimensional shape. This geometric pattern defines the majority of functionality within an organism, *i.e.*, the macroscopic properties, function, and behavior of a given protein. For instance, the hemoglobin is able to carry oxygen to the blood stream due to its 3D geometric pattern. However, contrary to the mapping from DNA to the amino acids sequence, the complex folding of this last sequence still remains not well-understood. Moreover, the determination of 3D protein structure from the amino acid linear sequence, that is to say, the exact computational search for the optimal conformation of a molecule, is completely unfeasible. It is due to the astronomically large number of possible 3D protein structures for a corresponding primary sequence of amino acids [HCS09]: the computation capability required even for handling a moderately-sized folding transition exceeds drastically the computational capacity currently accessible. Additionally, the forces involved in the stability of the protein conformation are currently not modeled with enough accuracy [HCS09], and we can even wonder if a fully accurate model is possible to find one day.

Then it is impossible to compute exactly the 3D structures of the proteins. Indeed, the Protein Structure Prediction (PSP) problem is a NP-complete one [CGP+98]. This is why the 3D conformations of proteins are *predicted*: the most stable energy-free states are looked for by using computational intelligence tools like genetic algorithms [HSHS10], ant colonies [SHeb], particle swarm [PHRVGJ10], memetic algorithms [IC09], or neural networks [DMHK95]. This search is justified by the Afinsen's "Thermodynamic Hypothesis", claiming that a protein's native structure is at its lowest free energy minimum [Anf73]. The use of computational intelligence tools coupled with proteins energy approximation models (like AMBER, DISCOVER, or ECEPP/3), come from the fact that finding the exact

minimum energy of a 3D structure of a protein is a very time consuming task. Furthermore, in order to tackle with the complexity of the PSP problem, authors that try to predict the protein folding process use models of various resolutions. In low resolution models, atoms into the same amino acid can for instance be considered as a same entity. These low resolution models are used as the first stage of the 3D structure prediction: the backbone of the 3D conformation is determined. Then, high resolution models come next for further exploration. Such a prediction strategy is commonly used in PSP softwares like ROSETTA [BB01, CKM$^+$05] or TASSER [ZAS05].

In [BCG11a] and its extension [BCGS12a], we have mathematically demonstrated that a particular dynamical system, used in low resolutions models to predict the backbone of the protein, is chaotic according to the Devaney's formulation. Chaos in protein folding has been already investigated in the past years. For instance, in [Bö91], the Lyapunov exponent of a folding process has been experimentally computed, to show that protein folding is highly complex. More precisely, they have established that the crambin protein folding process, which is a small plant seed protein constituted by 46 amino acids from *Crambe Abyssinica*, has a positive Lyapunov exponent. In [ZW96], an analysis of molecular dynamics simulation of a model α-helix indicates that the motion of the helix system is chaotic, *i.e.*, has nonzero Lyapunov exponents, broad-band power spectra, and strange attractors. Finally, in [BUAM97], authors investigated the response of a protein fragment in an explicit solvent environment to very small perturbations of the atomic positions, showing that very small changes in initial conditions are amplified exponentially and lead to vastly different, inherently unpredictable behavior. These research works study experimentally the dynamics of protein folding and state that this process exhibit some chaotic properties, where "chaos" refers to various physical understandings of the phenomenon. They note the complexity of the process in concrete cases, without offering a study framework making it possible to understand the origins of such a behavior.

The approach presented in [BCG11a, BCGS12a] is different for the two following reasons. First, we have focused on mathematical aspects of chaos. Second, we do not have studied the biological folding process, but the protein folding one as it is described in the 2D hydrophobic-hydrophilic (HP) lattice model [BL98]. In other words, we have mathematically studied the folding dynamics used in this model, and we wondered if this model is stable through small perturbations. For instance, what are the effects in the 2D model of changing a residue from hydrophobic to hydrophilic ? Or what happens if we do not realize exactly the good rotation on the good residue, at one given stage of the 2D folding process, due to small errors in the knowledge of the protein ? Let us recall that the 2D HP square lattice model is a popular model with low resolution that focuses only upon hydrophobicity by separating the amino acids into two sets: hydrophobic (H) and hydrophilic (or polar P) [Dil85]. This model has been used several times for protein folding prediction [IC10, UM93, BUAM97, HSHS10, HC10]. In [BCG11a] and its extension [BCGS12a], we have shown that *the folding process is unpredictable (chaotic) in the 2D HP square lattice model used for prediction*, and we have investigated the consequences of this fact. Chaos here refers to our inability to make relevant prediction with this model, which does not *necessary* imply that the biological folding dynamics is chaotic too. In particular, we do not claim that these biological systems must try a large number of conformations in order to find the best one. Indeed, the prediction model is proven to be chaotic, but this fact is not clearly related to the impact of environmental factors on true biological protein folding.

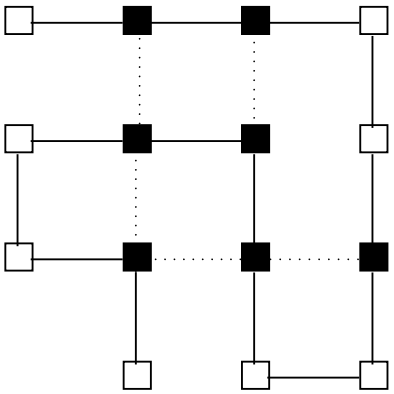


Figure 8.1: Hydrophilic-hydrophobic model (black squares are hydrophobic residues)

8.1.2/ 2D HYDROPHILIC-HYDROPHOBIC (HP) MODEL

8.1.2.1/ HP MODEL

In the HP model, hydrophobic interactions are supposed to dominate protein folding. This model was formerly introduced by Dill, who consider in [Dil85] that the protein core freeing up energy is formed by hydrophobic amino acids, whereas hydrophilic amino acids tend to move in the outer surface due to their affinity with the solvent (see Fig. 8.1).

As recalled in [BCG11a], in this model, a protein conformation is a "self-avoiding walk (SAW)" on a 2D or 3D lattice such that its energy E, depending on topological neighboring contacts between hydrophobic amino acids that are not contiguous in the primary structure, is minimal. In other words, for an amino-acid sequence P of length N and for the set $\mathcal{C}(P)$ of all SAW conformations of P, the chosen conformation will be $C^* = min\left\{E(C)/C \in \mathcal{C}(P)\right\}$ [SH05]. In that context and for a conformation C, $E(C) = -q$ where q is equal to the number of topological hydrophobic neighbors. For example, $E(c) = -5$ in Fig. 8.1.

8.1.2.2/ PROTEIN ENCODING

Additionally to the direct coordinate presentation, at least two other isomorphic encoding strategies for HP models are possible: relative encoding and absolute encoding. In relative encoding [HCS09], the move direction is defined relative to the direction of the previous move. Alternatively, in absolute encoding [BWC99], which is the encoding chosen in [BCG11a, BCGS12a], the direct coordinate presentation is replaced by letters or numbers representing directions with respect to the lattice structure.

For absolute encoding in the 2D square lattice, the permitted moves are: forward $\rightarrow$ (denoted by 0), down $\downarrow$ (1), backward $\leftarrow$ (2), and up $\uparrow$ (3). A 2D conformation C of N $+ 1$ residues for a protein P is then an element C of $\mathbb{Z}/4\mathbb{Z}^{\mathsf{N}}$, with a first component equal to 0 (forward) [HCS09]. For instance, in Fig. 8.1, the 2D absolute encoding is 00011123322101 (starting from the upper left corner). In that situation, at most 4^{N} conformations are possible when considering N $+ 1$ residues, even if some of them are invalid due to the SAW requirement.

8.1.3/ A DYNAMICAL SYSTEM FOR THE 2D HP SQUARE LATTICE MODEL

The objective of [BCG11a] was to state that the protein folding process, as it is described in the 2D model, has a chaotic behavior. To do so, this process has been firstly described as a dynamical system, as recalled below.

8.1.3.1/ INITIAL PREMISES

Let us firstly recall some preliminaries introduced in [BCG11a]. The primary structure of a given protein P with N $+ 1$ residues is coded by $00 \dots 0$ (N times) in absolute encoding. Its final 2D conformation has an absolute encoding equal to $0C_1^* \dots C_{N-1}^*$, where $\forall i, C_i^* \in \mathbb{Z}/4\mathbb{Z}$, is such that $E(C^*) = min\left\{E(C)/C \in \mathcal{C}(P)\right\}$. This final conformation depends on the distribution of hydrophilic and hydrophobic amino acids in the initial sequence.

Moreover, we suppose that, if the residue number $n + 1$ is forward the residue number n in absolute encoding (→) and if a fold occurs after n, then the forward move can only by changed into up (↑) or down (↓). That means, in the simplistic model of [BCG11a], only rotations of $+\frac{\pi}{2}$ or $-\frac{\pi}{2}$ are possible. Consequently, for a given residue that is supposed to be updated, only one of the two possibilities below can appear for its absolute move during a fold:

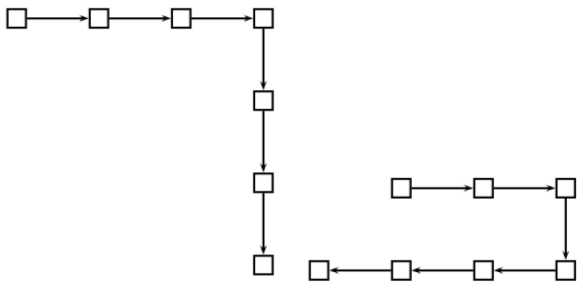
- $0 \longmapsto 1, 1 \longmapsto 2, 2 \longmapsto 3$, or $3 \longmapsto 0$ for a fold in the clockwise direction, or
- $1 \longmapsto 0, 2 \longmapsto 1, 3 \longmapsto 2$, or $0 \longmapsto 3$ for an anticlockwise.

This fact has led us to the following definition [BCG11a]:

Definition [44]. *The* clockwise fold function *is the function* $f : \mathbb{Z}/4\mathbb{Z} \longrightarrow \mathbb{Z}/4\mathbb{Z}$ *defined by* $f(x) = x + 1(mod\, 4)$.

Obviously the anticlockwise fold function is $f^{-1}(x) = x - 1(\mathrm{mod}\ 4)$. Thus at the n^{th} folding time, a residue k is chosen and its absolute move is changed by using either f or f^{-1}. As a consequence, all of the absolute moves must be updated from the coordinate k until the last one N by using the same folding function.

Example [6]. *If the current conformation is* $C = 000111$*, like in Figure 8.2(a), and if the*



(a) 000111 (b) 001222

Figure 8.2: Encoding folding operation

third residue is chosen to fold by a rotation of $-\frac{\pi}{2}$ *(mapping* f*), the new conformation will*

be:

$$(C_1, C_2, f(C_3), f(C_4), f(C_5), f(C_6)) = (0, 0, 1, 2, 2, 2).$$

That is, the one depicted in Figure 8.2(b).

These considerations have led us to a formalization of protein folding 2D model recalled thereafter.

8.1.3.2/ FORMALIZATION AND NOTATIONS

Let $\mathsf{N}+1$ be a fixed number of amino acids, where $\mathsf{N} \in \mathbb{N}^*$. We define

$$\check{\mathcal{X}} = \mathbb{Z}/4\mathbb{Z}^{\mathsf{N}} \times [\![-\mathsf{N}; \mathsf{N}]\!]^{\mathbb{N}}$$

as the phase space of all possible folding processes. An element $X = (C, F)$ of this dynamical folding space is constituted by:

- A conformation of the $\mathsf{N}+1$ residues in absolute encoding: $C = (C_1, \ldots, C_{\mathsf{N}}) \in \mathbb{Z}/4\mathbb{Z}^{\mathsf{N}}$. Note that we do not require self-avoiding walks here.
- A sequence $F \in [\![-\mathsf{N}; \mathsf{N}]\!]^{\mathbb{N}}$ of future folds such that, when $F_i \in [\![-\mathsf{N}; \mathsf{N}]\!]$ is k, it means that it occurs:
  - a fold after the $k-$th residue by a rotation of $-\frac{\pi}{2}$ (mapping f) at the $i-$th step, if $k = F_i > 0$,
  - no fold at time i if $k = 0$,
  - a fold after the $|k|-$th residue by a rotation of $\frac{\pi}{2}$ (*i.e.*, f^{-1}) at the $i-$th time, if $k < 0$.

On this phase space, the protein folding dynamic in the 2D model can be formalized as follows [BCG11a].

Denote by i the map that transforms a folding sequence in its first term (*i.e.*, in the first folding operation):

$$\begin{array}{rccc} i: & [\![-\mathsf{N}; \mathsf{N}]\!]^{\mathbb{N}} & \longrightarrow & [\![-\mathsf{N}; \mathsf{N}]\!] \\ & F & \longmapsto & F^0, \end{array}$$

by σ the shift function over $[\![-\mathsf{N}; \mathsf{N}]\!]^{\mathbb{N}}$, that is to say,

$$\begin{array}{rccc} \sigma: & [\![-\mathsf{N}; \mathsf{N}]\!]^{\mathbb{N}} & \longrightarrow & [\![-\mathsf{N}; \mathsf{N}]\!]^{\mathbb{N}} \\ & \left(F^k\right)_{k\in\mathbb{N}} & \longmapsto & \left(F^{k+1}\right)_{k\in\mathbb{N}}, \end{array}$$

and by $sign$ the function:

$$sign(x) = \begin{cases} 1 & \text{if } x > 0, \\ 0 & \text{if } x = 0, \\ -1 & \text{else.} \end{cases}$$

Remark that the shift function removes the first folding operation from the folding sequence F once it has been achieved, and that this modeling is quite close to the one presented in the information hiding chapter. Consider now the map $G : \check{\mathcal{X}} \to \check{\mathcal{X}}$ defined by:

$$G\left((C, F)\right) = \left(f_{i(F)}(C), \sigma(F)\right),$$

where $\forall k \in [\![-\mathsf{N};\mathsf{N}]\!]$, $f_k : \mathbb{Z}/4\mathbb{Z}^{\mathsf{N}} \to \mathbb{Z}/4\mathbb{Z}^{\mathsf{N}}$ is defined by:

$$f_k(C_1,\ldots,C_{\mathsf{N}}) = (C_1,\ldots,C_{|k|-1}, f^{sign(k)}(C_{|k|}),\ldots,f^{sign(k)}(C_{\mathsf{N}})).$$

Thus the folding process of a protein P in the 2D HP square lattice model, with initial conformation equal to $(0,0,\ldots,0)$ in absolute encoding and a folding sequence equal to $(F^i)_{i\in\mathbb{N}}$, is defined by the following dynamical system over $\check{\mathcal{X}}$ [BCG11a]:

$$\begin{cases} X^0 = ((0,0,\ldots,0),F) \\ X^{n+1} = G(X^n), \forall n \in \mathbb{N}. \end{cases}$$

In other words, at each step n, if $X^n = (C,F)$, we take the first folding operation to realize, that is $i(F) = F^0 \in [\![-\mathsf{N};\mathsf{N}]\!]$, we update the current conformation C by rotating all of the residues coming after the $|i(F)|-$th one, which means that we replace the conformation C with $f_{i(F)}(C)$. Lastly, we remove this rotation (the first term F^0) from the folding sequence F: F becomes $\sigma(F)$.

Example [7]. *Let us reconsider Example 6. The unique iteration of this folding process transforms a point of $\check{X}$ having the form $\big((0,0,0,1,1,1);(3,F^1,F^2,\ldots)\big)$ in $G\left((0,0,0,1,1,1),(+3,F^1,F^2,\ldots)\right)$, which is equal to $\big((0,0,1,2,2,2),(F^1,F^2,\ldots)\big)$.*

Rem [3]. *Such a formalization allows the study of proteins that never stop to fold, for instance due to never-ending interactions with the environment.*

Rem [4]. *A protein P that has finished to fold, if such a protein exists, has the form $(C,(0,0,0,\ldots))$, where C is the final 2D structure of P. In this case, we can assimilate a folding sequence that is convergent to 0,* i.e.*, of the form $(F^0,\ldots,F^n,0\ldots)$, with the finite sequence $(F^0,\ldots,F^n)$.*

We then have introduced in [BCGS12a] the SAW requirement inside the formulation of the folding process in the 2D model [BCG11a].

8.1.3.3/ THE SAW REQUIREMENT

Let $\mathcal{P}$ denotes the 2D plane and

$$\begin{array}{rccc} p: & \mathbb{Z}/4\mathbb{Z}^{\mathsf{N}} & \to & \mathcal{P}^{\mathsf{N}+1} \\ & (C_1,\ldots,C_{\mathsf{N}}) & \mapsto & (X_0,\ldots,X_{\mathsf{N}}) \end{array}$$

where $X_0 = (0,0)$ and

$$X_{i+1} = \begin{cases} X_i + (1,0) & \text{if } C_i = 0, \\ X_i + (0,-1) & \text{if } C_i = 1, \\ X_i + (-1,0) & \text{if } C_i = 2, \\ X_i + (0,1) & \text{if } C_i = 3. \end{cases}$$

The map p transforms an absolute encoding in its 2D representation. For instance, $p((0,0,0,1,1,1))$ is $((0,0);(1,0);(2,0);(3,0);(3,-1);(3,-2);(3,-3))$, that is, the first figure of Example 12. Now, for each $(P_0,\ldots,P_{\mathsf{N}})$ of $\mathcal{P}^{\mathsf{N}+1}$, we denoted by

$$support((P_0,\ldots,P_{\mathsf{N}}))$$

the set (with no repetition): $\{P_0, \ldots, P_{\mathsf{N}}\}$. For instance,

$$support\left((0,0);(0,1);(0,0);(0,1)\right) = \{(0,0);(0,1)\}.$$

Then [BCGS12a],

Definition 45. *A conformation* $(C_1, \ldots, C_{\mathsf{N}}) \in \mathbb{Z}/4\mathbb{Z}^{\mathsf{N}}$ *satisfies the* self-avoiding walk (SAW) requirement *iff the cardinality of* $support(p((C_1, \ldots, C_{\mathsf{N}})))$ *is* $\mathsf{N}+1$.

We can remark that Definition 45 concerns only one conformation, and not a *sequence* of conformations that occurs in a folding process. This definition is compliant with the self-avoiding walks of the discrete mathematics community. However, we have discovered in [BCGS12a], and further investigated in [GCBB] that the self-avoiding walk property in protein folding can be interpreted in various non-equivalent ways, which will be debated in a next section.

8.1.3.4/ A METRIC FOR THE FOLDING PROCESS

We have defined in [BCG11a] a metric d over $\mathcal{X} = \mathfrak{S}_{\mathsf{N}} \times [\![-\mathsf{N};\mathsf{N}]\!]^{\mathbb{N}}$ by:

$$d(X, \check{X}) = d_C(C, \check{C}) + d_F(F, \check{F}).$$

where

$$\begin{cases} \delta(a,b) = 0 \text{ if } a = b, \text{ and } \delta(a,b) = 1 \text{ otherwise,} \\ d_C(C, \check{C}) = \sum_{k=1}^{\mathsf{N}} \delta(C_k, \check{C}_k) 2^{\mathsf{N}-k}, \\ d_F(F, \check{F}) = \frac{9}{2\mathsf{N}} \sum_{k=0}^{\infty} \frac{|F^k - \check{F}^k|}{10^{k+1}}. \end{cases}$$

This distance for the dynamical description of the protein folding process in the 2D HP square lattice model can be justified as follows. The integral part of the distance between two points $X = (C, F)$ and $\check{X} = (\check{C}, \check{F})$ of $\mathcal{X}$ measures the differences between the current 2D conformations of X and $\check{X}$. More precisely, if $d_C(C, \check{C})$ is in $[\![2^k; 2^{k+1}]\!]$, then the first k terms in the acceptable conformations C and $\check{C}$ (the absolute encoding) are equal, whereas the $k+1^{th}$ terms differ: their 2D conformations will differ after the $k+1-$th residue. If the decimal part of $d(X, \check{X})$ is between 10^{-k} and $10^{-(k+1)}$, then the next k foldings of C and $\check{C}$ will occur in the same place (residue), same order, and same angle. The decimal part of $d(X, \check{X})$ will then decrease when the duration the folding process will be similar increase. More precisely, $F^k = \check{F}^k$ (same residue and same angle of rotation at the $k-$th stage of the 2D folding process) if and only if the $k+1^{th}$ digit of this decimal part is 0. Lastly, $\frac{9}{2\mathsf{N}}$ is just a normalization factor.

For instance, if we know where are now the $\mathsf{N}+1$ residues of our protein P in the lattice (knowledge of the correct conformation), and if we have discovered what will be its k next foldings, then we know that the point $X = (C, F)$ describing the folding process of the considered protein in the 2D model, will be "somewhere" into the ball $\mathcal{B}(C, 10^{-k})$, that is, very close to the point (C, F) if k is large [BCGS12a].

Example 8. *Let us consider two points*

- $X = ((0,0,0,1,1,1);(3,-4,2))$,

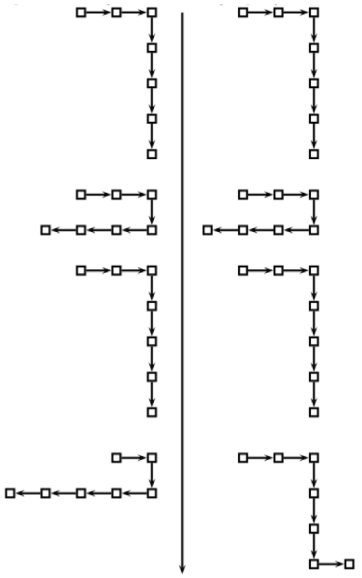


Figure 8.3: Representation of the two "points" $X = ((0,0,0,1,1,1);(3,-4,2))$ and $X' = ((0,0,0,1,1,1);(3,-4,-6))$ of the phase space $\mathcal{X}$ (X is in left part of the figure, X' is its right part).

- *and* $X' = ((0,0,0,1,1,1);(3,-4,-6))$

of $\mathcal{X}$. *We note* $X = (C,F)$ *and* $X' = (C',F)$. $d_C(C,C') = 0$, *then these two points have the same current (first) conformation. As* $d_F(F,F') = \frac{9}{2\times 6}\frac{|2-(-6)|}{10^3} = 0.006$ *is in* $\left[10^{-2};10^{-3}\right[$, *we can deduce that the two next foldings of* X *and of* X' *will lead to identical conformations, whereas the third folding operation will lead to different conformations. A possible way to represent these two points of the phase space is to draw the successive conformations induced by these points, as illustrated in Figure 8.3.*

Example [9]. *Figure 8.4 contains the representation of the two "points"* $X = ((0,0,0,1,1,1);(3,-4,2))$ *and* $X' = ((0,0,1,2,2,2);(-4,-5))$. *Let* $(C,F) = X$ *and* $(C',F') = X'$. *We have* $d_C(C,C') = 2^{6-3} + 2^{6-4} = 12$ *and* $d_F = \frac{9}{12}\left(\frac{|-4-3|}{10} + \frac{|-5+4|}{100} + \frac{2}{1000}\right) = 0.534$, *then* $d(X,X') = 12.534$. *As 12 is in* $\left[2^3;2^4\right[$, *we can conclude that the absolute encoding of the two initial conformations start to differ in the third residue.*

8.1.4/ FOLDING PROCESS IN 2D MODEL IS CHAOTIC

8.1.4.1/ MOTIVATIONS

In our topological description of the protein folding process in the 2D model, all the information is embedded into the folding sequence F. Indeed, roughly speaking, it is as if Nature has a function $\mathcal{N}$ that translates a protein P having a linear conformation $(0,...,0)$ into an environment E, in a folding sequence F, *i.e.*, $F = \mathcal{N}(P,E)$ [BCGS12a]. Having this "natural" folding sequence F, we are able to obtain its true conformation in the 2D model, by computing $G^n((0,\ldots,0);F)$, where n is the size of F. On our side, we have only a

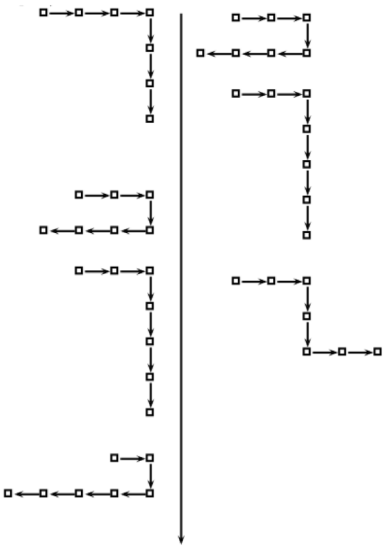


Figure 8.4: Representation of the two "points" $X = ((0,0,0,1,1,1);(3,-4,2))$ and $X' = ((0,0,1,2,2,2);(-4,-5))$ of the phase space $\mathcal{X}$ (X is in left part of the figure, X' is its right part).

partial knowledge of the environment E and of the protein P (exact interactions between atoms). We thus consider $\check{E}$ and $\check{P}$, as close as we can from E and P respectively. Moreover, we have only a model $\check{\mathcal{N}}$ of $\mathcal{N}$ as, for instance, we use various approximations: models for free energy, approximations of hydrophobic/hydrophilic areas and electro-polarity, etc. This is why we can only deduce an approximation $\check{F} = \check{\mathcal{N}}(\check{P}, \check{E})$ of the natural folding sequence $F = \mathcal{N}(P, E)$. One important motivation of [BCG11a] is to determine whether, having an approximation $\check{F}$ of F, we obtain a final conformation $\check{C} = G^{\check{n}}\left((0,\ldots,0);\check{F}\right)_0$ close to the natural conformation $C = G^n\left((0,\ldots,0);F\right)_0$ or not. In this last sentence, n and $\check{n}$ are the sizes of F and $\check{F}$ respectively, and the terms "approximation" and "close" can be understood by using d_F and d_C respectively. To sum up, even if we cannot have access with an infinite precision to all of the forces that participate to the folding process, *i.e.*, even if we only know an approximation $X'^0 = \left((0,\ldots,0),\check{F}\right)$ of $X^0 = ((0,\ldots,0),F)$, can we claim that the predicted conformation $X'^{n_1} = G^{n_1}\left((0,\ldots,0),\check{F}\right)$ still remains close to the true conformation $X^{n_2} = G^{n_2}\left((0,\ldots,0),F\right)$? Or, on the contrary, do we have a chaotic behavior, a kind of butterfly effect that magnifies any error on the evaluation of the forces in presence ?

Raising such a question has led us to the study of the dynamical behavior of the folding process [BCG11a, BCGS12a].

8.1.4.2/ CHAOS OF THE FOLDING PROCESS

We then have given in [BCG11a] two proofs of the chaotic behavior of the protein folding dynamics in the 2D model. For the first one, we have firstly established that,

Proposition [10]. *G is a continuous map on $(\mathcal{X}, d)$.*

It thus has been possible to study the chaotic behavior of the folding process. We have

successively proven the regularity and strong transitivity in [BCG11a], leading to the result:

Theorem 9. *The folding process G in the 2D model is chaotic according to Devaney.*

Strong transitivity states that being as close as possible of the true folding process (2D model) is not a guarantee of success. Indeed, let P a protein under interest and F its natural folding process in the 2D model. Then, for any possible conformation C of the square lattice, there exists a folding sequence $\check{F}$ very close to F leading to C. More precisely, for any $\varepsilon > 0$ (as small as possible), an infinite number of folding sequences are in $\mathcal{B}_{d_F}(F, \varepsilon)$ and lead to C. The strong transitivity property implies that without the knowledge of the *exact* initial condition (the natural folding process, and thus the exact free energy), all the conformations are possible. Additionally, no conformation of the square lattice can be discarded when studying a protein folding in the 2D HP square lattice model: the dynamical system obtained by such a formalization is intrinsically complicated and cannot be decomposed or simplified. Furthermore, this trend to visit the whole space of acceptable conformations is counteracted by elements of regularity stated before: it is even impossible to dress a kind of qualitative description of the dynamics in the 2D model, as two points close to each other can have fundamental different behaviors [BCGS12a].

A consequence of Theorem 9 is that this process is highly sensitive to its initial condition. If the 2D model can accurately describe the natural process, then this theorem implies that even a minute difference on an intermediate conformation of the protein, in forces that act in the folding process, or in the position of an atom, can lead to enormous differences in its final conformation, even over fairly small timescales. In particular, it seems very difficult to predict, in this 2D model, the structure of a given protein by using the knowledge of the structure of similar proteins. Let us remark that the whole 3D folding process with real torsion angles is obviously more complex than this 2D HP model. And finally, that chaos refers to our incapacity to make good prediction, it does not mean that the biological process is a random one [BCGS12a].

Before studying some practical aspects of this unpredictability in Section 8.1.7, we outline in the next subsection a second proof of the chaotic behavior of this process and we deepen its chaotic properties, as established in [BCG11a].

8.1.5/ Outlines of a second proof

We have proven in [BCG11a] that the folding dynamics can be modeled as chaotic iterations (CIs). Due to this relation between protein folding in 2D HP model and CIs, we have inherited topological properties from chaotic iterations to protein folding, and have compared the ability of neural networks to predict CIs (recalled in Chapter 3) with the capacity of artificial intelligence tools to predict the conformation of a protein using the HP model. Let us finally remark that it is easy to study processes such that more than one fold occur per time unit, by using CIs.

The use of chaotic iterations in order to model protein folding can be summarized as follows [BCG11a]. At each iteration, the same process is applied to the system (*i.e.*, to the conformation), that is the folding operation. Additionally, it is not a necessity that all of the residues fold at each iteration: indeed it is possible that, at a given iteration, only some of these residues folds. Such iterations, where not all the cells of the considered system are to be updated, are exactly the iterations modeled by CIs. Indeed, as stated

in [BCG11a], the protein folding process with folding sequence $(F^n)_{n\in\mathbb{N}}$ consists in the following chaotic iterations: $C^0 = (0, 0, \ldots, 0)$ and,

$$C_{|i|}^{n+1} = \begin{cases} C_{|i|}^n & \text{if } i \notin S^n \\ f^{sign(i)}(C^n)_i & \text{else} \end{cases},$$

where the chaotic strategy is defined by $\forall n \in \mathbb{N}, S^n = [\![-\mathsf{N}; \mathsf{N}]\!] \backslash [\![-F^n; F^n]\!]$. Thus, to prove that the protein folding process is chaotic as defined by Devaney, is equivalent to prove that the graph of iterations of the CIs defined above is strongly connected. This last fact is obvious, as it is always possible to find a folding process that map any conformation $(C_1, \ldots, C_\mathsf{N}) \in \mathfrak{C}_\mathsf{N}$ to any other $(C'_1, \ldots, C'_\mathsf{N}) \in \mathfrak{C}_\mathsf{N}$ (this is a lemma established in [BCG11a]).

We will now investigate some consequences resulting from the chaotic behavior of the 2D folding dynamical system.

8.1.6/ Qualitative and quantitative evaluations

First of all, the transitivity property implies the indecomposability of the system. Thus it is impossible to reduce, in the 2D model, the set of protein foldings in order to simplify its complexity. Furthermore, the folding process has the instability property. This property, which is implied by the sensitive dependence to the initial condition, leads to the fact that in all of the neighborhoods of any x, there are points that are separated from x under iterations of f. We thus can claim that the behavior of the folding process is unstable. We then have proven in [BCGS12a] that,

Proposition [11]. *Folding process in the 2D model has sensitive dependence on initial conditions on $(\mathcal{X}, d)$ and its constant of sensitivity is at least equal to $2^{\mathsf{N}-1}$. Furthermore, this process is an expansive chaotic system on $(\mathcal{X}, d)$. Its constant of expansiveness is at least equal to 1.*

8.1.7/ Consequences

Results established in [BCG11a] only concern the folding process in the 2D HP square lattice model. At this point, it is natural to wonder if such a model, being a reasonable approximation of the true natural process, is chaotic because this natural process is chaotic too. Indeed, claiming that the natural protein folding process is chaotic seems to be contradictory with the fact that only approximately one thousand folds have been discovered this last decade. The number of proteins that have an understood 3D structure increases largely year after year. However the number of new categories of folds seems to be limited by a fixed value approximately equal to one thousand. Indeed, there is no contradiction as a chaotic behavior does not forbid a certain form of order. As stated before, chaos only refers to limitations in prediction. For example, seasons are not forbidden even if weather forecast has a non-intense chaotic behavior. A same regularity appears in brains: even if hazard and chaos play an important role in a microscopic scale, a statistical order appears in the neural network.

That is, a certain order can emerge from a chaotic behavior, even if it is not a rule of thumb. More precisely, we have argued in [BCGS12a] that these thousand folds can be

related to basins of attractions or strange attractors of the dynamical system, objects that are well described by the mathematical theory of chaos. Thus, it should be possible to determine all of the folds that can occur, by refining our model and looking for its basins of attractions with topological tools. However, this assumption still remains to be further investigated.

Finally we have wondered in [BCG11a, BCGS12a] whether artificial intelligence is able to deal with chaotic dynamics as the one found in the 2D folding process. Recalling and adapting our previous work on neural networks (Chapter 3), and helped by our CI model, we deduced that considered neural networks can neither learn nor predict the folding dynamics in the 2D model with a sufficient accuracy, and discussed investigative ways to tackle this problem.

8.2/ Folded Self-Avoiding Walks Applied to Protein Folding

The first version of our proof of the chaotic behavior of the folding process in the 2D model, published in [BCG11a], was erroneous, due to a subtlety resulted from the absence of a clear definition of the "self-avoiding walk requirement" in bioinformatics, which does not always correspond the the self-avoiding walks (SAWs) studied in the enumerative combinatorics community [BBM11, BFGG12]. The correction of our error in [BCGS12a] has led us to investigate the various ways to understand such a requirement and to discover their relationships in [GCBB]. Since then, we have more systematically studied these particular folded SAWs. [BGNP13] contains a general presentation of our investigations, [BGG13] proves our major result in this field, while [BGMP13] presents computational aspects. All these research papers are summarized thereafter.

8.2.1/ Introduction

Self-avoiding walks (SAW) have been studied over decades, both for their interest in mathematics and their applications in physics: standard model of long chain polymers [Flo49], fundamental example in the theory of critical phenomena in equilibrium statistical mechanics [Sla11, dG72], and so on. They are the source of very difficult problems in probabilities and enumerative combinatorics [BBM11, BFGG12], regarding among other things the number of $n-$step SAW, their mean-square displacement, and the so-called scaling limit. We shown previously that the self-avoiding walks naturally appear in bioinformatics, during the prediction of the 3D conformation of a protein of interest. Frequently, the two dimensional backbone of the protein is looked for in a first stage, and then this 2D structure is refined step by step to obtain the final 3D conformation.

Protein Structure Prediction (PSP) software can be separated into two categories. On the one hand, some algorithms construct the proteins' structures on the 2D or 3D square lattice by adding, at each iteration, a new amino acid at the queue of the protein. Most of the time, various positions are possible for this amino acid, and the chosen position is the one that optimizes a given functional (for instance, the number of neighboring hydrophobic amino acids). On the other hand, some algorithms start from the straight line having the size of the considered protein, and they iterate pivot moves on this structure, pivot amino acids and angles being chosen to optimize another time a well-defined functional. *We*

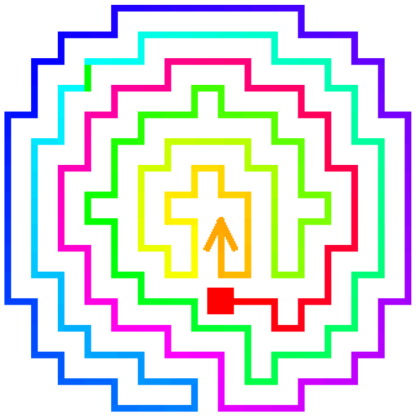


Figure 8.5: The first SAW shown to be not connected to any other SAW by 90° rotations (Madras and Sokal, [MS88]), that is, the first discovered unfoldable SAW.

have pointed out, in our research papers on the dynamics of the protein folding process recalled in the previous section, that these two categories of protein structure prediction software cannot produce the same conformations [GCBB]. More precisely, in the first category, all the conformations can be attained whereas it is not the case in the second one.

Indeed this result, which is ignored by bioinformaticians, has been formerly discovered by the community of mathematicians that studies the self-avoiding walks (SAWs), even though the connection with the PSP problem has not been signaled. In their article introducing the pivot algorithm [MS88], Madras and Sokal have demonstrated a theorem showing that, when starting from the straight line of length n, and iterating the 180° rotation and either both 90° rotations or both diagonal reflections, all the $n-$step self-avoiding walks on $\mathbb{Z}^2$ can be obtained (or, in other words, their pivot algorithm is ergodic for this set of transformations). As a counterexample, they depicted in this article a 223-step SAW in $\mathbb{Z}^2$ that is not connected to any other SAW by 90° rotations (their counterexample is represented in Figure 8.5). This first apparition of an "unfolded" SAW was indeed the unique one in the literature, and the study of (un)folded SAWs has not been deepened before our work in [GCBB]. In this section, we recall our first results and questionings about various sets of self-avoiding walks that can (or cannot) be attained by $\pm 90°$ pivot moves, and we deduce consequences regarding the PSP software.

8.2.2/ A SHORT OVERVIEW OF SELF-AVOIDING WALKS

We firstly recall usual notations and well-known results regarding self-avoiding walks. The objective of this section is not to realize a complete state of the art about established or conjectured results on SAWs, but only to present a few list of properties that are connected to our first investigations regarding the folded self-avoiding walks. For instance, the well-known pattern theorem [MS93] is not presented here. For further results about SAWs, readers can consult for instance [Sla11, MS93].

In the remainder of this chapter, the $n-$th term of a sequence s will denoted by $s(n)$, to be coherent with notations usually used in the enumerative combinatorics field.

Definition [46] (Self-Avoiding Walk)**.** *Let* $d \geqslant 1$*. A* $n-$*step* self-avoiding walk *from* $x \in \mathbb{Z}^d$ *to*

$y \in \mathbb{Z}^d$ *is a map* $w : [\![0, n]\!] \to \mathbb{Z}^d$ *with:*

- $w(0) = x$ *and* $w(n) = y$*,*
- $|w(i+1) - w(i)| = 1$*, where* $|x|$ *stands for the Euclidean norm,*
- $\forall i, j \in [\![0, n]\!]$*,* $i \neq j \Rightarrow w(i) \neq w(j)$ *(self-avoiding property).*

Let $d \in \mathbb{N}^*$. $\mathcal{S}_n(x)$ is the set of $n-$step self-avoiding walks on $\mathbb{Z}^d$ from 0 to x, $c_n(x) = \sharp\mathcal{S}_n(x)$ is the Cardinality of this set, $\mathcal{S}_n = \cup_{x\in\mathbb{Z}^d}\mathcal{S}_n(x)$ is constituted by all $n-$step self-avoiding walks that start from 0, whereas $c_n = \sum_{x\in\mathbb{Z}^d} c_n(x)$ is the number of $n-$step self-avoiding walks on $\mathbb{Z}^d$ starting from 0, that is, $c_n = \sharp\mathcal{S}_n$ [Sla11].

A first result concerning the number of $n-$step self-avoiding walks can be easily obtained by remarking that, when $m-$step SAWs are concatenated to $n-$step SAWs, we found all $(m+n)-$step self-avoiding walks *and* other walks having intersections. In other words,

Proposition [12]. $\forall m, n \in \mathbb{N}^*, c_{m+n} \leqslant c_m c_n$.

The existence of the so-called *connective constant* is a consequence of such a proposition.

Proposition [13]. *The limit* $\lim_{n\to\infty} c_n^{1/n}$ *exists. It is called the* connective constant *and is denoted by* μ*. Moreover, we have* $\mu^n \leqslant c_n$ *and* $d \leqslant \mu \leqslant 2d-1$.

Various bounds or estimates can be found in the literature [Jen04a, Sla11], like $c_n \approx A\mu^n n^{\gamma-1}$ for A and γ to determine (predicted asymptotic behavior) and

$$\mu \in [2.625662, 2.679193].$$

The pivot algorithm is a dynamic Monte Carlo algorithm that produces self-avoiding walks using the following basic approach [MS88]. Firstly, a point p on the walk w is picked randomly and used as a pivot. Then a random symmetry operation of the lattice, like a rotation, is applied to the second part (suffixes) of the walk, using p as origin. If the resulting walk is a SAW, it is accepted, else it is rejected and w is counted once again in the sample. A more detailed and precise algorithm can be found in [MS88]. In this article, it is shown that, quoting Madras and Sokal,

Theorem [10]. *The pivot algorithm is ergodic for self-avoiding walks on* $\mathbb{Z}^d$ *provided that all axis reflections, and either all 90° rotations or all diagonal reflections, are given nonzero probability. In fact, any* $N-$*step SAW can be transformed into a straight rod by some sequence of* $2N-1$ *or fewer such pivots.*

The pivot algorithm is ergodic too for SAWs on the square lattice [MS88], provided that the 180° rotation, and either both 90° rotations or both diagonal reflections, are given nonzero probability, whereas 90° rotations alone are not enough, due to Fig. 8.5.

8.2.3/ INTRODUCING THE (UN)FOLDED SELF-AVOIDING WALKS

8.2.3.1/ PROTEIN FOLDING AS PRELIMINARIES

The overriding problem in PSP is: *how to find such a minimal conformation, given all the* $n-$*step self-avoiding walks and the sequence of hydrophobicity of the protein ?*

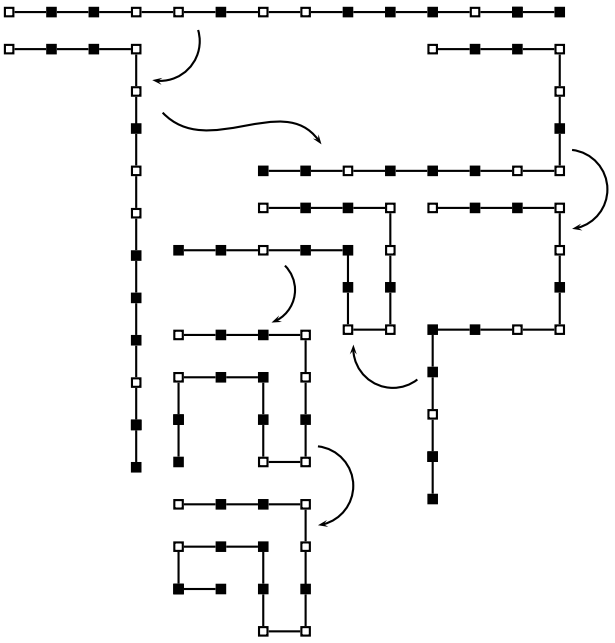


Figure 8.6: Protein Structure Prediction by folding SAWs

To find the best 2D conformation of a protein, given its sequence of hydrophobicity, is really not an easy task. Indeed authors of [CGP$^+$98] have proven that, considering the set of self-avoiding walks having $n-$steps and whose vertices are either black (hydrophobic) or white squares (hydrophilic residues), to determine the SAWs of this set that maximize the number of neighboring black squares is NP-hard. Given a sequence of amino acids, such statement leads to the use of heuristics to predict (and not to determine exactly) the most probable conformation of the protein. These heuristics operate as in the real biological world, folding or increasing the length of SAWs in order to minimize the free energy of the associated conformation: by doing so, the protein synthesis in aqueous environment is reproduced *in silico*. As stated previously, we have shown in a previous work that such investigations potentially lead to various subsets of self-avoiding walks [BCGS12a, BCG11a, GCBB].

In the first approach, starting from the straight line, we obtain by a succession of pivot moves of 90° a final conformation being a self-avoiding walk. In this approach, it is not regarded whether the intermediate walks are self-avoiding or not. Such a method corresponds to programs that start from the initial conformation, fold several times the linear protein, according to their embedded scoring functions, and then obtain a final conformation on which the SAW requirement is verified. It is easy to be convinced that, by doing so, the set of final conformations is exactly equal to the set of self-avoiding walks having n steps. As the conformations obtained by such methods coincide exactly to the well-studied global set of all SAWs, such an approach is not further investigated in what follows [GCBB].

In the second approach, the same process is realized, except that all the intermediate conformations must be self-avoiding walks (see Fig. 8.6). The set of $n-$step SAWs reachable by such a procedure is denoted by $fSAW_n$ in what follows. Such a procedure is one of the two most usual translations of the so-called "SAW requirement" in the bioinformatics literature, leading to proteins' conformations belonging into $fSAW_n$. For instance, PSP

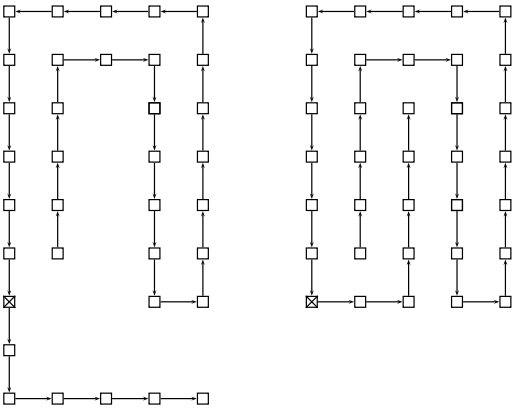


Figure 8.7: Pivot move acceptable in $fSAW$ but not in $fSAW'$

methods presented in [IC10, UM93, BUAM97, HSHS10, HC10] follow such an approach. We have shown in [GCBB] that $fSAW_n \subsetneq \mathcal{S}_n$ [MS88]. In other words, *in this first category of PSP software, it is impossible to reach all the conformations of* $\mathcal{S}_n$.

Other approaches in the same category can be imagined, like the following one. We can act as above, requiring additionally that no intersection of vertex or edge during the transformation of one SAW to another occurs. For instance, the pivot move of Figure 8.7 is authorized in the previous $fSAW$ approach, but it is refused in the current one: during the rotation around the residue having a cross, the rigid structure after this residue intersects the remainder of the "protein" (see Fig. 8.8). In this two dimensional approach denoted by $fSAW'$, it is impossible for a protein folding from one plane conformation to another plane one to use the 3D space to achieve this folding. A reasonable modeling of the true natural folding dynamics of an already synthesized protein can be obtained by extending this requirement to the third dimension. However, due to its complexity, this requirement is actually never used by tools that embed a 2D HP square lattice model for protein structure prediction. This is why these particular SAWs are not really investigated in this section. Let us just emphasize that $fSAW'_n$ is obviously a subset of $fSAW_n$, but there is *a priori* no reason to consider them equal. Indeed, Figure 8.9 shows that,

Proposition 14. *For all* $n \in \mathbb{N}^*$, $fSAW'_n \subset fSAW_n$. *However,* $\exists n \in \mathbb{N}^*$, $fSAW'_n \neq fSAW_n$.

Proof 2. *In Figure 8.9, the unique possible pivot move is the red dot, and obviously such move leads to the intersection between the head and the queue of the structure during the transformation.*

Note that we only studied pivot moves of $\pm 90°$ in our research. But to consider other sets of transformations could be interesting in some well-defined contexts, which can potentially lead to different new subsets of SAWs.

A last bioinformatics approach of protein structure prediction using self-avoiding walks starts with an $1-$step SAW, and at iteration k, a new step is added at the queue of the walk, in such a way that the new $k-$step self-avoiding walk presents the best value for the considered scoring function (see Fig 8.10). The protein is thus constructed step by step, reaching the best local conformation at each iteration. It is easy to see that such an approach leads, another time, to all the possible self-avoiding walks having the length of the considered protein [GCBB].

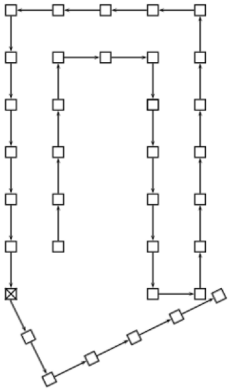


Figure 8.8: An intersection appears between the head and the queue during the transformation, thus this pivot move is refused in $fSAW'$.

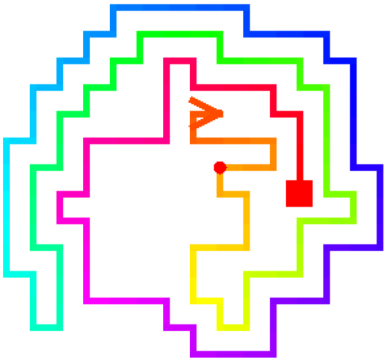


Figure 8.9: $fSAW_n \neq fSAW'_n$

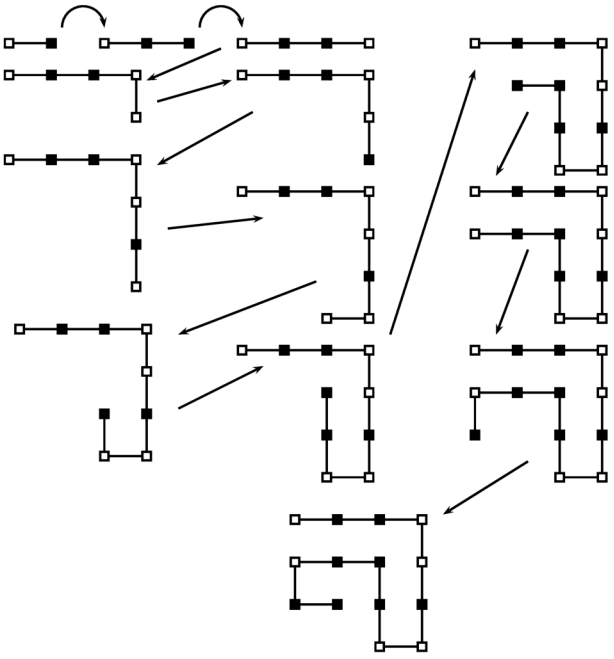


Figure 8.10: Protein Structure Prediction by stretching SAWs

In the remainder of this chapter, we give a more rigorous definition of the $fSAW_n$ set, we initiate its study, and compare it to the well-known $\mathcal{S}_n$ SAWs set.

8.2.3.2/ NOTATIONS

One of the easiest way to define the folded self-avoiding walks described previously, that appear during the realization of the SAW requirement in PSP algorithms, is to use the absolute encoding of a walk introduced previously in this chapter. In this encoding, a $n+1-$step walk $w = w(0), \ldots, w(n) \in \left(\mathbb{Z}^2\right)^{n+1}$ with $w(0) = (0,0)$ is a sequence $s = s(0), \ldots, s(n-1)$ of elements belonging into $\mathbb{Z}/4\mathbb{Z}$, such that:

- $s(i) = 0$ if and only if $w(i+1)_1 = w(i)_1 + 1$ and $w(i+1)_2 = w(i)_2$, that is, $w(i+1)$ is at the East of $w(i)$.
- $s(i) = 1$ if and only if $w(i+1)_1 = w(i)_1$ and $w(i+1)_2 = w(i)_2 - 1$: $w(i+1)$ is at the South of $w(i)$.
- $s(i) = 2$ if and only if $w(i+1)_1 = w(i)_1 - 1$ and $w(i+1)_2 = w(i)_2$, meaning that $w(i+1)$ is at the West of $w(i)$.
- Finally, $s(i) = 3$ if and only if $w(i+1)_1 = w(i)_1$ and $w(i+1)_2 = w(i)_2 + 1$ ($w(i+1)$ is at the North of $w(i)$).

8.2.3.3/ A GRAPH STRUCTURE FOR SAWS FOLDING PROCESS

We can now recall a graph structure, formerly introduced in [GCBB], which describes well the iterations of $\pm 90°$ pivot moves on a given self-avoiding walk. Given $n \in \mathbb{N}^*$, the graph $\mathfrak{G}_n$ is defined as follows:

- its vertices are the $n-$step self-avoiding walks, described in absolute encoding;
- there is an edge between two vertices s_i, s_j if and only if s_j can be obtained by one pivot move of $\pm 90°$ on s_i, that is, if there exists $k \in [\![0, n-1]\!]$ s.t.:
  - either $s_i(0), \ldots, s_i(k-1), f(s_i(k)), \ldots, f(s_i(n)) = s_j$
  - or $s_i(0), \ldots, s_i(k-1), f^{-1}(s_i(k)), \ldots, f^{-1}(s_i(n)) = s_j$.

Such a digraph is depicted in Figure 8.11. The circled vertex is the straight line whereas strikeout vertices are walks that are not self-avoiding. Depending on the context, and for the sake of simplicity, $\mathfrak{G}_n$ will also refer to the set of SAWs in $\mathfrak{G}_n$ (*i.e.*, its vertices).

Using this graph, the folded SAWs introduced in the previous section can be redefined more rigorously [GCBB].

Definition [47]. *$fSAW_n$ is the connected component of the straight line $00\ldots0$ (n times) in $\mathfrak{G}_n$, whereas $\mathcal{S}_n$ is constituted by all the vertices of $\mathfrak{G}_n$.*

Figure 8.5 shows that the connected component $fSAW(223)$ of the straight line in $\mathfrak{G}_{223}$ is not equal to the whole graph: $\mathfrak{G}_{223}$ is not connected. More precisely, this graph has a connected component of size 1: Figure 8.5 is totally unfoldable, whereas SAW of Fig. 8.9

can be folded exactly once [GCBB]. Indeed, to be in the same connected component is an equivalence relation $\mathcal{R}_n$ on $\mathfrak{G}_n, \forall n \in \mathbb{N}^*$, and two SAWs w, w' are considered equivalent (with respect to this equivalence relation) if and only if there is a way to fold w into w' such that all the intermediate walks are self-avoiding [BGNP13]. When existing, such a way is not necessarily unique.

These remarks have led us to introduce following definitions in [BGNP13].

Definition 48. *Let $n \in \mathbb{N}^*$ and $w \in \mathcal{S}_n$. We say that:*

- w *is* unfoldable *if its equivalence class, with respect to $\mathcal{R}_n$, is of size 1;*
- w is a folded self-avoiding walk *if its equivalence class contains the $n-$step straight walk $000\ldots0$ ($n-1$ times);*
- w can be folded k times *if a simple path of length k exists between w and another vertex in the same connected component of w.*

Moreover, we have introduced the following sets in [BGNP13]:

- $fSAW(n)$ *is the equivalence class of the $n-$step straight walk, or the set of all folded SAWs.*
- $fSAW(n,k)$ *is the set of equivalence classes of size k in $(\mathfrak{G}_n, \mathcal{R}_n)$.*
- $USAW(n)$ *is the set of equivalence classes of size 1 $(\mathfrak{G}_n, \mathcal{R}_n)$, that is, the set of unfoldable walks.*
- $f^1SAW(n)$ *is the complement of $USAW(n)$ in $\mathfrak{G}_n$. This is the set of SAWs on which we can apply at least one pivot move of $\pm 90\check{r}$.*

Example 10. *Figure 8.12 shows the two elements of a class belonging into $fSAW(219,2)$ whereas Fig. 8.5 is an element of $USAW(223)$.*

8.2.4/ A SHORT LIST OF RESULTS ON (UN)FOLDED SELF-AVOIDING WALKS

We now give a first collection of easy-to-obtained results concerning the particular SAW sets introduced in the previous section [BGNP13]. These results have been either obtained mathematically [BGG13] or by using computers [BGMP13].

We firstly show that,

Proposition 15. *The cardinality ϕ_n of $fSAW_n$ satisfies: $2^{n+2} \leqslant \phi_n \leqslant 4 \times 3^n$.*

This result is a consequence of the following lemma.

Lemma 1. *The 2^n $n-$step walks that take steps only in the positive coordinate directions are in $fSAW(n)$.*

This lemma can be proven using the number of cranks of a self-avoiding walk, defined below.

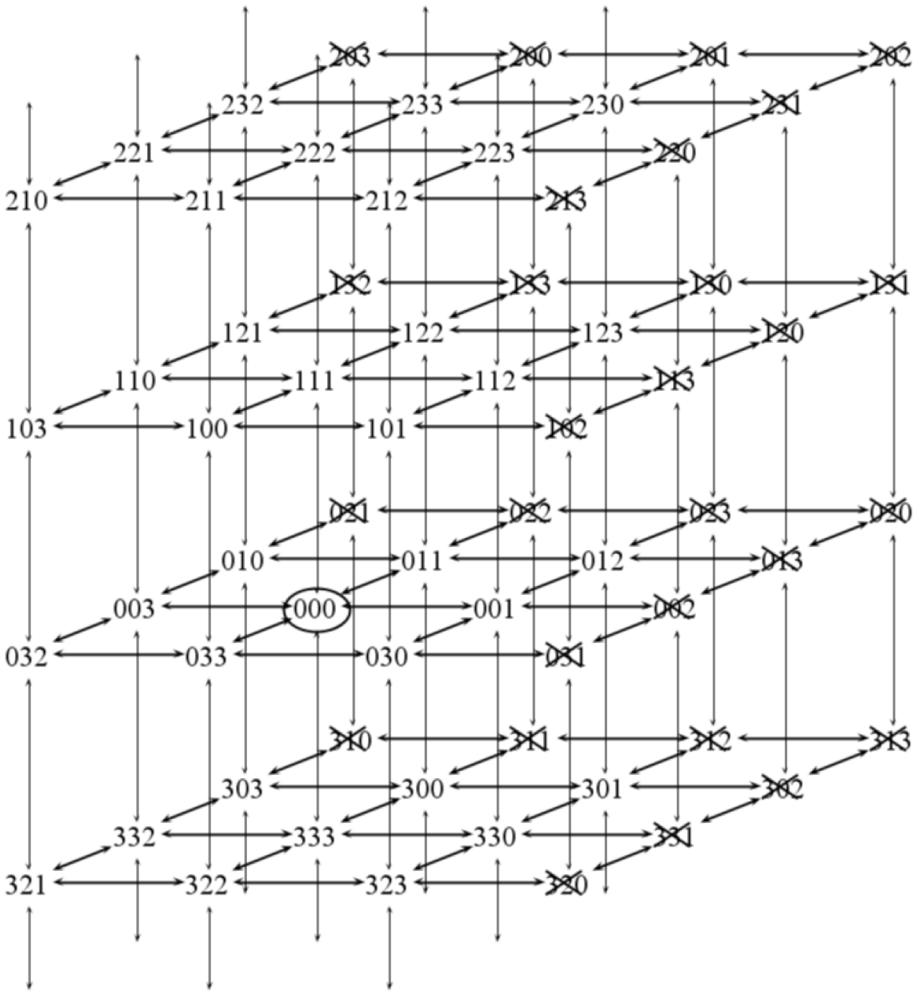


Figure 8.11: The digraph $\mathfrak{G}_3 = fSAW(3)$

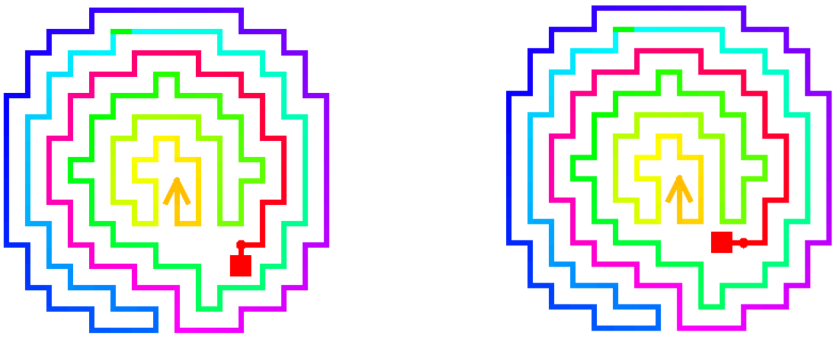


Figure 8.12: The two self-avoiding walks in $fSAW(219, 2)$

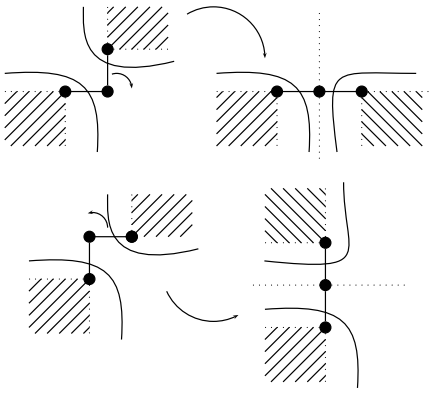


Figure 8.13: Walks that contain only 3 and 0 in their absolute encoding are folded SAWs: reducing the number of cranks does not introduce intersections in the walk.

Definition [49] (Crank)**.** *Let w be a $n-$step self-avoiding walk on $\mathbb{Z}^2$ of absolute encoding s. w contains a crank at position $k \in [\![1, n]\!]$ if $s(k-1) \neq s(k)$.*

For a proof of these results, see [BGNP13]. In particular, SAWs whose absolute encoding is only constituted by 0's and 1's are folded SAWs. It is quite possible that a few 2's or 3's can be added without breaking the folded character of the walk, meaning that the lower bound could be increased.

We can now give a result regarding the $USAW(n)$ set of self-avoiding walks.

Theorem [11]. *There is an infinite number of n such that $USAW(n)$ is nonempty. In particular, the number of unfoldable SAWs is infinite.*

Proof [3]. *A proof of this result, too long to be contained in this manuscript, can be found in [BGG13]. It consists in creating a recursive construction process of unfoldable self-avoiding walks, as depicted in Figure 8.19.*

Proposition [16]. *$\forall n \leqslant 14, fSAW(n) = \mathfrak{G}_n$ whereas $fSAW(107) \subsetneq \mathfrak{G}_{107}$ (see Figure 8.15).*

In other words, let ν_n the smallest $n \geqslant 2$ such that $USAW(n) \neq \varnothing$. Then $15 \leqslant \nu_n \leqslant 107$.

Proof [4]. *We have computed a program that constructs the connected component of the $n-$step straight line for $n \leqslant 14$, and at each time, we have obtained the whole $\mathfrak{G}_n$ (see [BGMP13]). Additionally, we have obtained using a backtracking method the walk depicted in Figure 8.16, which justifies the upper bound of 107: we have verified using a systematic program that no pivot move can be realized in that walk without breaking the self-avoiding requirement. These programs, their explanations and justifications can be found in [BGMP13].*

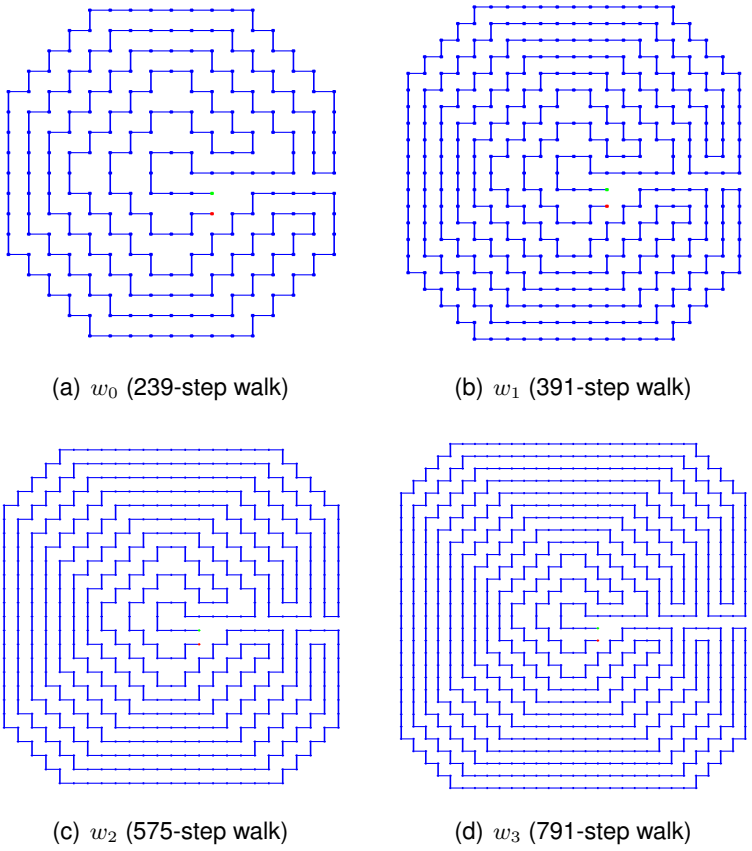
Proposition [17]. *$\forall n \leqslant 28, f^1SAW(n) = \mathfrak{G}_n$.*

Proof [5]. *Obtained experimentally, see [BGMP13].*

The results contained into the two previous propositions are summarized, with all intermediate computations, in Table 8.1. The $\sharp\mathfrak{G}_n$ values, obtained in [Jen04b], are recalled here for comparison.

n	$\sharp\mathfrak{G}_n$	$\sharp f^1SAW(n)$	$\sharp USAW(n) = \sharp\overline{f^1SAW(n)}$	$\sharp fSAW(n)$
1	4	4	0	4
2	12	12	0	12
3	36	36	0	36
4	100	100	0	100
5	284	284	0	284
6	780	780	0	780
7	2172	2172	0	2172
8	5916	5916	0	5916
9	16268	16268	0	16268
10	44100	44100	0	44100
11	120292	120292	0	120292
12	324932	324932	0	324932
13	881500	881500	0	881500
14	2374444	2374444	0	2374444
15	6416596	6416596	0	?
16	17245332	17245332	0	?
17	46466676	46466676	0	?
18	124658732	124658732	0	?
19	335116620	335116620	0	?
20	897697164	897697164	0	?
21	2408806028	2408806028	0	?
22	6444560484	6444560484	0	?
23	17266613812	17266613812	0	?
24	46146397316	46146397316	0	?
25	123481354908	123481354908	0	?
26	329712786220	329712786220	0	?
27	881317491628	881317491628	0	?
28	2351378582244	2351378582244	0	?
29	6279396229332	?	?	?
30	16741957935348	?	?	?
31	44673816630956	?	?	?
⋮	⋮	⋮	⋮	⋮
107	?	?	$\geqslant 3$	?
108	?	?	$\geqslant 1$	?
111	?	?	$\geqslant 5$	?
112	?	?	$\geqslant 1$	?
113	?	?	$\geqslant 2$	?
114	?	?	$\geqslant 2$	?
115	?	?	$\geqslant 5$	?
116	?	?	$\geqslant 3$	?
117	?	?	$\geqslant 4$	?
118	?	?	$\geqslant 2$	?
119	?	?	$\geqslant 2$	?
121	?	?	$\geqslant 4$	?
122	?	?	$\geqslant 5$	?
123	?	?	$\geqslant 1$	?
132	?	?	$\geqslant 7$	?
133	?	?	$\geqslant 6$	?
134	?	?	$\geqslant 95$	?
135	?	?	$\geqslant 165$	?
136	?	?	$\geqslant 40$	?
137	?	?	$\geqslant 50$	?
138	?	?	$\geqslant 175$	?
139	?	?	$\geqslant 179$	?
140	?	?	$\geqslant 66$	?
141	?	?	$\geqslant 119$	?
142	?	?	$\geqslant 322$	?
143	?	?	$\geqslant 476$	?
144	?	?	$\geqslant 8$	?
145	?	?	$\geqslant 18$	?
146	?	?	$\geqslant 54$	?
235	?	?	$\geqslant 1$	?
239	?	?	$\geqslant 1$	?
391	?	?	$\geqslant 1$	?
575	?	?	$\geqslant 1$	?
791	?	?	$\geqslant 1$	?

Table 8.1: Cardinality of various subsets of SAWs



(a) w_0 (239-step walk) (b) w_1 (391-step walk)

(c) w_2 (575-step walk) (d) w_3 (791-step walk)

Figure 8.14: Generating walks that cannot be folded out

Connected components presented previously either have the straight line, or are of size 1 or 2. A reasonable questioning is to wonder whether it is possible to have larger connected components different from the one of the straight line. We have shown that,

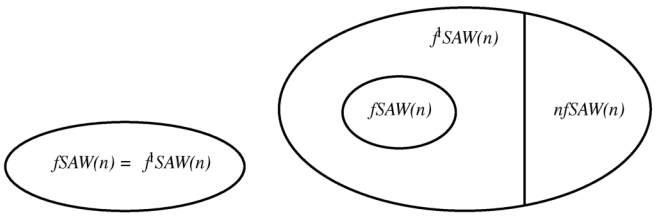
Proposition 18. *It exists $k > 2$ such that $fSAW(n, k)$ is nonempty.*

In other words, connected components different from $fSAW(n)$ and larger than 1 or 2 elements exist. The result, which has been experimentally obtained in [BGMP13], can be proven by exhibiting a counterexample: Figure 8.17 shows a connected component of size 5.

We have thus defined a diameter function D on the connected components of $\mathfrak{G}_n$, such that $D(C)$ is the length of the longest shortest path in the connected component C of $\mathfrak{G}_n$. Consider the connected component of the straight line $fSAW(n)$, we have the result [BGNP13],

Proposition 19. *The diameter of $fSAW(n)$ is equal to $2n$: $D(fSAW(n)) = 2n$.*

Example 11. *In $fSAW(2)$, this diameter corresponds, for instance, to the shortest path $03 \to 00 \to 11 \to 12 \to 23$ (see Figure 8.18).*



(a) $\mathfrak{G}_n$ for $n \leqslant 14$ (b) Diagram of $\mathfrak{G}_n$ for $n = 107$

Figure 8.15: Vien diagram for $\mathfrak{G}_n$

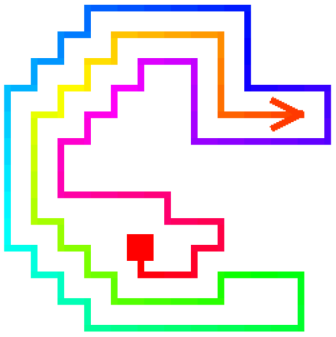


Figure 8.16: Current smallest (107-step) SAW that cannot be folded out

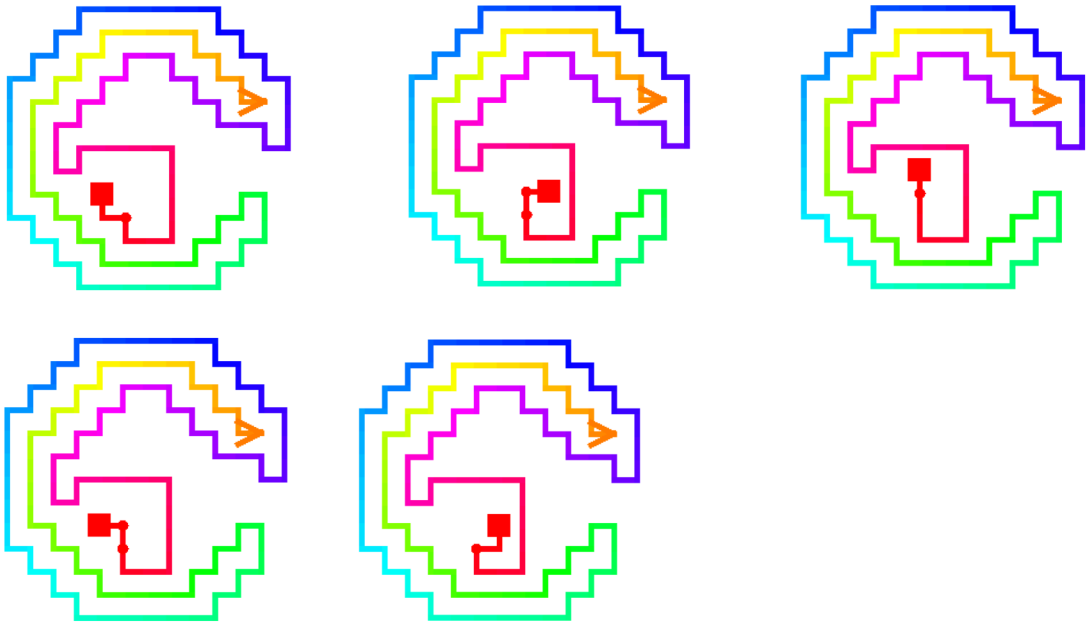


Figure 8.17: A connected component with 5 elements

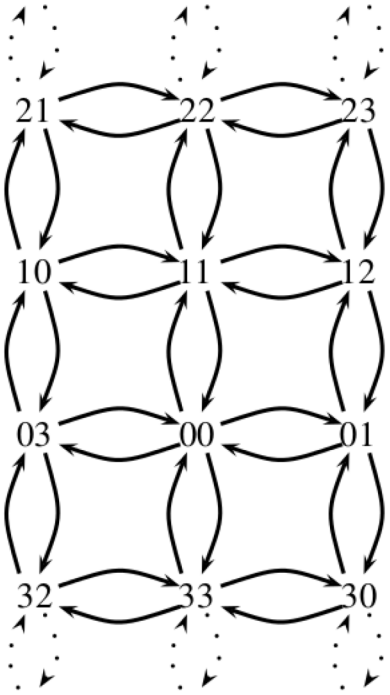


Figure 8.18: The digraph $\mathfrak{G}_2 = fSAW(2)$

8.2.5/ A LIST OF OPEN QUESTIONS

Our last investigations in the field of unfolded self-avoiding walks have consisted in enumerating in [BGNP13] a list of open questions that have appeared to us as interesting. Some of them should be very easy to solve, whereas other ones may involve a degree of difficulty.

In the following we define $fSAW^d(n)$ as the class of equivalency of the $n-$step straight walk on $\mathbb{Z}^d$ and $\mathfrak{G}_n^d$ is the equivalent of $\mathfrak{G}_n$ in $\mathbb{Z}^d$. Note that $fSAW^2(n)$ is equal to $fSAW(n)$, as introduced in Definition 8.2.3.3.

1. For any dimension d, do we have the existence of $n \in \mathbb{N}^*$ such that $fSAW^d(n) \subsetneq \mathfrak{G}_n^d$?
2. $fSAW^2(2)$ and $fSAW^2(3)$ are obviously connected graphs, but they are not Eulerian. Indeed, more than two vertices have an odd degree both in $fSAW^2(2)$ and $fSAW^2(3)$ (see Figures 8.18 and 8.11). Is it the case for all $fSAW^d(n)$?
3. $fSAW^2(2)$ and $fSAW^2(3)$ are Hamiltonian graphs, with the following Hamiltonian circuits:
   - $00 \to 03 \to 32 \to 23 \to 10 \to 11 \to 22 \to 33 \to 30 \to 21 \to 12 \to 01 \to 00$ for $fSAW^2(2)$ (see Figure 8.18).
   - $000 \to 003 \to 010 \to 011 \to 012 \to 001 \to 030 \to 323 \to 330 \to 301 \to 300 \to 333 \to 322 \to 321 \to 332 \to 303 \to 232 \to 233 \to 230 \to 223 \to 212 \to 211 \to 210 \to 221 \to 222 \to 111 \to 110 \to 121 \to 122 \to 123 \to 112 \to 101 \to 100 \to 103 \to 032 \to 033 \to 000$ for $fSAW^2(3)$ (see Figure 8.11).

Is it a coincidence, or is it the case for every $fSAW^d(n)$?

4. What is the exact value of the diameter $D(fSAW^d(n))$?

5. Do we have a connective constant for $fSAW^d(n)$. That is, does the limit $\lim_{n\to+\infty} \phi_n^{1/n}$ exist, and can we bound it ?

6. $u_n = \sharp USAW^d(n)$ is an increasing sequence (for $d = 2$, or for any d)? Does it grow at a given (linear or exponential) rate?

7. Let $k \in \mathbb{N}$. Is the sequence $v_n = \sharp fSAW(n, k)$ increasing with n ? If so, at which rate, and does it depend on the dimension d? And what about the sequence $w_k = \sharp fSAW(n, k)$ for a given n ?

8. More simply, is there an unfoldable walk in $\mathbb{Z}^3$?

9. Are the connected components of $\mathfrak{G}_n^d$ convex ? In other words, given two SAWs in a same component C. Are all (or at least one) the shortest paths connecting them on $\mathbb{Z}^d$ in C?

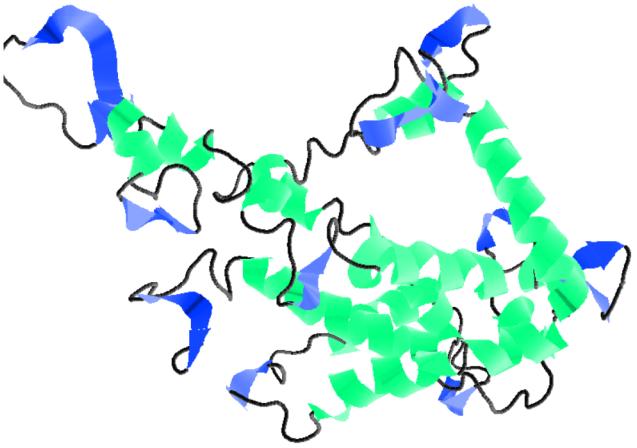
10. Is there a generating function expressing the folded self-avoiding walks more simply, making it possible to enumerate them on the square lattice (like what has been realized in [CEG93]).

11. When we can fold a self-avoiding walk until a straight line, is it possible to fold it in such a way that the number of cranks decreases ? And for two given self-avoiding walks w_i and w_j of the same connected component of $\mathfrak{G}_n$, such that w_i has more cranks than w_j, is there a path from w_i to w_j whose vertices' number of cranks is decreasing ? Is there a relation between the vertex depth and the number of cranks in $\mathbb{Z}^d$?

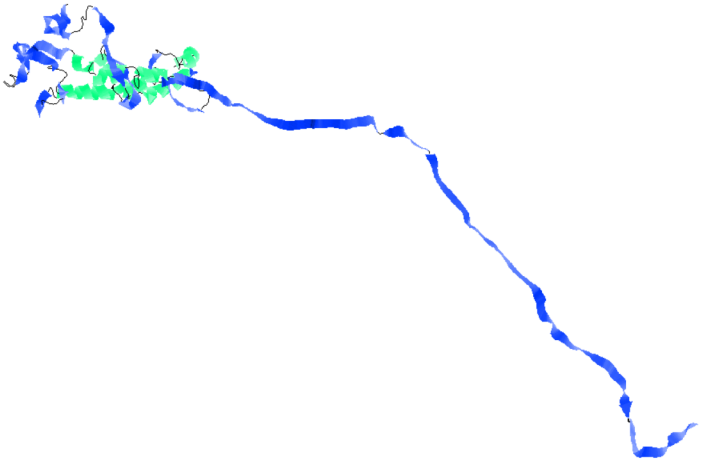
8.2.6/ CONSEQUENCES ON PROTEIN FOLDING

This first theoretical study about folded self-avoiding walks raises several questions regarding the protein structure prediction problem and the current ways to solve it [BGNP13]. In one category of PSP software, the protein is supposed to be synthesized first as a straight line of amino acids, and then this line of a.a. is folded out until reaching a conformation that optimizes a given scoring function. By doing so, the obtained backbone structures all belong into $fSAW(n)$, where n is the number of residues of the protein. The second category of PSP software consider that, as the protein is already in the aqueous solvent, it does not wait the end of the synthesis to take its 3D conformation. So they consider SAWs whose number of steps increases from 1 to the number of amino acids of the targeted protein and, at each step k, the current walk is stretched (one amino acid is added to the protein) in such a way that the pivot k is placed in the position that optimizes the scoring function they consider. By doing so, the possible predicted backbones are the whole $\mathfrak{G}_3$. The two sets of possible conformations are different, at least when considering 2D low resolution models.

We have shown in our research that (1) to take place in the first situation (folding the straight line by a succession of pivot moves) can be interesting as the number of possible SAW conformations is smaller than $\sharp\mathfrak{G}_n$. Indeed this interest is directly related to the rate



(a) Conformation having best score (27)



(b) Second best conformation (score 24)

Figure 8.19: Illustration of chaos in protein folding (conformations have been predicted using RaptorX)

$\frac{\sharp fSAW(n)}{\sharp \mathfrak{G}_n} < 1$. If this rate decreases dramatically when n increases, then the computational advantage is obvious. However, we have currently no idea of such a gain, that is, of the growing rate of $\sharp fSAW(n)$ compared to $\sharp \mathfrak{G}_n < 1$. (2) The use of heuristics instead of exact methods (like SAT solvers for instance) is *a priori* not justified for PSP software that fold the straight line. Indeed, the PSP problem has been proven NP hard on the set $\mathfrak{G}_n$ of all possible SAWs. As they consider a strict subset of it, the complexity of the problem might be reduced due to a lower number of cases to consider. However, Proposition 15 tends to indicate that this problem still remains difficult in $fSAW(n)$, which nevertheless necessitates a rigorous complexity proof. (3) Biologically speaking, to suppose that the proteins wait to be completely synthesized before starting to fold appears as unrealistic, as the synthesis occurs in an aqueous solvent. Indeed, the protein starts to fold during its synthesis. Furthermore, in our opinion, it is restrictive to consider that the head of the protein definitively stops to fold after having synthesized. Such a supposition is equivalent to make a confusion between local (the SAW at step k) and global (the final optimal SAW)

optimization. Indeed, we recognize honestly that we have no idea to determine if this third approach (continuously folding the walk while stretching it) is more reasonable than the previous ones, and if it is equivalent to either $fSAW(n)$ or to $\mathfrak{G}_n$ (or if it constitutes a third different subset of SAWs).

Our goal is only to point out the importance to determine the best dynamical system to model protein folding before programming it in PSP software, as this model determines which conformations can be predicted. A last remark to emphasize the importance of such a study: as recalled previously, we have proven in [BCG11a] that the dynamical system used in the "folding the straight line" category is chaotic according to Devaney, meaning that any wrong choice of pivot move (due to approximations in the scoring function, for instance) can potentially become dramatic. Other research works ([BUAM97] for instance) tend to show that the protein folding process intrinsically embeds a certain amount of chaos. Thus, to use a more or less erroneous model to predict the conformation could have grave consequences in prediction quality. Figure 8.19 shows the two best conformations predicted by RaptorX [PX11], a well-known PSP software. We can see that using twice a same model, but with different parameters can potentially lead to quite different conformations, illustrating a possible effect of some chaotic properties exhibited by the chosen model. We can reasonably wonder what is the effect of a wrong model in such a prediction.

9

Study of Genomic Recombinations

After proteins, we naturaly have studied the complex evolution of DNA sequences over time. This evolution has firstly been described with a discrete dynamical system in [BGPa], before investigating the particular cases of mutations [BGP12a, BGP12b, BGPb] and of transposable elements.

9.1/ Chaos Properties in Genomic Evolution

9.1.1/ Introduction

Due to mutations or recombination, some variations occur in the frequency of each codon, and these codons are thus not uniformly distributed into a given genome. Since the late '60s, various genome evolutionary models have been proposed to predict the evolution of a DNA sequence as the generations pass. Mathematical models allow the prediction of such an evolution, in such a way that statistical values observed into current genomes can be at least partially recovered from hypotheses on past DNA sequences. Moreover, it can be attractive to study the genetic patterns (blocs of more than one nucleotide: dinucleotides, trinucleotides...) that appear and disappear depending on mutation parameters.

A first model for genomes evolution has been proposed in 1969 by Thomas Jukes and Charles Cantor [JC69]. This first model is very simple, as it supposes that each nucleotide A, C, G, T has the probability m to mutate to any other nucleotide, as described in the following mutation matrix,

$$\begin{pmatrix} * & m & m & m \\ m & * & m & m \\ m & m & * & m \\ m & m & m & * \end{pmatrix}.$$

In that matrix, the coefficient in row 3, column 2 represents the probability that the nucleotide G mutates in C during the next time interval, *i.e.*, $P(G \to C)$. As diagonal elements can be deduced by the fact that the sum of each row must be equal to 1, they are omitted here.

This first attempt has been followed up by Motoo Kimura [Kim80], who has reasonably considered that transitions ($A \longleftrightarrow G$ and $T \longleftrightarrow C$) should not have the same mutation

rate than transversions ($A \longleftrightarrow T$, $A \longleftrightarrow C$, $T \longleftrightarrow G$, and $C \longleftrightarrow G$), leading to the following mutation matrix:

$$\begin{pmatrix} * & b & a & b \\ b & * & b & a \\ a & b & * & b \\ b & a & b & * \end{pmatrix}.$$

This model has been refined by Kimura in 1981 (three constant parameters, to make a distinction between natural $A \longleftrightarrow T$, $C \longleftrightarrow G$ and unnatural transversions), leading to:

$$\begin{pmatrix} * & c & a & b \\ c & * & b & a \\ a & b & * & c \\ b & a & c & * \end{pmatrix}.$$

Joseph Felsenstein [Fel80] has then supposed that the nucleotides frequency depends on the kind of nucleotide A,C,T,G. Such a supposition leads to a mutation matrix of the form:

$$\begin{pmatrix} * & \pi_C & \pi_G & \pi_T \\ \pi_A & * & \pi_G & \pi_T \\ \pi_A & \pi_C & * & \pi_T \\ \pi_A & \pi_C & \pi_G & * \end{pmatrix}$$

with $3\pi_A, 3\pi_C, 3\pi_G$, and $3\pi_T$ denoting respectively the frequency of occurrence of each nucleotide. Masami Hasegawa, Hirohisa Kishino, and Taka-Aki Yano [HKY85] have generalized the models of [Kim80] and [Fel80], introducing in 1985 the following mutation matrix:

$$\begin{pmatrix} * & \beta\pi_C & \alpha\pi_G & \beta\pi_T \\ \beta\pi_A & * & \beta\pi_G & \alpha\pi_T \\ \alpha\pi_A & \beta\pi_C & * & \beta\pi_T \\ \beta\pi_A & \alpha\pi_C & \beta\pi_G & * \end{pmatrix}.$$

These efforts have been continued by Tamura, who proposed in [Tam92, TN93] a simple method to estimate the number of nucleotide substitutions per site between two DNA sequences, by extending the model of Kimura (1980). The idea is to consider a two-parameter method, for the case where a GC bias exists. Let us denote by π_{GC} the frequency of this dinucleotide motif. Tamura supposes that $\pi_G = \pi_C = \dfrac{\pi_{GC}}{2}$ and $\pi_A = \pi_T = \dfrac{1-\pi_{GC}}{2}$, which leads to the following rate matrix:

$$\begin{pmatrix} * & \kappa(1-\pi_{GC})/2 & (1-\pi_{GC})/2 & (1-\pi_{GC})/2 \\ \kappa\pi_{GC}/2 & * & \pi_{GC}/2 & \pi_{GC}/2 \\ (1-\pi_{GC})/2 & (1-\pi_{GC})/2 & * & \kappa(1-\pi_{GC})/2 \\ \pi_{GC}/2 & \pi_{GC}/2 & \kappa\pi_{GC}/2 & * \end{pmatrix}.$$

In the last model of Tamura [TN93], the two different types of transversions ($A \leftrightarrow T, C \leftrightarrow G$) can have a different rate, whereas transversions are all assumed to occur at the same rate (but that rate is allowed to be different from both of the rates for transitions):

$$\begin{pmatrix} * & \kappa_1\pi_C & \pi_G & \pi_T \\ \pi_A & * & \pi_G & \kappa_1\pi_T \\ \kappa_2\pi_A & \pi_C & * & \pi_T \\ \pi_A & \pi_C & \kappa_2\pi_G & * \end{pmatrix}.$$

All these models lead to the so-called GTR model [Yan94], in which the mutation matrix has the form (using obvious notations):

$$\begin{pmatrix} * & f_{AC}\pi_C & f_{AG}\pi_G & f_{AT}\pi_T \\ f_{AC}\pi_A & * & f_{CG}\pi_G & f_{CT}\pi_T \\ f_{AG}\pi_A & f_{CG}\pi_C & * & \pi_T \\ f_{AT}\pi_A & f_{CT}\pi_C & \pi_G & * \end{pmatrix}.$$

A second category of models focus on di or trinucleotides evolution. It has been initiated by the Markov approach of [GY94], which has used a 61×61 rate matrix for protein-coding DNA sequences, and at the same time in [MG94]. More recently, Didier Arquès, Jean-Paul Fallot, and Christian Michel have proposed in 1998 a first evolutionary model on the $\{A, C, G, T\}$ alphabet that is based on trinucleotides [AFM98]. With such a model, the mutation matrix has now a size 64×64 (there are 64 trinucleotides). In this model, the 3 parameters p, q, r correspond, for a given trinucleotide XYZ, to the probability p of mutation of the first nucleotide X, the mutation probability q of Y, and the probability r that Z mutates. As for the nucleotides based models, this new approach has taken into account only constants parameters. In 2004, Jacques M. Bahi and Christian Michel have published a novel research work in which the model of 1998 has been improved by replacing constants parameters by new time dependent parameters [BM04]. The common point of all the models studied by Michel *et al.* is that almost all their mutation matrices are symmetric. Finally, Jacques M. Bahi and Christian Michel have recently introduced in [BG08], a last model with 3 constant parameters, but *whose evolution matrix evolves over time*. In other words, trinucleotides that have to mutate are not fixed, but they are randomly picked among a subset of potentially mutable trinucleotides. CM model outperforms largely the standard models, being closer to the observed frequencies of trinucleotides (for the evidences of such a claim, see [BM08a]).

The starting point of [BGPa] is to investigate possible reasons justifying the performance of the CM model. Obviously, to suppose that not all of the trinucleotides have to mutate at each time is reasonable as, for instance, the stop codons have very small mutation probabilities. However, such a biological claim is not sufficient to explain the success of the CM model to simulate with accuracy the dynamics of mutations into genomes. Our proposal is that *the dynamics of genomes evolution is indeed chaotic*, as it is defined by the Devaney's formulation. This is why linear non-chaotic models of evolution are far from what they attempt to model, leading to a poor accuracy in their prediction. Contrarily, we have recalled that discrete dynamical systems in chaotic iterations mode satisfy the Devaney's definition of chaos. Thus the CM model, which is based on chaotic iterations, uses a chaotic dynamical system to describe a chaotic behavior, leading to a nucleotides mutation model of the same nature than the phenomenon under study.

We have demonstrated in [BGPa] that, contrary to inversions and transpositions, mutations occurring in genomes have a chaotic dynamics. In particular, such results imply that linear models for nucleotides evolution prediction are quite irrelevant. Let us remark that, as emphasized in the partial but representative literature survey presented above, mathematical models for DNA evolutionary changes have majorly focused on stochastic models for nucleotide mutations, whereas [BGPa] is more largely concerned by modeling the dynamical behavior of the "evolutionary history" of a DNA sequence. In other words, existing researches use probabilistic models, while we have regarded in [BGPa] the predictable character of DNA evolution under mutations, inversions, and transpositions. This is why the proposed approach necessitates to redefine the well-known op-

erations of mutations, inversions, and transpositions as dynamical systems on relevant topological spaces, leading to formulations of DNA evolution quite different from existing well-established researches. The contribution [BGPa] is summarized thereafter.

9.1.2/ GENOMICS MUTATIONS AS A DISCRETE DYNAMICAL SYSTEM

9.1.2.1/ PRESENTATION OF THE PROBLEM

As stated previously, the question raised by this research work is to determine whether the evolution of a DNA sequences under evolution can be predicted or not. Obviously, this questioning is related to the determination of the possible chaotic nature of such an evolution.

In this section, we will more specifically focus on the following problems. Firstly, given a genome (or any DNA sequence) G of interest, and a more or less precise idea of mutations that it will probably face in future (for instance, some areas into the genomes are known to mutate more frequently than other ones), is it possible to infer a set of the most probable genomes that can result, in the future, from this original sequence G after mutations ? Secondly, given a sequence known at the current generation (say, at time t^n), is it possible to determine what was the most probable aspect of this sequence in the past (at time $t^m, m < n$)? Thirdly, given two DNA sequences, the second one being the result of some mutations on the first one, is it possible to discover the mutations sequence that has changed the first sequence in the second one (taking into account the fact that a given nucleotide can mutate several times).

Obviously, with no information about the mutation rate and history of the considered DNA sequence, this prediction is quite impossible. But what appends if we can follow the DNA sequence on some generations, learning by doing so information about the possible form of its mutations sequence ? For instance, following a lineage of *Escherichia coli* during 40000 generations [LM08] gives us a lot of information concerning the behavior of mutations in the genomes of the considered lineage. Is it possible to use this knowledge to predict the genome of this lineage at generation number 45000 ? In other words, knowing the initial DNA sequence G^0 at time t^0 and the 40000 first terms of the mutations sequence, can we predict the DNA sequence at time t^{45000} ?

With the knowledge of G^0 and the whole mutations sequence $S = (S^0, \ldots, S^{45000})$, the genome G^{45000} can be obtained without prediction, but what happens to our capability to make prediction when using only the head $(S^0, \ldots, S^{40000})$ of this sequence ? This head can be seen as an approximation of the true mutations sequence S, and if the evolution dynamics of the mutations is quite stable through approximations, then this prediction makes sense. Following [BGPa], to measure the stability of the mutations dynamics through small errors or approximations, and the capability to predict the evolution of genomes under mutations, we must firstly write this mutation operation as a dynamical system, provide an accurate distance that corresponds to the “approximation” cited below, and measure the effects of our ignorance on the complete mutations sequence on the prediction of genomes evolution.

9.1.2.2/ FORMALIZATION OF DNA MUTATION EVOLUTION

A genome having N nucleotides is formalized here as a sequence of N integers belonging into $\{1, 2, 3, 4\}$, where 1 (resp. $2, 3$, and 4) refers to the adenine (resp. cytosine, guanine, and thymine). An evolution under nucleotides mutations of this genome is a sequence of couples of $[\![1, \mathsf{N}]\!] \times [\![1, 4]\!]$, where we suppose that [BGPa]:

- time has been divided into a sequence $t^0, t^1, \ldots, t^n, \ldots$ such that at most one mutation can occur between two time intervals,
- the $i-$th couple of the mutation sequence is equal to (m, n) if and only if the $m-$th nucleotide of the genome is replaced into the nucleotide n. If the $m-$th nucleotide was n, then no mutation has occurred at time t^i.

Such a sequence will be called "mutations sequence" in the remainder of this chapter. $\mathcal{S}_{\mathsf{N}} = \bigcup_{n\in\mathbb{N}} ([\![1, \mathsf{N}]\!] \times [\![1, 4]\!])^n$ will denote the (infinite) set of all possible mutations (finite) sequences. We introduce the phase space $\mathcal{X}_{\mathsf{N}} = [\![1, 4]\!]^{\mathsf{N}} \times \mathcal{S}_{\mathsf{N}}$ as the set of mutating genomes. It is constituted by couples of points that store the information of a genome *and* its future evolution: the first coordinate of the couple is the current DNA sequence whereas the second coordinate is the sequence of mutations that will appear in the future (the problem is that this sequence can only be, in the best case, approximated concretely).

Example [12]. *For instance, the point* $((1, 1, 2, 1, 3), ((2, 2), (2, 3), (1, 4))) \in \mathcal{X}_5$ *corresponds to the evolution* $\{AACAG, ACCAG, AGCAG, TGCAG\}$*: the left coordinate* $(1, 1, 2, 1, 3)$ *means that we start with the sequence* $AACAG$*, whereas the second coordinate* $((2, 2), (2, 3), (1, 4))$ *explains that:*

1. *the first mutation* $(2, 2)$ *is a substitution of the second nucleotide by* C*,*
2. *the second mutation* $(2, 3)$ *is a substitution of the second nucleotide by* G*,*
3. *the third and last mutation* $(1, 4)$ *refers to the substitution of the first nucleotide by* T*, which is designed here by* 4*.*

Before describing the mutation operation on the phase space $\mathcal{X}_{\mathsf{N}}$, let us highlight the following points [BGPa].

- Mutation sequences like $S = ((2, 2), (2, 2), (2, 2))$ are accepted, even if there actually is no change happening. Such sequences are useful to describe an absence of mutation during two time units, which is, biologically speaking, relevant.
- Multiple changes cannot occur simultaneously. However, such a situation can be taken into account by considering sequences of sets of couples, instead of sequences of couples. This generalization is realizable by adapting, *mutatis mutandis*, the remainder of this section by considering such sets sequences. However, we do not see fit to burden the proposed model, as it is easy to check that proofs presented in what follows continue to hold in this more general set.

Let us now introduce the *initial* and *shift* operators i and σ defined respectively by

$$\begin{array}{rccc} i: & \mathcal{S}_{\mathsf{N}} & \longrightarrow & [\![1,\mathsf{N}]\!]\times[\![1,4]\!] \\ & (s^0,s^1,\ldots) & \longmapsto & s^0 \end{array}$$

and

$$\begin{array}{rccc} \sigma: & \mathcal{S}_{\mathsf{N}} & \longrightarrow & \mathcal{S}_{\mathsf{N}} \\ & (s^0,s^1,\ldots) & \longmapsto & (s^1,s^2,\ldots). \end{array}$$

With this material introduced in [BGPa], the mutation operation $\mathcal{M}$ over $\mathcal{X}_{\mathsf{N}}$ can be written as follows: $\mathcal{M}:\mathcal{X}_{\mathsf{N}}\longrightarrow\mathcal{X}_{\mathsf{N}}$, s.t.

$$\mathcal{M}\left((G_1,\ldots,G_{\mathsf{N}}),S\right)=\left((G_1,\ldots,G_{i(S)_1-1},i(S)_2,G_{i(S)_1+1},\ldots,G_{\mathsf{N}}),\sigma(S)\right).$$

In other words, the nucleotide at position $i(S)_1$ in the genome $(G_1,\ldots,G_{\mathsf{N}})$ is replaced by the nucleotide $i(S)_2$, and the first substitution $i(S)$ in the mutation sequence S is removed (as the mutation has already been achieved). Thus the DNA evolution as the generations pass can finally be written as the following discrete dynamical system [BGPa]:

$$\left\{\begin{array}{l} X^0=(G^0,S)\in\mathcal{X}_{\mathsf{N}} \\ X^{n+1}=\mathcal{M}(X^n). \end{array}\right. \tag{9.1}$$

Example [13]. *Let us consider Example 12 another time. As stated before,* $X^0=((1,1,2,1,3),((2,2),(2,3),(1,4)))\in\mathcal{X}_5$. *Then* $X^1=\mathcal{M}(X^0)=((1,2,2,1,3),((2,3),(1,4)))$, $X^2=\mathcal{M}(X^1)=((1,3,2,1,3),((1,4)))$, *and* $X^3=\mathcal{M}(X^2)=((4,3,2,1,3),\varnothing)$. *The last DNA sequence, obtained after 3 mutations (3 iterations of the dynamical system), is thus equal to* $G^3=X_1^3=(4,3,2,1,3)$, *which is* $TGCAG$.

Using this natural formalism, it has been possible to study whether the genomic evolution under mutations can be predicted or not in [BGPa]. A question that can immediately strike one is how chaotic behavior can be studied in the space of genomic sequences of predetermined length N, which is a discrete finite space. That is, how would topological analysis be pursued ? Indeed, a careful look reveals that the space is not really that of a genomic sequence, but instead the space of the *evolutionary history* of such a sequence, which is a discrete but infinite space. As explained previously in this manuscript, the discrete character of the infinite phase space is not a problem, as the sole requirement when studying the chaotic behavior of a recurrent sequence on a set $\mathcal{X}$ is that $\mathcal{X}$ is a topological space and that the iteration function is continuous for the associated topology.

9.1.2.3/ A METRIC FOR MUTATION BASED GENOMES EVOLUTION

A relevant metric has then been introduced in [BGPa], in order to measure the correctness of the prediction, and to give consistency to the notion of approximation that has been used several times in the previous section. Let us remark that this metric is not a measure of similarity between two genes or DNA sequences, like the Needleman-Wunch or the Smith-Waterman measures. It intends to measure the distance between two observed evolutions of DNA sequences.

This metric must be defined on the set $\mathcal{X}_{\mathsf{N}}$, to detail how close is a predicted DNA evolution to the real one. It has been constructed as follows [BGPa]. Given $X=(X_1,X_2),Y=(Y_1,Y_2)\in\mathcal{X}_{\mathsf{N}}$, the number $d(X,Y)$:

- has an integral part that computes the differences between the two DNA sequences X_1 (for instance, the predicted or approximated genome) and Y_1 (the real genome), that is, the number of nucleotides that do not correspond in the two genomes (Hamming distance).
- has a fractional part that must be as small as the evolution processes X, Y will correspond for a large duration. More precisely, the $k-$th digit of $d(X, Y)$ will be equal to 0 if and only if, after k generations, the same position (nucleotide) will be changed in both X_1 and Y_1 genomes, and the same nucleotide is inserted in each case.

Such requirements has led to the introduction of the following function [BGPa]:

$$\forall X, Y \in \mathcal{X}_{\mathsf{N}}, d(X, Y) = d_G(X_1, Y_1) + d_S(X_2, Y_2)$$

where

$$\begin{cases} d_G(X_1, Y_1) = \sum\limits_{k=1}^{\mathsf{N}} \delta(X_1^k, Y_1^k), \\ d_S(X_2, Y_2) = \dfrac{9}{\mathsf{N}} \sum\limits_{k=0}^{\infty} \dfrac{\mathcal{F}(X_2^k - Y_2^k)}{10^{k+1}}, \end{cases}$$

where δ is the discrete metric on $\mathbb{R}$, that is, for $x, y \in \mathbb{R}$, $\delta(x, y) = \begin{cases} 1 \text{ if } x \neq y, \\ 0 \text{ else,} \end{cases}$ and $\mathcal{F} : \mathbb{R}^2 \longrightarrow \mathbb{R}^+$ is given by $\mathcal{F}(x_1, x_2) = |x_1| + \delta(0, x_2)$. It has been possible to prove that [BGPa],

Proposition [20]. *Function d is a metric on $\mathcal{X}_{\mathsf{N}}$.*

9.1.2.4/ THE TOPOLOGICAL STUDY OF MUTATIONS

We have firstly proven in [BGPa], using the sequential characterization of the continuity, that,

Proposition [21]. *The mutation operation $\mathcal{M}$ is a continuous function on $(\mathcal{X}_{\mathsf{N}}, d)$.*

It has then been demonstrated that $\mathcal{M}$ is regular and (strongly) transitive on $(\mathcal{X}_{\mathsf{N}}, d)$, and that it has a constant of sensitivity equal to $\mathsf{N} + \dfrac{\lfloor \frac{\mathsf{N}}{2} \rfloor + 1}{\mathsf{N}}$. We thus have claimed that [BGPa],

Theorem [12]. *Mutations that occur into genomes have a chaotic behavior according to Devaney.*

Further investigations of the topological behavior of mutations has then been realized in [BGPa]. We have firstly deduced that genomic mutations possess the instability property: in all neighborhoods of any genome evolution (G, S) there are points that can be separated with distance bigger than ε in the future through mutations. We then have shown that the mutation operator $\mathcal{M}$ is expansive, and its constant of expansiveness is at least equal to 1. Additionally, topological transitivity implies indecomposability: reducing the size of the genome or DNA sequence in order to simplify its complexity, is impossible. And, lastly, genomic mutations are topologically mixing, and chaotic according to Knudsen on $(\mathcal{X}_{\mathsf{N}}, d)$. See [BGPa] for detailed proofs of these results.

9.1.2.5/ DISCUSSION

Conclusion of this study of mutations is that they present a chaotic behavior leading to the impossibility to measure the long term effect of an error in predicting the location and frequency of mutations into genomes. In the worst case scenario, this error will be amplified until having a completely different genome (all the nucleotides are different, as the constant of sensibility is greater than the length of the genome). However this case is rather marginal, mutations do not occur so frequently as the generations pass, and a mutation implies a change of only one nucleotide, leading to the opinion that, at least in short term, the general aspect of the genome under consideration still remains under control when only mutations occur.

Inversion and transpositions are other genomics rearrangements that mostly affects more than one nucleotide. Thus an error in the prediction of these operations can potentially impact more largely the genome evolution. This is why their dynamics have been studied in [BGPa], to measure the impact of error prediction.

9.1.3/ INVESTIGATING THE DYNAMICS OF TWO OTHER GENOMICS REARRANGEMENTS

We have firstly regarded in [BGPa] the case of inversions. To do so, some definitions useful to formalize them have been introduced, they are recalled below.

Definition 50. *The complementary function* $c : [\![1,4]\!] \longrightarrow [\![1,4]\!]$ *is defined by* $c(1) = 4$, $c(4) = 1$, $c(2) = 3$, *and* $c(3) = 2$.

Then the complement of adenine A is thymine T, and $c(2) = 3$ means, for instance, that the complement of cytosine is guanine. We then have defined the inversion process on a chromosome:

Definition 51. *Let* $\mathsf{N} \in \mathbb{N}^*$, *and* $(n_1, \ldots, n_{\mathsf{N}})$ *a chromosome.* Inversions *have the form:* $(n_1, \ldots, n_{i-1}, \underline{n_i, n_{i+1} \ldots, n_{j-1}, n_j}, n_{j+1}, \ldots, n_{\mathsf{N}}) \longrightarrow$

$$(n_1, \ldots, n_{i-1}, \underline{c(n_j), c(n_{j-1}) \ldots, c(n_{i+1}), c(n_i)}, n_{j+1}, \ldots, n_{\mathsf{N}}).$$

For instance, $ACCT\underline{GTAA}TGTTA$ is a possible inversion of $ACCT\underline{TTAC}TGTTA$. Obviously, it is impossible to map the DNA sequence $AAAAAAAA$ into $CCCCCCCC$ using only inversions, as the complement of A is T. This fact is in contradiction with the property of transitivity, leading to the statement that [BGPa],

Theorem 13. *The inversion rearrangement is not chaotic on the set of all genomes of size* N.

We have then investigated the case of transposons. Transposons are DNA sequences that can move into a given genome following a cut and paste mechanism[1]: $(n_1, \ldots, n_{i-1}, \underline{n_i, \ldots, n_j}, n_{j+1}, \ldots, n_k, n_{k+1}, \ldots, n_{\mathsf{N}}) \longrightarrow (n_1, \ldots, n_{i-1}, n_{j+1}, \ldots, n_k, \underline{n_i, \ldots n_j}, n_{k+1}, \ldots, n_{\mathsf{N}})$. Obviously this transposition cannot fit the requirements of

[1] Transposons will be more systematically studied at the end of this chapter

transitivity, as the number of adenines, thymines, guanines, and cytosines are preserved. Then, for instance, it is impossible to join a genome with an high rate of thymine, starting transpositions on a genome with a low rate of T. Thus [BGPa],

Theorem [14]. *Transposition of transposons is not chaotic according to Devaney.*

We thus have stated in [BGPa] that transpositions and inversions *alone* do not exhibit a chaotic behavior, without constructing rigorously a related dynamical system (as explicit counter-examples have been provided).

9.2/ The Specific Case of Nucleotide Mutations

9.2.1/ Introduction

We have presented at the beginning of this chapter a short review in mutations modeling. Other works of interest have been published on related models (codon-substitution model) in [YNH98, Miy11]. However, due to mathematical complexity, the matrices that have been investigated in state of the art to model evolution of DNA sequences are always limited, either by the hypothesis of symmetry or by the desire to reduce the number of parameters under consideration. These hypotheses allow their authors to solve theoretically the DNA evolution problem by computing directly the successive powers of their mutation matrix. However, one can wonder whether such restrictions on the mutation rates are realistic.

Focusing on this question, we have used in [BGP12a] a recent research work of Lang and Murray [LM08], in which the per-base-pair mutation rates of the Yeast *Saccharomyces cerevisiae* have been experimentally measured. The results of [LM08], which are summarized in Table 9.1, as led in [BGP12a] to the following mutation matrix for gene $ura3$:

$$\begin{pmatrix} 1-m & \frac{6m}{14} & 0 & \frac{8m}{14} \\ \frac{40m}{67} & 1-m & \frac{11m}{67} & \frac{16m}{67} \\ \frac{26m}{63} & \frac{9m}{63} & 1-m & \frac{28m}{63} \\ \frac{14m}{23} & \frac{4m}{23} & \frac{5m}{23} & 1-m \end{pmatrix},$$

where m is the mutation rate per generation in $ura3$ gene, which is equal to 3.80×10^{-10}/bp/generation, or to 3.0552×10^{-7}/generation for the whole gene [LM08]. Similarly, the mutation matrix for $can1$ gene can be computed, which is equal to:

$$\begin{pmatrix} 1-m & \frac{5m}{10} & \frac{m}{10} & \frac{4m}{10} \\ \frac{21m}{50} & 1-m & \frac{9m}{50} & \frac{20m}{50} \\ \frac{40m}{72} & \frac{12m}{72} & 1-m & \frac{20m}{72} \\ \frac{9m}{18} & \frac{4m}{18} & \frac{5m}{18} & 1-m \end{pmatrix},$$

Mutation	$ura3$	$CAN1$
$T \to C$	4	4
$T \to A$	14	9
$T \to G$	5	5
$C \to T$	16	20
$C \to A$	40	21
$C \to G$	11	9
$A \to T$	8	4
$A \to C$	6	5
$A \to G$	0	1
$G \to T$	28	20
$G \to C$	9	12
$G \to A$	26	40
Transitions	46	65
Transversions	121	85

Table 9.1: Summary of sequenced $ura3$ and $can1$ mutations [LM08]

with $m = 6.44 \times 10^{-10}$/bp/generation, or 1.1418×10^{-6}/generation for the whole $can1$ gene.

We thus have deduced, in the Third International Conference on Computational Systems-Biology and Bioinformatics (CSBio 2012 [BGP12a]), that none of the existing genomes evolution models can fit such mutation matrices. This deduction leads to the fact that hypotheses must be relaxed, even if this relaxation implies less ambitious models: current models do not match with what really occurs in concrete genomes, at least in the case of this yeast. Having these considerations in mind, the data obtained by Lang and Murray have been used in [BGP12a, BGP12b] and further deepened in [BGPb], in order to predict the evolution of the rates or purines and pyrimidines in the particular case of $ura3$. Mathematical investigations and numerical simulations have been proposed, focusing on this particular gene and its associated matrix of size 2×2 (purines *vs.* pyrimidines), and of size 3×3 (cytosines and thymines compared to purines). [BGP12a, BGP12b] focus on two particular matrices, while the extension [BGP12a] investigates systematically all the possible mutation matrices of sizes 2×2 and 3×3. These research works are summed up in this section.

9.2.2/ NON-SYMMETRIC MODEL OF SIZE 2×2

In this section, our first general genome evolution model focusing on two populations of interest is recalled [BGP12a, BGP12b]. These two populations can be “purines and pyrimidines”, “cytosine and other nucleotides”, or “stop codons and other codons” for instance. This first model has been introduced in [BGP12a, BGP12b] to illustrate the generality of the proposed method, and as a pattern for further investigations. It is applied to the case of purines versus pyrimidines rates in the yeast *Saccharomyces cerevisiae*.

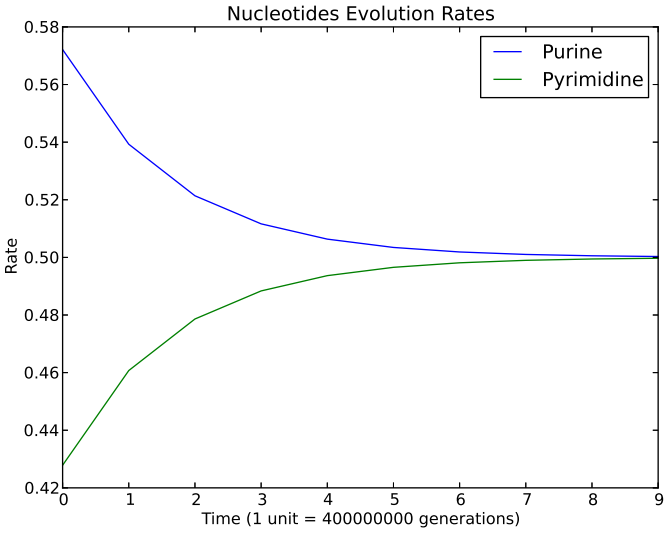


Figure 9.1: Prediction of purine/pyrimidine evolution of *ura3* gene in symmetric Cantor model.

9.2.2.1/ THEORETICAL STUDY

Let X and Y denote respectively the occurrence frequency of the two populations of interest in a biological sequence (nucleotides, trinucleotides, etc.), and $M = \begin{pmatrix} a & b \\ c & d \end{pmatrix}$ the associated mutation matrix, with $a = P(X \to X)$, $b = P(X \to Y)$, $c = P(Y \to X)$, and $d = P(Y \to Y)$ satisfying

$$\begin{cases} a + b = 1, \\ c + d = 1, \end{cases} \tag{9.2}$$

and thus $M = \begin{pmatrix} a & 1-a \\ c & 1-c \end{pmatrix}$.

The initial probability is denoted by $P_0 = (X_0 \;\; Y_0)$, where X_0 and Y_0 denote respectively the initial frequency of the two populations. So the occurrence probability at generation n is $P_n = P_0 M^n$, where $P_n = (X(n) \;\; Y(n))$ is a probability vector such that $X(n)$ (resp. $Y(n)$) is the rate of the first (resp. second) population after n generations.

We have proven in [BGPb] the following result.

Theorem [15]. *Consider a DNA sequence under evolution, whose mutation matrix is* $M = \begin{pmatrix} a & 1-a \\ c & 1-c \end{pmatrix}$ *with* $a = P(X \to X)$ *and* $c = P(Y \to X)$.

- *If* $a = 1, c = 0$*, then the frequencies of* X *and* Y *do not change as the generation pass.*
- *If* $a = 0, c = 1$*, then these frequencies oscillate at each generation between* $(X_0 \;\; Y_0)$ *(even generations) and* $(Y_0 \;\; X_0)$ *(odd generations).*
- *Else the value* $P_n = (X(n) \;\; Y(n))$ *of frequencies at generation* n *is convergent to*

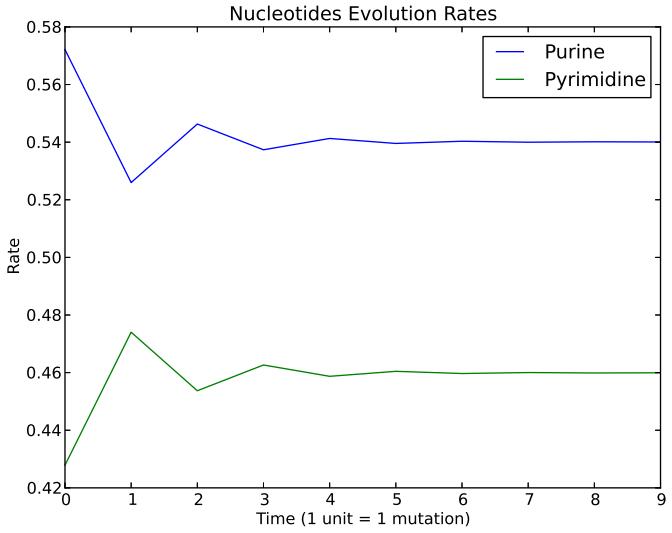


Figure 9.2: Prediction of purine/pyrimidine evolution of *ura3* gene in non-symmetric Model of size 2×2.

the following limit:

$$\lim_{n\to\infty} P_n = \frac{1}{c+1-a} \begin{pmatrix} c & 1-a \end{pmatrix}.$$

Let us finally remark that this theorem encompasses and generalizes the well-known "RY-coding", as used for instance in [PLHP01].

9.2.2.2/ NUMERICAL APPLICATION

For numerical application, we have considered in [BGP12a, BGP12b] mutations rates in the *ura3* gene of the Yeast *Saccharomyces cerevisiae*, as obtained by Gregory I. Lang and Andrew W. Murray [LM08]. As stated before, they have measured phenotypic mutation rates, indicating that the per-base pair mutation rate at *ura3* is equal to $m = 3.0552 \times 10^{-7}$/generation. For the majority of Yeasts they studied, *ura3* is constituted by 804 bp: 133 cytosines, 211 thymines, 246 adenines, and 214 guanines. So $R_0 = \dfrac{246+214}{804} \approx 0.572$, and $Y_0 = \dfrac{133+211}{804} \approx 0.428$. Using these values in the historical model of Jukes and Cantor [JC69], we have obtained the evolution depicted in Figure 9.1. Theorem 15, for its part, has allowed us to compute the limit of the rates of purines and pyrimidines [BGP12a, BGP12b]:

- **Computation of probability a:** $a = P(R \to R) = (1-m)+P(A \to G)+P(G \to A)$. The use of Table 9.1 implies that $a = (1-m) + m\left(\dfrac{0+26}{x}\right)$, where x is such that $1 - a = P(R \to Y) = m\left(\dfrac{6+8+28+9}{x}\right)$, *i.e.*, $x = 77$, and so $a = 1 - \dfrac{51m}{77}$.
- **Computation of probability c:** Similarly, $c = P(Y \to R) = P(C \to A) + P(C \to G) + P(T \to A) + P(T \to G) = \dfrac{70m}{y}$, whereas $1 - c = 1 - m + \dfrac{20m}{y}$. So $c = \dfrac{7m}{9}$.

The purine/pyrimidine mutation matrix that corresponds to the data of [LM08] is thus equal to [BGP12a, BGP12b]:

$$M = \begin{pmatrix} 1 - \frac{51m}{77} & \frac{51m}{77} \\ \frac{7m}{9} & 1 - \frac{7m}{9} \end{pmatrix}.$$

Using the value of m for the $ura3$ gene leads to $1 - a = 2.02357 \times 10^{-7}$ and $c = 2.37627 \times 10^{-7}$, which can be used in Theorem 15 to conclude that the rate of pyrimidines is convergent to 45.992% whereas the rate of purines converge to 54.008%. Numerical simulations using data published in [LM08] are given in Figure 9.2, leading to a similar conclusion [BGP12a, BGP12b].

9.2.3/ A FIRST NON-SYMMETRIC GENOMES EVOLUTION MODEL OF SIZE 3×3 HAVING 6 PARAMETERS

In order to investigate the evolution of the frequencies of cytosines and thymines in the gene $ura3$, a model of size 3×3 compatible with real mutation rates of the yeast *Saccharomyces cerevisiae* has been presented in [BGPb].

9.2.3.1/ FORMALIZATION

Let us consider a line of yeasts where a given gene is sequenced at each generation, in order to clarify explanations. The $n-$th generation is obtained at time n, and the rates of purines, cytosines, and thymines at time n are respectively denoted by $P_R(n), P_C(n)$, and $P_T(n)$.

Let a be the probability that a purine is changed into a cytosine between two generations, that is: $a = P(R \to C)$. Similarly, denote by b, c, d, e, f the respective probabilities: $P(R \to T)$, $P(C \to R)$, $P(C \to T)$, $P(T \to R)$, and $P(T \to C)$. Contrary to what is often required, $P(R \to C)$ is not supposed to be equal to $P(C \to R)$, and the same statement holds for the other probabilities. For the sake of simplicity, we will consider in [BGPb] that a, b, c, d, e, f are not time dependent. Let

$$M = \begin{pmatrix} 1-a-b & a & b \\ c & 1-c-d & d \\ e & f & 1-e-f \end{pmatrix}$$

be the mutation matrix associated to the probabilities mentioned above, and P_n the vector of occurrence, at time n, of each of the three kind of nucleotides. In other words, $P_n = (P_R(n) \;\; P_C(n) \;\; P_T(n))$. Under that hypothesis, P_n is a probability vector: $\forall n \in \mathbb{N}$,

- $P_R(n), P_C(n), P_T(n) \in [0, 1]$,
- $P_R(n) + P_C(n) + P_T(n) = 1$,

Let $P_0 = (P_R(0) \;\; P_C(0) \;\; P_T(0)) \in [0, 1]^3$ be the initial probability vector. We have obviously:

$$P_R(n+1) = P_R(n)P(R \to R) + P_C(n)P(C \to R) + P_T(n)P(T \to R).$$

Similarly, $P_C(n+1) = P_R(n)P(R \to C) + P_C(n)P(C \to C) + P_T(n)P(T \to C)$ and $P_T(n+1) = P_R(n)P(R \to T) + P_C(n)P(C \to T) + P_T(n)P(T \to T)$. This equality yields the following one,

$$P_n = P_{n-1}M = P_0M^n. \tag{9.3}$$

In [BGPb], we have wondered if, given the parameters a, b, c, d, e, f as in [LM08], one can determine the frequency of occurrence of any of the three kind of nucleotides when n is sufficiently large, in other words if the limit of P_n is accessible by computations.

9.2.3.2/ RESOLUTION

Determination of M^n in the general case. The characteristic polynomial of M is equal to [BGPb]

$$\chi_M(x) \quad = (x-1)\left(x^2 + (s-2)x + (1-s+p)\right),$$

where

$$s = a+b+c+d+e+f,$$
$$p = ad+ae+af+bc+bd+bf+ce+cf+de,$$
$$det(M) = 1-s+p.$$

The discriminant of the polynomial of degree 2 in the factorization of χ_M is equal to $\Delta = (s-2)^2 - 4(1-s-p) = s^2 - 4p$. Let x_1 and x_2 the two roots (potentially complex or equal) of χ_M, given by

$$x_1 = \frac{-s+2-\sqrt{s^2-4p}}{2} \text{ and } x_2 = \frac{-s+2+\sqrt{s^2-4p}}{2}. \tag{9.4}$$

Let $n \in \mathbb{N}, n \geqslant 2$. As χ_M is a polynomial of degree 3, a division algorithm of X^n by $\chi_M(X)$ leads to the existence and uniqueness of two polynomials Q_n and R_n, such that

$$X^n = Q_n(X)\chi_2(X) + R_n(X), \tag{9.5}$$

where the degree of R_n is lower than or equal to the degree of χ_M, *i.e.*, $R_n(X) = a_nX^2 + b_nX + c_n$ with $a_n, b_n, c_n \in \mathbb{R}$ for every $n \in \mathbb{N}$. By evaluating (9.5) in the three roots of χ_M, we find the system

$$\begin{cases} 1 & = a_n + b_n + c_n \\ x_1^n & = a_nx_1^2 + b_nx_1 + c_n \\ x_2^n & = a_nx_2^2 + b_nx_2 + c_n \end{cases}$$

This system is equivalent to

$$\begin{cases} c_n & + & b_n & + & a_n & = 1 \\ & & b_n(x_1-1) & + & a_n(x_1^2-1) & = x_1^n - 1 \\ & & b_n(x_2-1) & + & a_n(x_2^2-1) & = x_2^n - 1 \end{cases}$$

If we suppose that $x_1 \neq 1$, $x_2 \neq 1$, and $x_1 \neq x_2$, then standard algebraic computations have led in [BGPb] to

$$\begin{cases} a_n = \dfrac{1}{x_2-x_1}\left[\dfrac{x_2^n-1}{x_2-1} - \dfrac{x_1^n-1}{x_1-1}\right], \\ \\ b_n = \dfrac{x_1+1}{x_1-x_2}\dfrac{x_2^n-1}{x_2-1} + \dfrac{x_2+1}{x_2-x_1}\dfrac{x_1^n-1}{x_1-1}, \\ \\ c_n = 1-a_n-b_n. \end{cases}$$

Using for $i = 1, 2$ and $n \in \mathbb{N}$ the following notation,

$$X_i(n) = \frac{x_i^n - 1}{x_i - 1}, \tag{9.6}$$

and since $x_2 - x_1 = \sqrt{\Delta}$, the system above can be rewritten as [BGPb]

$$\begin{cases} a_n = \dfrac{X_2(n) - X_1(n)}{\sqrt{\Delta}}, \\ \\ b_n = \dfrac{(x_2+1)X_1(n) - (x_1+1)X_2(n)}{\sqrt{\Delta}}, \\ \\ c_n = 1 + \dfrac{x_1X_2(n) - x_2X_1(n)}{\sqrt{\Delta}}. \end{cases} \tag{9.7}$$

By evaluating (9.5) in M and due to the theorem of Cayley-Hamilton, we finally have for every integer $n \geqslant 1$,

$$M^n = a_nM^2 + b_nM + c_nI_3, \tag{9.8}$$

where I_3 is the identity matrix of size 3, a_n, b_n, and c_n are given by (9.7), and M^2 is given by [BGPb]

$$M^2 = \left(\begin{array}{c|c|c} \begin{array}{c} a^2 + 2ab + ac - 2a \\ +b^2 + be - 2b + 1 \end{array} & \begin{array}{c} -a^2 - ab - ac \\ -ad + 2a + bf \end{array} & \begin{array}{c} -ab + ad - b^2 \\ -be - bf + 2b \end{array} \\ \hline \begin{array}{c} -ac - bc - c^2 \\ -cd + 2c + de \end{array} & \begin{array}{c} ac + c^2 + 2cd - 2c \\ +d^2 + df - 2d + 1 \end{array} & \begin{array}{c} bc - cd - d^2 \\ -de - df + 2d \end{array} \\ \hline \begin{array}{c} -ae - be + cf \\ -e^2 - ef + 2e \end{array} & \begin{array}{c} ae - cf - df \\ -ef - f^2 + 2f \end{array} & \begin{array}{c} be + df + e^2 + 2ef \\ -2e + f^2 - 2f + 1 \end{array} \end{array}\right).$$

Determination of M^n in particular situations. Formulations of (9.7) only hold for $x_1 \neq x_2$, $x_1 \neq 1$, and $x_2 \neq 1$. We then have investigated in [BGPb] these latter cases.

Preliminaries. Let us firstly remark that, as the mutation matrix M is stochastic, we have necessarily $0 \leqslant a + b \leqslant 1$, $0 \leqslant c + d \leqslant 1$, and $0 \leqslant e + f \leqslant 1$. These inequalities imply that $s \in [0, 3]$. Consequently from the definition of p one can check that $p = ad + a(e + f) + b(c + d) + bf + c(e + f) + de \leqslant ad + a + b + bf + c + de \leqslant s$, as each parameter is in $[0, 1]$. To sum up,

$$0 \leqslant p \leqslant s \leqslant 3. \tag{9.9}$$

Suppose now that $\Delta \geqslant 0$. Then (9.4) and (9.9) imply that [BGPb]

$$x_1 = \frac{-s + 2 - \sqrt{\Delta}}{2} \in [-2; 1], x_2 = \frac{-s + 2 + \sqrt{\Delta}}{2} \in \left[-\frac{1}{2}; \frac{5}{2}\right]. \tag{9.10}$$

Suppose that $x_1 = 1$. Then $-s = \sqrt{s^2 - 4p} \iff s = p = 0$. So $a = b = c = d = e = f = 0$, and the mutation matrix is equal to the identity of size 3. Conversely, if $a = b = c = d = e = f = 0$, then $x_1 = 1$.

In that situation, the system does not evolve [BGPb].

Suppose that $x_2 = 1$ (and $x_1 \neq 1$). Then $s = \sqrt{s^2 - 4p} \iff p = 0$. In that situation, $x_1 = 1 - s$. Let us consider (9.5) another time: $X^n = Q_n(X)\chi_2(X) + a_nX^2 + b_nX + c_n$. 1 is root of multiplicity 2 of χ_2, whereas $x_1 = 1 - s$ is its third root. As the case $x_1 = 1$ has already been regarded, we can consider that $s \neq 0$. These facts lead to the following system:

$$\begin{cases} 1 &= a_n + b_n + c_n, \\ n &= 2a_n + b_n, \\ (1-s)^n &= (1-s)^2 a_n + (1-s)b_n + c_n. \end{cases}$$

Standard computations, not detailed here, have led us to the following formula [BGPb]:

$$\begin{cases} a_n = \dfrac{-1 + sn + (1-s)^n}{s^2}, \\ b_n = \dfrac{(3-s) + (s^2 - 2s)n + (s-3)(1-s)^n}{s}, \\ c_n = \dfrac{(s-1)(2s-1) - s(s-1)^2 n - (s^2 - 3s + 1)(1-s)^n}{s^2}. \end{cases} \tag{9.11}$$

Case $x_1 = x_2 \neq 1$ ($\Delta = 0$). Then (9.10) implies that $x_1 = 1 - s/2 \in \left[-\frac{1}{2}, 1\right)$. From a differentiation of (9.5) one deduces that x_1 satisfies the following system for every $n \in \mathbb{N}^*$,

$$\begin{cases} 1 &= a_n + b_n + c_n \\ x_1^n &= a_n x_1^2 + b_n x_1 + c_n \\ n x_1^{n-1} &= 2a_n x_1 + b_n \end{cases}$$

Standard algebraic computations, detailed in [BGPb], give, since $x_1 \neq 1$,

$$\begin{cases} a_n = n\dfrac{x_1^{n-1}}{x_1 - 1} - \dfrac{X_1(n)}{x_1 - 1} \\ b_n = X_1(n) - a_n(x_1 + 1) \\ c_n = 1 - a_n - b_n \end{cases} \tag{9.12}$$

where $X_1(n)$ is defined in (9.6).

9.2.3.3/ CONVERGENCE STUDY

Convergence study in the general case We suppose in this section that $x_1 \neq x_2$, $x_1 \neq 1$, and $x_2 \neq 1$. So formulations of (9.7) hold for a_n, b_n, and c_n. We have split the study convergence in several sub-cases in [BGPb]. All obtained results are recalled here without proof.

Theorem 16. *Suppose that $|x_1| < 1$ and $|x_2| < 1$. Then the frequencies $P_R(n), P_C(n)$, and $P_T(n)$ of occurrence at time n of purines, cytosines, and thymines in the considered gene, converge to the following values:*

- $P_R(n) \longrightarrow \dfrac{ce + cf + de + (df - bf)P_R(0)}{p - bf + df}$

- $P_C(n) \longrightarrow \dfrac{ae + af + df + (df - bf)P_C(0)}{p - bf + df}$
- $P_T(n) \longrightarrow \dfrac{ad + bc + bd + (df - bf)P_T(0)}{p - bf + df}$

Theorem [17]. *Suppose that $|x_1| > 1$ and $|x_2| < 1$, where x_1 and x_2 are given by (9.4). Then the evolutionary model is well formulated if and only if $-M^2 + (x_2 + 1)M - x_2 I_3 = 0$. In that case, we have*

- $P_R(n) \longrightarrow \dfrac{(1 - a - b - x_2)P_R(0) + cP_C(0) + eP_T(0)}{1 - x_2}$,
- $P_C(n) \longrightarrow \dfrac{aP_R(0) + (1 - c - d - x_2)P_C(0) + fP_T(0)}{1 - x_2}$,
- *and* $P_T(n) \longrightarrow \dfrac{bP_R(0) + dP_C(0) + (1 - e - f - x_2)P_T(0)}{1 - x_2}$.

Theorem [18]. *Suppose that $|x_1| < 1$ and $|x_2| > 1$, where x_1 and x_2 are given by (9.4). Then the evolutionary model is well formulated if and only if $M^2 - (x_1 + 1)M + x_1 I_3 = 0_3$. In that case, we have $(P_R(n)\ \ P_C(n)\ \ P_T(n)) \longrightarrow (P_R(0)\ \ P_C(0)\ \ P_T(0)) \times M_\infty$.*

Theorem [19]. *Suppose that $|x_1| > 1$ and $|x_2| > 1$, where x_1 and x_2 are given by (9.4). Then the evolutionary model does not evolve in time : $P_n = P_0$ for every $n \in \mathbb{N}$.*

Theorem [20]. *Using the notations as previously, suppose that $|x_1| = 1, x_1 \neq 1$, and $|x_2| \neq 1$. Then the evolutionary model is not convergent. More precisely, we have:*

- $P_R(2n) = (a^2 + 2ab + ac - 2a + b^2 + be - 2b + 1)P_R(0) + (-a^2 - ab - ac - ad + 2a + bf)P_C(0) + (-ab + ad - b^2 - be - bf + 2b)P_T(0)$,
- $P_R(2n+1) = (1 - a - b)P_R(0) + aP_C(0) + bP_T(0)$,
- $P_C(2n) = (-ac - bc - c^2 - cd + 2c + de)P_R(0) + (ac + c^2 + 2cd - 2c + d^2 + df - 2d + 1)P_C(0) + (bc - cd - d^2 - de - df + 2d)P_T(0)$,
- $P_C(2n+1) = cP_R(0) + (1 - c - d)P_C(0) + dP_T(0)$,
- $P_T(2n) = (-ae - be + cf - e^2 - ef + 2e)P_R(0) + (ae - cf - df - ef - f^2 + 2f)P_C(0) + (be + df + e^2 + 2ef - 2e + f^2 - 2f + 1)P_T(0)$,
- $P_T(2n+1) = eP_R(0) + fP_C(0) + (1 - e - f)P_T(0)$,

Theorem [21]. *If $|x_1| = |x_2|$, but $x_1, x_2 \in \mathbb{C}\backslash\mathbb{R}$, then $(P_R(n)\ \ P_C(n)\ \ P_T(n)) = (P_R(0)\ \ P_C(0)\ \ P_T(0)) \times (a_n M^2 + b_n M + c_n I_3)$, where*

- $a_n = -\dfrac{sin\left(\frac{n\theta}{2}\right) sin\left(\frac{(n-1)\theta}{2}\right)}{sin\left(\frac{\theta}{2}\right) sin(\theta)}$,
- $b_n = \dfrac{2\ sin\left(\frac{n\theta}{2}\right) sin\left(\frac{(n-2)\theta}{2}\right)\ cos\left(\frac{\theta}{2}\right)}{sin(\theta)\ sin\left(\frac{\theta}{2}\right)}$,
- $c_n = 1 - \dfrac{sin\left(\frac{n\theta}{2}\right) sin\left(\frac{(n-3)\theta}{2}\right)}{sin(\theta)\ sin\left(\frac{\theta}{2}\right)}$.

with $e^{-i\theta} = x_1$.

Convergence study in particular situations The case where $x_1 = 1$ has already been discussed, it implies that $a = b = c = d = e = f = 0$, and so the system does not evolve. Let us investigate the other particular situations.

Theorem 22. *Using the same notations as above, suppose that $p = 0$ (or $x_2 = 1$, which is equivalent). Then the system is well formulated if and only if $M^2 + s(s-2)M - (s-1)^2 I_3 \neq 0$. In that situation, we have:*

- *either $s \in]0, 2[$, and so $(P_R(n)\ P_C(n)\ P_T(n)) \longrightarrow (P_R(0)\ P_C(0)\ P_T(0)) \times \frac{1}{s^2}[-M^2 + s(3-s)M + (s-1)(2s-1)I_3]$.*
- *or $s = 2$, and so $(P_R(2n)\ P_C(2n)\ P_T(2n)) \longrightarrow (P_R(0)\ P_C(0)\ P_T(0))$ whereas $(P_R(2n+1)\ P_C(2n+1)\ P_T(2n+1)) \longrightarrow (P_R(0)\ P_C(0)\ P_T(0)) \times (-2M^2 + 4M + 2I_3)$.*

Theorem 23. *Using the same notations as previously, suppose that $(a+b+c+d+e+f)^2 = 4(ad + ae + af + bc + bd + bf + ce + cf + de)$.*

Then the probabilities $P_R(n), P_C(n)$, and $P_T(n)$ of occurrence at time n of a purine, cytosine, and thymine on the considered nucleotide, converge to the following values:

- $P_R(n) \longrightarrow \frac{4}{s^2}(ce + cf + de)(P_R(0) + P_C(0) + P_T(0))$,
- $P_C(n) \longrightarrow \frac{4}{s^2}(ae + af + bf)(P_R(0) + P_C(0) + P_T(0))$,
- $P_T(n) \longrightarrow \frac{4}{s^2}(ad + bc + bd)(P_R(0) + P_C(0) + P_T(0))$.

9.2.4/ APPLICATION IN CONCRETE GENOMES PREDICTION

We have considered another time in [BGP12a, BGP12b] the numerical values for mutations published in [LM08]. Gene *ura3* of the Yeast *Saccharomyces cerevisiae* has a mutation rate of 3.80×10^{-10}/bp/generation [LM08]. As this gene is constituted by 804 nucleotides, we can deduce that its global mutation rate per generation is equal to $m = 3.80 \times 10^{-10} \times 804 = 3.0552 \times 10^{-7}$. Let us compute the values of a, b, c, d, e, and f. The first line of the mutation matrix is constituted by $1-a-b = P(R \to R)$, $a = P(R \to T)$, and $b = P(R \to C)$. $P(R \to R)$ takes into account the fact that a purine can either be preserved (no mutation, probability $1 - m$), or mutate into another purine ($A \to G$, $G \to A$). As the generations pass, authors of [LM08] have counted 0 mutations of kind $A \to G$, and 26 mutations of kind $G \to A$. Similarly, there were 28 mutations $G \to T$ and 8: $A \to T$, so 36: $R \to T$. Finally, 6: $A \to C$ and 9: $G \to C$ lead to 15: $R \to C$ mutations. The total of mutations to consider when evaluating the first line is so equal to 77. All these considerations lead to the fact that $1 - a - b = (1 - m) + m\frac{26}{77}$, $a = \frac{36m}{77}$, and $b = \frac{15m}{77}$. A similar reasoning leads to $c = \frac{19m}{23}$, $d = \frac{4m}{23}$, $e = \frac{51m}{67}$, and $f = \frac{16m}{67}$.

In that situation, $s = a + b + c + d + e + f = \frac{205m}{77} \approx 8.134 \times 10^{-7}$, and $p = \frac{207488m^2}{118657} \approx 1.632 \times 10^{-13}$. So $\Delta = s^2 - 4p = \frac{854221m^2}{9136589} > 0$, $x_1 = 1 - \frac{m}{2}\left(\frac{205}{77} + \sqrt{\frac{854221}{9136589}}\right)$, and

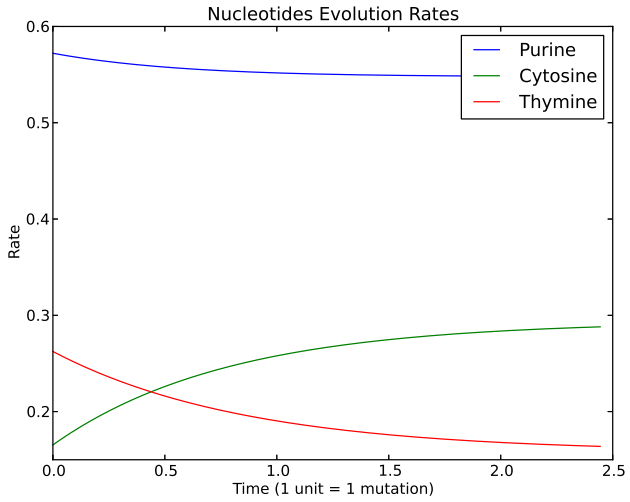


Figure 9.3: Prediction of evolution concerning the purine, thymine, and cytosine rates in *ura3*. Non-symmetric Model of size 3×3.

$x_2 = 1 - \frac{m}{2}\left(\frac{205}{77} - \sqrt{\frac{854221}{9136589}}\right)$. As $x_1 \approx 0.9999685 \in [0,1]$ and $x_2 \approx 0.9999686 \in [0,1]$, we have, due to Theorem 16:

- $P_R(n) \longrightarrow \dfrac{ce + cf + de + (df - bf)P_R(0)}{p - bf + df}$
- $P_C(n) \longrightarrow \dfrac{ae + af + df + (df - bf)P_C(0)}{p - bf + df}$
- $P_T(n) \longrightarrow \dfrac{ad + bc + bd + (df - bf)P_T(0)}{p - bf + df}$

Using the data of [LM08], we have found in [BGP12a, BGP12b] that $P_R(0) = \frac{460}{804} \approx 0.572$, $P_C(0) = \frac{133}{804} \approx 0.165$, and $P_T(0) = \frac{211}{804} \approx 0.263$. So $P_R(n) \longrightarrow 0.549$, $P_C(n) \longrightarrow 0.292$, and $P_T(n) \longrightarrow 0.159$. Simulations corresponding to this example are given in Fig. 9.3, they confirm these values.

9.3/ STUDYING THE TRANSPOSABLE ELEMENTS

We have continued to investigate the different ways the genomes evolve over time, which has led to first results regarding the (retro)transposition of transposable elements. This work in progress is realized with colleagues of the Laboratoire de Mathématique de Besançon (Alexeï Lozinsky, Romain Biard, and Landy Rabehasaina) and of Chrono-environnement (Antone Perasso).

9.3.1/ INTRODUCTION

A transposable element (TE, transposon or retrotransposon) is a DNA sequence that can change its position within the genome. TEs are thus mobile (they represent one of several types of mobile genetic elements), self-replicating, moderately repeated, and ubiquitous DNA sequences. They are powerful mutators by inserting themselves into genes and their regulatory regions, and by promoting chromosomal rearrangements, as they constitute a high proportion of genomes: they represent 40% of the human genome, 15% of the genome of Drosophila melanogaster, and up to 95% of the genome in some plants. These elements have thus played a significant role in species evolution and population adaptation.

TEs are assigned to one of two classes according to their mechanism of transposition, which can be described as either copy and paste (class I TEs) or cut and paste (class II TEs).

- **Retrotransposons (class I):** These elements, which are quite similar to retroviruses like HIV, are copied in three stages: (1) they are transcribed from DNA to RNA, and (2) the RNA produced is then reverse transcribed to DNA. (3) This copied DNA is finally inserted at a new position into the genome. Retrotransposons are commonly grouped into three main orders:
  - TEs with long terminal repeats (LTRs): like retroviruses, these elements encode their own reverse transcriptase, which is used in the reverse transcription step (stage number 2)
  - Long interspeared elements (LINEs): they encode reverse transcriptase, lack LTRs, and are transcribed by RNA polymerase II
  - Short interspeared elements (SINEs), for their part, do not encode reverse transcriptase and are transcribed by RNA polymerase III.
- **DNA transposons (class II):** These elements follow a cut-and-paste transposition mechanism, which does not involve a RNA intermediate.

Not all DNA transposons transpose through the cut-and-paste mechanism. In some cases, defined sometimes as a third class of TEs, a replicative transposition is observed in which a transposon replicates itself to a new target site, which occur for instance for the Helitron element.

9.3.2/ A FIRST PDE MODEL FOR TRANSPOSITION

9.3.2.1/ THEORETICAL FOUNDATIONS

In our mind, there was a strong link to discover between the transposition dynamics of TEs and partial differential equations like transport equations. This is why we have asked Alexeï Lozinski to model the evolution of TEs due to the aforementioned two kinds of modifications, namely the cut-and-paste and the copy-and-paste ones, and we currently develop algorithmic methods to provide relevant initial conditions and parameters to the obtained PDEs. For the sake of completeness, and as the establishment of these PDEs

shows the data to obtain computationally, we recall here this modeling, principally obtained by my colleague (helped by me).

In both cases, we assume that the elements' distribution at any time can be well described by a density function $\rho(t,x)$. In other words, the number of elements between positions a and b, $0 \leqslant a < b \leqslant 1$, at time t is supposed to be equal to $\int_a^b \rho(t,x)dx$. We assume that transpositions occur with rate λ in time, *i.e.*, the probability that a given element will be transposed during a time interval $(t, t+\Delta t)$ is $\lambda\Delta t + o(\Delta t)$. The probability density to jump from x to y at time t is denoted by $p(x,y)$. Remark that, in the cut-and-paste situation, the element moves from x to y, whereas in the copy-and-paste one, the element at x still continues to be in x while an additional copy is obtained in y.

Another important phenomenon to consider in the model is the death of TEs. We assume that it happens with rate γ in time, *i.e.*, the probability that a given element will be destroyed during a time interval $(t, t+\Delta t)$ is $\gamma\Delta t + o(\Delta t)$. The parameters λ, γ, and the function $p(x,y)$ must be obtained from experimental data. We now write the equations for ρ.

The cut-and-paste regime. Let us consider two real numbers $0 \leqslant a < b \leqslant 1$ and count the transposons between a and b at times t and $t+\Delta t$ respectively. When going from t and $t+\Delta t$, the elements can (1) move outside the interval, (2) come from the outside, or (3) die. Taking into account these 3 mechanisms for transposons gives:

$$\begin{aligned}\int_a^b \rho(t+\Delta t,x)dx = \int_a^b \rho(t,x)dx + \lambda\Delta t \Bigg[&-\int_{(0,1)\setminus(a,b)} \int_a^b p(x,y)\rho(t,x)dxdy \\ &+\int_a^b \int_{(0,1)\setminus(a,b)} p(x,y)\rho(t,x)dxdy \Bigg] - \gamma\Delta t \int_a^b \rho(t,x)dx.\end{aligned}$$

This equation can be rewritten as follows:

$$\int_a^b \frac{\rho(t+\Delta t,x)-\rho(t,x)}{\Delta t}dx = \lambda\left[-\int_a^b \rho(t,x)dx + \int_a^b \int_0^1 p(y,x)\rho(t,y)dydx\right] - \gamma\int_a^b \rho(t,x)dx.$$

Taking the limit $\Delta t \to 0$ and observing that a and b are arbitrary, we obtain equation for ρ (transposons case):

$$\frac{\partial\rho}{\partial t}(t,x) = \lambda\left[\int_0^1 p(y,x)\rho(t,y)dy - \rho(t,x)\right] - \gamma\rho(t,x). \tag{9.13}$$

Let us now reobtain the usual transport equation from the above one. For this, we should further assume that when a transposon is cut-and-pasted from position x, it arrives with an high probability to position $x+a(x)$, where the jump function $a(x)$ must also be determined experimentally. More specifically, we assume that,

$$p(x,y) = \delta_{x+a(x)}(y) + O(a^2), \tag{9.14}$$

where $O(a^2)$ means some term of order of $(\max a)^2$. We thus substitute this form for the probability inside the equation above, multiply it by any test function ϕ (that vanishes at 0 and 1) and integrate, to finally obtain:

$$\int_0^1 \frac{\partial\rho}{\partial t}(t,x)\phi(x)dx = \int_0^1 (-\lambda(\rho(t,x)a(x))' - \gamma\rho(t,x))\phi(x)dx + O(\lambda a^2).$$

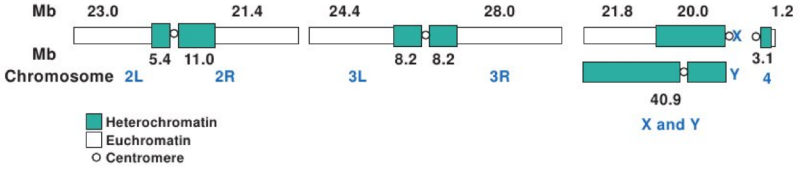


Fig. 1. Mitotic chromosomes of *D. melanogaster*, showing euchromatic regions, heterochromatic regions, and centromeres. Arms of the autosomes are designated 2L, 2R, 3L, 3R, and 4. The euchromatic length in megabases is derived from the sequence analysis. The heterochromatic lengths are estimated from direct measurements of mitotic chromosome lengths (*67*). The heterochromatic block of the X chromosome is polymorphic among stocks and varies from one-third to one-half of the length of the mitotic chromosome. The Y chromosome is nearly entirely heterochromatic.

Figure 9.4: *Drosophila melanogaster* chromosomes

Hence, if we neglect the terms of order λa^2, the density should satisfy:

$$\frac{\partial \rho}{\partial t} + \frac{\partial(\lambda a \rho)}{\partial x} = -\gamma \rho. \tag{9.15}$$

Note however that hypothesis on $p(x, y)$ (Equation (9.14)) can be very false. For example, the element at x may want to go to $x + a$ with probability 1/2 and to $x - a$ also with probability 1/2. Then, the simple transport equation above will not be adequate. The equation that holds under most general assumptions is Eq.(9.13).

The copy-and-paste regime. Consider any $0 \leqslant a < b \leqslant 1$ and count the retrotransposons between a and b at t and $t + \Delta t$. When going from t and $t + \Delta t$, the elements can be (1) added by a copy and paste, or (2) they can be destroyed. Taking into account these two mechanisms gives

$$\int_a^b \rho(t + \Delta t, x)dx = \int_a^b \rho(t, x)dx + \lambda \Delta t \int_a^b \int_0^1 p(y, x)\rho(t, y)dydx - \gamma \Delta t \int_a^b \rho(t, x)dx.$$

Taking the limit $\Delta T \to 0$, and observing that a and b are arbitrary, gives the equation for ρ (density of retrotransposons):

$$\frac{\partial \rho}{\partial t}(t, x) = \lambda \int_0^1 p(y, x)\rho(t, y)dy - \gamma \rho(t, x). \tag{9.16}$$

9.3.2.2/ DATA ACQUISITION: GENERAL APPROACH

In order to use the partial differential equations of (9.16) and (9.15) (or, at least (9.13)), which enable us to predict the spatial and temporal evolution of retrotransposons and transposons respectively, we need:

- The initial condition: current spatial distribution of each type of transposable element in each chromosome of *Drosophila melanogaster* as presented in Figure 9.4.
- The probability law $p(x, y)$ of jumps: the probability that a transposable element in x will later be in y, where x and y are two locations inside the same chromosome.

The species we have considered is the *Drosophila melanogaster* (FlyBase [MLS+13], release 5.51), as it is the most studied species regarding transposable elements.

To obtain an initial condition to the partial differential equations (9.15) and (9.16), we have:

1. divided each half chromosome in twenty parts (fifty parts in a second run of experiments),
2. counted the number of TEs of the considered type in each part, using FlyBase, and plotted histograms
3. interpolated these histograms in two manners, to obtain a continuous function:
    - Lagrange polynomial interpolation minimizing the squared errors at the middle of histograms (degree 50),
    - spline curves.

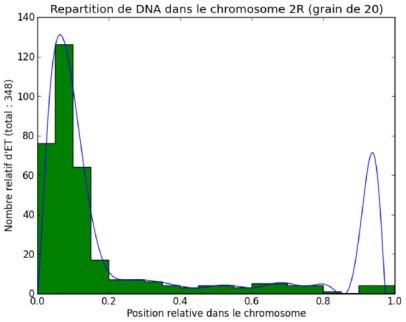
Indeed we can find in FlyBase a list of all the 5409 elements that have been discovered in the four chromosomes of *D.melanogaster*. For each element, the location (which place in which chromosome) is well specified. The sole difficulty with such an approach is that the file provided by FlyBase does not specify if the element is a retrotransposon with LTR, a retrotransposon without LTR, a Helitron, or a DNA transposon. To do so, we have contacted Emmanuelle Lerat (Laboratoire de Biométrie et de Biologie Evolutive, Lyon), who has sent us an Excel file in which the category of each of the 5409 elements is specified: we just needed to automatically cross-check the information given.

Examples of initial conditions are provided in Figures 9.5(a) and 9.5(b) for transposons of chromosome 2R, in Figures 9.6(a) and 9.6(b) for retrotransposons with LTR in chromosome 2L. For the sake of comparison, other interesting obtained results are provided in Appendix G.

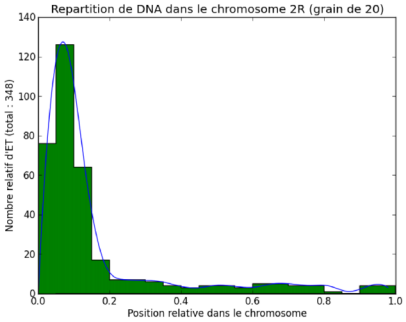
To obtain the probability law $p(x, y)$ of jumps, for each chromosome C and each location x, we have:

- obtained, with FlyBase [MLS+13], the LTR elements that are present in x;
- looked for in C the locations y containing the same retroelement, but less degraded,
    - the reference sequence, for each type of LTR retrotransposon (there are 1321 types of retrotransposons with LTRs: Dm88, Burdock, etc.), is obtained with Repbase Update [JKP+05],
    - the degraded rate between the sequence found and the reference one is obtained by a similarity rate computed from a Smith-Waterman local alignment (thanks to the *water* command of *emboss* package),
    - we thus consider that, when an element is more degraded, this is because it is older;
- plotted 3D histograms (the height of the bar at (x, y) is $p(x, y)$) and 3D interpolated splines.

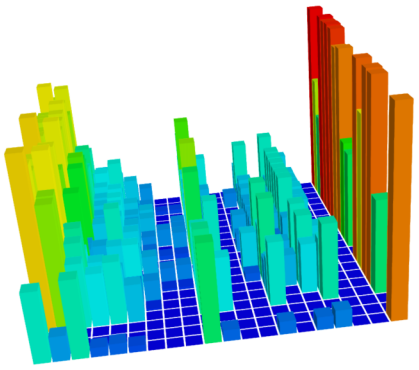
Examples of obtained results are depicted in Figures 9.5 and 9.6.



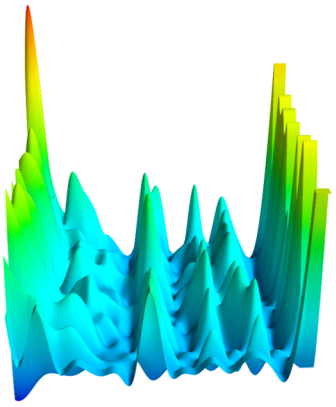
(a) Initial condition, polynomial version



(b) Initial condition, splines version



(c) Histograms of DNA jumps



(d) Splines of DNA jumps

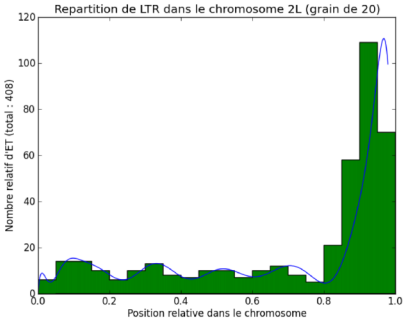
Figure 9.5: Chromosome 2R, DNA elements

9.3.3/ A BRANCHING PROCESS APPROACH

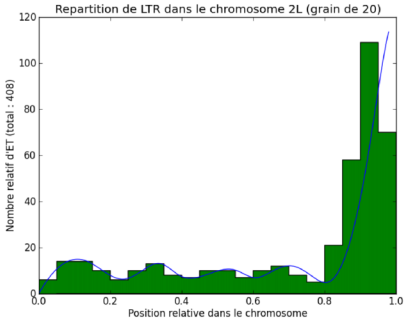
9.3.3.1/ CONCRETE CASE STUDY

We have regarded the cases of LTR, Non-LTR, and Helitrons elements in each chromosome of *D.melanogaster*, divided in either 20 parts or 50 parts. For each element of each kind, and each part of each chromosome, we have considered that this element has arrived at this location at generation $10k$ if it has a similarity score equal to $10k\%$ with the DNA sequence of this element referenced in Repbase Update [JKP+05]. Indeed we have supposed that nucleotide mutations are uniformly distributed in time and space (in the whole genome): the degradation of an element is thus proportional to the duration it has spent in the considered location.

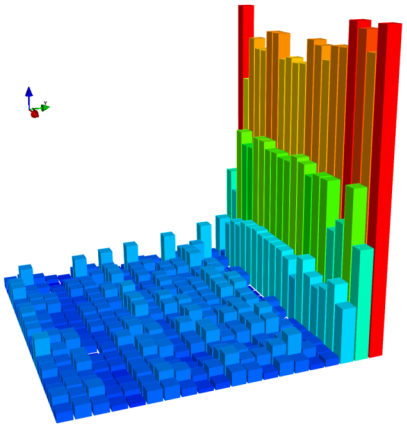
A few obtained results are depicted in Figure 9.7 for Helitron elements in chromosomes 2 and 4. We can see that, in chromosome 2, these elements are concentrated near the centromere while they are more uniformly distributed in time and space in chromosome 4. Abscissa represent 20 time generations (in arbitrary unity) while ordinates are the 20 parts of the considered chromosome. Finally, larger discs are for higher Helitrons concentration.



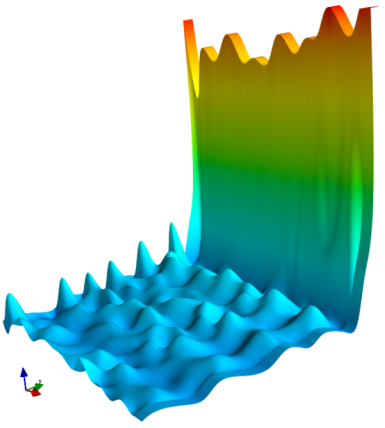
(a) Initial condition, polynomial version



(b) Initial condition, splines version



(c) Histograms of LTR jumps



(d) Splines of LTR jumps

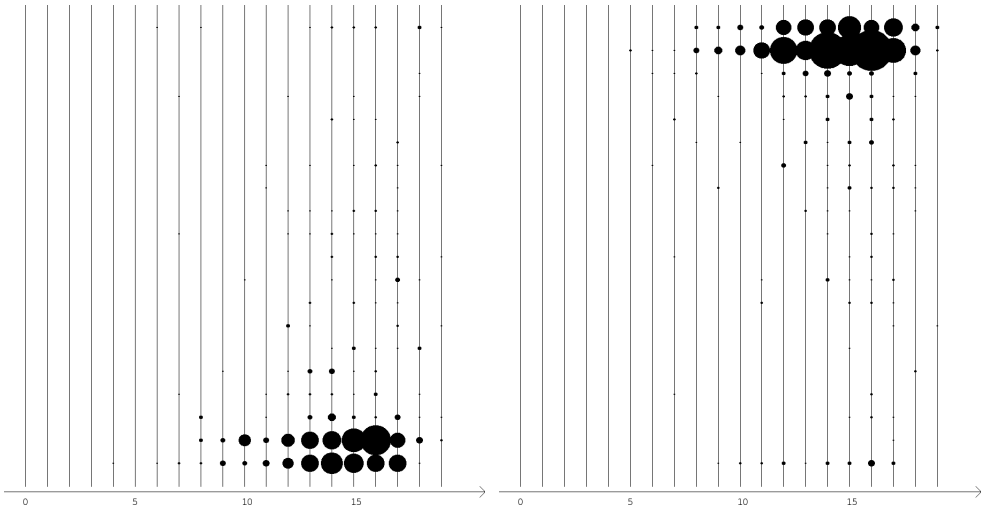
Figure 9.6: Chromosome 2L, LTR elements

9.3.3.2/ THEORETICAL MODELING

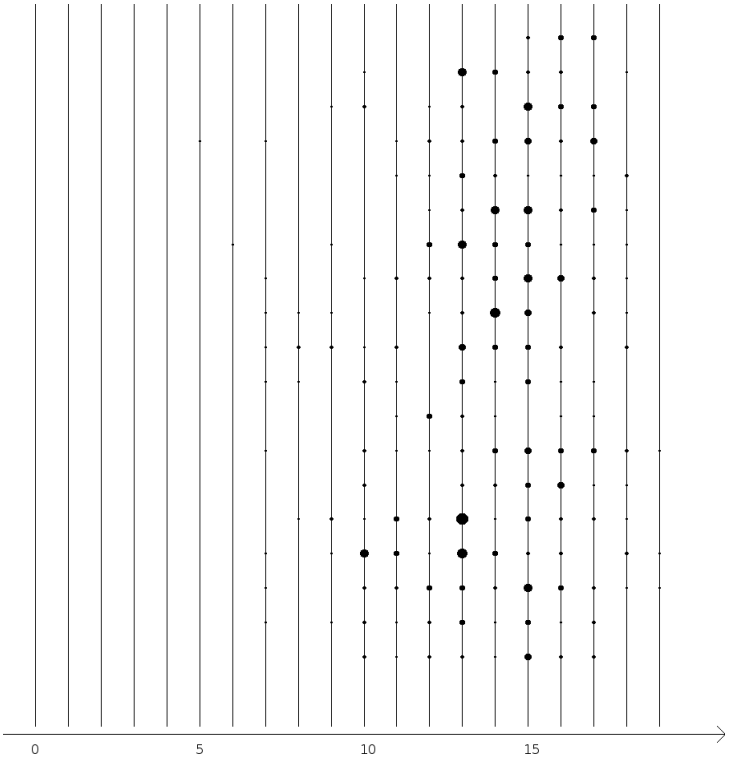
Motivations We can observe in some chromosomes a gaussian-like distribution of transposable elements centered on the centromere. Our idea was that, after the initial insertion of a retrotransposon in a given chromosome, as it has the same probability to be copied at its left or at its right, we finally obtain a gaussian distribution of presence for this element, and centered on its initial position. Then, when several retrotransposons are inserted in the chromosome, with an initial location close to the centromere (it is more likely to be destructed before its first transposition, when its first introduction is in coding sequences), we obtain a superposition of independent normal distributions, all centered near the centromere, which should finally lead to a global gaussian distribution for all the retrotransposons.

We have demanded to probabilistic colleagues of the LMB if such an hypothesis were correct, thus explaining the observed gaussian distribution. The answer has been provided by Romain Biard and Landy Rabehasaina, their modeling is recalled thereafter.

Model We consider a branching random walk on $\mathbb{R}$ of which particles model transposable elements. This process starts with one particle located at the origin at time 0. At each time k, every particle currently in the process dies (0 child) with probability p_0, survives and stays at the same location (1 child, itself) with probability p_1, or survives, stays



(a) Chromosome 2L (b) Chromosome 2R



(c) Chromosome 4

Figure 9.7: Helitron's evolution

at the same location and duplicates at one other random location (2 children, itself and one other) with probability p_2. We use the following notation:

- $V^{(n)}(x) \in \mathbb{R}$ be the position of the individual x belonging to generation n;
- Z_n the number of elements at generation n.

We also assume that the distance from one transposable element to its duplicate is randomly distributed as a common distribution V that we suppose centered and with variance σ_V^2, but independently from the other occurring displacements at a given time. One important aspect of the model is that $(Z_n)_{n\in\mathbb{N}}$ is a Galton Watson process. In particular, if

$$m := p_1 + 2p_2$$

denotes the mean number of offsprings by a transposable element, then one has the following almost sure convergence

$$\frac{Z_n}{m^n} \longrightarrow W, \quad n \to +\infty, \tag{9.17}$$

for some finite random variable W, see e.g. Theorem 1 p.9 of [AN04].

Gaussian framework

Gaussian aymptotic distribution We are interested in the behavior of how transposable elements are located as the number of generation n grows. The key to the asymptotic form of how they are distributed for large n lies in a Central Limit Theorem for branching random walks. It was proved in a refined form by Kaplan [Kap82] and states that

$$\frac{1}{m^n}\text{Card}\left\{x|\ V^{(n)}(x) \leqslant \sqrt{n}y\right\} \longrightarrow W\Phi(y),\ n \to +\infty, \quad \text{almost surely}$$

for all $y \in \mathbb{R}$, where W is defined by (9.17) and Φ is the cumulative distribution function of the $\mathcal{N}(0, \sigma_V^2)$ distribution. This implies that the empirical cumulative distribution of the transposable elements verifies, thanks to (9.17), and on the set of non extinction,

$$\frac{1}{Z_n}\text{Card}\left\{x|\ V^{(n)}(x) \leqslant \sqrt{n}y\right\} \longrightarrow \Phi(y),\ n \to \infty, \tag{9.18}$$

i.e., that one has that TEs have for large n a repartition that is roughly $\mathcal{N}(0, n\sigma_V^2)$ distributed. In the following sections, variance of displacements σ_V^2 is very small compared to the considered number of generations in the study, so that the limiting distribution of transposable elements can be considered $\mathcal{N}\left(0, \sigma_\infty^2\right)$ distributed with $\sigma_\infty^2 := n\sigma_V^2$.

Global gaussian distribution Let us now assume that the position at time 0 of one transposable element is random and also normally distributed with mean 0 (which corresponds to position of the centromere). We are now interested in the distribution of position $V(x)$ of a given element at generation n.

Proposition [22]. *Assume that position of one transposable element at $V(x_0) \sim \mathcal{N}(0, \sigma_0^2)$ and that the distribution of the position of one element x given it is a descendant from x_0 is normally distributed with mean $V(x_0)$ and variance σ_∞^2. Then $V(x) \sim \mathcal{N}\left(0, \sigma_0^2 + \sigma_\infty^2\right)$.*

Proof [6] (Biard & Rabehasaina)**.** *We proceed by proving that the characteristic function of* $V(x)$ *is that of a* $\mathcal{N}\left(0, \sigma_0^2 + \sigma_\infty^2\right)$ *random variable. We indeed have*

$$\begin{aligned}
\mathbb{E}e^{itV(x)} &= \mathbb{E}\left[\mathbb{E}e^{itV(x)}\,|V(x_0)\right] \\
&= \mathbb{E}\left[\exp\left\{iV(x_0)t - \frac{\sigma_\infty^2 t^2}{2}\right\}\right], \\
&= \mathbb{E}\left[\exp\left\{iV(x_0)t\right\}\right]\exp\left\{-\frac{\sigma_\infty^2 t^2}{2}\right\}, \\
&= \exp\left\{-\frac{\sigma_0^2 t^2}{2}\right\}\exp\left\{-\frac{\sigma_\infty^2 t^2}{2}\right\} = \exp\left\{-\frac{(\sigma_0^2 + \sigma_\infty^2)t^2}{2}\right\}.
\end{aligned}$$

V

Conclusion and Annexes

10

CONCLUSION

The study, the modeling, and/or the exploitation of complex discrete dynamics that appear in applications such that bioinformatics or information security, raise numerous questions rich in perspective. Some of them are detailed hereafter.

10.1/ THEORY OF COMPLEX SYSTEMS

The generalization of chaotic iterations (CIs) in iterative systems, proposed in our thesis, must be deeply studied and exploited. The ergodic theory is the natural extension of the mathematical theory of chaos: instead of using a topological description of the dynamics, ergodicity is based on measure theory. With this latter, complex dynamics we study should be described more quantitatively, and we could for instance provide more concrete and measured descriptions of the security that our algorithms exhibit.

Similarly, we are still at the beginning of the exploitation of the complexity theory as a description and evaluation of our complex dynamics. This theory should be more systematically investigated, both on the theory viewpoint and for security proofs. Additionally, CIs is a particular case of so-called “screw products” studied in mathematical topology: at each iterate, a function is picked in a predefined set of applications, and this latter is used to update the current state of the system. Finally, another mathematical community has defined a notion of random dynamical systems. Results taken from these theoretical domains of research could help us to explore and understand complex dynamics that occupy us.

10.2/ SENSOR NETWORKS

The importance of the network’s topology, of sensors scheduling politics, of faults tolerance, and of resistance against attacks... have just been taken into account in our project funded by the Région Franche-Comté regarding prognostics and diagnostics failures in industrial systems. This discovered importance should be proven both theoretically and experimentally, and good practices in this area should be emphasized.

The global approach of security within wireless sensor networks, recently initiated, should be completed and the use of chaos in these networks, proposed by us in the fields of video-surveillance and security, should be generalized and more concerted. Finally, we have started to use epidemiological models and predator-prey equations, for alert

spreading (in WSNs) or data survivability (in unattended WSNs). These promising models should be deepened.

10.3/ INFORMATION SECURITY

Our first research works in the use of complex dynamics for the purpose of information security should be pursued in various directions.

At PRNGs level, links between chaos and statistical quality of the generators should be better understood. Guaranteed results of success or failure for some statistical tests, in presence of well defined topological properties, should be established. Our good results of CIs post-treatment with the vectorial negation should be confirmed by using other functions making the CIs chaotic. Finally, we should better disseminate these good results to ensure that more people will use these generators.

In the field of information hiding, the cryptographic approach for steganography must be pursued and extended, by cryptanalyzing algorithms proposed in the literature and by developing various scenarii of attacks. Similarly, the promising approach of post-treatment on hash functions should be deepened, and other properties of cryptographical security should be proven as preservable, compatible with chaotic iterations.

10.4/ BIOINFORMATICS

We have proposed first models for describing the dynamics of transposable elements (TEs), using partial differential equations and branching processes. An approach using cellular automata has just been initiated and must be deepened. Data have been extracted from the genome of *D.melanogaster* to serve as parameters for these models (in case of copy-and-paste: retrotransposon category of TEs). These models must now be enriched and exploited, to deduce consequences at genetic diseases level and regarding genomes plasticity. These parameters must be found in other species too, and compared with currently obtained ones, to better identify the modification of TEs' dynamics depending on the regarded taxon. Phenomena such as the rapid contamination of P element in *D.melanogaster* must be explained, and the Ping-pong model for this P element must be justified too, either by using a percolation model or a predator-prey one. The presence or absence of long terminal repeat LTR sequences must be taken into account in our dynamical models, and the case of transposons (cut-and-paste) must be equated and studied. Similarly, we must continue to deepen the other operations involved in genomes evolution: mutation matrices must be of size 4 without symmetry, graphical models to infer mutation laws at genes scale must be finalized, whereas the dynamical systems we have written to describe different kind of genomics rearrangements (inversion, duplication, etc.) must be unified and studied. Finally, a numerical simulator integrating these mathematical models and exploiting a basis of knowledge must be realized (it is already started).

At protein folding level, folded self-avoiding walks must be investigated more deeply. Among other things, the shortest unfoldable SAW must be found, the proof of NP-completeness for stretching SAWs must be adapted to the folded ones, and chaos properties found in the folding dynamics must be related to the phenomenon of intrinsically

disordered proteins. Finally, other currently going on projects in bioinformatics raise numerous analysis problems of complex systems, whose solution will help us to better understand and simulate the evolution of genomes.

A

COMPLEMENTS REGARDING OUR PRNG RESEARCH WORK

A.1/ SOME WELL-KNOWN GENERATORS

We first introduce various well-known pseudorandom number generators. They have been used previously in this article, to evaluate the quality of a post-treatment based on chaotic iterations.

A.1.1/ BLUM BLUM SHUB

The Blum Blum Shub generator [BBS86] (usually denoted by BBS) takes the form:

$$x^0 \in [\![1, m-1]\!]$$

$$x^{n+1} = (x^n \times x^n) \; mod \; m, \;\; y^{n+1} = x^{n+1} \; mod \; (log(m)),$$

where m is the product of two prime numbers p and q, such that:

- p and q are congruent to 3 modulus 4
- $gcd(\phi(p-1), \phi(q-1))$ should be small[1]

y^n is the returned sequence, whereas log refers to the logarithm to base 2.

This generator is known to be secure for sufficiently large p and q. However, in this article, we do not focus on security, but on statistical improvement of defective generators: we want to show that deficient PRNGs can be improved using the chaotic iterations post-treatment. A way to find such defective generators is to use good ones like this BBS but in a wrong context (small prime numbers, in this situation).

A.1.2/ THE LOGISTIC MAP

The logistic map is given by:

[1] Euler's totient $\phi(n)$ is an arithmetic function that counts the number of positive integers less than or equal to n that are relatively prime to n.

$$x^{n+1} = \mu\, x^n(1 - x^n), \text{ with } x^0 \in (0, 1),\ \mu \in [0, 4],$$

where x is a real number. The logistic map was originally introduced as a demographic model by Pierre François Verhulst in 1838. In 1947, Ulam and Von Neumann [UN47] studied it as a PRNG. This essentially requires mapping the states of the system $(x^n)_{n\in\mathbb{N}}$ to $\{0, 1\}^{\mathbb{N}}$. A simple way for turning x^n to a discrete bit symbol r is by using a threshold function as it is shown in Algo.2. A second usual way to obtain an integer sequence from a real system is to chop off the leading bits after moving the decimal point of each x to the right, as it is obtained in Algo.3.

Algorithm 2: An arbitrary round of logistic map 1

Input: the internal state x (a decimal number)
Output: r (a 1-bit word)

$x \leftarrow 4x(1 - x)$
if $x < 0$ **then**
    $r \leftarrow 0$;
else
    $r \leftarrow 1$;
return r;

The logistic map is a famous example of Devaney's chaotic dynamical system for $\mu \in (3.99996, 4]$. However, it is statistically biased and its implementation on machines with finite precision raises a lot of problems. In this article, we have used it with the method of Algorithm 2 and a threshold equal to 0.5.

Algorithm 3: An arbitrary round of logistic map 2

Input: the internal state x (a decimal number)
Output: r (an integer)

$x \leftarrow 4x(1 - x)$
$r \leftarrow \lfloor 10000000x \rfloor$
return r;

A.1.3/ Linear Congruential Generator

The linear congruential generator (LCG) is defined by the recurrence:

$$x^0 \in [\![0, m - 1]\!], \quad x^n = (ax^{n-1} + c)\ mod\ m \tag{A.1}$$

where a, c, and x^0 are positive integers lesser than m, called respectively the multiplier, increment, and seed of the generator [SM02]. LCG is one of the oldest and best-known generator. It will have a full period for all seed values if and only if:

1. c and m are relatively prime,
2. $a - 1$ is divisible by all prime factors of m,
3. $a - 1$ is a multiple of 4 when m is a multiple of 4.

In this article, 2LCGs and 3LCGs refer as two (resp. three) combinations [SM02] of such LCGs, as follows:

- the first LCG s_1 has parameters (m_1, a_1, c_1) and the second one s_2 has parameters (m_2, a_2, c_2),
- the combination is $x^n = (s_1^n - s_2^n)\ mod\ (m_1 - 1)$, where s_1^n and s_2^n are the states of the two LCGs components at step n.

In other words:

$$\begin{cases} s_1^n = (a_1 \times s_1^{n-1} + c_1)\ mod\ m_1 \\ s_2^n = (a_2 \times s_1^{n-1} + c_2)\ mod\ m_2 \\ x^n = (s_1^n - s_2^n)\ mod\ (m_1 - 1). \end{cases}$$

These formulas can be easily adapted for the combination of 3 linear congruential generators. The inputted LCGs must satisfy the requirement recalled above, and one must also have $m1 > m2$. For further details, see [com88].

A.1.4/ Multiple Recursive Generators

The multiple recursive generators (MRGs) are based on higher order recursion k [SM02]:

$$x^0 \in [\![0, m-1]\!], \quad x^n = (a^1 x^{n-1} + \ ...\ + a^k x^{n-k})\ mod\ m, \tag{A.2}$$

where $a^1, \ldots, a^k \in [\![0, m-1]\!]$. Combination of two MRGs (referred as 2MRGs) has also been used in this article, they are defined like the multiple LCGs:

$$\begin{cases} s_1^n = (a_1^1 s_1^{n-1} + \ ...\ + a_1^k s_1^{n-k})\ mod\ m_1, \\ s_2^n = (a_2^1 s_2^{n-1} + \ ...\ + a_2^k s_2^{n-k})\ mod\ m_2, \\ x^n = (s_1^n - s_2^n)\ mod\ (m_1). \end{cases}$$

The combination method is thus obtained by subtracting the states modulo m_1. For reasons not debated in this document, usual implementations of this 2MRG that present correct statistics suppose that $k = 3$, $a_1^1 = 0, a_1^2 > 0, a_1^3 < 0, a_2^1 > 0, a_2^2 = 0$, $a_2^3 < 0$, and finally $a_1^j \times (m_1\ mod\ a_1^j) < m_1$ whereas $a_2^j \times (m_2\ mod\ a_2^j) < m_2$. These requirements have been followed in our experiments.

A.1.5/ UCARRY

UCARRY acronym refers to generators based on linear recurrences with carry. This includes the *add-with-carry* (AWC), *subtract-with-borrow* (SWB), and *shift-with-carry* (SWC) generators.

The add-with-carry generator, proposed by Marsaglia and Zaman, is based on the following linear recurrence with carry. Given a modulus m and two positive different integers r and s, and for integers initial values $c^0 \in \{0, 1\}$ and $x^0, ..., x^k \in [\![0, m-1]\!]$, where $k = max(r, s)$, compute for $n > k$:

$$\begin{aligned} x^n &= (x^{n-r} + x^{n-s} + c^{n-1})\ mod\ m, \\ c^n &= (x^{n-r} + x^{n-s} + c^{n-1})/m, \end{aligned} \tag{A.3}$$

and return at each iterate the output x^n/m, that is, their quotient.

The SWB generator, for its part, uses the same inputs and has the recurrence:

$$\begin{aligned} x^n &= (x^{n-r} - x^{n-s} - c^{n-1}) \; mod \; m, \\ c^n &= \begin{cases} 1 & \text{if } (x^{i-r} - x^{i-s} - c^{i-1}) < 0 \\ 0 & \text{else,} \end{cases} \end{aligned} \tag{A.4}$$

and the output is x^i/m another time.

Finally, the shift-with-carry SWC generator designed by R. Couture is based on the following recurrence:

$$\begin{aligned} x^n &= (a^1 x^{n-1} \oplus \; ... \; \oplus a^r x^{n-r} \oplus c^{n-1}) \; mod \; 2^w, \\ c^n &= (a^1 x^{n-1} \oplus \; ... \; \oplus a^r x^{n-r} \oplus c^{n-1}) \; / \; 2^w. \end{aligned} \tag{A.5}$$

with output equal to $x^n/2^w$. The initial values are $(x^0, ..., x^{r-1})$ and c is the initial carry. Restrictions: $0 < r$, and $w \leqslant 32$.

A.1.6/ Generalized Feedback Shift Register

By GFSR we referred to a particular generalized feedback shift register generator based on the recurrence:

$$x^n = x^{n-r} \oplus x^{n-k}. \tag{A.6}$$

Each x^n is a 32-bit vector, k and r are positive integers such that $r < k$. The output at step n is $u_n = \tilde{x}_n/2^l$, where $\tilde{x}_n$ is the integer formed by the first l bits of x_n, and $l \leqslant 32$. $x_0, \ldots, x_{k-1}$ must be provided as k initial bit vectors. Proper initialization techniques for this generator have been discussed in the literature, they have been respected during our implementations.

A.1.7/ Nonlinear Inversive Generator

Finally, INV stands for the nonlinear inversive generator, as defined in [SM02], which is:

$$x^n = \begin{cases} (a^1 + a^2/z^{n-1}) \; mod \; m & \text{if } z^{n-1} \neq 0 \\ a^1 & \text{if } z^{n-1} = 0. \end{cases} \tag{A.7}$$

The generator computes z via the modified Euclid algorithm (see [SM02]). If m is prime and if $p(x) = x^2 - a^1 x - a^2$ is a primitive polynomial modulo m, then the generator has maximal period m. Restrictions: $0 \leqslant z^0 < m, 0 < a^1 < m$ and $0 < a^2 < m$. Furthermore, m must be a prime number, preferably large.

A.1.8/ XORshift

XORshift is a category of very fast PRNGs designed by George Marsaglia [Mar03]. It repeatedly uses the transform of *exclusive or* (XOR) on a number with a bit shifted version of it.

The state of a XORshift generator is a vector of bits. At each step, the next state is obtained by applying 3 times the following operations to w-bit blocks in the current state, where $w = 32$ or 64: replace the w-bit block by a bitwise XOR of the original block with a shifted copy of itself by respectively a, b, and c positions, where $-w < a, b, c < w$. The direction of the circular shift is either the left or the right, depending on the signs of a, b, and c.

For instance, Algorithm 4 is the 32-bit XORshift with $(a, b, c) = (-13, 17, -5)$, which has a period of $2^{32} - 1 \approx 4.29 \times 10^9$.

Algorithm 4: An arbitrary round of XORshift algorithm

Input: the internal state z (a 32-bits word)
Output: y (a 32-bits word)

1: $z \leftarrow z \oplus (z \ll 13)$;
2: $z \leftarrow z \oplus (z \gg 17)$;
3: $z \leftarrow z \oplus (z \ll 5)$;
4: $y \leftarrow z$;
5: return y;

In this article, we have always supposed that the directions of the circular shifts are: left/right/left, which was not required in the original paper of Marsaglia. Other improved versions of this XORshift exist in the literature, we have chosen this historical one in our researches for its speed and statistical flaws.

A.1.9/ ISAAC

ISAAC is an array-based PRNG and a stream cipher designed by Robert Jenkins (1996) to be cryptographically secure [Jen96]. The name is an acronym for Indirection, Shift, Accumulate, Add, and Count. The ISAAC algorithm has similarities with RC4 [cit03]. It uses an array of 256 32-bit integers as the internal state, writes the results to another 256-integer array, from which they are read one at a time until empty, at which point they are recomputed. Since it only takes about 19 32-bit operations for each 32-bit output word, it is extremely fast on 32-bit computers.

We give the key-stream procedure of ISAAC in Algo.5. The internal state is x, the output array is r, and the inputs 32-bit words a, b, and c are those computed in the previous round. Normally a, b, c, and the array x are initialized with some random sequences.The value $f(a, i)$ in Algo.5 is a 32-bit word, defined for all a and $i \in \{0, \ldots, 255\}$ as:

$$f(a,i) = \begin{cases} a \ll 13 & \text{if } i \bmod 4 \equiv 0, \\ a \gg 6 & \text{if } i \bmod 4 \equiv 1, \\ a \ll 2 & \text{if } i \bmod 4 \equiv 2, \\ a \gg 16 & \text{if } i \bmod 4 \equiv 3. \end{cases} \tag{A.8}$$

Algorithm 5: An arbitrary round of ISAAC algorithm

Input: a, b, c, and the internal state x, they are 32-bit words
Output: an array r of 256 32-bit words

$c \leftarrow c + 1$;
$b \leftarrow b + c$;
while $i = 0, \ldots, 255$ **do**
  $s \leftarrow x_i$;
  $a \leftarrow f(a, i) + x_{(i+128)\ mod\ 256}$;
  $x_i \leftarrow a + b + x_{(x \gg 2)\ mod\ 256}$;
  $r_i \leftarrow s + x_{(x_i \gg 10)\ mod\ 256}$;
  $b \leftarrow r_i$;
return r;

A.2/ VARIOUS IMPROVEMENTS OF THE CIPRNG VERSION 1

In this section, a few results obtained at Qianxue Wang and Xiaole Fang thesis occasion are recalled.

A.2.1/ THE CIPRNG VERSION 2

After the proof of concept of CIPRNG version 1, a second version of generator based on chaotic iterations has been introduced in [WBGF10, BGW10a], in order to obtain outputs at the same speed than the inputted generators. The basic idea in this improvement is to prevent from changing a bit twice between two outputs, reducing by doing so the generation time. To do so, the meaning of sequence (m^n) must be changed: it now defines the number of bits to change between two outputs (instead of the number of chaotic iterations).

The output of the sequence PRNG2 is normally uniform in $[\![0, 2^{32} - 1]\!]$. However, we do not want the output of (m^n) to be uniform in $[\![0, N]\!]$, because in this case, the returns of our generator will not be uniform in $[\![0, 2^N - 1]\!]$, as it is illustrated in the following example. Suppose that $x^0 = (0, 0, 0)$. Then $m^0 \in [\![0, 3]\!]$: the number of bits we can change in x^0 is between 0 and 3.

Table A.1: Statistical results for the CIPRNG version 2

Test name	CIPRNG Version 2			
	Logistic + Logistic	XORshift + XORshift	ISAAC + XORshift	ISAAC + ISAAC
NIST (15)	15	15	15	15
DieHARD (18)	18	18	18	18
TestU01 (516)	516	516	516	516

- If $m^0 = 0$, then no bit must change between the first and the second output of our CI PRNG Version 2. Thus we have only 1 possibility for x^1, namely $x^1 = (0, 0, 0)$.
- If $m^0 = 1$, then exactly one bit must change, which leads to three possible values for x^1, that is, $(1, 0, 0)$, $(0, 1, 0)$, and $(0, 0, 1)$.
- *etc.*

Each value in $[\![0, 2^3 - 1]\!]$ must be returned with the same frequency, then the values $(0, 0, 0)$, $(1, 0, 0)$, $(0, 1, 0)$, and $(0, 0, 1)$ must occur for x^1 with the same probability. Finally we see that, in this example, $m^0 = 1$ must be three times more probable than $m^0 = 0$. This leads to the following general definition for the probability of $m = i$: $P(m^n = i) = \frac{C_N^i}{2^N}$, with $C_n^k = \dfrac{n!}{k!(n-k)!}$. Thus $\forall n \in \mathbb{N}, m^n = g(PRNG2()^n)$, where

$$g(y) = k \Leftrightarrow \sum_{i=0}^{k-1} C_N^i \leqslant y < \sum_{i=0}^{k} C_N^i.$$

We have adapted the outputs of PRNG2 to obtain a sequence corresponding to the number of changes between two outputs of the CIPRNG(PRNG1,PRNG2) version 2. We must also adapt the outputs of the strategy PRNG1: the strategy indicates the coordinates to change, and we do not want to change twice a given coordinate between two outputs of the CIPRNG. More precisely, the m^0 first terms of the strategy must be different, as we want to obtain m^0 changes between the initial state and the first output. Then, the terms in position $m^0 + 1, ..., m^1$ must be all different too for the same reason, and so one. However, PRNG1 does not necessarily provides such a particular sequence. This is why we must operate a decimation on it, as explained in [WBGF10, BGW10a].

Let $(d^1, d^2, \ldots, d^N) \in \{0, 1\}^N$ be a mark sequence, counting the number of occurrences of each integer between two outputs. It is such that whenever $\sum_{i=1}^{N} d^i = m^k$, then $\forall i, d_i = 0$: after each output of the CIPRNG, this counting sequence is reset. This mark sequence will control the PRNG1 sequence b as follows. Let b^j be the numbers produced by PRNG1:

- if $d^{b^j} \neq 1$, then $S^k = b^j$, $d^{b^j} = 1$, and $k = k + 1$,
- if $d^{b^j} = 1$, then b^j is discarded (it has already occurred in this output).

The basic design procedure of this optimized generator, released and tested in [WBGF10, BGW10a], is summed up in Algorithm 6, whereas the good obtained results are summarized in Table A.1.

A.2.2/ INVESTIGATING THE STATISTICAL IMPROVEMENTS OF CHAOS-BASED CIPRNGS POST-TREATMENT

CIPRNGs versions 1, 2, and XOR have been widely experimented these last three years, on various inputted pseudorandom generators, most of them being more or less defective. Obtained results are summarized thereafter.

The following well-known PRNGs have been considered for experiments:

Algorithm 6: An arbitrary round of the CIPRNG(PRNG1,PRNG2) version 2

Input: the internal state x (N bits)
Output: a state r (N bits)

1: **for** $i = 0, \ldots, N$ **do**
2: $\quad d_i \leftarrow 0$;
3: $i \leftarrow 0$;
4: $m \leftarrow g(PRNG2())$;
5: **while** $i < m$ **do**
6: $\quad S \leftarrow PRNG1()$;
7: $\quad$ **if** $d_S = 0$ **then**
8: $\quad\quad x_S \leftarrow \overline{x_S}$;
9: $\quad\quad d_S \leftarrow 1$;
10: $\quad\quad i \leftarrow i + 1$;
11: **return** x;

- LCG, MRG for linear congruential PRNGs;
- AWC, SWB, SWC, and GFSR for lagged ones;
- INV from type ICG (inversive congruential generators);
- lastly, 2LCG, 3LCG, and 2MRG to study the effects on mixed PRNGs.

We have performed various statistical tests on these generators, showing that they reveal several issues, as summarized in Table A.2. The tests studied here are the NIST suite [BR10] and DieHARD battery of tests [Mar96].

	linear		lagged				icg	mixed		
	lcg	mrg	awc	swb	swc	gfsr	inv	2lcg	3lcg	2mrg
NIST (11)	11	14	**15**	**15**	14	14	14	14	14	14
DieHARD (18)	16	16	15	16	**18**	16	16	16	16	16

Table A.2: NIST and DieHARD tests suite passing rates for PRNGs without CIPRNG method

Then we have performed statistical analyses on each of the aforementioned CIPRNGs. The results are reproduced in Table A.3, they have not yet been published (currently under reviewing process). An asterisk "*" means that the considered passing rate has been improved. We can observe that, except for the XOR CIPRNG, all of the CIPRNGs have passed the 15 tests of the NIST battery and the 18 tests of the DieHARD one. Moreover, considering these scores, we can deduce that both the single Version 1 CIPRNG and the single Version 2 CIPRNG are relatively steadier than the single XOR CIPRNG approach, when applying them to different PRNGs. However, the XOR CIPRNG is obviously the fastest approach to generate a CI random sequence, and it still improves the statistical properties relative to each generator taken alone, although the test values are not as good as desired.

Table A.3: NIST and DieHARD tests suite passing rates for PRNGs with CIPRNG method

Types	Linear		Lagged				icg	Mixed		
	lcg	mrg	awc	swb	swc	gfsr	inv	2lcg	3lcg	2mrg
Version 1										
NIST	**15***	**15***	**15**	**15**	**15***	**15***	**15***	**15***	**15***	**15**
DieHARD	**18***	**18***	**18***	**18***	**18**	**18***	**18***	**18***	**18***	**18***
Version 2										
NIST	**15***	**15***	**15**	**15**	**15***	**15***	**15***	**15***	**15***	**15**
DieHARD	**18***	**18***	**18***	**18***	**18**	**18***	**18***	**18***	**18***	**18***
XOR ciprng										
NIST	14*	**15***	**15**	**15**	14	**15***	14	**15***	**15***	**15**
DieHARD	16	16	17*	**18***	**18**	**18***	16	16	16	16

To have a realization of the XOR CIPRNG that can pass all the tests embedded into the NIST battery, we will now investigate the “Multiple XOR CIPRNG” variations of the XOR post-treatment in the following sections.

A.2.3/ VARIATIONS ON THE XOR CIPRNG

We now regard the possibility to use various successive terms of a given deficient generator S in order to improve its statistics. Such a desire, which still remains general chaotic iterations, leads to the definition of the multiple XOR CIPRNG, introduced in [BFG12a] and detailed below:

$$\begin{cases} x^0 \in [\![0, 2^{\mathsf{N}} - 1]\!], S \in [\![0, 2^{\mathsf{N}} - 1]\!]^{\mathbb{N}} \\ \forall n \in \mathbb{N}^*, x^n = x^{n-1} \oplus S^{nm} \oplus S^{nm+1} \ldots \oplus S^{nm+m-1}, \end{cases} \tag{A.9}$$

where S stands for the inputted PRNG. We show in Table A.4 that a threshold value m (called the functional power) can always be found such that the multiple XOR CIPRNG becomes able to pass the whole NIST battery. The existence of this threshold illustrates in a certain extend the progressive appearance of the effects of chaos.

The results presented in this section have reinforced our confidence in the capability for chaos to act as post-treatment on defective pseudorandom number generators, in order to improve their statistics. However, we can regret the following flaws for all the currently proposed CIPRNGs.

1. Up to now, speed performances are not really good, as in (single) CIPRNGs versions 1 and 2 we must call various times the inputted generators between two outputs. Similarly the XOR CIPRNG can satisfactorily improve defective generators only by grouping (xoring) a potentially large number of successive terms produced by the input (this is the Multiple XOR CIPRNG).

2. As presented here, XOR and multiple XOR can only handle one inputted generator. However, an interesting strategy when designing new generators using formerly

Table A.4: Functional power m making it possible to pass the whole NIST battery

Inputted $PRNG$	lcg	mrg	swc	gfsr	inv	2lcg	3lcg	2mrg
Threshold value m	19	7	2	1	11	9	3	4

released ones is to take the best of each input: speed of the first inputted PRNG and security of the second one, for instance.

3. CIPRNGs versions 1 and 2 and multiple XOR CIPRNGs have better statistical performances than XOR CIPRNG, because they use various successive terms of the inputs to produce one output: chaos has time to express itself and high correlations between two successive inputs of the deflated PRNGs are broken by doing so.

We will thus introduce two new methods to take the best of each version. These methods have been published in [BFGW13].

A.2.4/ "LUT" CIPRNG(XORSHIFT,XORSHIFT) VERSION 3

The LUT (Lookup-Table) CIPRNG version 3 is an improved, mixed version of both the CIPRNG version 2 and the XOR CIPRNG. The key-ideas, developed in [BFGW13], are:

1. To use a Lookup Table for a faster production of strategies than in CIPRNG version 2. These strategies satisfy the same property than the ones provided by the decimation process, reducing by doing so the correlations of successive terms in the inputted PRNG.
2. To operate as in XOR CIPRNG, by computing $x^{n+1} = x^n \oplus S^n$ directly (general chaotic iterations of the vectorial negation instead of unary chaotic iterations).

This generator will not be explain in details in this manuscript, only statistical tests results will be presented next pages. Readers interested by such a generator are referred to [BFGW13].

A.2.5/ THE VERSION 4 CATEGORY OF CIPRNGS

The CIPRNG version 4 is an improvement of the multiple XOR CIPRNG, in which we will use m PRNGs instead of m successive terms of one PRNG. Or, more precisely, subsets of these m PRNGs. By doing so, the problem of speed can be resolved by computing them in parallel, whereas the two other issues will no longer be problematic.

In the XOR CIPRNG $x^{n+1} = x^n \oplus S^n$, the k^{th} component of its state (a binary digit) changes if and only if the k^{th} digit in the binary decomposition of the $n-$th term S^n of the inputted generator is 1. In version 4, instead of updating only one cell at each iteration as the first versions of our CIPRNGs, a subset of components is chosen and updated. We have already shown that, taken alone, this XOR CIPRNG does not improve a lot the possibly defective inputted generator S. A first solution has been proposed in the multiple XOR CIPRNG by xoring various successive terms of S before xoring the result with the last state of the system. We have shown that this method is able to really improve the

inputted generator. However, its principal flaw is that, for a large number of generators, all the terms $S^{mn}, S^{mn+1}, ..., S^{mn+m-1}$ must be computed step by step, and m can be large for very defective PRNGs. A second but less critical flaw is that the XOR CIPRNG only receives one inputted generator. However, as stated before, some situations exist where we want to take benefits from various inputted generators: security of the first PRNG, speed of the second one, and so on.

It is possible to add more complexity and speed in the multiple XOR CIPRNG, by considering a set of M inputted generators, picking randomly a subset of them at each iteration, and xoring their xored values with the internal state of the system. This algorithm, based another time on general chaotic iterations and currently submitted, can be written as in Algo.7.

Algorithm 7: An arbitrary round of the version 4 CI generator

Input: the internal state x (N bits)
Output: a state r of N bits

1: **for** $i = 1, \ldots, M$ **do**
2: $\quad S(i) = PRNG2\_i()$;
3: $T = PRNG1()$;
4: $r = x \oplus h(T, S(1), S(2), ... S(M))$,
5: **return** r;

$S(1), S(2), ..., S(M)$ are the M inputted PRNGs, whereas $T^n \in [\![0, 2^M - 1]\!]$ gives which sequences must be considered at the current iteration, as follows. Let $(t_1^n, t_2^n, \ldots, t_M^n) \in \{0,1\}^M$ be the binary representation of the M-bit number T^n. Then the sequence $S^n(1), S^n(2), ..., S^n(M)$ is decimated with h function as follows: if $t_i^n = 0$, then $S^n(i)$ is discarded, else $S^n(i)$ is kept for *bitwise exclusive or* computing. In brief, the produced output sequence x^n, based on chaotic iterations, is updated by a *bitwise exclusive or* of an irregular decimation of $S(1), S(2), ..., S(M)$, according to the bits of T^n. Note that an efficient GPU implementation of this generator can be found in [BCGH11], which has successfully passed the stringent TestU01 battery of statistical tests [SM02].

A.3/ RANDOMNESS QUALITY OF CIPRNGS

In this section, we recall the last statistical investigations we have published. The detail of CIPRNGs tests results and references can be found in [Wan12, Fan13].

Table A.5 compares all the versions of CIPRNG (XORshift, XORshift) against the NIST and DieHARD batteries. We can see that XORshift alone fails both the two batteries, whereas the generator based on discrete chaotic iterations (CIPRNGs versions 1-4) can improve it.

Generators investigated in this second set of experiments are now respectively the BBS (with very bad security parameters: m of 32 bits and outputs of 4 bits), a Logistic map, XORshift, and ISAAC, while the NIST, DieHARD, and TestU01 test suites have been considered for statistical evaluation. Let us recall that Table 4.1 contains the statistical results obtained by the considered inputted generators. In Table A.6 are shown the results obtained by the version 3 of our CIPRNGs. These results confirm that the CIPRNGs

		CIPRNG version			
	XORshift	1	2	3	4
NIST (15 tests)	14	15	15	15	15
DieHARD (18 tests)	15	18	18	18	18

Table A.5: NIST and DieHARD results for XORshift alone and CIPRNG (XORshift, XORshift) versions 1-4.

Table A.6: Statistical results for the LUT CIPRNG version 3

Test name	LUT CIPRNG Version 3			
	Logistic + Logistic	XORshift + XORshift	ISAAC + XORshift	BBS + XORshift
NIST (15)	15	15	15	8
DieHARD (18)	18	18	18	8
TestU01 (516)	516	516	516	356

version 3 are all able to pass these tests, except when using the very deflated BBS generator. This issue is solved with the version 4, as shown in Table A.7. This last version of the CIPRNG family offer thus a great compromise among statistical performances and efficiency (Figure 4.1 provides a speed comparison between the slow BBS generator, the fast XORshift, and the CIPRNGs version 1-4). It can be considered as very suitable both for software and hardware implementations.

Table A.7: Statistical results for the CIPRNG version 4

Test name	CIPRNG version 4			
	Logistic + Logistic	XORshift + XORshift	ISAAC + XORshift	BBS + XORshift
NIST (15)	15	15	15	15
DieHARD (18)	18	18	18	18
TestU01 (516)	516	516	516	516

B

Further Developments in Information Hiding

B.1/ The $\mathcal{CIS}_2$ Chaotic Iteration based Steganographic Process

After the introduction of $\mathcal{CIW}_1$ in [GFB10], there were only two information hiding schemes being both stego-secure and topologically secure. The first one is based on a spread spectrum technique called Natural Watermarking. It is stego-secure when its parameter η is equal to 1 [CB08a]. Unfortunately, this scheme is neither robust, nor able to face an attacker in KOA and KMA setups, due to its lack of expansiveness [Guy10]. The second scheme both topologically secure and stego-secure has been presented in the previous section. However, this $\mathcal{CIW}_1$ process allows to embed securely only one bit per embedding parameters. The objective of [FGB11] was to improve the scheme studied in [GFB10], in such a way that more than one bit can be embedded. Such a study led to the definition of the $\mathcal{CIS}_2$ scheme presented here.

B.1.1/ The improved algorithm

Let us firstly recall the notations and terminologies introduced in [FGB11], which extend the ones presented in Chapter 2.

Definition 52. *Let* $\mathsf{k} \in \mathbb{N}^*$. *A* strategy adapter *is a sequence which elements belong into* $[\![0, \mathsf{k}-1]\!]$. *The set of all strategies with terms in* $[\![0, \mathsf{k}-1]\!]$ *is denoted by* $\mathbb{S}_\mathsf{k}$.

Intuitively, a strategy-adapter aims at generating a strategy $(S^t)^{t\in\mathbb{N}}$ where each term S^t belongs to $[\![1, n]\!]$.

Definition 53. *Let* $k \in \mathbb{N}^*$. *The* initial function *is the map* i_k *defined by:*

$$\begin{array}{rccc} i_k: & \mathbb{S}_k & \longrightarrow & [\![0, \mathsf{k}-1]\!] \\ & (S^n)_{n\in\mathbb{N}} & \longmapsto & S^0 \end{array}$$

Definition 54. *Let* $k \in \mathbb{N}^*$. *The* shift function *is the map* σ_k *defined by:*

$$\begin{array}{rccc} \sigma_k: & \mathbb{S}_k & \longrightarrow & \mathbb{S}_k \\ & (S^n)_{n\in\mathbb{N}} & \longmapsto & (S^{n+1})_{n\in\mathbb{N}} \end{array}$$

Let us additionally recall the following notations.

- $x^0 \in \mathbb{B}^{\mathsf{N}}$ the N least significant coefficients of a given cover media C.
- $m^0 \in \mathbb{B}^{\mathsf{P}}$ is the watermark to embed into x^0.
- $S_1 \in \mathbb{S}_N$ is a strategy called **place strategy**, giving the location (LCS) where to insert the message at each iteration.
- $S_2 \in \mathbb{S}_P$ is a strategy called **choice strategy**, providing which bits from the message must be inserted at the given iteration.
- Lastly, $S_3 \in \mathbb{S}_P$ is a strategy called **mixing strategy**, as it is required for chaos to mix the message at each iteration.

The information hiding scheme published in [FGB11] was formerly called Steganography by Chaotic Iterations and Substitution with Mixing Message (SCISMM in short), and has been renamed $\mathcal{CIS}_2$ in later publications. It is defined by $\forall (n,i,j) \in \mathbb{N}^* \times [\![0;\mathsf{N}-1]\!] \times [\![0;\mathsf{P}-1]\!]$:

$$\begin{cases} x_i^n = \begin{cases} x_i^{n-1} & \text{if } S_1^n \neq i \\ m_{S_2^n} & \text{if } S_1^n = i. \end{cases} \\ \\ m_j^n = \begin{cases} m_j^{n-1} & \text{if } S_3^n \neq j \\ \overline{m_j^{n-1}} & \text{if } S_3^n = j. \end{cases} \end{cases}$$

The stego-content is the Boolean vector $y = x^P \in \mathbb{B}^{\mathsf{N}}$.

B.1.2/ SECURITY STUDY OF THE $\mathcal{CIS}_2$

After having introduced the $\mathcal{CIS}_2$, we have studied its security in [FGB11].

B.1.2.1/ STEGO-SECURITY

We have proven in [FGB11] that,

Proposition 23. *$\mathcal{CIS}_2$ is stego-secure.*

Proof 7. *See [FGB11].*

B.1.2.2/ TOPOLOGICAL SECURITY

Topological model We have firstly proven in [FGB11] that $\mathcal{CIS}_2$ can be modeled as a dynamical system in a topological space, as follows. Let

$$\begin{array}{rl} F: & [\![0;\mathsf{N}-1]\!] \times \mathbb{B}^{\mathsf{N}} \times [\![0;\mathsf{P}-1]\!] \times \mathbb{B}^{\mathsf{P}} \longrightarrow \mathbb{B}^{\mathsf{N}} \\ & (k,x,\lambda,m) \longmapsto \left(\delta(k,j).x_j + \overline{\delta(k,j)}.m_\lambda\right)_{j \in [\![0;\mathsf{N}-1]\!]} \end{array}$$

where + and . are the boolean addition and product operations.

Consider the phase space $\mathcal{X}_2$ defined as follow:

$$\mathcal{X}_2 = \mathbb{S}_N \times \mathbb{B}^{\mathsf{N}} \times \mathbb{S}_P \times \mathbb{B}^{\mathsf{P}} \times \mathbb{S}_P,$$

where $\mathbb{S}_N$ and $\mathbb{S}_P$ are the sets introduced in Section B.1.1.

We define the map $\mathcal{G}_{f_0} : \mathcal{X}_2 \longrightarrow \mathcal{X}_2$ by:

$$\mathcal{G}_{f_0}(S_1, x, S_2, m, S_3) =$$
$$(\sigma_N(S_1), F(i_N(S_1), x, i_P(S_2), m), \sigma_P(S_2), G_{f_0}(m, S_3), \sigma_P(S_3))$$

$\mathcal{CIS}_2$ can be described by the iterations of the following discrete dynamical system:

$$\begin{cases} X^0 \in \mathcal{X}_2 \\ X^{k+1} = \mathcal{G}_{f_0}(X^k). \end{cases}$$

Then, by comparing $\mathcal{X}_2$ and the phase space $\mathcal{X}$ formerly introduced in this manuscript, we have verified in [FGB11] that.

Proposition 24. *The phase space $\mathcal{X}_2$ has, at least, the cardinality of the continuum.*

A new distance on $\mathcal{X}_2$ We have defined in [FGB11] a new distance on $\mathcal{X}_2$ as follows: $\forall X, \check{X} \in \mathcal{X}_2$, if $X = (S_1, x, S_2, m, S_3)$ and $\check{X} = (\check{S_1}, \check{x}, \check{S_2}, \check{m}, \check{S_3})$, then:

$$\begin{aligned} d_2(X, \check{X}) &= d_{\mathbb{B}^{\mathsf{N}}}(x, \check{x}) + d_{\mathbb{B}^{\mathsf{P}}}(m, \check{m}) \\ &+ d_{\mathbb{S}_N}(S_1, \check{S_1}) + d_{\mathbb{S}_P}(S_2, \check{S_2}) + d_{\mathbb{S}_P}(S_3, \check{S_3}). \end{aligned}$$

Continuity of $\mathcal{CIS}_2$ To prove that $\mathcal{CIS}_2$ is another example of topological chaos in the sense of Devaney, $\mathcal{G}_{f_0}$ must be continuous on the metric space $(\mathcal{X}_2, d_2)$. We thus have proven in [FGB11] that,

Proposition 25. *$\mathcal{G}_{f_0}$ is a continuous function on $(\mathcal{X}_2, d_2)$.*

$\mathcal{CIS}_2$ is chaotic Then, in [FGB11], $(\mathcal{X}_2, \mathcal{G}_{f_0})$ has been proven to be topologically transitive, regular, and sensitive dependence on initial conditions. We thus have the result [FGB11]:

Theorem 24. *$\mathcal{G}_{f_0}$ is a chaotic map on $(\mathcal{X}_2, d_2)$ in the sense of Devaney.*

So we can claim that $\mathcal{CIS}_2$ is topologically secure.

B.1.3/ CORRECTNESS AND COMPLETENESS STUDIES

Without attack, the $\mathcal{CIS}_2$ scheme has to ensure that the user can always extract a message and that this latter is the watermark, provided the user has the correct keys. These two demands correspond respectively to the study of completeness and of correctness for the proposed approach, which have been investigated in [BCF+13]. We have firstly established that,

Proposition [26]. *Let $\Im(S_p)$ be the set (without repetitions) $\{S_p^1, S_p^2, \ldots, S_p^l\}$ of cardinality k, $k \leqslant l$. This set contains all the elements of x that have been modified along the $\mathcal{CIS}_2$ iteration process. Let us consider $\Im(S_c)_{|D}$ defined by $\{S_c^{d_1}, S_c^{d_2}, \ldots, S_c^{d_k}\}$ where d_i is the last iteration that has modified the element $i \in \Im(S_p)$.*

Message can be extracted from the stego-content if and only if $\Im(S_c)_{|D} = [\![0; \mathsf{P}-1]\!]$.

Under this condition, one bit of index j of the original message m^0 is thus embedded at least twice in x^l. By counting the number of times this bit has been switched in S_m, the value of m_j can be deduced in many places. Without attack, all these values are equal and the message is immediately obtained. After an attack, the value of m_j is obtained as mean value of all its occurrences. The scheme is thus complete. Notice that if the cover is not attacked, the returned message is always equal to the original due to the definition of the mean function.

B.1.4/ Deciding whether a possibly attacked media is watermarked

Let us consider a first media y that is watermarked with a message m and a second one, namely y', which is an altered version of y, *i.e.*, where some bits have been modified. Let m' be the message that is extracted from y'.

We have checked in [BCF+13] how far the extracted message m' is from m. To achieve this, we have considered the set $M = \{i | m_i = 1\}$ of the Boolean vector message m and similarly the set M' for the message m'. Most of similarity measures depend on the functions a, b, c, and d, all from $\mathbb{B}^{\mathsf{P}} \times \mathbb{B}^{\mathsf{P}}$ to $\mathbb{N}$, and respectively equal to $a(m, m') = |M \cap M'|$, $b(m, m') = |M \backslash M'|$, $c(m, m') = |M' \backslash M|$, and $d(m, m') = |\overline{M} \cap \overline{M'}|$ ($|S|$ and $\overline{S}$ respectively denote the cardinality and the complementary of any set S). In what follows a, b, c, and d respectively stand for $a(m, m')$, $b(m, m')$, $c(m, m')$, and $d(m, m')$.

According to [RDBM03] the Fermi-Dirac measure S_{FD} is the one that has the highest discrimination power, *i.e.*, which allows a clear separation between correlated vectors and uncorrelated ones. The measure is recalled hereafter with respect to the previously defined scalars a, b, and c.

$$\begin{aligned} S_{FD}(\varphi) &= \frac{F_{FD}(\varphi) - F_{FD}(\frac{\pi}{2})}{F_{FD}(0) - F_{FD}(\frac{\pi}{2})}, \\ F_{FD}(\varphi) &= \frac{1}{1 + \exp(\frac{\varphi - \varphi_0}{\gamma})}, \end{aligned}$$

where $\varphi = \arctan(\frac{b+c}{a})$, φ_0 is $\pi/4$, and γ is 0.1.

The distance between m and m' is then computed in [BCF+13] as $1 - S_{FD}(m, m')$ and is thus a real number in $[0; 1]$. We have proposed in [BCF+13] that, if such a distance is lower than a given threshold, y' will be declared as watermarked and not watermarked otherwise. Next section presents a practical robustness evaluation of $\mathcal{CIS}_2$ using this decision rule.

B.1.5/ ROBUSTNESS STUDY OF THE PROCESS

This section is devoted to the recall of the robustness study of the $\mathcal{CIS}_2$ scheme realized in [BCF+13]. For the whole experiments, a set of 100 images has been randomly extracted from the database taken from the BOSS contest [PFB10b]. In this set, each cover is a 512×512 grayscale digital image. The considered watermark m is given in Fig. 5.2(b). Testing the robustness of the approach is achieved by successively applying on watermarked images attacks like cropping, compression, geometric transformations,... Differences between m and m' have been computed as described in the previous section.

We have firstly evaluate the robustness of the $\mathcal{CIS}_2$ approach by applying different percentages of cropping, from 0.25% to 90%. Results are recalled in Fig. B.1, which presents effects of such an attack. All the percentage differences are so far less than 97% and thus robustness is established.

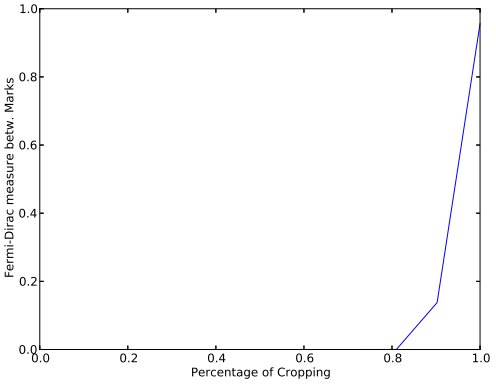
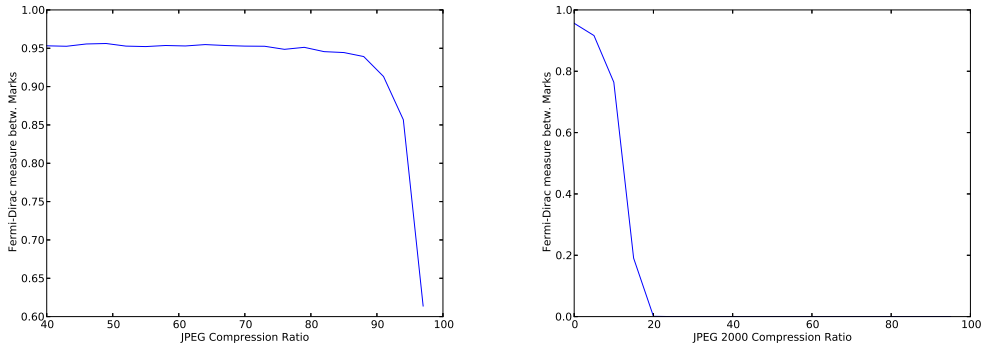


Figure B.1: Cropping Results

Robustness against compression has then been addressed in [BCF+13], by studying both JPEG and JPEG 2000 image compressions. Results are respectively presented in Fig. B.2(a) and Fig. B.2(b). It is not hard to see that robustness is well established for JPEG2000 compression: for all the ratios larger than 10%, the watermark is retrieved. However, as stated in [BCF+13], this scheme is not robust against JPEG compression for a ratio inferior to 90%. Remark that a potential solution can be to insert the watermark in least significant coefficient of the image described in frequency domain, for instance using either discrete cosine or with wavelet transform.

Among geometric transformations, we then focused on rotations, *i.e.*, when two opposite rotations of angle θ are successively applied around the center of the image. In these geometric transformations, angles range from 2 to 60 degrees. Results are presented in Fig. B.3: thanks to an efficient embedding, our scheme is resistant to all that type of attacks.

The first step of the $\mathcal{CIS}_2$ scheme studied in this subsection has defined x as the LSBs of the host image, it is thus based on LSB modifications. We have then considered in [BCF+13] two types of attacks modifying these LSB sets (see Fig B.4). The former consists in setting to zero a subset of this one. Results are expressed in Fig. B.4(a) and show that the scheme is robust, unless 95% of the LSB is erased. In this case the image is really damaged. The latter consists in applying again this scheme on the watermarked



(a) JPEG Effect (b) JPEG 2000 Effect

Figure B.2: Compression Results

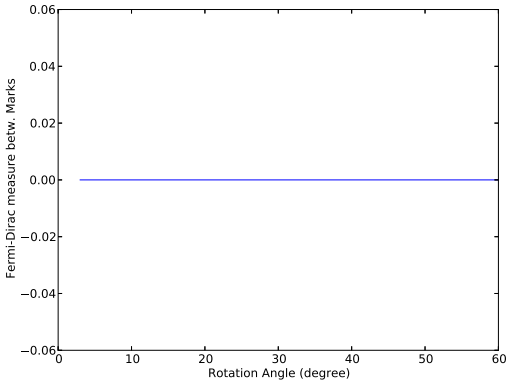


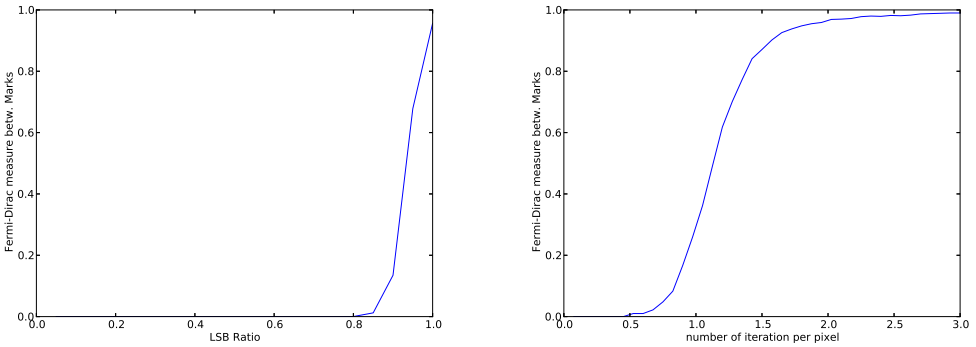
Figure B.3: Rotation Attack Results

image but with another message. Results of Fig. B.4(b) show that this scheme is robust against that type of attack, provided the number of iterations is lesser than 1.75 times the number of pixels. With more iterations, the image is dramatically modified: more than 50% of the LSB is switched.

B.1.6/ EVALUATION OF THE EMBEDDINGS

A Receiver Operating Characteristic (ROC) approach has finally been implemented in [BCF+13], to find the most adapted threshold w.r.t. the separation between watermarked images and other ones.

Figure B.5 recalls the obtained ROC curve. This latter is close to the ideal one that is without False Positive and False Negative answer. The threshold with best results is a distance equal to 0.97. With such a value, we can give some confidence intervals for most of evaluated attacks. The approach is resistant to all the cropping where percentage is less than 90%, to a JPEG2000 compression where quality ratio is greater than 5%, to all the rotation attacks, and to LSB erasing when less than 95% LSBs are set to 0.



(a) LSB erasing effects (b) Applying algorithm twice

Figure B.4: LSB Modifications

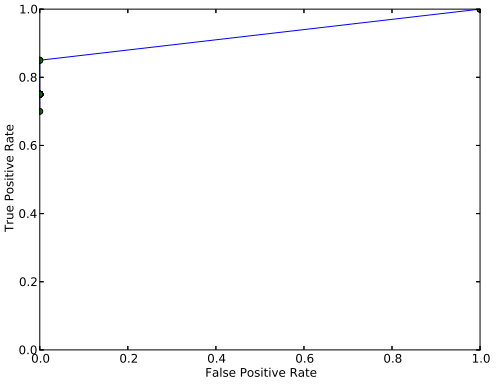


Figure B.5: ROC Curves for DWT or DCT Embeddings

B.1.7/ LYAPUNOV EVALUATION OF $\mathcal{CIS}_2$

We finally close the study of the $\mathcal{CIS}_2$ process by recalling the way we evaluated its Lyapunov exponent in Secrypt13 [BFG13].

B.1.7.1/ A TOPOLOGICAL SEMI-CONJUGACY BETWEEN $\mathcal{X}_2$ AND $\mathbb{R}$

In this section, by using a topological semi-conjugacy, we recall that $\mathcal{CIS}_2$ modeled by $\mathcal{G}_{f_0}$ on $\mathcal{X}$ can be described as iterations on a real interval. To do so, new notations and terminologies must be introduced.

Let $\mathcal{X}_{(\mathsf{N};\mathsf{P})} = \mathbb{S}_N \times \mathbb{B}^{\mathsf{N}} \times \mathbb{S}_P \times \mathbb{B}^{\mathsf{P}} \times \mathbb{S}_P$. In what follows and for easy understanding, we will assume that $\mathsf{N} = 3$ and $\mathsf{P} = 2$. So $\mathsf{N} + \mathsf{P} = 5$ and $\mathsf{NP}^2 = 12$. However, an equivalent formulation of the following can be easily obtained by replacing the bases 5 and 12 by any base $(\mathsf{N} + \mathsf{P})$ and (NP^2). N has only to be greater than P.

Definition [55]. *The function* $\psi : [\![1, \mathsf{N}]\!] \times [\![1, \mathsf{P}]\!] \times [\![1, \mathsf{P}]\!] \to [\![0, \mathsf{NP}^2 - 1]\!]$ *is defined by:* $\psi\left(S_p^i, S_c^i, S_m^i\right) = (S_p^i - 1)\mathsf{P}^2 + (S_c^i - 1)\mathsf{P} + (S_m^i - 1)$.

This function aims to convert a strategy of triplets in a simple strategy of integers expressed in a different basis, see Table B.1. Obviously, ψ is a bijective function, the reverse operation will be denoted by ψ^{-1}. The three projections of ψ^{-1} are denoted by: $\psi_1^{-1}\left(\psi\left(S_p^i, S_c^i, S_m^i\right)\right) = S_p^i$, $\psi_2^{-1}\left(\psi\left(S_p^i, S_c^i, S_m^i\right)\right) = S_c^i$, and $\psi_3^{-1}\left(\psi\left(S_p^i, S_c^i, S_m^i\right)\right) = S_m^i$.

Base N = 3	Base P = 2	Base P = 2	Base $NP^2 = 12$
S_p^i	S_c^i	S_m^i	$\psi\left(S_p^i, S_c^i, S_m^i\right)$
1	1	1	0
1	1	2	1
1	2	1	2
1	2	2	3
2	1	1	4
2	1	2	5
2	2	1	6
2	2	2	7
3	1	1	8
3	1	2	9
3	2	1	10
3	2	2	11

Table B.1: Some values for ψ (see Definition 55).

Definition 56. *Let us define* $\varphi : \mathcal{X}_{(3;2)} = \mathbb{S}_3 \times \mathbb{B}^3 \times \mathbb{S}_2 \times \mathbb{B}^2 \times \mathbb{S}_2 \longrightarrow \left[0, 2^5\right[$*, as follows. If* $(S_p, E, S_c, M, S_m) = \left((S_p^0, S_p^1, \ldots); (E_0, E_1, E_2, E_3); (S_c^0, S_c^1, \ldots); (M_0, M_1); (S_m^0, S_m^1, \ldots)\right)$*, then* $\varphi\left(S_p, E, S_c, M, S_m\right)$ *is the real number:*

- *whose integral part* e *is* $\sum_{k=0}^{2} 2^{4-k} E_k + \sum_{k=3}^{4} 2^{4-k} M_{k-3}$*, that is, the binary digits of* e *are* $E_0\ E_1\ E_2\ M_0\ M_1$*.*
- *whose decimal part* s *is equal to:* $s = 0, \psi\left(S_p^0, S_c^0, S_m^0\right)\ \psi\left(S_p^1, S_c^1, S_m^1\right)\ \psi\left(S_p^2, S_c^2, S_m^2\right)\ \ldots = \sum_{k=1}^{+\infty} 12^{-k} S^{k-1}$*.* s *is thus expressed in base* 12*.*

As notified in [BFG13], φ realizes the association between a point of $\mathcal{X}_{(3;2)}$ and a real number into $\left[0, 2^5\right[$. We must now translate the steganographic process $\mathcal{CIS}_2$, which is represented by $\mathcal{G}_{f_0}$, as iterations on this real interval. To do so, two intermediate functions over $\left[0, 2^5\right[$ denoted by e and s has been introduced in [BFG13].

Definition 57. *Let* $x \in \left[0, 2^5\right[$ *and:*

- $e_0, \ldots, e_4$ *the binary digits of the integral part of* x*:* $\lfloor x \rfloor = \sum_{k=0}^{4} 2^{4-k} e_k$*.*
- $(s^k)_{k \in \mathbb{N}}$ *the digits of* x*, expressed in base* 12*, where the chosen decimal decomposition of* x *is the one that does not have an infinite number of* 11*:* $x = \lfloor x \rfloor + \sum_{k=0}^{+\infty} s^k 12^{-k-1}$*.*

e and s are thus defined as follows:

$$\begin{array}{rccc} e: & \big[0,2^5\big[& \longrightarrow & \mathbb{B}^3\times\mathbb{B}^2 \\ & x & \longmapsto & ((e_0,e_1,e_2);(e_3,e_4)) \end{array}$$

and

$$\begin{array}{rccc} s: & \big[0,2^5\big[& \longrightarrow & [\![0,11]\!]^{\mathbb{N}} \\ & x & \longmapsto & (s^k)_{k\in\mathbb{N}} \end{array}$$

We have thus been able to define the function g, whose goal is to translate the steganographic process $\mathcal{CIS}_2$ represented by $\mathcal{G}_{f_0}$ on an interval of $\mathbb{R}$ [BFG13].

Definition 58. *$g:\big[0,2^5\big[\longrightarrow\big[0,2^5\big[$ is such that $g(x)$ is the real number of $\big[0,2^5\big[$ defined below:*

- *its integral part has a binary decomposition equal to $e'_0,\ldots,e'_4$, with $\forall i\in[\![0,2]\!]$:*

$$e'_i=\begin{cases} e(x)_i & \text{if } i\neq\psi_1^{-1}\left(s^0\right)\\ e(x)_{2+\psi_2^{-1}(s^0)} & \text{if } i=\psi_1^{-1}\left(s^0\right)\end{cases}$$

  and $\forall i\in[\![3,4]\!]$:

$$e'_i=\begin{cases} e(x)_i & \text{if } i\neq\psi_3^{-1}\left(s^0\right)\\ e(x)_i+1 \ (mod\ 2) & \text{if } i=\psi_3^{-1}\left(s^0\right),\end{cases}$$

- *whose decimal part is $s(x)^1,s(x)^2,\ldots$*

In other words, if $x=\sum_{k=0}^{4}2^{4-k}e_k+\sum_{k=0}^{+\infty}s^k\ 12^{-k-1}$, then:

$$g(x)=\sum_{k=0}^{2}2^{4-k}\left[e_k\left(\delta(k,\psi_1^{-1}(s^0))+1\ (\mathrm{mod}\ 2)\right)+e_{2+\psi_2^{-1}(s^0)}\left(\delta(k,\psi_1^{-1}(s^0))\right)\right]$$

$$+\sum_{k=3}^{4}2^{4-k}(e_k+\delta(k,\psi_3^{-1}(s^0)\ (\mathrm{mod}\ 2))+\sum_{k=0}^{+\infty}s^{k+1}12^{-k-1},$$

where δ is the discrete Boolean metric introduced previously.

Numerous metrics can be defined on the set $\big[0,2^5\big[$, the most usual one being the Euclidian distance $\Delta(x,y)=|y-x|^2$. However, this Euclidian distance does not reproduce exactly the notion of proximity induced by distance d_2 on $\mathcal{X}_2$ introduced in a previous section, which is more relevant for the targeted applications. Indeed d_2 is richer than Δ, this is why we have introduced the following map in [BFG13].

Definition 59. *Given $x,y\in\big[0,2^5\big[$, D denotes the function from $\big[0,2^5\big[^2$ to $\mathbb{R}^+$ defined by: $D(x,y)=D_e\left(e(x),e(y)\right)+D_s\left(s(x),s(y)\right)$, where:*

$$D_e(e,\check{e})=\sum_{k=0}^{4}\delta(e_k,\check{e}_k),\quad \text{and}\quad D_s(s,\check{s})=\sum_{k=1}^{\infty}\frac{|s^k-\check{s}^k|}{12^k}.$$

We have thus proven in [BFG13] that,

Proposition 27. *D is a distance on $\big[0,2^5\big[$.*

The convergence of sequences according to D is not the same than the usual convergence related to the Euclidian metric. For instance, if $x^n \to x$ according to D, then necessarily the integral part of each x^n is equal to the integral part of x (at least after a given threshold), and the decimal part of x^n corresponds to the one of x "as far as required". D is richer and more refined than the Euclidian distance, and thus is more precise.

φ has been constructed in order to be continuous and onto, so we obtained the following theorem in [BFG13].

Theorem 25. *The steganographic process $\mathcal{CIS}_2$ represented by $(\mathcal{G}_{f_0}, \mathcal{X}_2)$ can be considered as simple iterations on $\mathbb{R}$, which is illustrated by the semi-conjugacy given below:*

$$\begin{array}{ccc} \left(\mathcal{X}_{(3;2)}, d_2 \right) & \xrightarrow{\mathcal{G}_{f_0}} & \left(\mathcal{X}_{(3;2)}, d_2 \right) \\ \varphi \downarrow & & \downarrow \varphi \\ \left(\left[0, 2^5\right[, D \right) & \xrightarrow[g]{} & \left(\left[0, 2^5\right[, D \right) \end{array}$$

In other words, $\mathcal{X}_2$ is approximately equal to $\left[0, 2^{\mathsf{N}+\mathsf{P}}\right[$. We have thus remarked in [BFG13] that,

Proposition 28. *The process $\mathcal{CIS}_2$ represented by g defined on $\mathbb{R}$ has derivatives of all orders on $\left[0, 2^5\right[$, except on the 385 points in I defined by: $I = \left\{ \frac{n}{12} \ / \ n \in [\![0; 2^5 \times 12]\!] \right\}$.*

Furthermore, on each interval of the form $\left[\frac{n}{12}, \frac{n+1}{12} \right[$, with $n \in [\![0; 2^5 \times 12[\![$, g is a linear function having a slope equal to 12: $\forall x \notin I, g'(x) = 12$.

We are now able to recall the way to evaluate the Lyapunov exponent of $\mathcal{CIS}_2$.

B.1.7.2/ TOPOLOGICAL SECURITY OF $\mathcal{CIS}_2$ ON $\mathbb{R}$

$\mathcal{CIS}_2$ represented by the function $\mathcal{G}_{f_0}$ on $\mathcal{X}_2$ is topologically secure, that is to say $(\mathcal{G}_{f_0}, \mathcal{X}_2)$ is chaotic in the sense of Devaney. We can deduce the same property for $\mathcal{CIS}_2$ represented by the g function on $\mathbb{R}$ for the order topology. Indeed $(\mathcal{G}_{f_0}, \mathcal{X}_2)$ and $\left(g, [0, 2^5[_D\right)$ are semi-conjugate by φ as recalled below. So $\left(g, [0, 2^5[_D\right)$ is a chaotic system according to Devaney, because the semi-conjugacy preserves this character [For98]. However the topology generated by D is finer than the topology generated by the Euclidean distance Δ, which is the order topology. This is why we have proven in [BFG13] that,

Theorem 26. *Let $\mathcal{X}$ be a set, and τ, τ' two topologies on $\mathcal{X}$ such that τ' is finer than τ. Let $f : \mathcal{X} \to \mathcal{X}$, continue for both τ and τ'.*

If $(\mathcal{X}_{\tau'}, f)$ is chaotic in the sense of Devaney, then $(\mathcal{X}_{\tau}, f)$ is also chaotic.

Finally, according to Theorem 26, we have deduced in [BFG13] that the steganographic process $\mathcal{CIS}_2$ represented by g is chaotic in the sense of Devaney, for the order topology on $\mathbb{R}$. Having these assertions in mind, we have then formulated the following theorem:

Theorem 27. *The steganographic process $\mathcal{CIS}_2$ represented by g on $\mathbb{R}$ is chaotic in the sense of Devaney, when the usual topology of $\mathbb{R}$ is used (the order topology).*

This result is weaker than Theorem 24, which establishes the chaotic property of $\mathcal{CIS}_2$ for a finer topology. It is as if the chaos observed using usual tools like the Euclidian distance is still preserved when considering more powerful tools (higher resolution, *i.e.*, finer topologies). The result contained in Theorem 27 is however interesting, as it confirms that approach followed in [BFG13] does not lead to deflated properties.

Indeed, our studies take place in a system other than the one usually considered in computer science ($\mathcal{X}_2$ instead of $\mathbb{R}$), in order to be as close as possible to the targeted computer machines. By doing so, we prevent from any loss of chaotic properties when computing the scheme written in mathematical terms. However, it might be feared that the choice of a discrete mathematics approach leads to a disorder of lower quality. In other words, perhaps we have avoided a situation of great disorder *lost* during the computation into finite machines. But the cost of such success may be to obtain a weaker disorder ? Theorem 27 proves exactly the contrary.

B.1.7.3/ Evaluation of the Lyapunov exponent

Let $\mathcal{L} = \left\{x^0 \in \left[0, 2^5\right[\ / \ \forall n \in \mathbb{N}, x^n \notin I\right\}$, where I is the set of points in the real interval where g is not differentiable (as it is explained in Proposition 28). Then [BFG13].

Theorem [28]. *$\forall x^0 \in \mathcal{L}$, the Lyapunov exponent of $\mathcal{CIS}_2$ having x^0 for initial condition is equal to $\lambda(x^0) = \ln(12) > 0$.*

Rem [5]. *The set of initial conditions for which this exponent is not calculable is countable. This is indeed the initial conditions such that an iteration value will be a number having the form $\frac{n}{12}$, with $n \in \mathbb{N}$. Moreover, for a system having* N + P *cells (a number of LSCs equal to* N *and a secret message to embed of width equal to* P*), we will find,* mutatis mutandis*, an infinite uncountable set of initial conditions $x^0 \in \left[0; 2^{\mathsf{N}+\mathsf{P}}\right[$ such that $\lambda(x^0) = \ln(\mathsf{NP}^2)$.*

So, it is possible to make the Lyapunov exponent of the scheme $\mathcal{CIS}_2$ as large as possible, depending on the number of least significant coefficients of the cover media we decide to consider, and on the width of the message to embed. As proven in [GFB10], a large Lyapunov exponent makes it impossible to achieve the well-known "Estimated Original Attacks" [CB08a].

B.2/ The $\mathcal{DI}_3$ Steganographic Process

In [BCFG12a, BCFG12b], a new steganographic algorithm named $\mathcal{DI}_3$ is presented. It is inspired from $\mathcal{CIW}_1$ and $\mathcal{CIS}_2$ respectively published in [FGB11] and [GFB10], and recalled previously in this chapter. Compared to the first one, $\mathcal{DI}_3$ is a steganographic scheme, not just a watermarking technique. That is, in our understanding, it can embed more than one bit. Unlike $\mathcal{CIS}_2$, which requires embedding keys with three strategies, only one sequence is required for $\mathcal{DI}_3$, so it is easier to implement. Indeed $\mathcal{DI}_3$ is a faster instance of $\mathcal{CIS}_2$, as there is no message mixing in it. $\mathcal{DI}_3$ is well-defined mathematically and its security is evaluated in [BCFG12a], whereas [BCFG12b] provides algorithms and investigates its robustness, comparing it to some well-known watermarking schemes, namely the YASS [SSM07], nsF5 [FPK07], MMx [KDR06], and HUGO [PFB10a] algorithms detailed in the Appendix B.3.

B.2.1/ MATHEMATICAL DEFINITIONS AND NOTATIONS

New notations and terminologies must be introduced another time in order to be able to define mathematically the $\mathcal{DI}_3$ steganographic process. They are provided thereafter.

Definition 60. *The* support of a finite sequence S *of* n *terms is the finite set* $\mathsf{S}(S) = \left\{S^k, k < n\right\}$ *containing all the distinct values of* S*. Its cardinality is s.t.* $\#\mathsf{S}(S) \leqslant n$*.*

Definition 61. *A finite sequence* $S \in \mathbb{S}_{\mathsf{N}}$ *of* n *terms is* injective *if* $n = \#\mathsf{S}(S)$*. It is* onto *if* $N = \#\mathsf{S}(S)$*. Finally, it is bijective if and only if it is both injective and onto, so* $n = N = \#\mathsf{S}(S)$*.*

"S is injective" reflects the fact that all the n terms of the sequence S are distinct, while "S is onto" means that all the values of the set $[\![1; \mathsf{N}]\!]$ are reached at least once.

B.2.2/ THE NEW $\mathcal{DI}_3$ PROCESS

In this section, the new algorithm introduced in [BCFG12a] and studied in [BCFG12b] is recalled. Let $\mathsf{P} \in \mathbb{N}^*$ be the width, in term of bits, of the message to embed into the cover media. $\lambda \in \mathbb{N}^*$ is the number of iterations to realize, which is s.t. $\lambda > \mathsf{P}$. $x^0 \in \mathbb{B}^{\mathsf{N}}$ is for the N LSCs of a given cover media C supposed to be uniformly distributed. $m \in \mathbb{B}^{\mathsf{P}}$ is the message to hide into x^0. Finally, $S \in \mathbb{S}_{\mathsf{P}}$ is a strategy such that the finite sequence $\left\{S^k, k \in [\![\lambda - \mathsf{P} + 1; \lambda]\!]\right\}$ is injective.

Rem 6. *The width* P *of the message to hide into the LSCs of the cover media* x^0 *has to be far smaller than the number of LSCs.*

The proposed information hiding scheme is defined by:

Definition 62 ($\mathcal{DI}_3$ Data hiding scheme)**.** $\forall (n, i, j) \in \mathbb{N}^* \times [\![0; \mathsf{N} - 1]\!] \times [\![0; \mathsf{P} - 1]\!]$*:*

$$x_i^n = \begin{cases} x_i^{n-1} & \text{if } S^n \neq i \\ m_{S^n} & \text{if } S^n = i. \end{cases}$$

The stego-content is the Boolean vector $y = x^\lambda \in \mathbb{B}^{\mathsf{N}}$, which will replace the former LSCs, that is, LSCs of the cover media are replaced by the vector y.

B.2.3/ SECURITY STUDY

A security study of the $\mathcal{DI}_3$ steganographic process has been realized in [BCFG12a]. Conclusion of this study is summarized thereafter.

Proposition 29. $\mathcal{DI}_3$ *is stego-secure.*

This proof of this proposition, provided in [BCFG12a], holds for the following restrictive hypotheses:

- **Distribution of LSCs:** We have supposed that $x^0 \sim \mathcal{U}\left(\mathbb{B}^{\mathsf{N}}\right)$ to prove the stego-security of the data hiding process $\mathcal{DI}_3$. This hypothesis of the uniform distribution

of the least significant coefficients is obviously the most restrictive one, but it can be obtained at least partially in two possible manners. Either a channel that appears to be random (for instance, when applying a chi squared test, or for test batteries recalled in a previous chapter) can be found in the media. Or a systematic process can be applied on the images to obtain this uniformity, as follows. Before embedding the hidden message, all the original LSCs must be replaced by randomly generated ones, hoping so that such cover media will be considered to be noisy by any given attacker. Let us remark that, in the field of data anonymity for privacy on the Internet, we are in the "watermark-only attack" framework. As it has been recalled in Table **??**, in that framework, the attacker has only access to stego-contents, having so no knowledge of the original media (*i.e.*, before introducing the message in the LSCs random channel). These considerations, which have been deepened in later publications, will be discussed more largely at the end of this chapter.

- **Distribution of the messages m:** In order to prove the stego-security of the data hiding process $\mathcal{DI}_3$, we have supposed that $m \sim \mathcal{U}\left(\mathbb{B}^{\mathsf{P}}\right)$. This hypothesis of the uniform distribution of the message to hide is not really restrictive. Indeed, to encrypt the message before its embedding into the LSCs of cover media, which is usually required for obvious security reasons, is sufficient to achieve this goal. To say it different, in order to be in the conditions of applications of the process $\mathcal{DI}_3$, the hidden message must be encrypted.

- **Distribution of the strategies S:** To prove the stego-security of the data hiding process $\mathcal{DI}_3$, we have finally supposed that $S \sim \mathcal{U}\left(\mathbb{S}_{\mathsf{P}}\right)$. This hypothesis is not restrictive too, as any cryptographically secure pseudorandom generator (PRNG) satisfies this property. With such PRNGs, it is impossible in polynomial time, to make the distinction between random numbers and numbers provided by these generators. For instance, *Blum Blum Shub (BBS)* [Jun99], *Blum Goldwasser (BG)* [VV85], or *ISAAC* [Jen96], recalled in the chapter focusing on PRNGs, are convenient here.

After this theoretical study of the $\mathcal{DI}_3$ steganographic process realized in [BCFG12a], we have investigated practical aspects, discussing about its concrete implementation and evaluating its robustness in [BCFG12b], while article [BCFG12a] already mentioned deals with its ability to face steganalyzers. These practical aspects are summarized below.

B.2.4/ IMPLEMENTING THE $\mathcal{DI}_3$ SCHEME

In the algorithms recalled here, the following notations are used: S denotes the embedding and extraction strategy, H the host content or the stego-content depending of the context. LSC stands for the old or new LSCs of the host or stego-content H depending of the context too. N denotes the number of LSCs, λ the number of iterations to realize, M the secret message, and P the width of the message (number of bits).

The $\mathcal{DI}_3$ scheme theoretically presented in [BCFG12a] has been practically described by three main algorithms in [BCFG12b]:

1. Algorithm 8 generates the embedding strategy, part of the embedding key (with the LSCs and the number of iterations).

2. Algorithm 9 embeds the message into the LSCs of the cover media using the strategy. The strategy has been generated by the first algorithm and the same number of iterations is used.

3. Algorithm 10 extracts the secret message from the LSCs of the media (the stego-content) using the strategy, which constitutes with the message length the extraction key.

Two other complementary functions must be used:

1. Algorithm 11, which allows to extract MSCs, LSCs, and passive coefficients from the host content. Its implementation is based on the concept of signification function described previously.

2. Algorithm 12 rebuilds the new host content (the stego-content) from the corresponding MSCs, LSCs, and passive coefficients. This function realizes the opposite operation of Algorithm 11.

Rem [7]. *These two algorithms depend of the definition of the MSCs, LSCs, and passive coefficients, which can correspond to a spatial or frequency description of the host content. This is why they are not documented here.*

Algorithm 8: $strategy(N, P, \lambda)$

`/* S is a sequence of integers into` $[\![0, P-1]\!]$`, such that` $(S_{n_0}, \ldots, S_{n_0+P-1})$ `is injective on` $[\![0, P-1]\!]$`. */`

Result: S: The strategy, integer sequence $(S_0, S_1, \ldots)$.

begin

- $n_0 \longleftarrow L - P + 1$;
- **if** $P > N$ *OR* $n_0 < 0$ **then**
  - **return** *ERROR*
- $S \longleftarrow$ Array of width λ, all values initialized to 0;
- $cpt \longleftarrow 0$;
- **while** $cpt < n_0$ **do**
  - $S_{cpt} \longleftarrow$Random integer in $[\![0, P-1]\!]$.;
  - $cpt \longleftarrow cpt + 1$;
- $A \longleftarrow$ We generate an arrangement of $[\![0, P-1]\!]$;
- **for** $k \in [\![0, P-1]\!]$ **do**
  - $S_{n_0+k} \longleftarrow A_k$;
- **return** S

B.2.5/ EVALUATION AGAINST STEGANALYZERS

The steganographic scheme detailed in [BCFG12a] has been compared to state of the art steganographic approaches, namely YASS [SSM07], HUGO [PFB10a], and nsF5 [FPK07] detailed in the Appendix B.3. This study, realized in [BCFG12a], is summarized thereafter.

Algorithm 9: $embed(LSC, M, S, \lambda)$

Result: New LSCs with embedded message.

begin

$N \longleftarrow$ Number of LSCs in LSC;

$P \longleftarrow$ Width of the message M;

for $k \in [\![0, \lambda]\!]$ **do**

$i \longleftarrow S_k$;

$LSC_i \longleftarrow M_i$;

return LSC

Algorithm 10: $extract(LSC, S, \lambda, P)$

Result: The message to extract from LSC.

begin

$RS \longleftarrow$ The strategy S written in reverse order.;

$M \longleftarrow$ Array of width P, all values initialized to 0;

for $k \in [\![0, \lambda]\!]$ **do**

$i \longleftarrow RS_k$;

$M_i \longleftarrow LSC_i$;

return M

The steganalysis is based on the BOSS image database [BFP11], which consists in a set of 10 000 512x512 greyscale images. We have randomly selected 50 of them to compute the cover set. Since YASS and nsF5 are dedicated to JPEG support, all these images have been firstly translated into JPEG format thanks to the `mogrify` command line. To allow the comparison between steganographic schemes, the relative payload is always set with 0.1 bit per pixel. Under that constrain, the embedded message m is a sequence of 26214 randomly generated bits. This step has led to distinguish four sets of stego contents, one for each steganographic approach.

We have next used in [BCFG12a] the steganalysis tool developed by the HugoBreakers team [KF11,KFH11] based on AI classifier and which won the BOSS competition [BFP11]. Table B.2 summarizes these steganalysis results expressed as the error probabilities of the steganalyser, as they are given in [BCFG12a]. The errors are the mean of the false alarms and of the missed detection. An error that is closed to 0.5 signifies that deciding whether an image contains a stego content is a random choice for the steganalyser. Conversely, a tiny error denotes that the steganalyser can easily classify stego content and non stego content.

The best result is obtained by HUGO, which is closed to the perfect steganographic approach to the considered steganalyser, since the error is about 0.5. However, even if the approach detailed in [BCFG12a] has no optimization, these first experiments shown promising results.

B.2.6/ ROBUSTNESS STUDY

This section summarizes the robustness study presented in [BCFG12b]. Each experiment is build another time on a set of 50 images, which are randomly selected among

Algorithm 11: $significationFunction(H)$

Data: H: The original host content.
Result: MSC: MSCs of the host content H.
Result: PC: Passive coefficients of the host content H.
Result: LSC: LSCs of the host content H.
begin
    `/* Implemented by the user. */`
    return (MSC, PC, LSC)

Algorithm 12: $buildFunction(MSC, PC, LSC)$)

Result: H: The new rebuilt host content.
begin
    `/* Implemented by the user. */`
    return (MSC, PC, LSC)

database taken from the BOSS contest [BFP11]. Each cover is a 512×512 greyscale digital image. The relative payload is always set with 0.1 bit per pixel. Under that constrain, the embedded message m still remains a sequence of 26214 randomly generated bits.

According to previous similar work in the field of information hiding, we have conducted in [BCFG12b] our evaluation following a same canvas than other robustness studies documented previously in this chapter. We have firstly chosen some classical attacks like cropping, compression, and rotation ones. The robustness of $\mathcal{DI}_3$ has then been tested by successively applying on stego content these attacks. Differences between the message that is extracted from the attacked image and the original one are then computed and expressed as percentage.

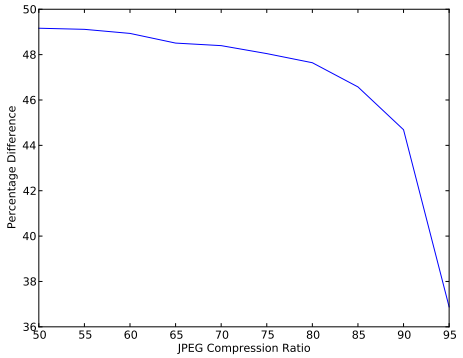
Different percentage of cropping (from 1% to 81%) have been firstly applied on the stego image in [BCFG12b], Fig. B.6 (c) recalls effects of such attacks. We have then addressed robustness against JPEG and JPEG 2000 compression, and results are summarized in Fig. B.6 (a-b). Attacks based on geometric transformations have finally been addressed through rotations: as presented previously in this chapter, two opposite rotations of angle θ are successively applied around the center of the image. In these geometric transformations, angles range from 2 to 20 degrees. Effects of such attacks are also recalled in Fig. B.6 (d).

From all these experiments, one can conclude that the steganographic scheme does not present obvious drawback and resists to all the attacks: all the percentage differences are so far less than 50%.

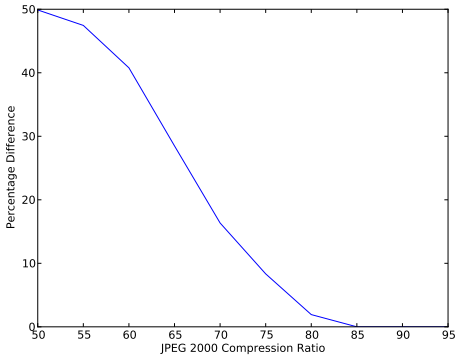
All researches presented in previous sections have started from the $\mathcal{CIW}_1$ process, proceeding by successively correcting its drawbacks. By doing so, we have had a retreat from chaotic iterations. At the same time, the chaotic iterations based information hiding (dhCI) process, whose the $\mathcal{CIW}_1$ scheme historically arises from, continued to be investigated in parallel. Results of these investigations are detailed in the next section.

Steganographic Tool	$\mathcal{DI}_3$	YASS	HUGO	NsF5
Error Probability	0.4133	0.0067	0.495	0.47

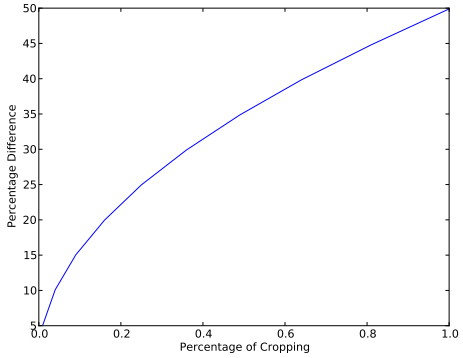
Table B.2: Steganalysis results of HugoBreakers steganalyser applied on steganographic scheme



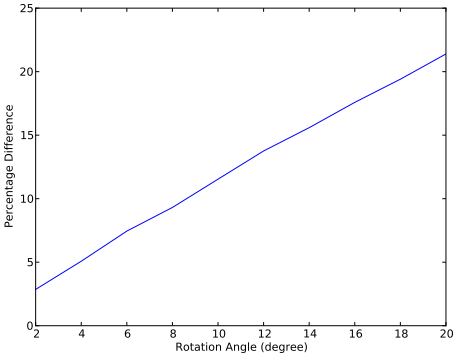
(a) JPEG effect.



(b) JPEG 2000 effect.



(c) Cropping attack.



(d) Rotation attack.

Figure B.6: Robustness of $\mathcal{DI}_3$ scheme facing several attacks (50 images from the BOSS repository)

B.3/ Some Well-known Steganographic Schemes

We recall in this appendix some state of the art information hiding schemes. One should find more details in [Fri09].

B.3.1/ YASS

YASS (*Yet Another Steganographic Scheme*) [SSM07] is a steganographic approach dedicated to JPEG cover. The main idea of this algorithm is to hide data into 8×8 randomly chosen inside $B \times B$ blocks (where B is greater than 8) instead of choosing standard 8×8 grids used by JPEG compression. The self-calibration process commonly embedded into blind steganalysis schemes is then confused by the approach. In the paper [SSM], further variants of YASS have been proposed simultaneously to enlarge the embedding rate

and to improve the randomization step of block selecting. More precisely let be given a message m to hide, a size B, $B \geqslant 8$, of blocks . The YASS algorithm follows:

1. computation of m' which is the Repeat-Accumulate error correction code of m
2. in each big block of size $B \times B$ of cover, successively:
   (a) random selection of an 8×8 block b using w.r.t. a secret key.
   (b) two-dimensional DCT transformation of b and normalisation of coefficient w.r.t a predefined quantization table. Matrix is further referred to as b'.
   (c) a fragment of m' is embedded in some LSB of b'. Let b'' be the resulting matrix.
   (d) The matrix b'' is decompressed back to the spatial domain leading to a new $B \times B$ block.

B.3.2/ nsF5

The nsF5 algorithm [FPK07] extends the F5 algorithm [Wes01]. Let us first have a closer look on this latter

First of all, as far as we know, F5 is the first steganographic approach that solves the problem of remaining unchanged a part (often the end) of the file. To achieve this, a subset of all the LSB is computed thanks to a pseudorandom number generator seeded with a user defined key. Next, this subset is split into blocks of x bits. The algorithm takes benefit of binary matrix embedding to increase it efficiency. Let us explain this embedding on a small illustrative example where a part m of the message has to be embedded into this x LSB of pixels which are respectively a 3 bits column vector and a 7 bits column vector. Let then H be the binary Hamming matrix

$$H = \begin{pmatrix} 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 \end{pmatrix}$$

The objective is to modify x to get y s.t. $m = Hy$. In this algebra, the sum and the product respectively correspond to the exclusive *or* and to the *and* Boolean operators. If Hx is already equal to m, nothing has to be changed and x can be sent. Otherwise we consider the difference $\delta = d(m, Hx)$ which is expressed as a vector :

$$\delta = \begin{pmatrix} \delta_1 \\ \delta_2 \\ \delta_3 \end{pmatrix} \text{ where } \delta_i \text{ is } 0 \text{ if } m_i = Hx_i \text{ and } 1 \text{ otherwise.}$$

Let us thus consider the jth column of H which is equal to δ. We denote by $\overline{x}^j$ the vector we obtain by switching the jth component of x, that is, $\overline{x}^j = (x_1, \ldots, \overline{x_j}, \ldots, x_n)$. It is not hard to see that if y is $\overline{x}^j$, then $m = Hy$. It is then possible to embed 3 bits in only 7 LSB of pixels by modifying on average $1 - 2^3$ changes. More generally, the F5 embedding efficiency should theoretically be $\frac{p}{1-2^p}$.

However, the event when the coefficient resulting from this LSB switch becomes zero (usually referred to as *shrinkage*) may occur. In that case, the recipient cannot determine whether the coefficient was -1, +1 and has changed to 0 due to the algorithm or was

initially 0. The F5 scheme solves this problem first by defining a LSB with the following (not even) function:

$$LSB(x) = \begin{cases} 1 - x \mod 2 \text{ if } x < 0 \\ x \mod 2 \text{ otherwise.} \end{cases} .$$

An next, if the coefficient has to be changed to 0, the same bit message is re-embedded in the next group of x coefficient LSB.

The scheme nsF5 focuses on steps of Hamming coding and ad'hoc shrinkage removing. It replaces them with a *wet paper code* approach that is based on a random binary matrix. More precisely, let D be a random binary matrix of size $x \times n$ without replicate nor null columns: consider for instance a subset of $\{1, 2^x\}$ of cardinality n and write them as binary numbers. The subset is generated thanks to a PRNG seeded with a shared key. In this block of size x, one choose to embed only k elements of the message m. By abuse, the restriction of the message is again called m. It thus remains $x - k$ (wet) indexes/places where the information shouldn't be stored. Such indexes are generated too with the keyed PRNG. Let v be defined by the following equation

$$Dv = \delta(m, Dx). \tag{B.1}$$

This equation may be solved by Gaussian reduction or other more efficient algorithms. If there is a solution, one have the list of indexes to modify into the cover. The nsF5 scheme implements such a optimized algorithm that is to say the LT codes.

B.3.3/ MMx

Basically, the MMx algorithm [KDR06] embeds message in a selected set of LSB cover coefficients using Hamming codes as the F5 scheme. However, instead of reducing as many as possible the number of modified elements, this scheme aims at reducing the embedding impact. To achieve this it allows to modify more than one element if this leads to decrease distortion.

Let us start again with an example with a $[7, 4]$ Hamming codes, *i.e*, let us embed 3 bits into 7 DCT coefficients, $D_1, \ldots, D_7$. Without details, let $\rho_1, \ldots, \rho_7$ be the embedding impact whilst modifying coefficients $D_1, \ldots, D_7$ (see [KDR06] for a formal definition of ρ). Modifying element at index j leads to a distortion equal to ρ_j. However, instead of switching the value at index j, one should consider to find all other columns of H, j_1, j_2 for instances, s.t. the sum of them is equal to the jth column and to compare ρ_j with $\rho_{j_1} + \rho_{j_2}$. If one of these sums is less than ρ_j, the sender has to change these coefficients instead of the j one. The number of searched indexes (2 for the previous example) gives the name of the algorithm. For instance in MM3, one check whether the message can be embedded by modifying each time 3 pixel or less.

B.3.4/ HUGO

The HUGO [PFB10a] steganographic scheme is mainly designed to minimize distortion caused by embedding. To achieve this, it is firstly based on an image model given as SPAM [PBF10] features and next integrates image correction to reduce much more distortion. What follows discuss on these two steps.

The former first computes the SPAM features. Such calculi synthesize the probabilities that the difference between consecutive horizontal (resp. vertical, diagonal) pixels belongs in a set of pixel values which are closed to the current pixel value and whose radius is a parameter of the approach. Thus a fisher linear discriminant method defines the radius and chooses between directions (horizontal, vertical...) of analyzed pixels that gives the best separator for detecting embedding changes. With such instantiated coefficients, HUGO can synthesize the embedding cost as a function $D(X, Y)$ that evaluates distortions between X and Y. Then HUGO computes the matrices of $\rho_{i,j} = \max(D(X, X^{(i,j)+})_{i,j}, D^-(X, X^{(i,j)-})_{i,j})$ such that $X^{(i,j)+}$ (resp. $X^{(i,j)+}$) is the cover image X where the the (i, j)th pixel has been increased (resp. has been decreased) of 1.

The order of modifying pixel is critical: HUGO surprisingly modifies pixels in decreasing order of $\rho_{i,j}$. Starting with $Y = X$, it increases or decreases its (i, j)th pixel to get the minimal value of $D(Y, Y^{(i,j)+})_{i,j}$ and $D^-(Y, Y^{(i,j)-})_{i,j}$. The matrix Y is thus updated at each round.

C

APPLICATION TO HASH FUNCTIONS

Hash functions are cryptographic tools involved, among other things, in integrity checking and password storage. They are of prior importance to improve security of exchanges through the Internet. However, as security flaws are regularly identified in standards in this domain, new ways to hash digital data must always be investigated.

During our thesis, we have initiated the use of chaotic iterations for hash functions [BG10a, BG10d]. The idea was to compose a new hash function by mixing elements of the SHA-1 and of chaotic iterations. The first work after our thesis was to rationalize and simplify this hash function, and to study its behavior computationally: diffusion and confusion have been obtained, justifying in our opinion the interest of adding chaos to existing hash functions. Since these first investigations, and due to our reflections on PRNG post-treatments as presented in Chapter 4, our research works have then taken another direction: instead of creating from scratch a new hash function, our approach is now to realize a post-treatment on existing hash functions that preserves their properties while adding chaos.[1]

C.1/ INTRODUCTION

Security or privacy of data exchanged through the Internet are guaranteed by protocols that make an adequate use of a few cryptographic tools as secure pseudorandom number generators or hash functions. Hash functions are applications that map words of any lengths to words of fixed lengths (often 256 or 512 bits). These hash functions allow, for instance, to store passwords in a secure manner or to check whether a download has occurred without any error. They can be designed to depend from a given parameter, called a key. According to their field of application, the requirements an hash function has to satisfy can change. They need at least: to be very fast, such that the diffusion of the digest into the set of hash values occurs (whatever the bias into the inputted message), and such that a link between a message and its digest is impossible to establish in practice (confusion). The possibility to use a key or to distribute the computation on numerous threads must often be offered in several applications. Finally, in the computer security field, stringent complexity properties have to be proven, namely the collision, first-preimage, and second-preimage resistances, the unpredictability, and the pseudo-

[1] This review chapter summarizes partially my research works regarding hash functions, which has led to 3 journal and 2 conference articles, 3 of them being published after my thesis [BCG11b, GB12, BCG12a]. They have been realized in collaborations with Jean-François Couchot, and Jacques Bahi.

randomness properties. Each of these latter have a rigorous formulation in terms of polynomial indistinguishability.

Several hash functions have been proposed as candidates to be a standard in computer science. Such standards are designed by the scientific community and selected, after peer studies, by administrations as the NIST one (National Institute for Standards and Technologies of the US government). SHA-1 is probably the most widely used hash functions. It is present in a large panel of security applications and protocols through the Internet. However, in the last decade, security flaws have been detected in this latter. As the SHA-2 variants are algorithmically close to SHA-1 and produce finally message digests on principles similar to the MD4 and MD5 message digest algorithms, a new hash standard has been defined during the SHA-3 contest.

Inspired by this contest, and as standards always finish to be broken which requires to always regard new investigative directions, we have formerly proposed our own family of hash functions during our thesis [BG10a, BG10d]. Being designed by using discrete dynamical systems, and taking benefits from the various established topological properties recalled previously, this new family of hash functions that mix SHA-1 and chaotic iterations is thus based on a different approach. In the first publications after our thesis, this new family has been dramatically simplified and studied both theoretically and practically [BCG11b, GB12, BCG12a]. However, creating from scratch a new hash function is a very hard task that requires strong knowledge in this field. Additionally, our more recent researches in the fields of chaotic finite state machines and of chaotic iterations based pseudorandom number generators have emphasized the interest to realize a post-treatment on existing objects, in order to add chaos properties while preserving existing properties. This is why in our most recent proposals, we have added an ingredient of chaos to existing hash functions, in order to reinforce their properties.

As in other fields of information security, the use of chaos to design hash functions is often disputed, for the following reasons [sZlC97, GWHC09]. Methods of existing chaos-based hash functions [WZZ03, XLW09a, XLW09b, XSL10] usually transform the initial message into its padded fixed length version and then translated into a real number. Next, with a chosen chaotic map, methods set the initial algorithm parameters according to the secret key and start iterations. It is then supposed that the final hash function preserves the properties of chaos. But, in our opinion, this claim is not so evident. Moreover, even if these algorithms are themselves proven to be chaotic, their implementations on finite machines can result to loss of chaos properties. The main reason already evoked in this manuscript is that chaotic functions (embedded in these researches) only manipulate real numbers, which do not exist in a computer.

The hash function we have proposed since our thesis does not simply integrate chaotic maps into algorithms hoping that the result remains chaotic; we have conceived algorithms and have mathematically proven that they are chaotic. To do both, as in Chapter 4, our theory and our implementation are based on finite integer domains and finite states iterations, where only one randomly chosen element is modified at each step. By doing so, the complete chaotic behavior of asynchronous chaotic iterations is capitalized to produce a truly chaotic keyed hash function.

This chapter summarizes our researches published in [BCG11b, GB12, BCG12a] and new works under submission. Compared to our thesis investigations [BG10a, BG10d], we have completely rethought, simplified, and fixed some drawbacks in the hash function proposed in our thesis. Then, in a second time, we have rethought it: our approach

consists now in realizing a post-treatment on existing hash functions, to improve their profile, while it was previously to design an hash function from scratch.

C.2/ BACKGROUND SECTION

Definition [63] (Secure Keyed One-Way Hash Function [BSNP96])**.** *Let Γ and Σ be two alphabets, let $k \in K$ be a key in a given key space, let l be a natural numbers which is the length of the output message, and let $h : K \times \Gamma^+ \to \Sigma^l$ be a function that associates a message in Σ^l for each pair of key, word in $K \times \Gamma^+$. The set of all functions h is partitioned into classes of functions $\{h_k : k \in K\}$ indexed by a key k and such that $h_k : \Gamma^+ \to \Sigma^l$ is defined by $h_k(m) = h(k, m)$, i.e., h_k generates a message digest of length l.*

A class $\{h_k : k \in K\}$ is a Secure Keyed One-Way Hash Function *if it satisfies the following properties:*

1. *the function h_k is keyed one-way. That is,*
   (a) *Given k and m, it is easy to compute $h_k(m)$.*
   (b) *Without knowledge of k, it is*
      - *difficult to find m when $h_k(m)$ is given; this property is referred as* preimage resistance*;*
      - *difficult to find $h_k(m)$ when only m is given.*
2. *The function h_k is keyed* collision free*, that is, without the knowledge of k it is difficult to find two distinct messages m and m' s.t. $h_k(m) = h_k(m')$. A weaker version of this property is the* second preimage resistance *which is established if for a given m it is difficult to find another message m', $m \neq m'$, such that $h_k(m) = h_k(m')$.*
3. *Images of function h_k has to be uniformly distributed in Σ^l in order to counter statistical attacks.*
4. *Length l of produced image has to be larger than 128 bits in order to counter birthday attacks.*
5. *Key space size has to be sufficiently large in order to counter exhaustive key search.*

Finally, hash functions have to verify the *avalanche criteria*, which means that a difference of one bit between two given medias has to lead to completely different digest. Intuitively, the topologically transitivity and the sensitivity on initial conditions respectively address the preimage resistance and the avalanche criteria. Section C.4 formalizes this intuition.

The next section presents the hash function we have published and studied in [BCG11b, GB12, BCG12a]. It is based on chaotic iterations and on SHA-1, and it simplifies and rationalizes our first proposal formerly introduced during our thesis.

C.3/ CHAOS-BASED KEYED HASH FUNCTION ALGORITHM

The hash value is obtained as the last configuration resulting from chaotic iterations of G_{f_0}. We then have to define the pair $X^0 = ((S^t)^{t \in \mathbb{B}}, x^0)$, *i.e.*, the strategy and the initial configuration x^0.

C.3.1/ COMPUTING x^0

The first step of the algorithm is to transform the message in a normalized $n = 256$ bits sequence x^0. This size n of the digest can be changed, mutatis mutandis, if needed. Here, this first step is close to the pre-treatment of the SHA-1 hash function, but it can easily be replaced by any other compression method.

To illustrate this step, we take an example, our original text is: "*The original text*".

Each character of this string is replaced by its ASCII code (on 7 bits). Following the SHA-1 algorithm, first we append a "1" to this string, which is then

```
10101001 10100011 00101010 00001101 11111100 10110100 11100111 11010011 10111011
00001110 11000100 00011101 00110010 11111000 11101001
```

Next we append the block 1111000, which is the binary value of this string length (120) and let R be the result. Finally another "1" is appended to R if and only if the resulting length is an even number.

```
10101001 10100011 00101010 00001101 11111100 10110100 11100111 11010011 10111011
00001110 11000100 00011101 00110010 11111000 11101001 1111000.
```

The whole string is copied, but in the opposite direction:

```
10101001 10100011 00101010 00001101 11111100 10110100 11100111 11010011 10111011
00001110 11000100 00011101 00110010 11111000 11101001 11110000 00111110 01011100
01111101 00110010 11100000 10001101 11000011 01110111 00101111 10011100 10110100
11111110 11000001 01010011 00010110 010101.
```

The string whose length is a multiple of 512 is obtained, by duplicating enough this string obtained above, and truncating it at the next multiple of 512. This string is further denoted by D. Finally, we split our obtained string into two blocks of 256 bits and apply to them the exclusive-or function, from the first two blocks to the last one. It results a 256 bits sequence, that is in our example:

```
00001111 00101111 10000010 00111010 00001110 01100111 01111000 10011101 01010111
00110101 11010100 01101001 11111001 00011011 01001110 00110000 11000111 00101101
10001001 11111001 01100010 10111010 11001110 10101011 10010001 11101110 01100111
00000101 11000100 00011111 01001111 00001100.
```

The configuration x^0 is the result of this pre-treatment and is a sequence of $n = 256$ bits. Notice that many distinct texts lead to the same string x^0. The algorithm detailed in [BCG11b, GB12] always appended "1" to the string R. However such an approach suffered from generating the same x^0 when R has length 128. In that case the size of its reverse is again 128 bits leading a message of length 256. When we duplicate the message, we obtain a message of length 512 composed of two equals message. Resulting Xor function is thus 0. This improvement, proposed in [BCG12a], allows thus to avoid this drawback.

Let us build now the strategy $(S^t)^{t\in\mathbb{B}}$ that depends on both the original message and a given key.

C.3.2/ COMPUTING $(S^t)^{t\in\mathbb{B}}$

To obtain the strategy S, the chaotic proven pseudorandom number generator, detailed in [BCGH11] and recalled in Chapter 4, is used. The seed of this PRNG is computed as follows: first the ASCII code (on 7 bits again) of the key is duplicated enough and truncated to the length of D. A xor between D and this chain gives the seed of the PRNG, that is left to generate a finite sequence of natural numbers S^t in $[\![1, n]\!]$ whose length is $2n$.

C.3.3/ COMPUTING THE DIGEST

To design the digest, chaotic iterations of G_{f_0} are realized with initial state $X^0 = ((S^t)^{t\in\mathbb{B}}, x^0)$ as defined above. The result of these iterations is a $n = 256$ bits vector. Its components are taken 4 per 4 bits and translated into hexadecimal numbers, to obtain the hash value:

`AF71542C90F450F6AE3F649A0784E6B16B788258E87654B4D6353A2172838032.`

As a comparison if we replace "*The original text*" by "*the original text*", the hash function returns:

`BAD8789AD6924B6460F8E7686A24A4228486DC8FDCAE15F1F681B91311426056.`

To sum up, this hash function consists in realizing chaotic iterations with the vectorial negation and an initial condition constituted by:

- a compression function for the internal state,
- a prng seeded with the media to hash for the strategy.

We then investigate qualitative properties of this algorithm.

C.4/ QUALITY ANALYSIS

We show in this section that, as a consequence of recalled theoretical results, this hash function tends to verify desired informal properties of a secure keyed one-way hash function [BCG12a, BCG11b].

C.4.1/ THE AVALANCHE CRITERIA

In our opinion, this criteria is implied by the topological properties of sensitive dependence to the initial conditions, expansiveness, and Lyapunov exponent. We recall that a function f has a constant of expansiveness equal to ε if an arbitrarily small error on any initial condition is *always* magnified till ε, while the function G_{f_0} verifies the *expansiveness*

property if there exists any constant $\varepsilon > 0$ such that for any X and Y in $\mathcal{X}$, $X \neq Y$, we can find a $k \in \mathbb{N}$ s.t. $d(G^k_{f_0}(X), G^k_{f_0}(Y)) \geqslant \varepsilon$. We have proven in previous works [GFB10] and recalled in Chapter 2 that $(\mathcal{X}, G_{f_0})$ is an expansive chaotic system. Its constant of expansiveness is equal to 1.

Next, some dynamical systems are highly sensitive to small fluctuations in their initial conditions. The constants of sensitiveness and of expansiveness have been historically defined to illustrate this fact. However, in some cases, these variations can become enormous, can grow in an exponential manner in a few iterations, and neither sensitiveness nor expansiveness are able to measure such a situation, which has led to the introduction of the Lyapunov constant recalled in the first chapter of this hdr. We recall that, by using a topological semi-conjugation between $\mathcal{X}$ and $\mathbb{R}$, we have proven in [Guy10] that, for almost all X^0, the Lyapunov exponent of asynchronous iterations G_{f_0} with X^0 as initial condition is equal to $\ln(n)$.

We can now justify why, in our opinion, the topological properties of the proposed hash function lead to the avalanche effect. Indeed, due to the sensitive dependence to the initial condition, two close media can possibly lead to significantly different digests. The expansiveness property implies that these similar medias mostly lead to very different hash values. Finally, a Lyapunov exponent greater than 1 lead to the fact that these two close media will always finish to have very different digests.

C.4.2/ PREIMAGE RESISTANCE

Let us now recall our topological justifications about the preimage resistance of our keyed hash function denoted by h [BCG11b]. As stated at the beginning of this chapter, an adversary given a target image D should not be able to find a preimage M such that $h(M) = D$. One reason (among many) why this property is important is that on most computer systems user passwords are stored as the cryptographic hash of the password instead of just the plain-text password. Thus an attacker who gains access to the password file cannot use it to then gain access to the system, unless it is able to invert target message digest of the hash function.

We now explain why, topologically speaking, our hash function is resistant to preimage attacks [BCG11b]. Let m be the message to hash, (S, x^0) its normalized version (*i.e.*, the initial state of our chaotic iterations), and $M = h(m)$ the digest of m by using our method. So chaotic iterations with initial condition (S, M) and iterate function G_{f_0} have x^0 as final state. Thus it is impossible to invert the hash process with a view to obtain the normalized message by using the digest. Such an attempt is equivalent to trying to forecast the future evolution of chaotic iterations by only using a partial knowledge of its initial condition. Indeed, as M is known but not S, the attacker has an incertitude on the initial condition. She/he only knows that this value is into an open ball of radius 1 centered at the point M, and the number of terms of such a ball is infinite.

With such an incertitude on the initial condition, and due to the numerous chaos properties possessed by the chaotic iterations (as these stated in Section C.4.1), this prediction is impossible. Furthermore, due to the transitivity property, it is possible to reach all of the normalized medias, when starting to iterate into this open ball. These qualitative explanations can be formulated more rigorously, when considering a more general, post-treatment oriented instance of the proposed hash function. Such a formulation, cor-

responding to our most recent thoughts in this discipline, is provided with the collision resistance property in Section C.6, while the next sections investigate some computational and experimental aspects of security.

C.4.3/ ALGORITHM COMPLEXITY

In this section, we recall the complexity of the hash function published in [BCG11b]:

Theorem [29]. *Let l be the size of the message to hash and n be the size of its hash value. The algorithm detailed along these lines requires $\mathcal{O}(l) + \mathcal{O}(n^2)$ elementary operations to produce the hash value.*

Proof [8]. *See [BCG11b].*

C.5/ EXPERIMENTAL EVALUATIONS

Let us now give some examples of hash values before statistically studying the quality of hash outputs.

C.5.1/ EXAMPLES OF HASH VALUES

Let us consider the proposed hash function with $n = 256$. We consider the key to be equal to "*my key*".

To give illustration of the confusion and diffusion properties, we will use this function to generate hash values in the following cases:

- **Case 1:** The original text message is the poem *Ulalume* (E.A.Poe), which is constituted by 104 lines, 667 words, and 3754 characters.
- **Case 2:** We change *serious* by *nervous* in the verse "*Our talk had been serious and sober*"
- **Case 3:** We replace the last point '.' with a coma ','.
- **Case 4:** In "*The skies they were ashen and sober*", skies becomes Skies.
- **Case 5:** The new original text is the binary value of the Figure C.1.
- **Case 6:** We add 1 to the gray value of the pixel located in position (123,27).
- **Case 7:** We subtract 1 to the gray value of the pixel located in position (23,127).

The corresponding hash values in hexadecimal format are:

- **Case 1:** 0B4730459FBB5E54A18A9CCD676C8396365B0104407D98C866FDAA51A07F0E45,
- **Case 2:** 752E28088150B98166D870BC2417734223A59463D44B83E9808383B30F8B8409,
- **Case 3:** C10EED0A9D44856847F533E5647D0CCD2C58A08643E4D3E5D8FEA0DA0E856760,

Figure C.1: The original plain-image.

- **Case 4:** 52BF23429EC3AD16A0C9DE03DF51C4204466285448D6D73DDFB42E7A839BEE80,
- **Case 5:** 5C639A55E2B26861EB9D8EADDF92F9355B6214ADC01197510586745D47C888B8,
- **Case 6:** E48989D48209143BAE306AC0563FFE31EAB02E5E557B49E3442A840996BECFC1,
- **Case 7:** EC850438A2D8EA95E691C746D487A75512BEE63F4DDB4466C11CD859671DFBEB.

These simulation results are coherent with the topological properties of sensitive dependence to the initial condition, expansiveness, and Lyapunov exponent: any alteration in the message causes a substantial difference in the final hash value.

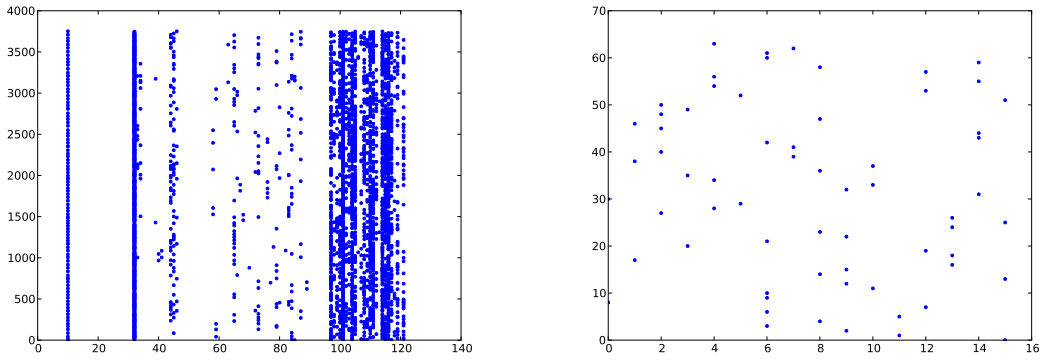
C.5.2/ Statistical evaluation of the algorithm

We focus now on the illustration of the diffusion and confusion properties [Sha49]. Let us recall that confusion refers to the desire to make the relationship between the key and the digest as complex and involved as possible, whereas diffusion means that the redundancy in the statistics of the plain-text must be "dissipated" in the statistics of the cipher-text. Indeed, the avalanche criterion is a modern form of the diffusion, as this term means that the output bits should depend on the input bits in a very complex way. This section summarizes the simulations provided in [BCG12a].

C.5.2.1/ Uniform distribution for hash values

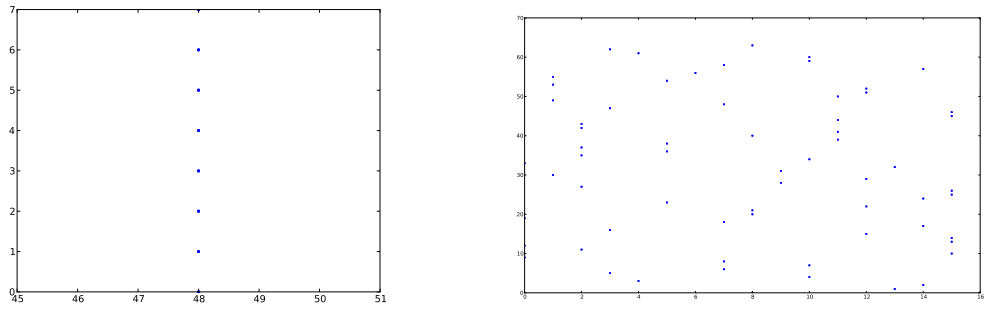
To show the diffusion and confusion properties verified by our scheme, we have firstly given an illustration of the difference of characters distribution between a plain-text and its hash value, when the original message is again the Ulalume poem. Such a distribution is recalled thereafter. In Figure C.2(a), the ASCII codes are localized within a small area, whereas in Figure C.2(b) the hexadecimal numbers of the hash value are more uniformly distributed.

A similar experiment has been realized with a message having the same size, but which is only constituted by the character "*0*". The contrast between the plain-text message and its digest are respectively presented in Figures C.3(a) and C.3(b). Even under this very extreme condition, the distribution of the digest still remains uniform. To conclude, these simulations tend to indicate that no information concerning the original message can be



(a) Original text (ASCII) (b) Digest (Hexadecimal)

Figure C.2: Values distribution of Ulalume poem



(a) Original text (ASCII) (b) Digest (Hexadecimal)

Figure C.3: Values distribution of the "*00000000*" message

found into its hash value, as it is recommended by the Shannon's diffusion and confusion requirements.

C.5.2.2/ BEHAVIOR THROUGH SMALL RANDOM CHANGES

We now consider the following experiment. A first message of 1000 bits is randomly generated, and its hash value of size $n = 256$ bits is computed. Then one bit is randomly toggled into this message and the digest of the new message is obtained. These two hash values are compared by using the hamming distance, to compute the number B_i of changed bits. This test is reproduced $t = 10000$ times. The corresponding distribution of B_i is shown in Figure C.4 [BCG12a].

As desired, Figure C.4 shows that the distribution is centered around 128, which reinforces the confidence put into the good capabilities of diffusion and confusion of the proposed hash algorithm. To analyze these results, the following common statistics have been used in [BCG12a].

- Mean changed bit number $\overline{B} = \frac{1}{t}\sum_{i=1}^{\mathsf{Nt}} B_i$.

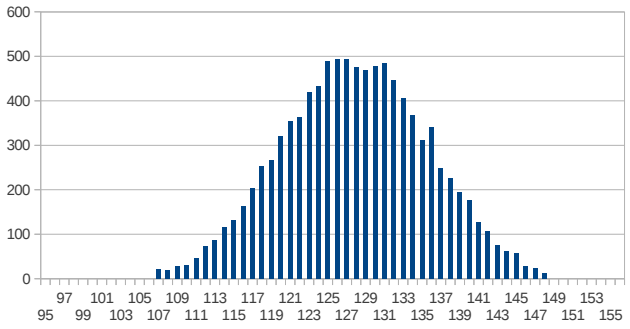


Figure C.4: Histogram

	B_{min}	B_{max}	$\overline{B}$	$P(\%)$	ΔB	$\Delta P(\%)$
$n = 256$	87	167	127.95	49.98	8.00	3.13
$n = 512$	213	306	255.82	49.97	11.29	2.21
$n = 1024$	446	571	511.54	49.96	15.97	1.56

Table C.1: Statistical performance of the proposed hash function

- Mean changed probability $P = \frac{\overline{B}}{n}$.
- $\Delta B = \sqrt{\frac{1}{t}\sum_{i=1}^{t}(B_i - \overline{B})^2}$.
- $\Delta P = \sqrt{\frac{1}{t}\sum_{i=1}^{t}(\frac{B_i}{n} - P)^2}$.

The obtained statistics are listed in Table C.1 where n belongs to $\{256, 512, 1024\}$. In that study, starting from a message of length 1000 and its digest, all the messages that have one bit of difference are further generated and the digest of the new message is obtained. Obviously, both the mean changed bit number $\overline{B}$ and the mean changed probability P are close to the ideal values ($\frac{n}{2}$ bits and 50%, respectively), which illustrates the diffusion and confusion capability of our algorithm. Lastly, as ΔB and ΔP are very small, these capabilities are very stable.

C.6/ Toward a Chaotic Iterations Based Post-Treatment for Hash Functions

In previous sections, we have explained what have been the improvements of the hash function formerly introduced during our thesis. This first version of a chaotic iteration based hash function was initially a kind of mixture between SHA-1 and some chaotic iterations. The first post-thesis stage of our investigations was to simplify this hash function, to make it appears exactly as chaotic iterations, to evaluate experimentally its diffusion and confusion properties, while relating them to topological properties [BCG11b, GB12, BCG12a]. The second most recent stage of our thoughts regarding

hash functions consists in showing that, given a secured hash function, it is possible to realize a post-treatment on the obtained digest using chaotic iterations that preserves the security of the hash function. Furthermore, if the media to hash is obtained frame by frame from a stream, the resulted chaotic hash machine inherits the chaos properties of the chaotic iterations presented previously. This second approach will be presented with more details in this manuscript, as this work is not yet accepted (only submitted).

Let us firstly introduce some definitions.

C.6.1/ DEFINITIONS

Definition 64 (Collision resistance). *For a keyed hash function $h : \mathbb{B}^k \times \mathbb{B}^* \longrightarrow \mathbb{B}^n$, define the advantage of an adversary* A *for finding a collision as*

$$Adv_{\mathsf{A}} = Pr\left[\begin{array}{c|c} K \xleftarrow{\$} \mathbb{B}^k & m \neq m' \\ (m, m') \leftarrow \mathsf{A}(K) & h(K, m) = h(K, m') \end{array}\right] \tag{C.1}$$

where $\$$ means that the element is picked randomly. The insecurity of h with respect to collision resistance is

$$InSec_h(t) = \max_{\mathsf{A}} \{Adv_{\mathsf{A}}\} \tag{C.2}$$

when the maximum is taken over all adversaries A *with total running time t.*

In other words, an adversary should not be able to find a collision, that is, two distinct messages m and m' such that $h(m) = h(m')$.

Definition 65 (Second-Preimage Resistance). *For a keyed hash function $h : \mathbb{B}^k \times \mathbb{B}^* \longrightarrow \mathbb{B}^n$, define the advantage of an adversary* A *for finding a second-preimage as*

$$Adv_{\mathsf{A}}(m) = Pr\left[\begin{array}{c|c} K \xleftarrow{\$} \mathbb{B}^k & m \neq m' \\ m' \xleftarrow{\$} \mathsf{A}(K) & h(K, m) = h(K, m') \end{array}\right] \tag{C.3}$$

The insecurity of h with respect to collision resistance is

$$InSec_h(t) = \max_{\mathsf{A}} \left\{ \max_{m \in \mathbb{B}^k} \{Adv_{\mathsf{A}}(m)\} \right\} \tag{C.4}$$

when the maximum is taken over all adversaries A *with total running time t.*

That is to say, an adversary given a message m should not be able to find another message m' such that $m \neq m'$ and $h(m) = h(m')$. Let us now give a post-operative mode that can be applied to a cryptographically secure hash function without loosing the cryptographic properties recalled above.

Definition 66. *Let*

- $k_1, k_2, n \in \mathbb{N}^*$,
- $h : (k, m) \in \mathbb{B}^{k_1} \times \mathbb{B}^* \longmapsto h(k, m) \in \mathbb{B}^n$ *a keyed hash function,*
- $S : k \in \mathbb{B}^{k_2} \longrightarrow \left(S(k)^i\right)_{i \in \mathbb{N}} \in [\![1, n]\!]^{\mathbb{N}}$*:*

  - *either a cryptographically secure pseudorandom number generator (PRNG),*
  - *or, in case of a binary input stream* $m = m^0||m^1||m^2||\dots$ *where* $\forall i, |m^i| = n$, $\left(S(k)^i\right)_{i\in\mathbb{N}} = \left(m^k\right)_{i\in\mathbb{N}}$.
- $\mathcal{K} = \mathbb{B}^{k_1} \times \mathbb{B}^{k_2} \times \mathbb{N}$ *called the* key space*,*
- *and* $f : \mathbb{B}^n \longrightarrow \mathbb{B}^n$ *a bijective map.*

We define the keyed hash function $\mathcal{H}_h : \mathcal{K} \times \mathbb{B}^* \longrightarrow \mathbb{B}^n$ *by the following procedure*

Inputs: $k = (k_1, k_2, n) \in \mathcal{K}$
$m \in \mathbb{B}^*$

Runs: $X = h(k_1, m)$, *or* $X = h(k_1, m^0)$ *if* m *is a stream*
for $i = 1, \dots, n$:
$X = G_f(X, S^i)$
return X

$\mathcal{H}_h$ is thus a chaotic iteration based post-treatment on the inputted hash function h. The strategy is provided by a secured PRNG when the machine operates in a vacuum whereas it is redetermined at each iteration from the input stream in case of a finite machine open to the outside. By doing so, we obtain a new hash function $\mathcal{H}_h$ with h, and this new one has a chaotic dependence regarding the inputted stream.

C.6.2/ SECURITY PROOFS

The two following lemma are obvious [GBC].

Lemma 2. *If* $f : \mathbb{B}^n \longrightarrow \mathbb{B}^n$ *is bijective, then* $\forall S \in [\![1, n]\!]$*, the map* $G_{f,S} : x \in \mathbb{B}^n \to G_f(x, S)_1 \in \mathbb{B}^n$ *is bijective too.*

Proof 9. *Let* $y = (y_1, \dots, y_n) \in \mathbb{B}^n$ *and* $S \in [\![1, n]\!]$*. Thus*

$$G_{f,S}(y_1, \dots, y_{S-1}, f^{-1}(y_S), y_{S+1}, \dots, y_n)_1 = y.$$

So $G_{f,S}$ *is a surjective map between two finite sets.*

Lemma 3. *Let* $S \in [\![1, n]\!]^{\mathbb{N}}$ *and* $N \in \mathbb{N}^*$*. If* f *is bijective, then* $G_{f,S,N} : x \in \mathbb{B}^n \longmapsto G_f^N(x, S)_1 \in \mathbb{B}^n$ *is bijective too.*

Proof 10. *Indeed,* $G_{s,f,n} = G_{f,S^n} \circ \dots \circ G_{f,S^0}$ *is bijective as a composition of bijective maps.*

We can now state that [GBC],

Theorem 30. *If* h *satisfies the collision resistance property, then it is the case too for* $\mathcal{H}_h$*. And if* h *satisfies the second-preimage resistance property, then it is the case too for* $\mathcal{H}_h$*.*

Proof 11. *Let* $A(k_1, k_2, n) = (m_1, m_2)$ *such that* $\mathcal{H}_h\left((k_1, k_2, n), m_1\right) = \mathcal{H}_h\left((k_1, k_2, n), m_2\right)$*. Then* $G_{f,S(k_2),n}\left(h(m_1)\right) = G_{f,S(k_2),n}\left(h(m_2)\right)$*. So* $h(m_1, k_1) = h(m_2, k_1)$*.*

For the second-preimage resistance property, let $m, k \in \mathbb{B}^* \times \mathcal{K}$*. If a message* $m' \in \mathbb{B}^*$ *can be found such that* $\mathcal{H}_h(k, m) = \mathcal{H}_h(k, m')$*, then* $h(k_1, m) = h(k_1, m')$*: a second-preimage for* h *has thus be found.*

Finally, as $\mathcal{H}_h$ simply operates chaotic iterations with strategy $\mathcal{S}$ provided at each iterate by the media, we have:

Theorem 31. *In case where the strategy $\mathcal{S}$ is the bitwise xor between a secured PRNG and the input stream, the resulted hash function $\mathcal{H}_h$ is chaotic.*

Rem 8. *$\mathcal{S}$ should be $m^k \oplus x^k$ where (x^k) is provided by a secured PRNG if security of $\mathcal{H}_h$ is required.*

C.7/ CONCLUSION

The hash function family formerly proposed during our thesis has been completely rethought and simplified. Drawbacks in the former version have been fixed, by considering that chaotic iterations should rather be used as a post-treatment on existing hash functions, instead of embedding them into *de novo* hash function design. Moreover, various cryptographic properties have been proven to be preserved during this post-treatment, leading to better experimental results for the proposed hash functions. These investigations are justified by the fact that, in 2004, MD5 and SHA-0 have been broken. An attack over SHA-1 has been achieved with only 2^{69} operations (CRYPTO-2005), that is, 2000 times faster than a brute force attack (that requires 2^{80} operations). Even if 2^{69} operations still remains impossible to realize on common computers, such a result based on a previous attack on SHA-0 is a very important one: it leads to the conclusion that SHA-2 is not secure. So, in continuation to the SHA-3 contest, new original hash functions, or improvements for existing ones, must be found.

In this proposal, security of existing hash functions is reinforced by the unpredictability of the behavior of the proposed post-treatment. The resulting hash function, a combination between an existing hash function and chaotic iterations, satisfies important properties of topological chaos such as sensitivity to initial conditions, uniform distribution (as a result of the transitivity), unpredictability, and expansiveness. Moreover, its Lyapunov exponent can be as great as needed. The results expected in our study have been experimentally checked these last three years. The choices made in these first studies are simple. But these simple choices lead to desired results, justifying that such a post-treatment can possibly improve the security of the inputted hash function. And, thus, such an approach should be investigated more largely. This is why, in future work, we will test other choices of iteration functions and strategies. We will try to characterize topologically the diffusion and confusion capabilities. Other properties induced by topological chaos will be explored and their interest for the realization of hash functions will be deepened. Furthermore, other security properties of resistance and pseudorandomness will be proven. We will thus compare the results of this post-treatment on several hash functions, among other things with the SHA-3 finalists.

D

EPIDEMIOLOGICAL APPROACHES FOR DATA SURVIVABILITY IN UNATTENDED WIRELESS SENSOR NETWORKS: CONSIDERING THE SENSORS LIFETIME

Our last investigations in the field of wireless sensor networks' security have regarded the particular case of data survivability in unattended WSNs, introduced in the next section. This work, currently submitted to New Generation Computing (Springer, [GMB]), emphasizes the importance of epidemiological models for tackling the difficulties raised by such a problem. This first apparition of biology in complex systems will be more systematically studied in the last part of this manuscript.

D.1/ DATA SURVIVABILITY IN UNATTENDED WSN

Unattended Wireless Sensor Networks (UWSNs), which have been introduced by Di Pietro *et al.* in [DPMS+08], are WSNs characterized by the sporadic presence of the sink. These UWSNs are useful for instance to detect poaching in a national park, or as a monitoring system to check the pressure of an underground pipeline, as stated in [PV13]. In such networks, nodes collect data from the area under consideration, and then they try to upload all the stored data when the sink comes around. Information survivability is a key problem in UWSNs as these latter are more subject to malicious attacks than traditional WSNs [MT08]: the dimension of the area is often prohibitive in such networks, while the absence of the sink facilitates the work of attackers [GMB].

Epidemic theory has already been considered for data survivability in UWSN in presence of attackers [DPV11, PV13]: SIS, SIR, and SIRS models have been investigated by authors of these research works, in order to derive the parameters that can assure information to survive. In these articles, $S(t)$ compartment is constituted by sensors that do not possess the datum at time t, while $I(t)$ is the compartment of sensors that possess it. Finally, the $R(t)$ compartment is constituted by sensors that have been compromised by the attacker, see [GMB] for further explanations.

However, as stated in [GMB], authors of [DPV11, PV13] surprisingly never consider that in a wireless sensor network, nodes' energy is provided by a battery that can be emptied

due to data acquisition, transmission, or simply functioning cost of keeping alive. More precisely, the topology of the networks they consider is static, the network's lifetime is unbounded, and sensors cannot die due to empty batteries [GMB]. Indeed, their work is more related to unattended wired sensor networks (on main power) but not with a battery as $S + I$ (SIS model) or $S + I + R$ (SIR and SIRS models) are constant. Our intention in [GMB] is to deepen their interesting work, by bringing their proposal from wired sensor networks to WSNs, refining theirs models, and producing more theoretical results on each model. This last contribution in the field of WSNs security is summarized in what follows.

D.2/ A SIR Model for Data Survivability in UWSNs

D.2.1/ Introducing the Kermack & McKendrick model

In this section, the SIR model formerly presented in [DPV11, PV13] is firstly recalled. Then, consumption hypotheses underlined in this model are precised, as in [GMB], while theoretical results on the behavior of the compartments of the network are further investigated.

In unattended wireless sensor networks the presence of the sink is sporadic. However the duration between two visits of the sink to the network (its absence) can sometimes be considered negligible, in a first approximation, compared to the time required to empty a sensor battery. In such UWSNs, the death processes of sensors can be neglected if the aim is to study the immediate consequences of an attack between two visits of the sink. Under such an assumption, the global network can be divided in three compartments, namely the sensors S susceptible to receive the datum of interest (intrusion detection, etc.), the ones that currently store it I, and the recovered sensors R that have been compromised by the attacker: their stored datum has been removed [GMB].

Suppose now that between S and I, the transmission rate is bI, where b is the contact rate, which is the probability of transferring the information in a contact between a susceptible sensor and a sensor having the datum. Indeed, as proven by Di Pietro *et al.*, such a situation occurs when the wireless sensor network is composed by n sensor, and if each sensor forwards the datum with probability $\frac{\alpha}{n}$ [DPV11, PV13]. Suppose additionally that between I and R, the rate of recovery is c: the attacker is able to individuate the sensors containing the target information, and to destroy each of them with this probability c. Notice that, if the duration of the information survivability is D, then $c = \frac{1}{D}$, as a sensor experiences one recovery in D units of time.

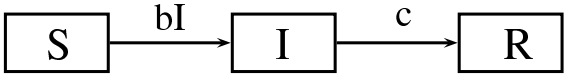


Figure D.1: SIR model

Under such hypotheses and as stated in [DPV11, PV13], the sensors population follows the so-called SIR model of Kermack & McKendrick [KM27] depicted in Figure D.1. Remark that the total sensors population is equal to $N = S + I + R = S_0 + I_0 + R_0$, which is a constant: as emphasizes in [GMB], the number of awaken, alive sensors does not evolve. In particular, only two of the three populations of sensors have to be studied.

D.2.2/ FIRSTS THEORETICAL RESULTS

Consider now that $x(t) = \frac{X(t)}{N}$ denotes the fraction of individuals in the compartment X. The SIR model can be expressed by the following set of ordinary non-linear differential equations [GMB]:

$$\begin{cases} \frac{ds}{dt} = -bis \\ \frac{di}{dt} = bis - ci \\ \frac{dr}{dt} = ci. \end{cases} \tag{D.1}$$

Obviously, the typical time between transmissions is $T_t = b^{-1}$ while the typical time until attack when having the information is equal to $T_e = c^{-1}$. Thus

$$\frac{T_t}{T_e} = \frac{c}{b}$$

is the average number of transmissions between a sensor having the datum and others before it lost this information due to the attacker [GMB]. Such a statement explain why, in the SIR historical model, the dynamics of the infectious class depends on the *reproduction ratio* defined by

$$R_0 = \frac{b}{c},$$

which corresponds here to the expected number of new informed sensors (so-called "secondary infections") providing a single sensor with the datum where all sensors are susceptible [GMB]. Furthermore, direct standard analyses manipulations (variables separation and then integration) lead to the following form for the susceptible sensors compartment: $s(t) = s(0)exp\left(-R_0(r(t) - r(0))\right)$.

As $\frac{di}{dt} = (R_0 s - 1)ci$, if the basic reproduction number satisfies $R_0 > \frac{1}{s(0)}$, there will be an information outbreak with an increasing number of sensors with the datum. In other words, R_0 determines whether or not the information will spread through the network.

All these facts are summarized in a proposition of [GMB] recalled below.

Proposition [30]. *Consider a sensor network that aims to monitor a given area, and that has to spread an alert or an information to a sink, whose presence is sporadic. Suppose that an attacker tries to remove the datum in sensors' memory, and that:*

1. *all sensor activities are negligible, in terms of energy,*
2. *when a sensor has the datum, it spreads the information to its neighbors with a probability b, until being attacked.*

Denote by T_t the typical time between transmissions, T_e the typical time an informed sensor loses its information due to the attacker, and by $s(0)$ the initial fraction of susceptible sensors. So the information will spread through the network if and only if $T_t < s(0)T_e$.

In other words, this proposition states that if the reproduction ratio is greater than one, then an "epidemic" occurs since the prevalence (the infective ratio) increases to a peak and then decreases to zero. Otherwise there is no epidemic since the prevalence decreases to zero [GMB].

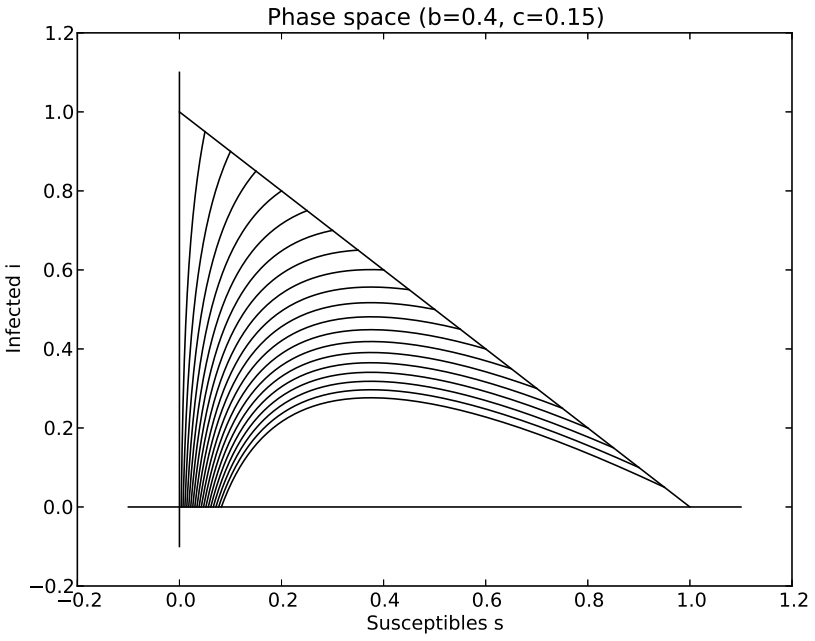


Figure D.2: Phase space (s, i) with $b = 0.4, c = 0.15$ (SIR model).

We then have shown in [GMB] that is possible to be more precise in the formulation of Proposition 30, following an approach similar to [Het00].

Proposition [31]. *The fraction $s(t)$ of sensors susceptible to receive the information is a decreasing function. The limiting value $s(\infty)$ is the unique root in $(0, \frac{T_e}{T_t})$ of the equation*

$$1 - r(0) - s(\infty) + \frac{T_e}{T_t} \ln\left(\frac{s(\infty)}{s(0)}\right).$$

Additionally,

- *if $T_t \geqslant s(0)T_e$, then the fractional number $i(t)$ of sensors having the datum decreases to zero as $t \to \infty$,*
- *else $i(t)$ first increases up to a maximum value equal to $1 - r(0) - \frac{T_e}{T_t}\left(1 + \ln\left(\frac{s(0)T_t}{T_e}\right)\right)$ and then decreases to zero as $t \to \infty$, where $\ln$ stands for the natural logarithm.*

Proof [12]. *See [GMB].*

The phase space of the solutions of the SIR system with given parameters is provided in Figure D.2 while the evolution of s and i is depicted in Figure D.3. Remark that the results

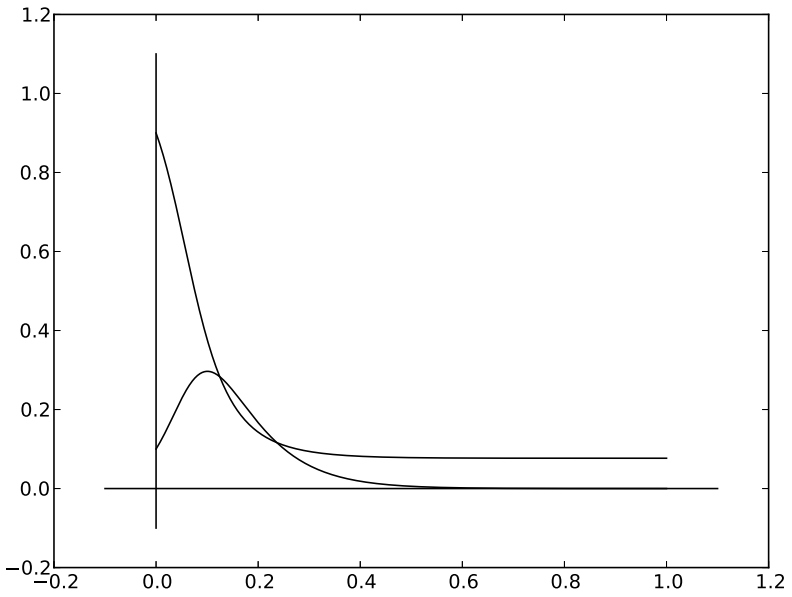


Figure D.3: Evolution of the fractions s and i of susceptible and having the datum sensors with $b = 0.4, c = 0.15, s(0) = 0.9$, and $i(0) = 0.1$ (SIR model).

presented in this section hold for a transition rate between susceptible and informed sensors having the form $F = ai$, which thus represents the force of information. Nonlinear forces of information, or infection, can be investigated too, to model more realistically the information survivability (see [GMB] for further details).

D.2.3/ ANOTHER UNDERSTANDINGS FOR THE RECOVERED COMPARTMENT

In the previous section, the R compartment was constituted by sensors that have been compromised by the attacker, which will be referred in what follows as Situation 1. As remarked in [GMB], it is possible to attribute at least two other understandings to this compartment, for an unattended wireless sensor network whose lifetime is dependent on energy consumption and in absence of attacks.

This compartment can be constituted by dead sensors, when considering that the sole action on the energy is the information transmission, and that the unique way to death for a sensor is to have too much transmitted the datum. In other words, in this Situation 2, sensors send information messages to their neighbors until emptying totally their batteries. The sink will receive the information when it will interrogate the network at time t if $I(t) \neq 0$.

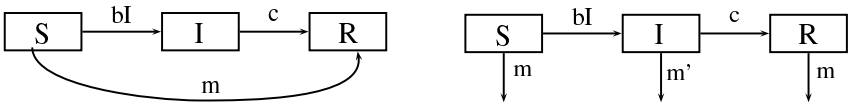
A third situation can be considered without any changes in formalization, except redefining the meaning of the R compartment. Indeed, it can be interesting to consider that a sensor is first susceptible to receive an information message for a while, then in a second time it owns and transmits the information, before finally entering into the third age of its life, the recovered state in which it will lose its ability to transmit the information. Materials of the previous section tackles too this scenario, when considering the network

lifetime sufficiently large compared to information spreading, in order to neglect sensors' death due to energy consumption. The question raised by [GMB] is then to determine the quantity of informed sensors on large timescales.

However, in a large amount of situations, energy consumption and the death of sensors cannot be neglected, this is why a "natural" death rate for all compartments is introduced in [GMB] and recalled in the next section. Such an approach generalizes the models presented in the current section.

D.3/ Considering Energy Consumption for Data Survivability in UWSNs

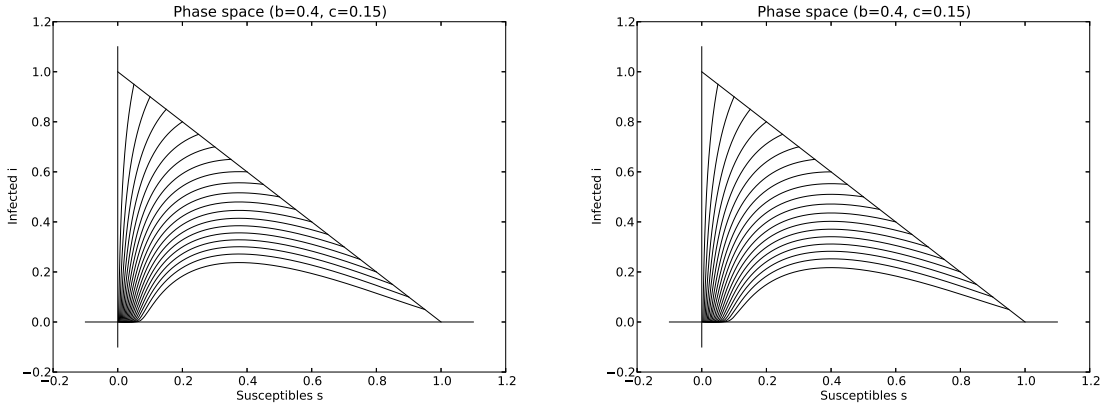
D.3.1/ A SIR model with natural death rate



(a) Situation 2 (b) Situations 1 and 3

Figure D.4: SIR models with natural death rate

The previous section considers that all sensor activities are negligible, in terms of energy, except the transmission of information in situations 2 and 3, which is reasonable in a first approximation. It is however possible to refine the SIR model in these two last situations, in order to consider that sensors' energy decreases too in absence of information transmission [GMB].



(a) Situation 2 (b) Situations 1 and 3

Figure D.5: Phase space (s, i) with $b = 0.4, c = 0.15, m = 0.01$, SIR model with natural death rate in the three situations.

In Situation 2, the R compartment of the SIR model is constituted by dead sensors. This compartment is populated by susceptible nodes that have naturally died (death rate m) without having received the datum and by sensors of the I compartment which die at

another rate c supposed to be greater than m, as they have to transfer the datum, an energy-consuming task. This situation, introduced in [GMB], is depicted in Figure D.4(a).

In the two other situations investigated in [GMB], the R compartment is constituted by living sensors that do not transmit the datum anymore, either because they have been corrupted and thus have lost it (first situation), or because their batteries must be preserved (third one). This new situation is closed to the SIR model of Figure D.1, except that a the new network is characterized by a death rate for each sensors compartment (see Figure D.4(b)). Notice that the death rate m' of the I compartment is *a priori* different from the one of S and R compartments, as it is reasonable to suppose that the datum transmission implies more energy consumption. However, setting $m' = m$ is possible too [GMB].

The SIR model of Equation (D.1) can be adapted as follows for Situation 2:

$$\begin{cases} \dfrac{ds}{dt} = -bis - ms \\ \dfrac{di}{dt} = bis - ci \\ \dfrac{dr}{dt} = ci + ms, \end{cases} \tag{D.2}$$

while it has the following form in Situations 1 and 3:

$$\begin{cases} \dfrac{ds}{dt} = -bis - ms \\ \dfrac{di}{dt} = bis - ci - m'i \\ \dfrac{dr}{dt} = ci - mr. \end{cases} \tag{D.3}$$

We have then investigated the long-term behavior of these models in [GMB]. Regarding Situation 2, it is natural to think that, for large timescales, all sensors will take place in the third R compartment of died sensors, as all the batteries are continually emptied (either due to natural consumption or because of the information transmission). It has been proven in [GMB] by considering that in an equilibrium point $(s^*, i^*, r^* = 1 - s^* - i^*)$, we have $\dfrac{ds}{dt} = \dfrac{di}{dt} = \dfrac{dr}{dt} = 0$, and so

$$\begin{cases} (bi^* + m)s^* = 0 \\ (bs^* - c)i^* = 0 \\ ci^* + ms^* = 0. \end{cases}$$

As $c > 0, m > 0, i^* \geqslant 0$, and $s^* \geqslant 0$, we can conclude from the third equation above that $s^* = i^* = 0$, and so $r^* = 1$. The Jacobian is equal to

$$J(s, i, r) = \begin{pmatrix} -bi - m & -bs & 0 \\ 0 & bs - c & 0 \\ m & c & 0 \end{pmatrix}$$

and its characteristic polynomial in $(0, 0, 1)$ is $\lambda(\lambda + c)(\lambda + m)$. The eigenvalues being negative, the equilibrium $(0, 0, 1)$ is attractive. These results are summarized in the following proposition [GMB].

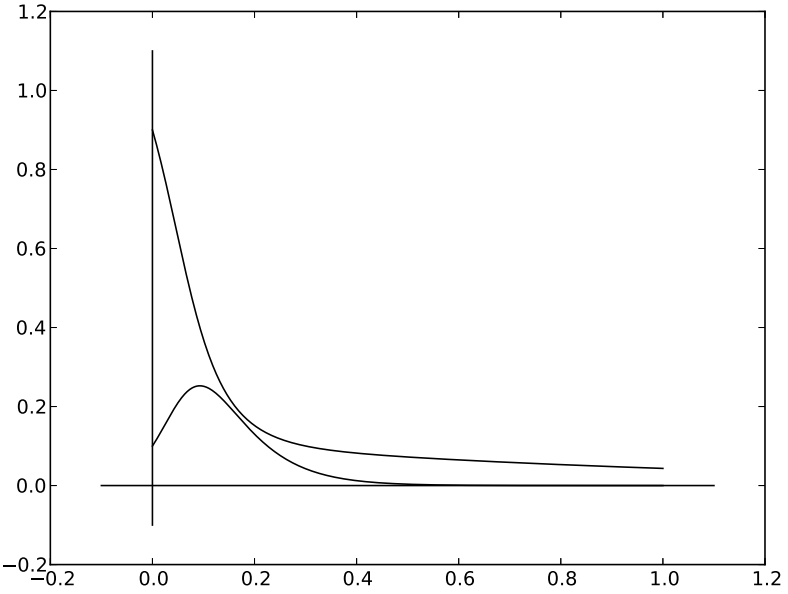


Figure D.6: Evolution of the fractions s and i of susceptible and having the datum sensors with $b = 0.4, c = 0.15, m = 0.01, s(0) = 0.9$, and $i(0) = 0.1$, SIR model with natural death rate in Situations 1 and 3.

Proposition 32. *Consider an unattended wireless sensor network divided in three sets of sensors, the first category S being susceptible to receive a given datum, the second one I having and transmitting this latter, and the third one R being constituted by dead sensors.*

Suppose that the death rate is m for S compartment and c for I's one, and that the transmission rate is bI between S and I. In that situation, for all initial conditions and all positive parameters $b, c,$ and m, the system is convergent to the equilibrium point $(0, 0, 1)$.

In particular, in that situation, the datum cannot survive a long time in the UWSN.

As remarked in [GMB], Equation D.3 can be resolved similarly: from $bi^*s^* + ms^*$, we deduce that $s^* = 0$ (as $b > 0$, $m > 0$, and $i^* \geqslant 0$). So $bi^*s^* - ci^* - m'i^*$ implies that $i^* = 0$ too. Finally, from the third line, we conclude that $r^* = 0$. Eigenvalues of the characteristic polynomial of the Jacobian in $(0, 0, 0)$ are $-m$ and $-c - m'$, which are negative. So this equilibrium point is attractive too, and a similar proposition than previously can be formulated, with the same conclusion, both for Situations 1 and 3 [GMB]. Phase spaces for the three situations are provided in Figure D.4 while Fig. D.8 depicts the evolution of the fractions s and i in Situations 1 and 3.

To put it in a nutshell, to achieve data survivalibity in UWSNs, the birth of awaken sensors must be considered, which is the subject of the next subsection.

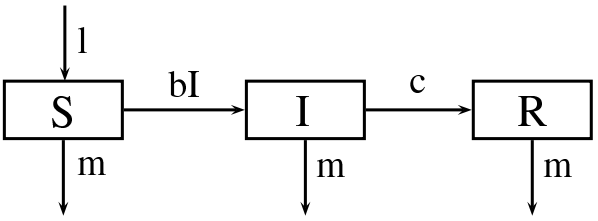


Figure D.7: SIR model with natural birth and death rates

D.3.2/ A SCHEDULING PROCESS IN DATA SURVIVABILITY

Suppose now that a scheduling process, like the ones presented in previous chapters, is planned to enlarge the network's lifetime. At the initial stage, only a small part of the sensor nodes is awakened. New sensors are then awakened periodically during the whole network's lifetime at a rate l, repopulating by doing so the S compartment. Along with this birth rate, a natural death rate m is considered in [GMB] for each of the three kind of sensors, while the R compartment is for corrupted sensors in the original situation 1, as depicted in Figure D.7. Remark that such a model is compatible with living and awaken nodes that have stopped to transfer the information in Situation 3.

To model such a scenario requires to rewrite the first line of Equation (D.2), leading to the following system [GMB]:

$$\begin{cases} \dfrac{ds}{dt} = l - bis - ms \\ \dfrac{di}{dt} = bis - ci - mi \\ \dfrac{dr}{dt} = ci - mr. \end{cases} \tag{D.4}$$

This updated system is the usual SIR model with vital dynamics, but we have not supposed the birth and death rates equal in [GMB]. Let us notice that it is possible to show that the problem is well formulated, as the triangle $T = \{(s,i) \mid s \geqslant 0, i \geqslant 0, s+i \leqslant 1\}$ still remains positively invariant.

A study of this system supposes to consider the Poincaré-Bendixon theorem in phase space and the use of Lyapunov functions [Het00]. However, as explained in [GMB], it can be understood by considering what will happen to the information in a long run: will it die out or will it establish itself in the network like an endemic situation in epidemiological models? The long-term behavior of the solutions, which depends largely on the equilibrium points that are time-independent solutions of the system, must be investigated to answer this question. Since these solutions do not depend on time, we have $s'(t) = i'(t) = r'(t) = 0$, which lead to the system [GMB]:

$$\begin{cases} 0 = l - bis - ms \\ 0 = bis - (c+m)i \\ 0 = ci - mr. \end{cases}$$

$r = \dfrac{c}{m} i$ from the last equation, and either $i = 0$ or $s = \dfrac{c+m}{b}$ from the second one. On

the one hand, if $i = 0$, then $r = 0$, and $s = \frac{l}{m}$ from the first equation. This leads to the equilibrium solution

$$\left(\frac{l}{m}, 0, 0\right).$$

As the number of sensors having the datum is 0 in this point, it means that if a solution of the system approaches this equilibrium, the fraction i will approach 0, and the datum tends to disappear from the network: an *information-free equilibrium*. Remark that the existence of this equilibrium is independent of the parameters of the system: it always exists [GMB].

On the other hand, if $i \neq 0$, then $s = \frac{c+m}{b} \neq 0$ from the second equation, and $\frac{l}{s} = bi + m$ according to the first equation. Substituting s and solving for i, we find

$$i = \frac{bl - m(c+m)}{b(c+m)} = \frac{R_0 l - m}{b},$$

with $R_0 = \frac{bl}{m(c+m)}$, which is a positive number iff $R_0 > 1$.

R_0 is the reproduction number of the information, which tells us how many secondary informed sensors will one informed sensor produces in an entirely susceptible network, as:

- a network which consists of only susceptible nodes in a long run has $\frac{l}{m}$ sensors;
- $c + m$ is the rate at which sensors leave the I compartment. In other words, the average time spent as an informed sensor is $\frac{1}{c+m}$ time units.
- The number of data transmissions per unit of time is given by the incidence rate bIS. If there is only one informed sensor ($I = 1$) and every other sensor is susceptible ($S = \frac{l}{m}$), then the number of transmissions by one "infective" node per unit of time is $\frac{bl}{m}$.

So the number of data transmissions that one informed sensor can achieve during the entire time it is not attacked, if all the remained sensors are susceptible, is equal to $\frac{bl}{m(c+m)}$, that is, R_0 [GMB].

So if $R_0 > 1$, the number of sensors having the datum is strictly positive in this equilibrium solution: if some other solutions of the system approach this equilibrium as time goes large, the number of sensors having the datum will remain strictly positive, and the information remains in the network and becomes endemic.

These statements are summarized in the following proposition [GMB].

Proposition 33. *If either $R_0 \leqslant 1$ or $s(0) = 0$, then any solution $(s(t), i(t))$ is convergent to the equilibrium without information $(1, 0)$.*

If $R_0 > 1$ then there are two equilibriums: the non attractive information-free equilibrium and the endemic equilibrium. This latter is attractive so that solutions of the ODE system approach it as time goes to infinity: the information remains endemic in the UWSN.

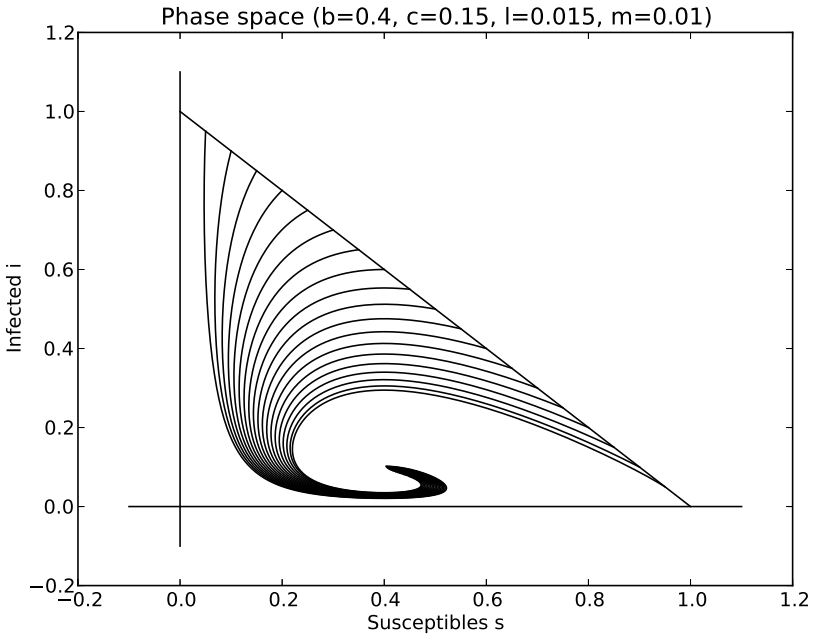


Figure D.8: Evolution of the fractions s and i of susceptible and having the datum sensors, SIR model with natural birth and death rates ($R_0 = 3.75$).

As remarked in [GMB], the attacker's desire is to have $R_0 < 1$ to tend to an information-free equilibrium, whereas R_0 must be greater than 1 for the sink to face such attack. If the attacker has the opportunity to observe the network running a certain duration, then he or she can infer the values of parameters b, c, m, and l. Let N be the number of data transmissions by one informed node per time unit, that is, $N = \frac{bl}{m}$. If the attacker is able to detect and infect the informed nodes in a time $\frac{1}{c+m}$ lower than $\frac{1}{N}$, then he or she is sure that $R_0 < 1$: the data will not survive in the network [GMB]. The sink interest, for its part, is to have $\frac{bl}{m}$ large and $\frac{1}{c+m}$ small, which can be achieved in the following manner:

- increasing the birth rate b,
- increasing the lifetime of sensors to reduce m,
- increasing the data transmission rate b, but m increases when b increases,
- if possible, reducing c by considering countermeasures against data removal.

D.4/ NUMERICAL SIMULATIONS

We then have verified experimentally the Proposition 33 on a basic wireless sensor network in [GMB]. In this simulation, the initial number of susceptible sensors is set to 300 while 3 nodes initially receive the datum.

```
from random import random
from pylab import *

S,I,R = [],[],[]
b,c,l,m=0.4,0.015,0.4,0.03

cpt,n,nn = 0,300,3
lifetime = 300

X=[(len(S),len(I),len(R))]
for t in range(lifetime):
    while random()<l:
        S.append(cpt)
        cpt += 1
    for sensor in S:
        if random()<m:
            S.remove(sensor)
        elif random()<b:#/(len(S)+len(I)+len(R)):
            S.remove(sensor)
            I.append(sensor)
    for sensor in I:
        if random()<m:
            I.remove(sensor)
        elif random()<c:
            I.remove(sensor)
            R.append(sensor)
    for sensor in R:
        if random()<m:
            R.remove(sensor)
```

Figure D.9: Python program to simulate a SIR-compartmented UWSN.

The simulator, written in Python language, is detailed in Listing D.9, while Figure D.10 shows the obtained result. We can see that the I compartment is never empty, leading to a data survivability in this wireless sensor network [GMB].

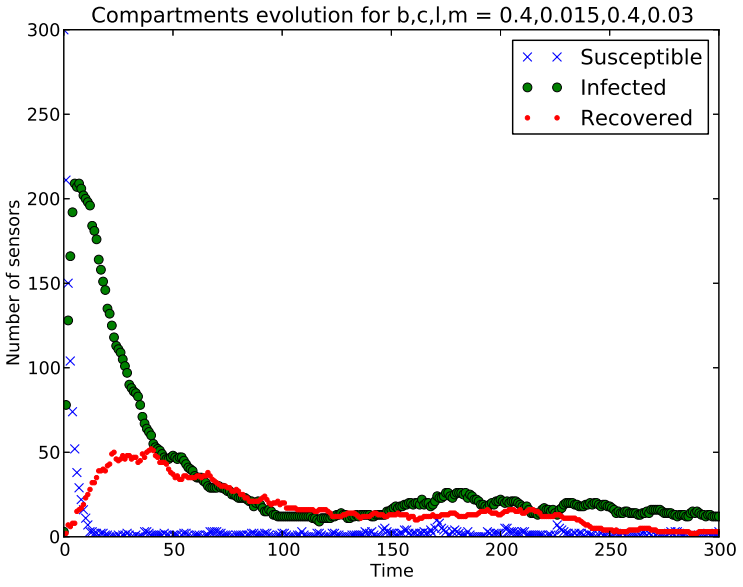


Figure D.10: Simulation of SIR model with birth and death rates and $R_0 > 1$

E

Other Complex Applications in Bioinformatics

Our studies of complex biological systems have led us to participate to projects with biologists of Chrono-environnement and of the Faculté de médecine de Besançon, in which some bioinformatics stages was too complicated to be tackled with usual available tools. Due to large amount of data and of their plurality of forms, they required to develop ad hoc programs and the use of the Mésocentre de calcul de Franche-Comté. Some of these projects are recalled thereafter, with more or less details.

E.1/ Investigating the Cestodes Evolution

E.1.1/ A molecular phylogeny of 33 Eucestoda species based on complete mitochondrial genomes

We have firstly actively participate to the elaboration of a biomolecular phylogeny of a large collection of Eucestoda genomes. This research work, currently submitted [CLBGB], is a collaboration with Jacques M. Bahi and two colleagues of Chrono-environnement (Nathalie Côté and Matthieu Le Bailly), in the framework of the PEG project[1] founded by the Région Franche-Comté. Our major contributions to [CLBGB] are summarized thereafter.

E.1.1.1/ Introduction

Cestoda (Cestoidea) is a class over a thousand species of parasitic flatworms of the phylum Platyhelminthes, whose members live in the digestive tract of vertebrates as adults, and often in the bodies of various animals as juveniles. All vertebrate species can be parasitised by at least one species of tapeworm.

The phylogeny of the Eucestoda subclass of Platyhelminthes has evolved through the ages, moving from morphological characteristics to the recent use of molecular data, leading to a more and more precise knowledge of the respective relationship between species within this subclass. The latest morphological and ecological based phylogenies of the Taenia has been realized one decade ago by Hoberg *et*

[1] Paléoparasitologie et application à l'évolution des génomes

al. [HAQJ01, HJR+00, Hob06]. They have considered various characteristics related to morphology, geographical spread, intermediate and definitive hosts, in the largest set of *Taenia* that has ever looked at (see Table E.2). However, inferred phylogenies were often contradictory when changing the studied characteristics, and the appearance of sequenced genes and genomes has led researchers to specify problematic relationships using DNA or amino acid sequences. Finally, the most recent and one of the most complete study about molecular phylogeny of *Taeniidae* has been published in [NLI+]. In this article, interesting proposals to retought the phylogenetic relationship of species within this family have been suggested, but some weaknesses in their tree topologies make that these proposals must be further investigated, which is one objective of our research work summarized in this chapter.

More precisely, *Taenia* and *Echinococcus* have been considered for a long time as the two valid genera in the Eucestoda family *Taeniidae*, due to morphological similarities. However, even though the members of the *Echinococcus* genus are highly similar both for the features of development and ecology, it is not the case for *Taenia*. Indeed, this genus was formerly divided into various genera remarkably diverse in terms of morphology and other characteristics usually used to establish a phylogeny. The recent development of molecular phylogeny has brought elements of response to the questioning of the monophyly of the *Taenia* genus. This is why, in [NLI+], two new genera in the *Taeniidae* have been proposed, namely the resurrection of *Hydatigera* Lamarck, 1816 and the creation of a new genus *Versteria*. Authors of this previous research work have arrived to this conclusion of the *Taenia* paraphyly thanks to molecular phylogenetic analyses using molecular phylogenetic trees of 18S ribosomal DNA and concatenated exon regions of protein-coding genes (*pepck* and *pold*).

Objective of [CLBGB] summarized in this chapter was twofold. On the one hand, our intention has been to question the Nakao *et al.* hypotheses regarding the resurrection of *Hydatigera* Lamarck, 1816 and the creation of a new genus *Versteria* [NLI+]. On the other hand, our goal was to improve the knowledge of the Eucestoda phylogeny with molecular analyses.

The first objective has been achieved by considering sets of data different from [NLI+]. In our study recalled here, the *Taenia* and *Echinococcus* genera are regarded twice, namely by considering the complete mitochondrial genomes and by extracting their twelve genes. The first approach is motivated by the desire to limit the amount of data treatments, to be as close as possible to the DNA information: only a multiple global alignment has been performed on the complete genomics sequence using M-Coffee [NHH00]. By doing so, sources of errors inferred by any sequence post-treatment, like annotation errors for instance, are as reduced as possible. This first approach has been possible due to the fact that all the Eucestoda mitochondrial genomes share the same genes in the same order: large rearrangements of sequences have thus not occurred in this set of data. The second approach was more classical: the 12 genes are taken from the NCBI annotated genomes. All alleles of each gene are converted into amino acids and then aligned with M-Coffee. In the two approaches, both maximum likelihoods and Bayesian inferences are realized to produce similar phylogenetic trees. This second approach was similar to [NMS+07], but the set of species, the tools used during analyses, and the questioning are different.

Our study is complementary to the Nakao *et al.* one, as (1) the set of data is much greated and not similar (18S ribosomal DNA and concatenated exon regions of two protein-

Species	Family	Order	Accession
Diph.latum	*Diphyllobothriidae*	*Pseudophyllidea*	NC_008945
Diph.nihonkaiense	*Diphyllobothriidae*	*Pseudophyllidea*	NC_009463
Dipl.balaenopterae	*Diphyllobothriidae*	*Pseudophyllidea*	NC_017613
Dipl.grandis	*Diphyllobothriidae*	*Pseudophyllidea*	NC_017615
Dipy.caninum	*Dipylidiidae*	*Cyclophyllidea*	NC_021145
Echi.canadensis	*Taeniidae*	*Cyclophyllidea*	NC_011121
Echi.equinus	*Taeniidae*	*Cyclophyllidea*	NC_020374
Echi.felidis	*Taeniidae*	*Cyclophyllidea*	NC_021144
Echi.granulosus	*Taeniidae*	*Cyclophyllidea*	NC_008075
Echi.multilocularis	*Taeniidae*	*Cyclophyllidea*	NC_000928
Echi.oligarthrus	*Taeniidae*	*Cyclophyllidea*	NC_009461
Echi.ortleppi	*Taeniidae*	*Cyclophyllidea*	NC_011122
Echi.shiquicus	*Taeniidae*	*Cyclophyllidea*	NC_009460
Echi.vogeli	*Taeniidae*	*Cyclophyllidea*	NC_009462
Hyme.diminuta	*Hymenolepididae*	*Cyclophyllidea*	NC_002767
Spir.erinaceieuropaei	*Diphyllobothriidae*	*Pseudophyllidea*	NC_011037
Taen.asiatica	*Taeniidae*	*Cyclophyllidea*	NC_004826
Taen.crassiceps	*Taeniidae*	*Cyclophyllidea*	NC_002547
Taen.hydatigena	*Taeniidae*	*Cyclophyllidea*	NC_012896
Taen.krepkogorski	*Taeniidae*	*Cyclophyllidea*	NC_021142
Taen.laticollis	*Taeniidae*	*Cyclophyllidea*	NC_021140
Taen.madoquae	*Taeniidae*	*Cyclophyllidea*	NC_021139
Taen.martis	*Taeniidae*	*Cyclophyllidea*	NC_020153
Taen.multiceps	*Taeniidae*	*Cyclophyllidea*	NC_012894
Taen.mustelae	*Taeniidae*	*Cyclophyllidea*	NC_021143
Taen.ovis	*Taeniidae*	*Cyclophyllidea*	NC_021138
Taen.parva	*Taeniidae*	*Cyclophyllidea*	NC_021141
Taen.pisiformis	*Taeniidae*	*Cyclophyllidea*	NC_013844
Taen.saginata	*Taeniidae*	*Cyclophyllidea*	NC_009938
Taen.serialis	*Taeniidae*	*Cyclophyllidea*	AB731674
Taen.solium	*Taeniidae*	*Cyclophyllidea*	NC_004022
Taen.taeniaeformis	*Taeniidae*	*Cyclophyllidea*	NC_014768
Taen.twitchelli	*Taeniidae*	*Cyclophyllidea*	NC_021093
Schi.mekongi (*Trematoda*)	*Schistosomatidae*	*Strigeidida*	NC_002529

Table E.1: Eucestoda + outgroup taxa and their accession numbers

coding genes versus complete mitochondrial genomes and complete 12 mit. genes), (2) alignment tools are different (T-Coffee [NHH00] and R-Coffee vs. M-Coffee and Muscle [Edg04]), and (3) Bayesian analyses tools are not the same too (MrBayes [HR01] and PhyML [GDL+10] vs. PhyloBayes [LLB09] and PhyML verified with MrBayes and RAxML [SL05]), while the conclusions are convergent. Statements and hypotheses of [NLI+] have been confirmed, whereas elements of response have been proposed to the questionings raised in this former article focusing on the *Taeniidae* phylogeny.

Regarding the second objective, the number of species we have studied in this molecular analysis makes that the knowledge of the Eucestoda phylogeny is improved. In the *Taeniidae* family, some species have been considered by only one or two molecular studies in the state of the art phylogenies, namely *T.pisiformis* and *T.krepkogorski*. Our research has improved the knowledge of their sister relationship with other species within this family. Additionally, other Eucestoda families have been added in our study, leading for the first time to a global vision of respective relationships between various families and genera of this subclass. More precisely, the seven species not related to the *Taeniidae* family that have been regarded by us are: *Diphyllobothrium latum*, *Diphyllobothrium nihonkaiense*, *Diplogonoporus balaenopterae*, *Diplogonoporus grandis*, *Dipylidium caninum*, *Hymenolepis diminuta*, and *Spirometra erinaceieuropaei*. The large amount of data has led us to develop original solutions on the Mésocentre to achieve our goals.

E.1.1.2/ MATERIALS AND METHODS

Taxa details 33 complete mitochondrial genomes of *Eucestoda* representing respectively two *Diphyllobothria*, two *Diplogonoporus*, one *Dipylidium*, nine *Echinococcus*, one *Hymenolepis*, one *Spirometra*, and seventeen *Taenias* were used to construct a molecular phylogeny with *Schistosoma mekongi* (*Trematoda: Digenea: Strigeidida: Schistosomatoidea: Schistosomatidae*) as outgroup. All these genomes are listed with their accession numbers and other taxonomic information in Table E.1.

A large part of species studied in previous molecular-based phylogenic researches are represented in this sample used in [CLBGB], while the introduction of new other ones, never investigated in a molecular phylogeny context, has been possible due to recent mitochondrial genome releases, see Table E.2.

Complete mitochondrial genomes analyses In the first series of experiments, a multiple sequence alignment of the 34 complete mitochondrial genomes has been realized using the optional M-Coffee of the alignment program T-Coffee [NHH00]. M-Coffee is a multiple sequence alignment package, part of the T-Coffee distribution. Instead of computing a multiple sequence alignment on its own, M-Coffee uses other packages to compute the alignments, namely: clustalw [LBB+07], partial order alignment (poa [LGS02]), muscle [Edg04], probcons [DMBB05], mafft [KiKTM05], pcma [PSG03], and T-Coffee. It then uses T-Coffee to combine all these alignments into one unique final alignment. This multiple approach enables the use of computation center like the Mésocentre one. Quoting the T-Coffee's author, "in practice we have shown that the combined alignments are on average better than the initial alignments. Furthermore, the regions where they agree tend to be correctly aligned." An analysis of Eucestoda phylogeny based on a multiple sequence alignment provided by Muscle only has also been achieved, see supplementary materials of [CLBGB].

	Hoberg *et al.* 2000 [HJR+00]	Hoberg *et al.* 2001 [HAQJ01]	Hoberg 2006 [Hob06]	Nakao *et al.* 2010 [NYO+10]	Lavikainen *et al.* 2010° [LHL+10]	Knapp *et al.* 2011° [KNY+11]	Nakao *et al.* 2013° [NLI+]	This study°	Total of studies
Diphyllobothrium latum								X	1
Diphyllobothrium nihonkaiense								X	1
Diplogonoporus balaenopterae								X	1
Diplogonoporus grandis								X	1
Dipylidium caninum						*	*	X	1
Echinococcus canadensis				X^2		X	+	X	3
Echinococcus equinus				X		X	+	X	3
Echinococcus felidis				X		X	+	X	3
Echinococcus granulosus				X		X	+	X	3
Echinococcus multilocularis				X		X	+	X	3
Echinococcus oligarthrus				X	*	X	+	X	3
Echinococcus ortleppi				X		X	+	X	3
Echinococcus shiquicus				X		X	+	X	3
Echinococcus vogeli				X		X	+	X	3
Hymenolepis diminuta								X	1
Spirometra erinaceieuropaei								X	1
Taenia acinonyxi	X	X	X						3
Taenia asiatica	X	X	X	X	X	X	X	X	8
Taenia brachyacantha	X	X							2
Taenia crassiceps	X	X	X	X	X	X	X	X	8
Taenia crocutae	X	X	X	X					4
Taenia dinniki	X	X							2
Taenia endothoracicus	X	X	X						3
Taenia gonyamai	X	X	X						3
Taenia hyaenae	X	X	X	X					4
Taenia hydatigena		X	X		X	X	X	X	6
Taenia ingwei	X	X							2
Taenia intermedia			X						1
Taenia krabbei					X				1
Taenia krepkogorski							X	X	2
Taenia laticollis	X	X				X	X	X	5
Taenia macrocystis	X	X	X						3
Taenia madoquae	X	X	X		X	X	X	X	7
Taenia martis	X	X	X		X	X	X	X	7
Taenia multiceps	X	X	X	X	X	X	X	X	8
Taenia mustelae	X	X	X		X	X	X	X	7
Taenia olngojinei	X	X	X						3
Taenia omissa	X	X	X						3
Taenia ovis	X	X	X^2		X	X	X	X	7
Taenia parenchymatosa	X	X	X^3						3
Taenia parva	X	X	X		X	X	X	X	7
Taenia pencei			X						1
Taenia pisiformis	X	X	X		X			X	5
Taenia polyachantha	X	X			X^2				3
Taenia pseudolaticollis	X	X							2
Taenia regis	X	X	X		X				4
Taenia rileyi	X	X	X						3
Taenia saginata	X	X	X	X	X	X	X	X	8
Taenia selousi	X	X	X						3
Taenia serialis	X	X	X^2	X^2	X	X	X	X	8
Taenia simbae	X	X	X	X					4
Taenia solium	X	X	X	X	X	X	X	X	8
Taenia taeniaeformis	X	X	X	X	X^2	X	X	X	8
Taenia taxidiensis	X	X	X						3
Taenia twitchelli	X	X	X		X	X	X	X	7
Total 55	35	35	31	19	18	24	16	33	

Table E.2: Eucestoda in state of the art phylogenies (* when serving as outgroup; + when present in dataset but not used in phylogenetic analyses; X^n when n representents of the species).

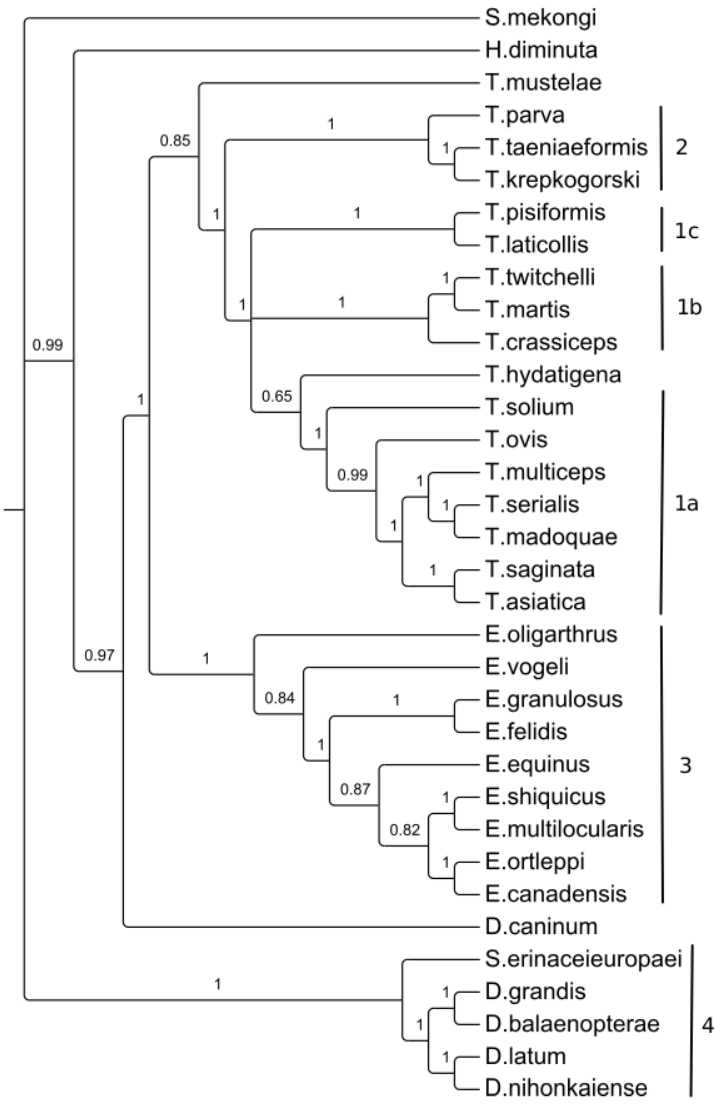


Figure E.1: Cestodes phylogeny using Bayesian inference (PhyloBayes) on amino acid sequences.

Contrary to previous studies, mitochondrial sequences used in our first series of experiments are the complete genomes (not only gene features). Such an approach requires more computational resources but it reduces the number of treatments on data (alignment of subsequences, concatenations whose appropriateness must be checked, and so on), reducing by doing so the risk of errors [CLBGB]. The unique pre-treatment realized on these genomes is an artificial circular rotation on the sequence. Indeed we have verified that all these *Platyhelminthes* (the 33 *Cestodes* and the *Trematoda* serving as outgroup) have the same twelve genes in the same order, which is: *cox1*, *cox2*, *nad6*, *nad5*, *cox3*, *cytb*, *nad4l*, *nad4*, *atp6*, *nad2*, *nad1*, and finally *nad3*. So, to facilitate the multiple sequence alignment and to improve the quality of the M-Coffee result, beginning location of *cox1* gene has been obtained from NCBI, and circular rotation of genomes have been achieved in [CLBGB], in such a way that each data used in M-Coffee starts at the beginning of a *cox1* gene. 12222 distinct alignment patterns have been obtained, with

a proportion of gaps and completely undetermined characters in this alignment equal to 40.31%.

FindModel (http://www.hiv.lanl.gov/content/sequence/findmodel), which is based on the program ModelTest, has been used in order to identify the best substitution model. A subset of 20 aligned genomes (without the outgroup) has been extracted from the output of M-Coffee and sent to the FindModel website. All possible 28 models have been investigated and Weighbor has been chosen for the initial tree construction method. The General Time Reversible GTR model with gamma-distributed rate variation across sites has been identified as the best model among all tested ones.

Phylogeny has then been reconstructed in [CLBGB] with the maximum-likelihood (ML) optimality criterion method as implemented in PhyML version 20120412 website [GDL+10] using the multiple genome alignment provided by M-Coffee. Based on the ModelTest study, the general time reversible model with a proportion of invariable sites and gamma-distributed rate variation across sites (GTR+I+G) was identified as the best suited model in the dataset under consideration. Both the proportion of invariable sites and the Gamma distribution parameter have been estimated by PhyML, the nucleotide equilibrium frequencies were empirical. The initial phylogenetic tree was estimated using the BIONJ algorithm [Gas97], NNIs were selected as tree topology search, whereas the optimise tree topology option has been set as yes. Finally, the robustness of ML trees has been checked by 1000 BS replicates. All the trees where rooted using the *Schistosoma mekongi* genome as outgroup taxon, see [CLBGB] for further details.

Alternative phylogenies have been inferred in [CLBGB] by using RAxML version 7.2.8 [SL05]. This program is another well-known ML method, using the same multiple sequence alignment. GTRGAMMA has been chosen as first nucleotide substitution model. All model parameters have been estimated by RAxML (the GTRGAMMA implementation uses 4 discrete rate categories). The sequences have been added one by one in random order and RAxML have inferred the best starting tree using the parsimony optimality criterion. The bootstrap analysis to get support values for the tree with the best likelihood has been achieved using a random number seed and by letting RAxML to automatically determinate a sufficient number of bootstrap replicates. The standard bootstrap search has stopped after 150 replicates with FC Bootstopping criterion: after every 50 bootstrap replicates, the RAxML performs 100 random splits of the bootstrap replicate set into two halves and computes the Pearson and Sierk correlation coefficient in the two halves from the 100 splits. Bootstrapping (BS) stops if there are at least 99 splits whose halves show a correlation coefficient greater than 0.99. Having computed the bootstrap replicate trees, they have been used to draw bipartitions on the best ML tree. Let us remark that GTRCAT, GTRCATI, and GTRGAMMAI models have been tested too in [CLBGB]. The same result, provided in supplementary data, have finally been obtained.

Bayesian inference methods have completed the study with the use of PhyloBayes MPI version 1.4f [LLB09], a message-passing-interface system for multi-core computing of the PhyloBayes 3.3f Monte Carlo Markov Chain (MCMC) sampler for phylogenetic reconstruction. According to their authors, its main distinguishing feature compared to other phylogenetic MCMC samplers is the use of non-parametric methods for modeling among-site variation in nucleotide or amino-acid propensities. The global exchange rates has been inferred from the data (CAT-GTR settings). Two runs have been launched with 32 processes on the Mésocentre de calcul de Franche-Comté computer facilities on the same alignment file (34 taxa, 22777 sites) than above and with random seeds until

reaching a largest discrepancy observed across all bipartitions (maxdiff) lower than 0.1 (using a burn-in of 1000, and sub-sampling every 10 trees).

For the robustness of the process, a last Bayesian inference using the well-known MrBayes [HR01] has been finally achieved in [CLBGB]. MrBayes is a Bayesian program for phylogeny. It has received the same data than PhyloBayes, RAxML, and PhyML, and the same substitution model. The Markov Chain Monte Carlo analysis was run for 2 millions generations. In order to estimate the posterior probabilities of phylogenetic trees, the run was sampled every 100 generations (4002 trees have been produced). New generations have been computed in the Bayesian approach until obtaining a standard deviation between the two runs launched by MrBayes lower than 0.005. Tracer (http://beast.bio.ed.ac.uk/Tracer) has been used to define a burn-in of 500 generations. The obtained results, reported in the suplementary data of [CLBGB], are coherent with PhyML, RAxML, and PhyloBayes.

Each analyses has been conducted twice, whereas FigTree 1.4.0 (http://tree.bio.ed.ac.uk) and TreeGraph 2.0.47-206 beta (http://treegraph.bioinfweb.info/) have been used as a tree plotter and a tree editor respectively.

Amino acids analyses Amino acid sequences have been used in the second series of experiments. These sequences have been obtained from the NCBI website and then aligned using M-Coffee another time. We do not provide supplementary information on these analyses based on amino acids, as obtained results are similar to the genomic approach, see [CLBGB] for further details.

E.1.1.3/ PHYLOGENY OF EUCESTODA CLASS

Figure E.1 provides a phylogeny inferred from the whole 34 complete mitochondrial genomes. Even though bootstrapping and aLRT values tend to indicate that the tree topology is not totaly robust in a few places, three monophyletic grouping clearly appear: *Taenia* spp. in clades 1a, 1b, 1c, and 2, *Echinococcus* spp. in clade 3, and the remainder of *Eucestoda* spp. in clade 4. The sister relationship between *T.mustelae* and *Echinococcus* spp. appeared in the tree of [NLI$^+$] is not supported here, and *T.mustelae* appears within the *Taenia* clade. However, bootstrapping does not support totally this assessment of the positioning of this species regarding both *Taenia* and *Echinococcus*, and this questionning raised in [CLBGB] must be confirmed in further researches.

Clade 2 consisting of *T.taeniaeformis*, *T.krepkogorski*, and *T.parva* is identical to clade 2 of [NLI$^+$]. In both cases, bootstrapping are sufficiently large to consider the phylogenetic relationship between these species as resolved: *T.taeniaeformis* and *T.krepkogorski* are sister species and their clade is sister to *T.parva*. The clade 2 retains a sister position to the other members of *Taenia* spp. except for *T.mustelae*. This assertion of [CLBGB], which is along the state-of-the-art (both molecular [NLI$^+$] and morphological [HAQJ01, HJR$^+$00, Hob06] analyses, except that *T.krepkogorski* has not been regarded by Hoberg *et al.*), is highly supported by Bayesian analyses and ML.

The remainder of *Taenia* species constitutes the clades 1a-c. Each clade alone has a well-supported tree topology, but the relative positions of these three clades is not highly supported by bootstrapping: the relationship of *T.hydatigena* regarding to these three clades cannot be obtained with ML or Bayesian analyses. Nodes' clusters separated by

Accession	Species	Cestodes' order	Order 2
NC_012147	*Clonorchis sinensis*	X	
NC_002546	*Fasciola hepatica*	X	
NC_011127	*Opisthorchis felineus*	X	
NC_002354	*Paragonimus westermani*	X	
NC_008074	*Schistosoma haematobium*		X
NC_002544	*Schistosoma japonicum*	X	
NC_002545	*Schistosoma mansoni*		X
NC_002529	*Schistosoma mekongi*	X	
NC_008067	*Schistosoma spindale*		X
NC_009680	*Trichobilharzia regenti*	X	

Table E.3: Gene order in Trematoda species. Order 2 is: *cox1*, *cox2*, *nad6*, *atp6*, *nad2*, *nad5*, *cox3*, *cytb*, *nad4l*, *nad4*, *nad3*, *nad1*.

long branches in *Taenia* species claim in [CLBGB] for a taxonomy revision within *Taenia*, as it has already been suggested by previous biomolecular phylogenic studies (see [NLI+] for instance).

Clade 3 is only constituted by *Echinococcus* spp. This phylogenetic relationships of *Echinococcus* is identical to the cladogram inferred by the nucleotide data of mitochondrial genes [NMS+07] (ML and partitioned Bayesian analyses using concatenated data sets of nucleotide and amino acid sequences) and [HNW+07]. Among other things, sister species relationships between *E. ortleppi* and *E. canadensis*, and between *E. multilocularis* and *E. shiquicus* have been confirmed in [CLBGB]. It is very closed to the one of [SJM+09], inferred by the nucleotide data of nuclear genes, the sole differences are the relative position of *E.shiquicus* and *E.multilocularis* regarding the remainder of the clade.

The remainder of the Eucestoda phylogenetic tree is constituted by clade 4 which illustrates that *Diplogonoporus* spp. (*Dipl.latum* and *Dipl.nihonkaiense*) have a sister relationship with *Diphyllobothrium* spp. (*Dipl.balaenopterae* and *Dipl.grandis*) and that *Spirometra erinaceieuropaei* has a sister position to the other members of this clade. ML and Bayesian analyses yield a robust support for this clade [CLBGB].

E.1.1.4/ DISCUSSION

Phylogeny of Eucestoda class cannot be inferred using gene order, as all species studied here share the same arrangement of genes (*i.e.*, same order). In addition of using an amino acids multi-locus approach for investigating Eucestoda phylogeny, we have considered the whole mitochondrial genomes in [CLBGB], to reduce sequences pre-treatments leading to possible fakes and noises in deduced relationships. This way to proceed allows to present a complementary approach for *Taenia* and *Echinococcus* species compared to previous studies, reinforcing by doing so the confidence in convergent results. Furthermore, the number of investigated taxa is larger than previous molecular phylogenies, even for the *Taeniidae* family.

The choice of *Schistosoma mekongi* as outgroup is justified as follows [CLBGB]. As there was no molecular phylogeny of cestodes encompassing both *Pseudophyllidea* and *Cyclophyllidea* orders, it would not be possible to determine without ambiguity the first cestoda which has gone away from the remainder of this class, that is, the most divergent

species of this class. Furthermore, our desire was to establish a molecular phylogeny of the whole cestodes whose mitochondrial genome was available, we could not lost a cestode species by taking it as outgroup. Trematode species have the double advantage to be a separated class but which is sufficiently close to cestodes. When investigating the genes order of this class, we discovered in [CLBGB] that all the species whose complete mitochondrial genome was available shared a same genes order with cestodes, except for some species within the *Schistosoma* genus (see Table E.3). The fact that trematodes and cestodes are two different classes is sometimes disputed due to close morphological and ecological characteristics. This is why the outgroup is within the *Schistosoma* genus, because the presence of a second genes order that can be found neither in the remainder of trematodes nor in cestodes claims for an old divergence of this genus. However we wanted to realize global alignment of the whole mitochondrial genomes, so we choosed in [CLBGB] the outgroup between members of this genus that share the same gene order than the cestodes, that is, between *Schi.japonicum* and *Schi.mekongi*. This choice is probably not optimal, and distance between *Schi.mekongi* taxa and all Eucestoda must be investigated more deeply.

Previous studies suggested a possibility that at least two cryptic species might exist within the *T.taeniaeformis* taxon, which uses felids as definitive and rodents as intermediate hosts [JYL+12, NLI+, LHL+08]. The phylogenetic study presented in [CLBGB] illustrates that these species could be *T.krepkogorski* and *T.parva*.

The previous molecular phylogeny of Nakao *et al.* inferred from a data set of 18S rDNA on the one hand, and on the protein-coding genes *pepck* and *pold* has demonstrated *Taenia* to be paraphyletic [NLI+]. This study has led to the following monophyletic clades within members of *Taenia*.

Clade 1a, constituted by very closed species, is characterized by the utilization of *Bovidae* and *Suidae* as intermediate hosts. It contains human-taenia spp. This clade 1a corresponds exactly to the clade 1a of Nakao *et al.* [NLI+]. Clade 1b, which corresponds to *Taenia* using members of the order *Rodentia* as intermediate hosts, are equal in [NLI+] and in the present analysis. It contains *T.martis*, *T.twitchelli*, and *T.crassiceps*.

These two clades appear for the second time in a molecular phylogeny study and using various sets of data, while they have no correspondence in the previously proposed genera within the *Taeniidae*. As in [NLI+], such a result strongly suggests that the former proposed genera are invalid. Another common result with Nakao *et al.* is that both the pattern of their host selection and their phylogenetic positions are quite approved by morphological phylogenies ([HJR+00, Hob06]).

A third clade (1c) sister to clade 1a were not present in [NLI+]. It contains *T.pisiformis* and *T.laticollis*. These species have closed geographical range, intermediate, and definitive hosts, as stated in [HJR+00]. This result provides the position of *T.laticollis* compared to membes of clades 1a and 1b, a question raised by Nakao *et al.* [NLI+]. However, *T.hydatigera* continues in [CLBGB] to have a problematic cladistic relationship with other species within the genus *Taenia*.

In the last phylogeny study inferred from nuclear protein-coding genes [NLI+], clade 2 consisting in *T.parva*, *T.krepkogorski*, and *T.taeniaeformis* that utilize rodents as intermediate hosts, and viverrids and felids as definitive carnivore hosts, appeared as sister to all the other *Taeniidae* taxa including *Echinococcus*, whereas analyses on both nuclear 18S rDNA and mitochondrial protein-coding genes consider that clade 2 is a sister of *Taenia* s.s. The ML and Bayesian analyses on complete mitochondrial genomes or cod-

Current name	Nakao *et al*'s proposal [NLI$^+$]
Taenia krepkogorski	*Hydatigera krepkogorski*
Taenia parva	*Hydatigera parva*
Taenia taeniaeformis	*Hydatigera taeniaeformis*
Taenia mustelae	*Versteria mustelae*

Table E.4: Summary of taxonomic changes in [NLI$^+$]

ing sequences presented in [CLBGB] validate the second hypothesis. More precisely, this study confirms the need of the resurrection of *Hydatigera* for clade II, as it has been formerly proposed in [KNY$^+$11, NLI$^+$]. This genus currently contains *Hydatigera parva* (Baer, 1926) Wardle & McLeod, 1952, *H.taeniaeformis* (Batsch, 1786) Lamarck, 1816, and *H.krepkogorski* Landa & Schultz, 1934. This genus *Hydatigera* Lamarck, 1816 was historically composed by 6 species, namely *Taenia balaniceps* Hall, 1910, *T.laticollis*, *T.lyncis* Skinker, 1935, *T.macrocystis* (Diesing, 1850), *T.parva*, and *T.taeniaeformis*, while *T.krepkogorski* and *T.hyperborea* von Linstow, 1905 have been added later into this genus in [Yam59] and [Abu64] respectively. They majorly share in common the primary use of felids as definitive hosts and the particular structure of their strobilocescus. As in [NLI$^+$], we conclude in [CLBGB] to the fact that *T.laticollis* and *T.crassiceps* do not belong into the *Hydatigera* genus, even if they are not very distant together in the *Taeniidae* family.

The basal position of *T.mustelae* has been already stated using both morphological [HJR$^+$00] and both mitochondrial protein-coding and nuclear genes [NLI$^+$] analyses. As in Nakao *et al.*, *T.mustelae* appears as sister to *Echinococcus* in [CLBGB]. The molecular phylogenetic analysis presented in our article thus confirms that *T.mustelae* is more linked to *Echinococcus* than to other *Taenia* species. This assessment is consistent with the erection of a new genus, namely *Versteria*, formerly proposed in *Nakao13*. This erection has been proposed after convergent molecular and morphological observations: quoting Nakao *et al.*, "*T.mustelae* differs distinctively from the members of *Taenia* ss. in its morphological miniaturization, especially in its very small rustellar hooks."

E. oligarthrus and *E. vogeli*, whose definitive hosts are derived from carnivores that immigrated from North America after the formation of the Panamanian land bridge [NMS$^+$07], occupies the the basal positions of the phylogenetic tree. Like in [NMS$^+$07], our study [CLBGB] reinforces the theory suggesting that the ancestral homeland of Echinococcus was North America or Asia, depending on whether the ancestral definitive hosts were canids or felids.

The remainder of our phylogenetic analysis of the Eucestoda class brings the relative relationship of *Diphyllobothrium*, *Diplogonoporus*, *Dipylidium*, *Hymenolepis*, and *Spirometra* genera with regards to the *Taeniidae* family. Three families of Eucestoda are represented in the data set of [CLBGB], namely the *Taeniidae*, *Hymenolepididae*, and *Diphyllobothriidae* ones. These three families are clearly apparent in our study: *Spirometra*, *Diphyllobothrium*, and *Diplogonoporus* species are grouped together in 4, all the *Taeniidae* sp. (clades 1, 2, and 3) constitute the upper side of the tree, and its remained part is constituted by the two species of the *Hymenolepididae* family, that is, *Dipylidium caninum* and *Hymenolepis diminuta*. The order level is respected: clade 4 constituted of genera *Spirometra*, *Diplogonoporus*, and *Diphyllobothrium* are within the *Diphyllobothriidae* family, so they belong in the *Pseudophyllidea* order. This clade presents a sister relationship with the remainded species, which are all in the *Cyclophyllidea* order: *Dipylidium caninum*, *Hymenolepis diminuta*, and the *Taeniidae*, another family of *Cyclophyllidea* or-

der [CLBGB].

E.1.2/ ANCESTOR RECONSTRUCTION

E.1.2.1/ ALGORITHMIC METHOD

Having available a phylogeny of the *Cestodes* considered as reliable by us, our next research work in the study of complex evolution of these species were to reconstruct the mitochondrial genome of each ancestor of the *Taeniae* genus. Our approach is detailed thereafter.

Consider the two closest species[2] having a sister relationship in the phylogenetic tree of Figure E.1. We call brother1 and brother2 these two species, while cousin1, cousin2, and cousin3 are the three closest genomes of brother1 and brother2. We achieve two runs of Muscle [Edg04] alignment tool on this set of mitochondrial genomes (complete sequences): the first run allows us to realize a circular shift on these genomes, as a pre-treatment, to improve the accuracy of the second final alignment of length n.

Denote by $b_1[k]$ (resp. $b_2[k]$, $c_1[k]$, $c_2[k]$, $c_3[k]$, and $a[k]$) the k-th letter of the DNA sequence (mitochondrial genome) of brother1 (resp. brother2, cousin1, cousin2, cousin3, and the ancestor of brother1 and brother2). Then, for $k = 1..n$:

1. if $b_1[k] = b_2[k]$, then $a[k] = b_1[k]$,
2. elif $b_1[k] = c_1[k] = c_2[k] = c_3[k]$, then $a[k] = b_1[k]$,
3. elif $b_2[k] = c_1[k] = c_2[k] = c_3[k]$, then $a[k] = b_2[k]$,
4. elif $b_1[k] = c_1[k] = c_2[k]$, then $a[k] = b_1[k]$,
5. elif $b_2[k] = c_1[k] = c_2[k]$, then $a[k] = b_2[k]$,
6. elif $\{b_1[k], b_2[k], c_1[k], c_2[k]\} \subset \{A, G\}$, then $a[k] = R$ (purine),
7. elif $\{b_1[k], b_2[k], c_1[k], c_2[k]\} \subset \{A, C\}$, then $a[k] = M$,
8. elif $\{b_1[k], b_2[k], c_1[k], c_2[k]\} \subset \{T, C\}$, then $a[k] = Y$ (pyrimidine),
9. elif $\{b_1[k], b_2[k], c_1[k], c_2[k]\} \subset \{A, T\}$, then $a[k] = W$,
10. elif $\{b_1[k], b_2[k], c_1[k], c_2[k]\} \subset \{G, T\}$, then $a[k] = K$,
11. elif $\{b_1[k], b_2[k], c_1[k], c_2[k]\} \subset \{G, C\}$, then $a[k] = S$,
12. elif '-' in $\{b_1[k], b_2[k]\}$ then $a[k] =?$ (a nucleotide or an insertion -),
13. else $a[k] = N$ (any nucleotide).

The process is iterated again by replacing the two brothers by their ancestor.

We have applied this procedure to the subtree of *Taeniae* depicted in Figure E.2. By doing so, four ancestors have been reconstructed, the oldest one being the common ancestor of the clade constituted by *T.asiatica*, *T.saginata*, *T.multiceps*, *T.madoquae*, and *T.serialis*.

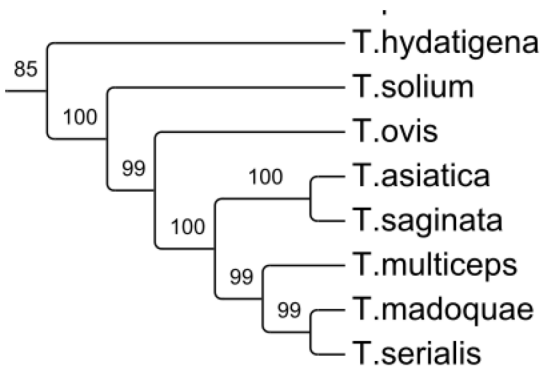


Figure E.2: Subtree whose ancestors have been reconstructed

b_1	b_2	c_1	c_2	c_3	a
T.saginata	*T.asiatica*	*T.multiceps*	*T.serialis*	*T.madoquae*	ancestor1
T.serialis	*T.madoquae*	*T.multiceps*	*T.saginata*	*T.asiatica*	ancestor2
ancestor2	*T.multiceps*	*T.saginata*	*T.asiatica*	*T.ovis*	ancestor3
ancestor1	ancestor3	*T.ovis*	*T.solium*	*T.hydatigena*	ancestor4

Table E.5: “Brothers” and “cousins” in *Taeniae* genus

For each couple of brother used in our simulations, the selected cousins and the name of the ancestor are detailed in Table E.5.

The number of ambiguous nucleotides (not ’ATCG-’ but ’RNYWKSN?’) in each ancestor are given in Table E.6, while the aNy column counts specifically the occurrences of ’N’ in the ancestor. The single nucleotide polymorphisms between the couples of brothers, and between each brother and its ancestor, are provided too as an interval: the first value focuses on the ’ATCG-’ alphabet while the second one is for the extended ’ATCGRNYWKSN?-’ alphabet. In other words, the real number of SNPs is between these two extremal values.

Table E.7, for its part, gives the number of mutations *per trinucleotide site* between a species and its direct ancestor. The position of the first ’?’ specified, and the number of mutations are only considered before this position, as the location of these mutations become impossible after having a doubt between a nucleotide and an indel ’-’, which is signaled in the ancestor with a character ’?’. In the obtained results, it is not obvious that the third trinucleotide site is preferred. However, we should determine whether an higher mutation rate on the third site in Table E.7 is correlated to the nature of the sequence (coding or not): indeed, due to our circular rotation of some genomes in the first run of alignments, we do not know this nature, which may change from one row to another one

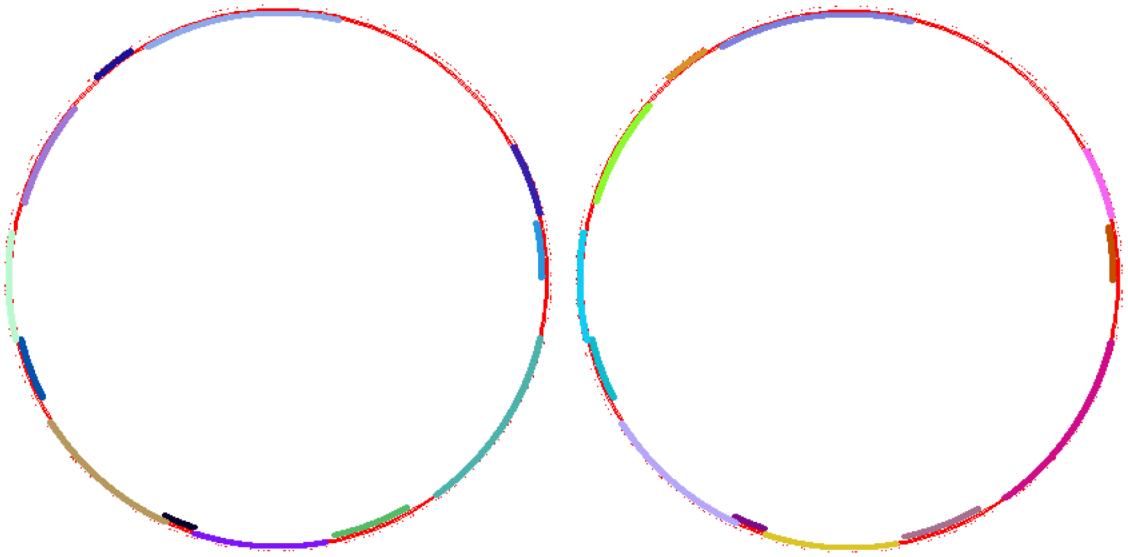
[2]Number of SNPs (single nucleotide polymorphisms) in the global alignment of the whole genomes

b_1	b_2	Ambiguous	aNy	$snp(b_1, b_2)$	$snp(b_1, a)$	$snp(b_2, a)$
T.saginata	*T.asiatica*	286	96	819	299..554	296..551
T.serialis	*T.madoquae*	502	156	1229	450..913	355..818
ancestor2	*T.multiceps*	604	487	728	198..673	427..996
ancestor1	ancestor3	742	523	$\geqslant 630$	211..902	276..942

Table E.6: SNPs between couple of genomes and their ancestor

Species	Situation known until position	Distribution of SNPs per trinucleotide site
ancestor1	718	(2,33,7)
ancestor2	787	(1, 30, 7)
ancestor3	718	(6,35,11)
T.asiatica	3581	(31, 20, 73)
T.madoquae	1888	(59, 49, 14)
T.multiceps	787	(5, 41, 12)
T.saginata	3581	(43, 43, 27)
T.serialis	1888	(58, 52, 7)

Table E.7: Mutations per trinucleotide site



(a) Asiatica (b) Saginata

Figure E.3: SNPs locations compared to the ancestor

in this table.

Finally, Figure E.3 indicates the locations of genes in *T.asiatica* and in *T.saginata*, and the locations of SNPs they share with their common ancestor (that is, ancestor1): no correlation clearly appears between the coding character of a sequence and the mutations' density, even though silent mutations (without modification of the associated amino acid, due to the recurrent character of the genetic code) may be more frequent in coding sequences.

The last common ancestor (ancestor4) is provided in Appendix F. The next sections are devoted to our first investigations in the attempts to provide a confidence score for this ancestor.

E.1.2.2/ MATHEMATICAL FOUNDATIONS

For the sake of simplicity, we now consider only one cousin:

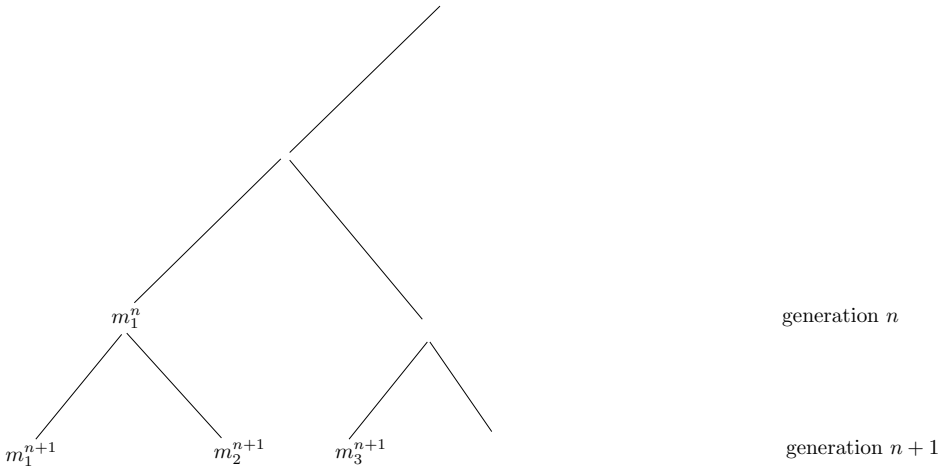


Figure E.4: Tree of ancestors

1. if the two brothers b_1 and b_2 are equal in position l, then we put $b_1[k]$ in $a[k]$
2. elif $b_1[k] = c[k]$, then $a[k] = b_1[k]$
3. elif $b_2[k] = c[k]$, then $a[k] = b_2[k]$
4. else $a[k] = N$.

Our reconstruction confidence can be evaluated as follows[3]. Let $\{X_n,\ n \in \mathbb{N}\}$ be the Markov chain such that, at generation n, $X_n = (X_n(i))_{i=1,...,2^n}$ is the set of individuals $i \in \{1, ..., 2^n\}$ with genetic material $X_n(i)$. $X_0 = X_0(1)$ is the original ancestor. We let $\mathcal{S}$ the set of values of the $X_n(i)$'s, so that for all n, X_n takes its values in set $\mathcal{S}^{2^n}$. It is assumed that the transition between X_n and X_{n+1} is such that

$$\mathbb{P}(X_{n+1} = (m_i^{n+1})_{i=1,...,2^{n+1}} |\ X_n = (m_i^n)_{i=1,...,2^n}) = \prod_{k=1}^{2^n} [p(m_{2k-1}^{n+1}, m_k^n).p(m_{2k}^{n+1}, m_k^n)]$$

$$= \prod_{k=1}^{2^n} [\mathbb{P}(X_{n+1}(2k-1) = m_{2k-1}^{n+1} |\ X_n(k) = m_k^n).\mathbb{P}(X_{n+1}(2k) = m_{2k}^{n+1} |\ X_n(k) = m_k^n)]$$

for some given function $p(.,.)$ that explains the probabilistic mechanism that gives expression of genes from one parent to its immediate two children. It is clear that, if $(k_n)_{n\in\mathbb{N}}$ is a path along the tree partially represented in Figure E.4, with $k_n \in \{1, ..., 2^n\}$ for all n and $k_{n+1} \in \{2k_n - 1, 2k_n\}$, then $\{X_n(k_n),\ n \in \mathbb{N}\}$ is a Markov chain with state space $\mathcal{S}$. For example, $\{X_n(1),\ n \in \mathbb{N}\}$ is a Markov chain, which corresponds to the leftmost path on the tree. However, if $(k_n)_{n\in\mathbb{N}}$ and $(k'_n)_{n\in\mathbb{N}}$ are two paths, corresponding Markov chains $\{X_n(k_n),\ n \in \mathbb{N}\}$ and $\{X_n(k'_n),\ n \in \mathbb{N}\}$ have same distribution, but are not independent (as they are issued from the same starting point $X_0(1)$).

In practice, we observe $X_{n+1} = (X_{n+1}(i))_{i=1,...,2^{n+1}}$ and we want to "reconstruct" one ancestor $X_n(1)$. To this end, if said observations are m_i^{n+1}, $i \in \{1, ..., 2^{n+1}\}$, we construct

[3]This proof is due to our colleagues Landi Rabehasaina and Romain Biard (LMB), “mandated” by us to solve the problem of the pertinence of the reconstructed ancestors.

deterministically a candidate $\tilde{m}_1^n$ for $X_n(1)$ thanks to some (deterministic) function

$$\tilde{m}_1^n := f(m_1^{n+1}, m_2^{n+1}, m_3^{n+1})$$

that takes into account both immediate descendants $X_{n+1}(1)$ and $X_{n+1}(2)$ of $X_n(1)$, but also its "cousin" $X_{n+1}(3)$. The goal is to choose f adequately such that

$$p_f(m_1^{n+1}, m_2^{n+1}, m_3^{n+1}) := \mathbb{P}(X_n(1) = f(m_1^{n+1}, m_2^{n+1}, m_3^{n+1}) | \, X_{n+1}(1) = m_1^{n+1}, X_{n+1}(2) = m_2^{n+1}) \tag{E.1}$$

is close to 1. A close form of this quantity is available if the distribution π_n of $X_n(1)$ is known. In that case, (E.1) is equal to

$$\begin{aligned} & p_f(m_1^{n+1}, m_2^{n+1}, m_3^{n+1}) \\ = \; & \frac{\mathbb{P}(X_{n+1}(1) = m_1^{n+1}, X_{n+1}(2) = m_2^{n+1} | \, X_n(1) = f(m_1^{n+1}, m_2^{n+1}, m_3^{n+1})).\mathbb{P}(X_n(1) = f(m_1^{n+1}, m_2^{n+1}, m_3^{n+1}))}{\mathbb{P}(X_{n+1}(1) = m_1^{n+1}, X_{n+1}(2) = m_2^{n+1})} \\ = \; & \frac{p\left(m_1^{n+1}, f(m_1^{n+1}, m_2^{n+1}, m_3^{n+1})\right).p\left(m_2^{n+1}, f(m_1^{n+1}, m_2^{n+1}, m_3^{n+1})\right).\pi_n(f(m_1^{n+1}, m_2^{n+1}, m_3^{n+1}))}{\sum_{m \in \mathcal{S}} p(m_1^{n+1}, m).p(m_2^{n+1}, m)\pi_n(m)}. \end{aligned} \tag{E.2}$$

There are two approaches:

- Either n is small and π_n is an *a priori* distribution fixed by an expert,
- Or n is large, in which case one can suppose that X_n has reached its stationary distribution, i.e.

$$\pi_n \approx \pi_{n+1} \approx \frac{1}{2^{n+1}} \sum_{i=1}^{2^{n+1}} \delta_{X_n(i)}. \tag{E.3}$$

E.1.2.3/ EXPERIMENTAL EVALUATION

We have measured the confidence we can put in *ancestor1*, the direct ancestor of *T.asiatica* and *T.saginata*. Listing E.5 details how we have defined the ancestor function f that appears in the previous section.

```
1 def f(X,Y,Z):
2     if X==Y: return X
3     elif X==Z: return X
4     elif Y==Z: return Y
5     else: return 'N'
```

Figure E.5: The ancestor function f

Then, we have computed the mutation matrix between *ancestor1* and its two children. It is equal to[4]

$$\begin{pmatrix} 0.767945544554 & 0.0210396039604 & 0.0717821782178 & 0.139232673267 \\ 0.0763052208835 & 0.746987951807 & 0.0542168674699 & 0.122489959839 \\ 0.0702247191011 & 0.0147471910112 & 0.78441011236 & 0.130617977528 \\ 0.0650429799427 & 0.0309455587393 & 0.0547277936963 & 0.849283667622 \end{pmatrix}$$

[4] Remark that this mutation matrix is compatible with none of the state-of-the-art nucleotides evolutionary models recalled at the beginning of this part.

where rows and columns follow the 'ACGT' order and, at row X and column Y, we found $p(X, Y)$, which is equal to the probability a nucleotide X in *ancestor1* becomes Y in either *T.asiatica* or *T.saginata*. To obtain this matrix, we started with a 4×4 matrix initiated by zeros, and for each position k:

- we add 0.5 at row X and column Y_1, where $a[k] = X$ and $b_1[k] = Y_1$,
- we add 0.5 at row X and column Y_2, where $a[k] = X$ and $b_2[k] = Y_2$.

Finally, the matrix is renormalized to have a sum of each row equal to 1.

π_n, for its part, is equal to 0.24221, 0.07908, 0.47257, and 0.20614 for respectively A, C, T, and G. To obtain it, we have simply computed the frequency of each nucleotide in the genome of *T.saginata*.

With all this material, it has been possible to compute $p_f(b_1, b_2, c)$, which measures the confidence we can put in our ancestor $a(b_1, b_2)$. The obtained results are provided in Table E.8. In this table, a p_f probability close to 1 indicates that we can have an high confidence in the associated ancestor reconstruction. Low probabilities are due to the simplifications we have operate on our process to be able to compute mathematically these probabilities. In particular, probabilities equal to 0 are due to situations where we put an any 'N' nucleotide in the ancestor: in that case, the letter put in the reconstructed ancestor is never the nucleotide (letter) that was really present in the true real ancestor. Deeper investigations in that direction, to be closer to the process really implemented (as it is described in Section E.1.2.1), will improve the values contained in Table E.8.

E.1.2.4/ POSSIBLE IMPROVEMENT: TO INFER MUTATION LAW ON GENE SCALE

Study presented in previous sections is a proof of concept showing that it is possible to reconstruct the ancestors of closed species, and that a confidence score can be deduced from the reconstruction. However, this first approach presents various flaws, the most important one being probably to suppose that the mutation rate is constant whatever the location, leading to a constant mutation matrix.

We have regarded how to relax this hypothesis using the so-called graphical model approach [KFGT07]: the initial hypothesis is that any nucleotide's probability mutation depends on all the other nucleotides of the considered gene, as follows. Consider a gene of length n. The Markov chain is supposed *a priori* to be a complete graph with 4^n nodes, having each vertex connected to all other ones: each gene can *a priori* becomes any other gene due to mutations. However, almost all the edges have a probability close to 0. The graphical model approach needs a large collection of data, which is used to remove, in this graph with $(4^n)^2$ edges, a very large amount of edges, in such a way that the remained ones are those whose probability may be not negligible, as inferred by the observed data.

With Stéphane Chrétien from the LMB, we have started to study the graphical model approach for genes mutations, in the particular case of NAD3 gene in the *Cestodes*, *Nematodes*, and *Trematodes* families. To do so, we have extracted its DNA sequence in 100 genomes of these families, realize a global alignment using Muscle, computed a similarity distance between each aligned sequence (using the same measure than in the Needleman-Wunch algorithm, with EDNAFULL amino acid matrix, and penalties of -10

b_1	b_2	c	$p_f(b_1, b_2, c)$	b_1	b_2	c	$p_f(b_1, b_2, c)$
A	A	A	0.9762	A	A	C	0.9762
A	A	G	0.9762	A	A	T	0.9762
A	C	A	0.4082	A	C	C	0.4702
A	C	G	0.0	A	C	T	0.0
A	G	A	0.4997	A	G	C	0.0
A	G	G	0.4250	A	G	T	0.0
A	T	A	0.4740	A	T	C	0.0
A	T	G	0.0	A	T	T	0.4778
C	A	A	0.4082	C	A	C	0.4702
C	A	G	0.0	C	A	T	0.0
C	C	A	0.9864	C	C	C	0.9864
C	C	G	0.9864	C	C	T	0.9864
C	G	A	0.0	C	G	C	0.4742
C	G	G	0.3531	C	G	T	0.0
C	T	A	0.0	C	T	C	0.3484
C	T	G	0.0	C	T	T	0.5982
G	A	A	0.4997	G	A	C	0.0
G	A	G	0.4250	G	A	T	0.0
G	C	A	0.0	G	C	C	0.4742
G	C	G	0.3531	G	C	T	0.0
G	G	A	0.9776	G	G	C	0.9776
G	G	G	0.9776	G	G	T	0.9776
G	T	A	0.0	G	T	C	0.0
G	T	G	0.4588	G	T	T	0.4771
T	A	A	0.4740	T	A	C	0.0
T	A	G	0.0	T	A	T	0.4778
T	C	A	0.0	T	C	C	0.3484
T	C	G	0.0	T	C	T	0.5982
T	G	A	0.0	T	G	C	0.0
T	G	G	0.4588	T	G	T	0.4771
T	T	A	0.9731	T	T	C	0.9731
T	T	G	0.9731	T	T	T	0.9731

Table E.8: p_f values in all situations

and of -0.5 for gap opening and extending respectively). Additionally to this similarity matrix of size 100×100, we have translated all the obtained alignments as binary sequences, having a 1 in position k in the work in row i, column j, if and only if the NAD3 gene in genomes i and j present the same nucleotide in position k. With all this material, we are currently computing the graphical model method. It will allow us to obtain the first global mutation law of a gene, and to compare it to the constant mutation matrix obtained previously.

E.2/ TOWARDS THE ANCESTOR OF THE *Mycobacterium Tuberculosis* COMPLEX

E.2.1/ GENERAL PRESENTATION

The *Mycobacterium tuberculosis* complex (MTBC) consists of a few bacterial species inside the *Mycobacterium* genus that evolve in a clonal manner. This complex contains *M.tuberculosis*, *M.bovis*, *M.bovis BCG*, and *M.africanum*. The most commonly accepted scenario is that MTBC has appeared 40,000 years ago in the Horn of Africa, its progenitor species (or last common ancestor) being probably an evolved and mutated *M.canettii*, the smooth group of *M.tuberculosis* [BHS+12].

We currently have available in the NCBI website 65 complete genomes of *Mycobacterium* genus, among which 31 belong in the strict MTBC and 5 are *M.canettii* strains. However, during our first investigations of this complex [GCR+], only some of them were accessible, that is, the 18 genomes of Table E.9. Our final objective regarding MTBC was to validate (or not) the hypothesis considering that the last common ancestor of this complex is related to *M.canettii*. Remark that *M.canettii* still remains an existing bacteria, we thus have a complex of species having numerous different strains that have evolved these last 40,000 years from an archaic *M.canettii* on the one hand, and the current strains of *M.canettii* that have evolved too from their ancestral compositions on the other hand. Our proposal in [GCR+] was to reconstruct the ancestors of both the MTBC and of the genomes of *M.canettii* currently available, and to compare the two obtained ancestors.

To achieve this goal, we have chosen the following steps for the two groups of species [GCR+]:

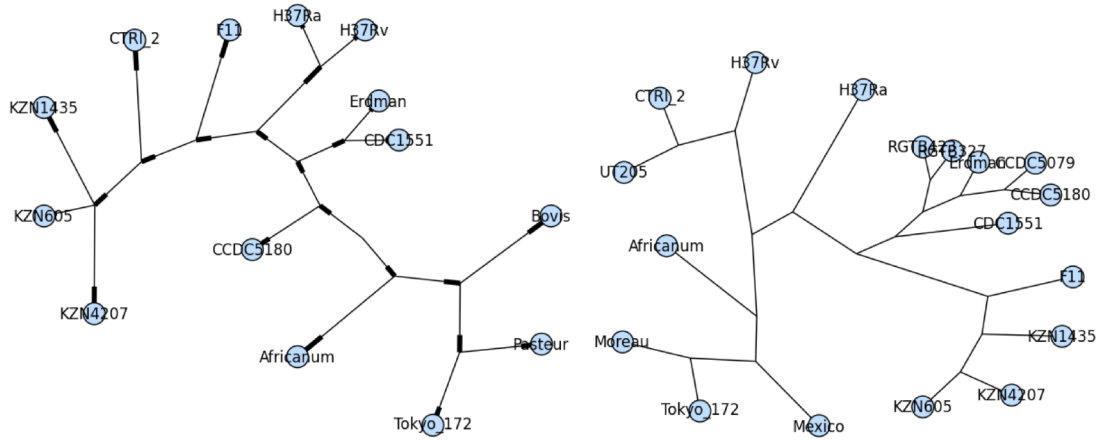
1. To determine genes constituting the core genome of these species, that is, genes belonging in all the genomes.

2. To extract genes in the core genome in order to reconstruct a reliable phylogeny of this collection of strains (and validate or not the phylogenetic proposal of [BHS+12] recalled in Figure E.6(a)).

3. To reconstruct each ancestor in this phylogenetic tree, until the last common one.

4. Finally, to compare the last common ancestors of the two groups of strains (MTBC and *M.canettii*).

Each of these steps is detailed hereafter.



(a) Minimum spanning tree on 13382 SNPs [BHS+12] (b) Maximum Likelihood on genes order and content (EPFL)

Figure E.6: Phylogenetic trees of the MTBC complex

Accession nb.	Species	Strain	Nb. of transposases
NC_016804.1	*M.bovis BCG*	Mexico	54
AM412059.2	*M.bovis BCG*	Moreau RDJ	54
NC_002755.2	*M.tuberculosis*	CDC1551	45
NC_009525.1	*M.tuberculosis*	H37Ra	73
CP001642.1	*M.tuberculosis*	CCDC5180	76
NC_000962.2	*M.tuberculosis*	H37Rv	82
CP003233.1	*M.tuberculosis*	RGTB327	82
NC_012943.1	*M.tuberculosis*	KZN 1435	85
CP002992.1	*M.tuberculosis*	CTRI-2	77
CP001976.1	*M.tuberculosis*	KZN 605	83
CP001662.1	*M.tuberculosis*	KZN 4207	76
AP012340.1	*M.tuberculosis*	Erdman	57
NC_016934.1	*M.tuberculosis*	UT205	0
NC_015758.1	*M.africanum*	GM041182	55
CP001641.1	*M.tuberculosis*	CCDC5079	79
CP003234.1	*M.tuberculosis*	RGTB423	77
NC_009565.1	*M.tuberculosis*	F11	87
NC_012207.1	*M.bovis BCG*	Tokyo 172	52

Table E.9: Transposases per genome in the MTBC

E.2.2/ CORE AND PAN GENOME

To set the ideas, let us explain in a short example what are the core and pan genomes. Consider a genome G_1 constituted by three genes: g_1, g_2, and g_3 while a second genome G_2 is constituted by g_1, g_3, g_4, and g_5. The core genome of G_1 and G_2 is their intersection, that is, $\{g_1, g_3\}$, while the pan genome is their union: $\{g_1, g_2, g_3, g_4, g_5\}$.

Core genome is useful to determine which genes are necessary for the bacteria, that is, the fundamental functionality of the species, whereas pan genome helps to understand the variability inside the strains. Additionally, molecular phylogenies compare DNA sequences shared in common in the collection of regarded species. So, in order to construct a phylogeny of MTBC, we need to find genes shared by the whole species, that is, genes belonging in the core genome.

To determine the core genome of our 18 genomes of MTBC, we have downloaded the coding sequences of the 18 complete genomes from the NCBI website [GCR+]. These DNA sequences contain approximately 4,000 genes (protein coding sequences, exons and introns) in a fasta file having the following form:

```
>lcl|NC_002755.2_cdsid_NP_334410.1 [gene=dnaA] [protein=chromosomal
replication initiation protein] [protein_id=NP_334410.1] [location=1..1524]
TTGACCGATGACCCCGGTTCAGGCTTCACCACAGTGTGGAACGCGGTCGTCTCCGAACTTAACGGCGACC
CTAAGGTTGACGACGGACCCAGCAGTGATGCTAATCTCAGCGCTCCGCTGACCCCTCAGCAAAGGGCTTG
GCTCA...CGTCAAAGAACTCACCACTCGCATCCGTCAGCGCTCCAAGCGCTAG
>lcl|NC_002755.2_cdsid_NP_334411.1 [gene=dnaN] [protein=DNA polymerase
III subunit beta] [protein_id=NP_334411.1] [location=2052..3260]
ATGGACGCGGCTACGACAAGAGTTGGCCTCACCGACTTGACGTTTCGTTTGCTACGAGAGTCTTTCGCCG
ATGCGGTGTCGTGGGTGGCTAAAAATCTGCCAGCCAGGCCCGCGGTGCCGGTGCTCTCCGGCGTGTTGTT
GACCGGCT...
```

Genes are thus constituted by an headline starting by $>$, which contains various information like the gene's name and location, and followed by the DNA coding sequence.

A first idea to constitute the core genome may be to use gene names: core genome is constituted by genes whose name are present in all the genomes whereas pan genome has genes whose name appears at least once in the collection of genomes. However, this simple approach cannot work, as:

- Annotation tools used by researchers making the submission of their genome in the NCBI website are manifold, leading to different names for a same gene in the collection of genomes (a same sequence identified as two different genes using two different tools).
- The same gene may be well-identified, but naming conventions may be different in two different annotation tools (spelling mistakes, various conventions for using capital letters or abbreviations).
- Finally, some genes have no name, but simply an indication like "putative protein" as in the UT-205 genome.

Our approach in [GCR+] has then consisted in extracting the $\approx 80,000$ DNA coding sequences from these 18 genomes, and to compare them two by two using the *needle*

command of the Emboss package, which is based on the so-called Needleman-Wunch global alignment algorithm (the computation of the $80,000^2$ similarity scores has required 2 months using 50 cores on the Mésocentre de calcul de Franche-Comté). Roughly speaking, two DNA sequences will be considered as two alleles of a given gene if their similarities are greater than a predefined threshold. The idea at the basis of our approach was that:

- All strains in the complex are supposed to derive from a common ancestor, and their clonal evolution reduces horizontal genes transfer. So, apart a few gene loss or gain due to mutations, genomic recombination, or transposases (TEs), all strains must share approximately the same collection of genes, which is the one of the ancestor. To say this another way, the core genome of the complex should be close to the pan genome.
- This MTBC complex is very recent, so the various alleles of each genes must be close one to each other.

The result of this computation is an undirected graph whose vertices are the $80,000$ coding sequences $g_1, ..., g_{80000}$, each vertex being connected to the $80,000$ other vertices. The edge between two vertices g_i and g_j, for its part, is the similarity score $r_{i,j}$ obtained after the global alignment of these two DNA coding sequences. Finally, all edges having a similarity score lower than a given threshold (set at 90% in our experiments) are removed from the graph, and the remained connected components of the disconnected graph are called *genes*.

- Genes whose connected component contains at least one representative of each genome belong in the core genome.
- Pan genome is the collection of all the connected components.

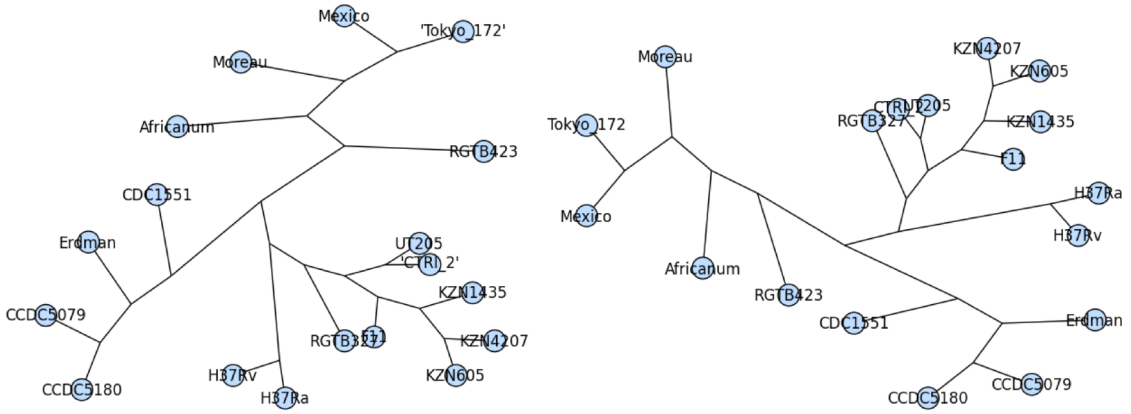
For the MTBC and a threshold set at 90%, we found a core genome of 1349 genes and a pan genome of 7323 genes. $1,177,378$ SNP locations were identified after having realized a global alignment of each gene (18 alleles) and counting the variations in these alignment (Hamming distance).

Rem 9. *In our approach [GCR+], two coding sequences s_1 and s_2 with a low similarity score can still be two alleles of a same gene. Indeed, the necessary and sufficient condition is that a chain $g_1, g_2, \ldots, g_k$ can be found in the $80,000$ coding sequences such that all the similarity scores of (s_1, g_1), of (g_i, g_{i+1}), and of (g_k, s_2) are greater than the threshold.*

E.2.3/ INVESTIGATING THE MTBC PHYLOGENY

The ancestor reconstruction must be based on a relevant phylogenetic tree of the MTBC: as in Section E.1.2, we need to determine the closest species and their neighboring strains, to be able to produce a relevant direct ancestor of these sister species.

A first phylogenetic study can be found in [BHS+12]: the authors analyzed 435 *M.tuberculosis* complex isolates of the same clade. By focusing on the H37Rv genome, they produce 13382 SNPs. Later, they compare 44 genomes to this one regarding these SNPs (the way they extract this phylogenetic tree is not completely detailed). Their obtained tree is depicted in Figure E.6(a). Even though this work can be considered as



(a) SNPs of strict core genome (neighbor joining tree)

(b) SNPs of strict core genome (FastME)

(c) Genes content (neighbor joining tree)

(d) Genes content (FastME)

Figure E.7: New phylogenetic trees of the MTBC complex

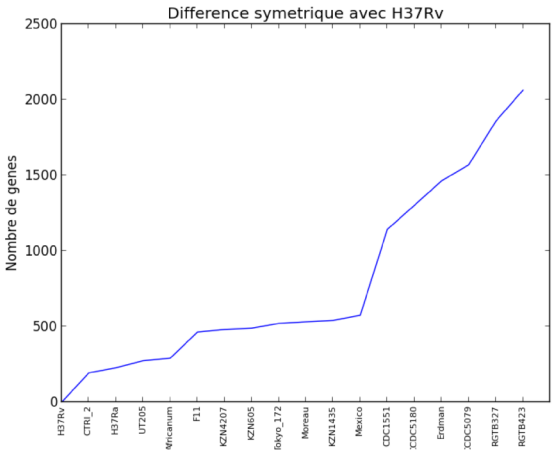
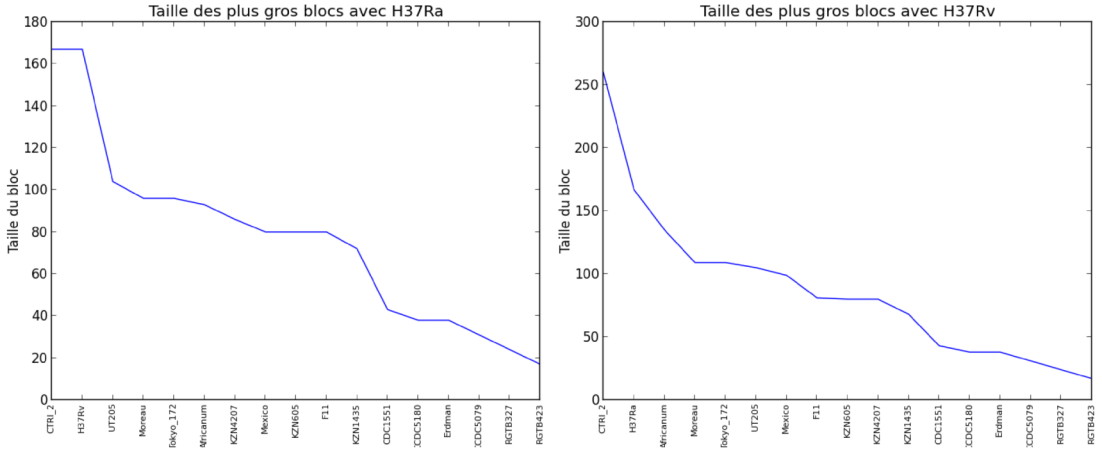


Figure E.8: Symmetric difference with H37Rv



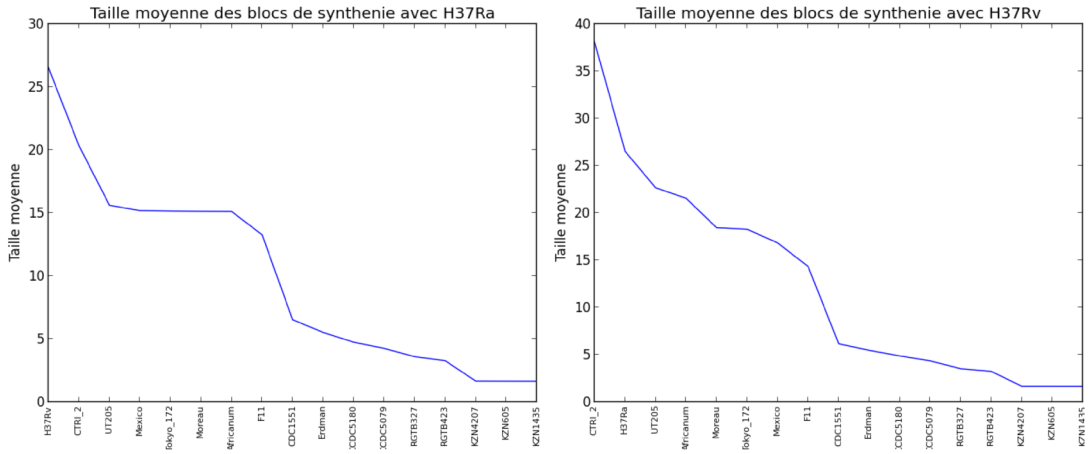
(a) H37Ra (b) H37Rv

Figure E.9: Largest synteny block

thorough and sound, it is only based on SNPs, and we believe that its conclusion should be reinforced by other approaches, focusing on gene order or content, before launching the reconstruction process.

A first tree based both on gene content and order has been obtained during our collaboration with Yu Lin and Bernard Moret (EPFL), whose research works concentrating especially on such phylogenetic reconstruction. Their produced tree is depicted in Figure E.6(b). As their result does not correspond to the tree based on SNPs, we decided in [GCR+] to further investigate the phylogenetic relationship inside MTBC using the DNA coding sequences similarity classes obtained previously.

Our obtained results are depicted in Figure E.7 for SNPs in our core genomes and for gene contents. Other results for metrics related to synteny blocs (size of the largest synteny bloc, the average size of synteny blocs, and the number of synteny blocs respectively) have been obtained too [GCR+], but for the sake of concision they are not documented in this manuscript. At each time, the trees have been obtained by launching either FastME [DG02] turn key program for phylogenetic reconstruction, or a single well-known neighbor joining tree algorithm [SN87].



(a) H37Ra (b) H37Rv

Figure E.10: Average size of synteny blocks

In almost all obtained trees, H37Ra and H37Rv appear as sister species, which is coherent regarding their history. See Figure E.11 for an annular comparison between these two genomes, based on the similarities computation of the previous section. We then have obtain the following neighboring lists: the first strains in each list are the genomes closest to H37Ra and H37Rv.

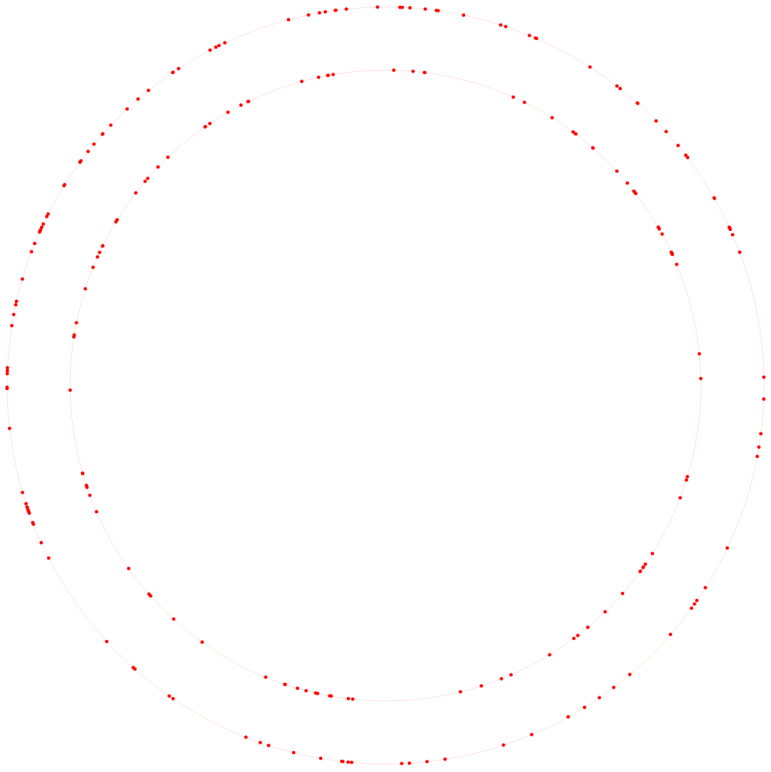
- **Genes order:** CTRI_2, UT205, Africanum, Mexico, Moreau, Tokyo_172, KZN1435, KZN4207, KZN605, F11, CDC1551, Erdman, CCDC5180, CCDC5079, RGTB327, RGTB423,

- **Genes content:** CTRI_2, UT205, Africanum, F11, KZN4207, KZN605, KZN1435, Tokyo_172, Moreau, Mexico, CDC1551, CCDC5180, Erdman, CCDC5079, RGTB327, RGTB423, see Figure E.8,

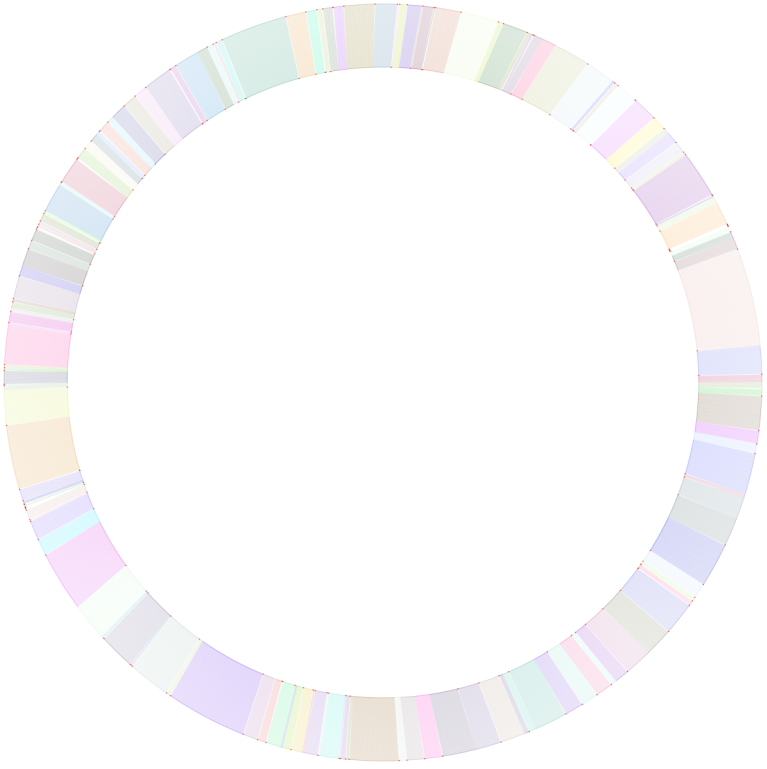
- **SNPs of the core genome:** UT205, CTRI_2, F11, KZN605, KZN4207, KZN1435, RGTB327, CDC1551, Erdman, Tokyo_172, Mexico, Moreau, Africanum, CCDC5180, RGTB423, CCDC5079,

which lead to the following cousins ranking: CTRI_2 (1), UT205 (2), F11 (14), KZN4207 (15), KZN605 (16), Africanum (16), KZN1435 (17), Tokyo_172 (21), Mexico (22), Moreau (23), CDC1551 (27), Erdman (31), RGTB327 (34), CCDC5180 (36), CCDC5079 (41), RGTB423 (44). This list is a good compromise, taken into account the size of the largest synteny block (Fig. E.9), the average size of synteny blocks (Fig. E.10), or the number of synteny blocks (Fig. E.12).

We thus have launched in [GCR$^+$] a reconstruction process of the ancestor of H37Ra and H37Rv, and with the list of cousins above. The size of each class of equivalency in the pan genome of either (H37Ra,H37Rv) or (H37Ra, H37R, CTRI2, UT205, F11, KZN4207v) comfort us, as they are always equal to 2 in the first case and mainly equal to 6 in the second case, see Figure E.13.

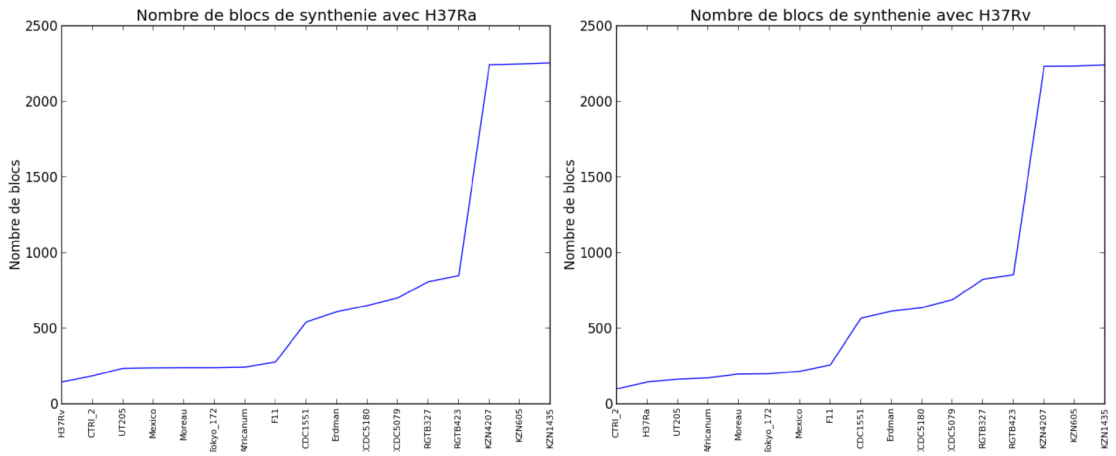


(a) Differences



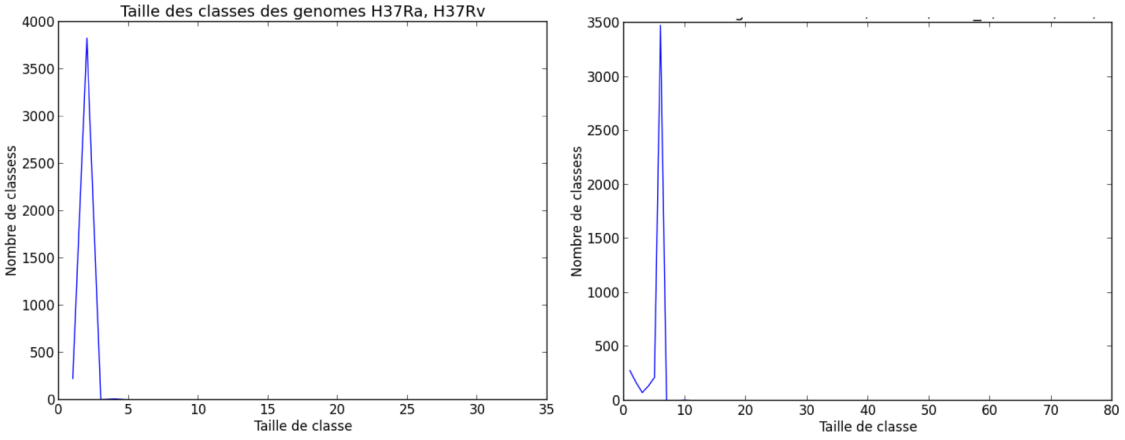
(b) Similarities

Figure E.11: Annular comparison between H36Ra and H37Rv



(a) H37Ra (b) H37Rv

Figure E.12: Number of synteny blocks



(a) Between H37Ra and H37Rv (b) H37Ra, H37Rv, CTRI2, UT205, F11, KZN4207

Figure E.13: Sizes of equivalency classes

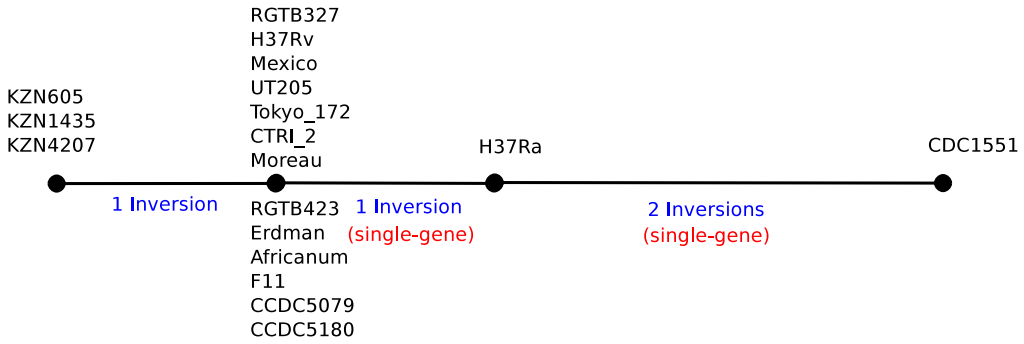
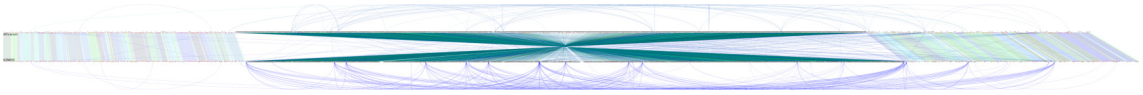
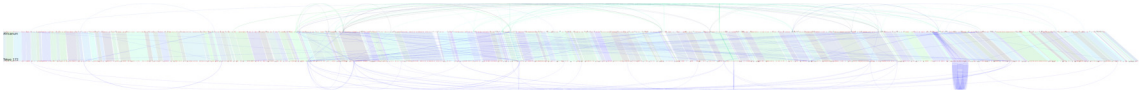


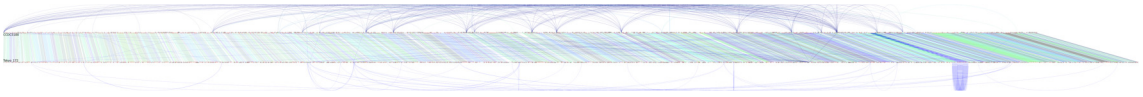
Figure E.14: Inversions found in the set of data (EPFL)



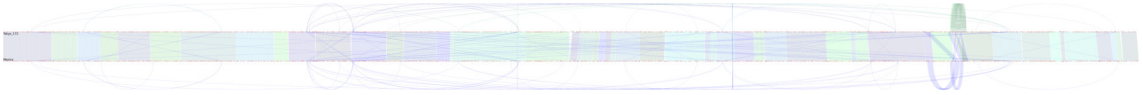
(a) Africanum vs KZN605



(b) Africanum vs Tokyo_172



(c) CCDC5180 vs Tokyo_172



(d) Tokyo_172 vs Mexico

Figure E.15: Genomes comparison

E.2.4/ Reconstruction of the ancestor of H37Ra and H37Rv

Two ways are possible for reconstructing the direct common ancestor of two bacterial genomes like H37Ra and H37Rv. The first approach consists in processing as in the *Cestodes* case (Section E.1.2): to realize a global alignment of the two genomes, and to iterate on this alignment nucleotide per nucleotide (for each position, if the nucleotides in H37Ra and H37Rv are A, then put A in the ancestor, etc.) The problem with this approach is that a mitochondrial genome of *Cestode* has approximately $16,000$ nucleotides while H37Rv (bacterial genome) has for instance $4,411,532$ base pairs. Common global alignment tools are unable to tackle such large genomes, but Mummer [KPD+04] software seems to be capable to achieve such thing. We are currently investigating this tool, regarding whether its results are sufficiently accurate to achieve an ancestor reconstruction. If so, a particular strategy must be discovered to face genomic recombination we have discovered in the MTBC genomes, depicted in Figures E.15, E.16, and E.14: duplication and inversions have obviously occurred in the MTBC [GCR+].

The second approach consists in the three following steps:

1. Work first on genes level, by determining genes content and order in the ancestor using the data provided by its children and their cousins. The method used in the *Cestodes* case can be directly adapted to integer lists (each integer being the id of the similarity class).

2. Then, for each gene, produce the ancestor following an identical approach than in the *Cestodes* case.

3. Finally, fill the hollow cavities (non coding sequences) with exactly the same method than in the second step above.

Such a reconstruction has been realized on 3 couples of genomes. However, we do not have reach the last common ancestor of the MTBC, due to an important problem found at the first stage of our reconstruction process. This problem, presented in the position paper [GCR+], is emphasized in the next section.

E.2.5/ NCBI ANNOTATIONS PROBLEM

We have computed the symmetric difference Δ between each couple of genomes G_1 and G_2 (sets of classes of equivalency), which is defined by:

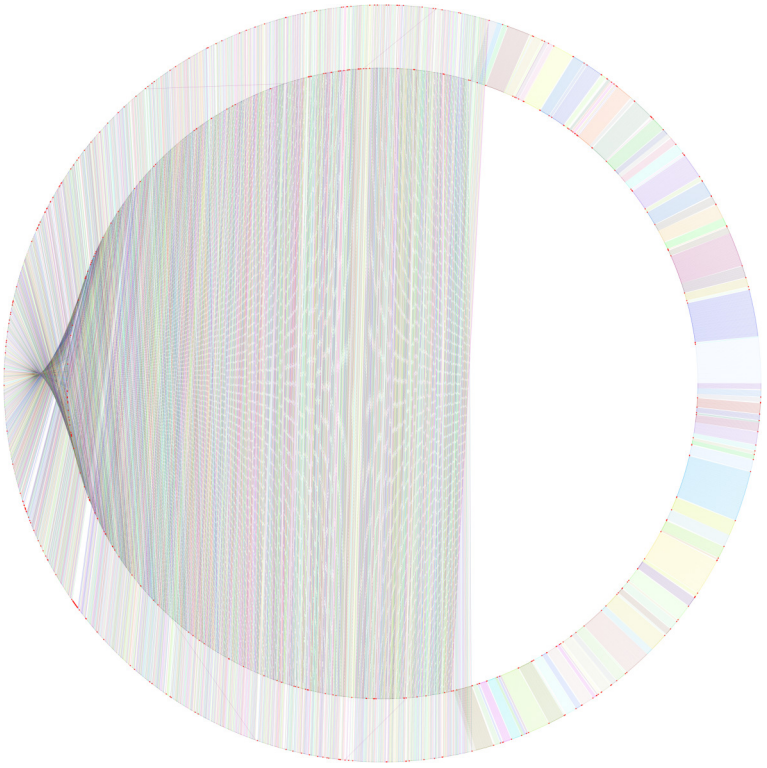
$$G_1 \Delta G2 = (G1 \cup G_2) \backslash (G1 \cap G_2) = (G1 \backslash G_2) \cup (G_2 \backslash G1).$$

The cardinality of each symmetric difference is provided in Figure E.17. Surprising symmetric differences have appeared in this complex. For instance, the symmetric difference between H37Rv and RGTB423 is equal to $2,063$, while H37Rv has approximately $4,000$ genes. This difference is surprisingly large for two strains belonging in the same species. As a comparison, the symmetric difference between H37Rv and Africanum strains, which belong in two separated species, is equal to 290. Indeed, RGTB327 and RGTB423 are both very distant from the remainder *M.tuberculosis* strains.

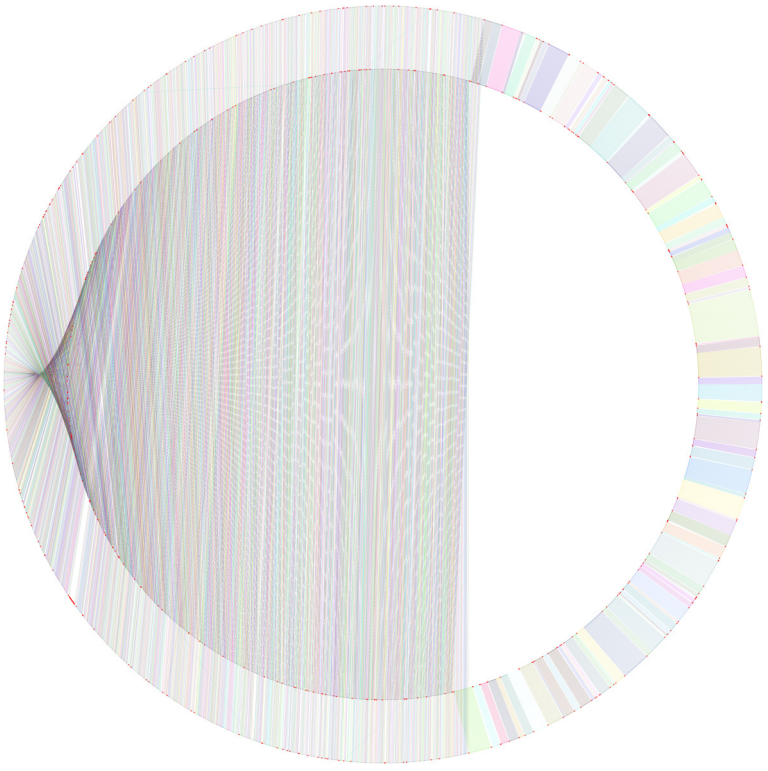
Other illustrations of this problem can be found in Figures E.12, E.10, or in E.8 for instance. Note that H37Rv can be considered as a reference of *M.tuberculosis* species, as it is in revision 3 and because it is majorly humanly annotated (the scientific community focus their efforts in improving the annotation quality of this genome). We found some *M.tuberculosis* strains in the left part of these graphs (so, these genomes are close to H37Rv, which is natural, as they belong into the same species), then strains from *M.bovis* and *M.Bovis BCG*, reasonably at a larger distance from H37Rv, as we consider different species. Finally, after an obvious break-off (see Fig. E.8), we find again 6 *M.tuberculosis* strains: CDC1551, CCDC5180, Edrman, CCDC5079, RGTB327, and RGTB423.

Other problems can be emphasized [GCR+]:

- As stated previously, core and pan genomes should have approximately the same size, due to the clonal evolution and recent history of the MTBC complex. However we have obtained $1,349$ genes in the core genome and $7,323$ genes in the pan genome.



(a) H37Ra



(b) H37Rv

Figure E.16: Annular comparison between KZN4207 and...

	Africanum	CCDC5079	CCDC5180	CDC1551	CTRI_2	Erdman	F11	H37Ra	H37Rv	KZN1435	KZN4207	KZN605	Mexico	Morean	RGTB327	RGTB423	Tokyo_172	UT205
Africanum	0	1597	1340	1244	307	1549	567	409	290	660	594	604	474	431	1894	2093	422	371
CCDC5079	1597	0	683	2091	1552	1546	1488	1624	1569	1645	1579	1609	1731	1720	2691	2804	1705	1588
CCDC5180	1340	683	0	1822	1279	1219	1215	1363	1302	1368	1302	1326	1462	1449	2516	2631	1434	1307
CDC1551	1244	2091	1822	0	1129	1817	1075	1079	1142	1106	1064	1064	1372	1383	2386	2555	1368	1209
CTRI_2	307	1552	1279	1129	0	1454	400	312	193	455	385	543	543	498	1849	2076	483	222
Erdman	1549	1546	1219	1817	1454	0	1432	1466	1463	1485	1487	1471	1653	1662	2543	2652	1653	1492
F11	567	1488	1215	1075	400	1432	0	506	463	365	283	331	751	736	1961	2176	719	518
H37Ra	409	1624	1363	1079	312	1466	506	0	227	549	493	501	617	604	1873	2060	589	384
H37Rv	290	1569	1302	1142	193	1463	463	227	0	540	480	488	574	531	1858	2063	520	273
KZN1435	660	1645	1368	1106	455	1485	365	549	540	0	192	188	812	813	1948	2189	798	599
KZN4207	594	1579	1302	1054	385	1487	283	493	480	192	0	158	766	761	1954	2179	744	525
KZN605	604	1609	1326	1064	405	1471	331	501	488	188	158	0	766	763	1928	2135	746	539
Mexico	474	1731	1462	1372	543	1653	751	617	574	812	766	766	0	95	2038	2193	104	605
Morean	431	1720	1449	1383	498	1662	736	604	531	813	761	763	95	0	2041	2192	55	560
RGTB327	1894	2691	2516	2386	1849	2543	1961	1873	1858	1948	1954	1928	2038	2041	0	2559	2028	1831
RGTB423	2093	2804	2631	2555	2076	2652	2176	2060	2063	2189	2179	2135	2193	2192	2559	0	2181	2088
Tokyo_172	422	1705	1434	1368	483	1653	719	589	520	798	744	746	104	55	2028	2181	0	543
UT205	371	1588	1307	1209	222	1492	518	384	273	599	525	539	605	560	1831	2088	543	0

Figure E.17: Symmetric differences

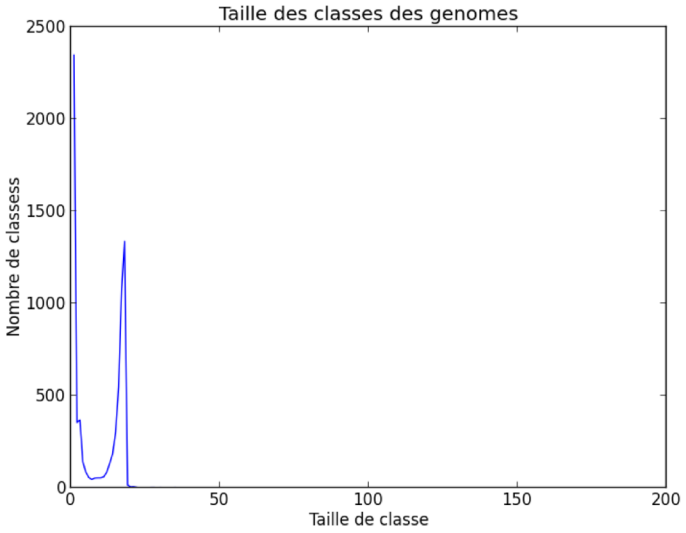


Figure E.18: Sizes of equavalency classes

- We have found $2,347$ classes of size 1, that is, $2,347$ coding sequences are distant of at least 10% of each of the other $77,653$ coding sequences. A sequence so distant to all the other ones can be explained by an horizontal gene transfer from a different bacterial species. But supposing that $2,347$ such transfers have occurred these last $40,000$ years in a clonal complex (in which an horizontal transfer is quite impossible) is not realistic.
- Similarly, we have 354 classes of size 2, 368 classes of size 3, and 140 classes of size 4. The scarcity of such coding sequences in the collection of genomes can only be explained by horizontal transfers, which are impossible in this clonal complex.
- Finally, we have found 112 classes of transposable elements (DNA transposases) in the man genome, and 45 classes in the core genome. The largest class has 189 representatives. As shown in Table E.9, their spreading seems correlated to their human virulence, while they seem to be at the origin of the inversion in the KZN strains (related to an epidemic of tuberculosis in India). However, some strains in this table seem to have a number of transposases incompatible with such statements, and indeed surprisingly low (like CDC1551, Erdman, or UT205).

Our explanation of this situation is that either the annotation is wrong, or the DNA sequence is problematic (assembling errors, for instance). Such a statement invalidates our work achieved until now. It necessitates to:

1. Use relevant tools to evaluate the quality of the complete DNA sequence.
2. Use exactly the same (good) annotation tool on all the unannotated genomes, like GeneMaskS which achieves to rediscover 2/3 of the coding sequences of H37Rv, see Table E.10.
3. Start over the whole reconstruction process.

Gene prediction software	Good ORFs
Glimmer	2558
Genemask	2768
Rast	2560

Table E.10: Gene prediction scores of the best GPS on H37Rv

E.3/ OTHER PROJECTS

E.3.1/ *Escherichia coli*

The epidemiology of ESBL-producing *Escherichia coli* (ESBLEC) is complex. It combines their spread into the community setting, nosocomial acquisition, and the horizontal transfer of plasmid carrying bla_{ESBL} genes. The determinants of the spread of these strains in the community remain poorly known, particularly the role of environmental dissemination. The objective of [BPS$^+$] was to (i) to quantify the ESBLEC throughout the wastewater (WW) network of the city of Besançon, eastern France, (ii) to compare the ESBLEC loads between hospital and community WW, (iii) to assess the clonal diversity of ESBLEC and the bla_{ESBL} diversity throughout the WW network. Our work in this project consisted in genotyping by multi-locus sequence typing (MLST), and in analysing the MLST data.

MLST was performed according to the protocol described in the *E. coli* MLST website http://mlst.ucc.ie. Nucleotide sequences were determined for internal fragments of the adk, fumC, gyrB, icd, mdh, purA, and recA genes, on both strands, and were compared with sequences in the above-mentioned E. coli MLST website for the assignment of allele numbers and sequence types (ST). Clonal complexes (CC) are defined as a group of STs sharing at least 5 loci, using the START2 software. All chromatograms were imported, assembled, edited, and trimmed in Geneious Pro (5.3.6, Biomatters Ltd, Auckland, New Zealand). We concatenated the sequences of adk, fumC, gyrB, icd, mdh, purA, and recA genes to form a 3,423-bp sequences alignment. The best-fit nucleotide substitution model for this data was GTR+G+I, as determined with jModelTest 0.1.1. We used the Shigella dysenteriae Sd197 as the outgroup strain. Maximum likelihood tree was constructed with RaxML 7.2.8 [SL05] and visualized with dendroscope. In every case, 1000 bootstrap repetitions gave values above 900 for most branches.

E.3.2/ *Pseudomonas*, CHLOROPLASTS, ETC.

We have achieved the same work than in the MTBC, but with 98 chloroplastic genomes. All the complete genomes have been annotated with Dogma, core and pan genomes have been obtained, and the results have been compared with those obtained with annotated genomes from the NCBI website. Results are satisfactory in the first situation, they are currently studied by a phylogenetician, while we have started a phylogenetic study of this collection of plants.

MLST data have been used too for *Pseudomonas aeruginosa*. The project consists in studying the genotypic and phenotypic evolution of an epidemic strain during 13 years of its hospital dissemination. Establishing core and pan genomes of this species is under study too, with an approach similar to the one applied on the *Mycobacterium tuberculosis* complex.

Other projects currently going on encompasses the study of the dissemination of transposases in the MTBC, to the evaluation of the danger related to the inhalation of mycotoxines produced by *Stachybotrys chartarum*, and to the characterization of heavy metals poilluted lands using transporter genes in fungi.

F

LAST COMMON ANCESTOR OF *T.asiatica*, *T.saginata*, *T.multiceps*, *T.madoquae*, AND *T.serialis*

ATGCTGTTAGTATAAG-TTATTATRTTTCTTTTCCAAAGAAAAGATCTA-TTTTTGTAGACNGTATA--TA
AATGTCTATTGTTCCTRTTTTTAATGCTTCTTTTGTTGGTTTTTTTTTTRGTTGGTTTATTTTTATGAAAGG
TTAAATTNTTTTTGATATTTTTAATNTGTGCTATTTTATCAATAGTRATATTTGTATTTGATGGGTTNGGT
AAGRYTNTTCATTATGAATCTGCNTTTTGGTTGTTTGTTTTTAGNGANGTTATGATTTTTGGTAGATTTTT
AACATGTTGTTTATTTTTTGATTCTTGATCTTATGAAAAATTGTCTAGATCTTTNGAGATACCTTTTGTTG
GGTGTTTTGTTTTATTAGGTTCTAGTATTACTGTTACTGCGTTTCATCATTTATTAGGTTGNAAGTATTGT
GATTTTTTTTTGTTTTTGACTGTNRTTTTAGGGTTNAGTTTTGTNGTTTTGCAGATTTCTGARATGGAAGA
TATAAGTGTTAATATATTTGATACNAGATTTCATGCTAGAAGATTTTGTACTGTTGGNTTACATTTTAGTC
ATGTTTTATTAGGTGTNGTTGGATTNAGTACNATTTTATTRGTWGGNAGNAGTANGTTTGGNGTTTATCGT
TGTACTGTTTTNACATGRTATTGRCATTTTGTNGATTATATATGRTTACTTGTTTATACTATAGTNTATGT
NTGTTART??T???-----TTACTAATAGGTTATGTAAACTARTAGATTGTGGTTCTATTGAATACTTGTT
AATTTNN?GT-AGTATTAGTAAATT-NATGGTTAGGTTATTTCGACGTAATTTGATAGATTTGCCAATTAG
TTATTCTCTTAATTATTATTGAAGTAGGGGATTTGTNCTTTCTGNTTTTATGNTTATNCAAATATTAACTG
GTGTAGTNTTGTCTTTTTTATATGTTGCNGATTATTTGTGTAGTTTTTTNATGGTTATGAGNTTRTCAAAA
GATTCTTTTTTTACTTGATGTTTGCGGTATTGGCATATGATAGGTGTTAAAGTGTTGTTTGGTTTACTTTT
TGTTCATATGGCTCGTGCTTTRTATTATTCNAGTTATAAAAAGAAGGGTGTATGAAATGTAGGGTTTGTTT
TRTATTTATTNGTTATGGGTGAGGCTTTTACTGGATATATATTNCCTTGNCATCAAATGTCTTATTGRGCT
GCTACTGTTTTAACATCTATAGTNGATAGATTGCCTATTTTTGGTAGTGTTGTTTATAAGTATGTNGTTGG
TGGATTTTCTGTNTCAGGTATAACTTTAATTCGTGTGTTRTCTGTGCATATTTGTTTNGGTTTTGTTATTT
TNGGNTTNATGRTTNTTCATATGTTTTATTTACATAAGAGTGGTAGNAGTAAACCTTTATTTTCNTTTAAT
TATTTNAGNGATGTAATTTATTTTCATTCTTATTTTACGGTTAAGGATTTTGTNTTGTTTATGATAGTTGN
TAGGTTTGTAGTTTTTTGATTATTTRTAAGACCTGATGCTTTAGTTGATATAGAGGCRTATTTAGAGGCTG
ATCCNTTGAATACTCCTGTTTCAATTAAGCCTGARTGATATTTTTTAKCATTTTATGCTATTTTACGTTGT
ATAGGNTCTAAGATTGGTGGTTTNGTGTTNATTNTAGCGTTTTTGTTTTTTTTTGTGAGTNCCTACTAATAG
TGGTTCRAGTGTNTATAAWGTATGRCGTCANGTTAAATTTTGNTTNATTGTAAGTTTNTTTTTTTCTTTAA
TTTATTTNGGTGGTTGTCATCCTGAGTATCCTTATCTTTTTNTATGTCAGTTATTTAGYATTAGTATGGTT
ATNCTTATGTTTNTTTTTAAGATTTATTAATTNGTTTTNTAANTNA??--------ATGGTTAGTTTATTT
TTAATTTTTTGTTCTGTAATWGGNGTTAGTTTTTTTTTATCTATTACTCGATTTTTGAATAGTTTAATANT
NTTGGAAAATTTTAATGTTTTAATTTTRATGTTTTGTTTGATTTTTTCTTCTTTTGATAGTCATATGATTT
TTATGGCATTAATGGTTATTTCTACTGTGGAGATAATTGTRGGTTTAGTTGTNTTRACACGNGTTTGNGAR
TGTTCTTCTTCATTAGANTTAATAGRTTTTTAATAGTTTTANTNTGTATATTTNTATCNTTATTATTTAGT
TGTGGKGTTAAATGTATGAGTATRTTTAGTATGAAAGTTTTTAANGGAATGTTTATATTTGATTCTATTAG

TTTTTATTTAATTATTTTRGTGTTRTTATTAGGTTTATATAGACAGATTATGTTTTATANTATGTTNAGTR
ATGTTGTTCGTGTNTATTTATTTTTTAGATTRGGTTTTACTGTNTTNAGNTTTTGTGTNAATCATTGTGTN
GTTTTTTGATGTGTNTATGAGTTRTCNATGTTRCCTTTACTTTAYTTRATTTTTAGAGANTCTCCTTATTC
TGAGCGATTTTTGGCNGGTTGATATTTTTGTGGTTATCTTTTAAGTACTAGATTNCCATTAATTTTRATTT
TATTATATTTGTCTTTANTTAAAGATTCTTTTTTNTTTAGTGAGTGGARTTNTAATGGTNGTGTTTCTTTA
NTTGTTTATTATTTGTTGTCTTTTATTTTTTTTACTAAGGTGCCGTTNGTTCCTTTTCATACGTGATTGCC
TATAGTACATGCTGAAGCTATAAGTATAGTTTCAATATTTTTAAGNGGTTATATAATGAAGCTTGGATTAT
TAGGNGTTTATCGTTGTGCTAGTTTTATTTTTAGTGGNRGNTTTTTGTGATATTTRTTTTTATGNTGTATT
NTATCTATATTTTTTTTAATTACRTCATGTAGTGARTTAGATGGNAAGCGTTGATTAGCATTTTTAAGTTT
AGCTCATATAGTRGTTCCNTTTTTAGGNTTTTATATTAGWGATTGRTCTAATATAAAATTTTCTTTTTTTT
ATTGTTTAGGTCATGGNTTAAGTGCTGGTATAGTGTTTGGNTTATTGTGNTGTTTTTATGATGTTTCGCAT
ACTCGTAAATGAATTTTATTGAAGTCAAGTATTAAAGGTGTARGTTTAATGAGAATTGTGGTTTTTAGNTT
ATTAAGATTGTGTTCGTTTCCTACTACTATTCANTTTTTTTGTGAGGTNGGGTTAGTTAGACANAGGTTTG
GATTTTTNATNTATTTATTATTTTGATGTTTTTATTTATTTTTTGGTGGTCTTGTGCCGTTNATATTATGT
GGNCATTTRTTAATTCGTAGRGAGTGTTATGAATCTGTTGNTGCTTGTTATTATTCTCATTTTTATTTTTT
GNTTTTTCTTTGTTTGTGATGTTATTTTGGNATATTAGTTTTATAA?GTNN--TRTTTAATGTGGTGTGTG
N-TATGCATTTTACATTTTGGTTGTAAAGGTGATTAGTAATCCGTTAAATTTCTCTTAGCTTAAG?TTTAA
AGCGTCAATTTGAAGCGTTGGAGATAATNTAATTAGAGAGACTGGTAAGTTAANTTAAACTGTGGGGTTCA
TGTCTCCATTATACAC??TTTT----GTGTCTGGTTGARTTTA?--NTATGTTTNRTAATGTTAATGATTT
TAGTTCTTTAATGATTTTNATAAATAAGTTTGTTTTAGGTGATGTTGTTTATTATTATTTTAGTGTTTTAG
GTTGGGTTTTRTTTTTGTTTTTGTGTTATCGTTTNCCTTATTGTTATAGTCCTTTTTTATTTAGTGTNNTT
TTANTTAGTGTNGTNTTTAGNTGTTTTGTNTCTTTTTTTTTTAAGTCGTATTTGTGATAAARTRAATTTATT
TTTTAGTTCNTTTATTCCTGTNGGTACTCCTATTTATATATGTCCATTAGTGTGTGTTGCTGAGTTAATAA
GTTATATTATTCGTCCNGTAGTATTGATTTTACGTCCTTTTATNAATATAAGTTTGGGTTGTTTTGGAGCN
GTTGCATTAGGTAAATTAAGNTTAATTAGNNGTTGRTGATGTATAGTGNTATTTTTTTTATTTTTTTATGA
AGTTTTTGTTGCTTTAGTTCATTGATTTATTGTGACTAGAATTTTNGCGTTTTCAGTTGATCATTAATART
GRNTATTGTTCGTTKGGGTTATTTAGATGTNGTTNTTTTTTCTTTAATTTTTTCTNTATTTTTTTGTTTTT
TGTGTTGTGTNGTTGATAGTTTATTAGGRTTTTGAGTTTTTTTAGANTTGTGTGGATTNTCNATTATACCT
TCATTNTTTTTTAATGTTAGRTCTATGTCTTATAAATTTTATAATTCTATTCTTTGTTATGTRATAATGTC
TGGATTRTCTTCTGTATTATTAGTTTCTGGGTTATTANTRRTNRGRTTATATTATTTTGTTTATTTTGGGT
TTGTRGTTAAGTTTGGATTATTTCCTTTTATGTTTTGGGTTTATCGAGTTTTTAGTATTGGTAAATGAGTN
TTTATATATTTATTAAGTGTTGTAATGAAGTTTCCNGTTATATTTTTTTGTTTTTTNTATGAGACNAAWAA
TTTAAAGATTGTATATATTGATTGTTTTTTTACTATTTTTTTNTGTTGTTTTTTNGTNTGGTTTTTTAGTT
TTAGTTGGGANTATATTTGGTGTCATATTTCNTTGACTTCTGTTGCTACATTAGTTGTGGCNTGTTTTTGT
AGNAGNATTGAAGTGTGNTTTTTTATTTATTGTTATTATTTTATTTGGGCTAGNTTRANNATAGTTTATTT
TGTNGTNGTATCTAGTAGTANAGATTTNAAGRGTTATNTATTTTGGGTGTTTTGTTTTTTNTTATTNGTTA
CTCCTGTNTCTTTTCCTTTATTTTATAAGTTGAGTGTTAGTNTTGGTATTTTATATTCNTCTATTTATTTA
TTGTTNGTATGAAGAATATATAGATTTTCTGAACAGYTNTTTCTTTATAAGTTGGCTAGTGAATATTTTTA
TGTTAATGTTTTTAAAAATTGAGTATAANNNTT-----?????TATATATGTGATGTAGTTTATTNAAAAT
ACTTGTTTTACACTCAAGAGAACTCTTTATGTGGAGCTTTACTANNNTTNNTTT?TTTTATATA-AACAAA
ATAGTTTAAT-TAAAAATATTGGGTTTGCGTCTCGAAGATGGAT?-TWNNTTCTTTTGTTAT??????????
????--TGTGTGGCTTTAGTTTAATNAAAATRGTGATTTGTCTAGTCATAGATGGTAGTTTAGTAACCAAG
TCRCT?GTT-----------TATGATTATTTTTGGGT--TTATTAGTGGTTTAACAGGTTTGTTAGTTAGT
TTATTNATTATAGCTTTTTTTATTTTAGGNGAGCGTAAGATTTTGGGTTATTCTCAGTTTCGTAAGGGTCC
TAAAAAGGTTGGTATTATTGGGTTGCTTCARAGATTTTCTGATTTGTTAAAGTTNATAGTTAAGTTTAAGA
ATTATGGTTTTCAAAGTCGTAGATGAGTTGGTTTATTTGGNGTTATATTNTTNGTTGGTTTAGTTATTTAT
TATTCNTTTGTNTATGGTGGTTATTATAGRTATAGTTTTAATTCNCTTTCNTTRTTATGNTTTTTGGTTAT
NACYAGNTTTTGTAGTTATTCTATATTRTGTACAGGTTGGGGTAGTTATAAAAGTTATTCGTTTTTAAGTT
CAATTCGTTGTGCTTTTGGNTCTATAAGGTTTGAGGCTTGTTTTATGTGTATTRTTATATTTTCTGCATTG

TGTTATTGTAGNTATAGTTTGGTGGATTTTTTTTTAAGTGGTTGGTTRTCNGTTRTGATTTTTCCTTGTNT
GTTAATTTTNTATATAATATGTATTTTATGTGAAACTAATCGTACTCCNTTTGATTATGGTGAGGCTGAAA
GNGAGTTAGTTAGTGGTTTTAAAGTTGAGTATAGTGGTATATATTTTACATGTTTNTTTGCATGTGAGTAT
ATAATTATATTTATTTTTTCATGNTTAGGNATGATTTTNATGTTTGGTGGTGGGTTTATTGGTAGTATATT
TTTATTNTTTTGTTTGNTTTTTTTTATGTGGGCTCGAGCTACATTACCNCGTGTTCGTTATGATTATTTTG
TNAAWTTTTTTTGNGANGTATGTTTATGTTTNTTNATTTTNAGNTTTTTTGTTATNGTTAATT-----GTC
TATATAGATTA-TTTTAAATCGTGATGCTGTTAACTTCAGGAAATGGTTNNRTACCATTATAGTCG-TGTN
TG---TTNTGNTN----?TNTCTAATCTTAGTTTAATTTTAGAATNNAGRTTTTGGGGATCTTTGGTCTCA
A-TTTGAGAGATTTGRNGTTAATAGGGCTGCATTAGCAGGTCACTTTGATATAGTGAATTGTGAATTTG--
ATATTCCGTTAATATTA---NGT???T?-TTTTATCTATGTATTCTAA?NTNNAAGAGGCTGAATTCTTAC
TTCAGTAATGTGAG-TTTCACTATAGGTTAATGATTTTATTNNTTTTTAGTAGTATTTTNTTTTTTGGGTT
NTGTTTTNTTATTTATTTTTTTTGTTCTGGTTTRTTNAAAAAGATTGTGGATGTTGGTTTTGGGTGAGGTA
GNTCTTATGAGTGTGGNTTTTTTTCTAGTGTNTTNAATTTGAATTGTTTTAGTTTTACTTATTTTTTTTTG
TTGATTATGTTTGTNATATTTGATCTTGANATTTCTTTACTTTTRAATATGCCTANNCANGGNTTRTTATT
TTNTAAATTTNGTTATTATTATNTTTTTTTAGTTTTNTTGTTGGTTGGTTTTGTGTTTGAGTTATTTAGTG
GTTATGTNCGTTGATTNTATTANT---TTAGAGGAAATTGTGAAGTTACTGCTAATAATTTCGTGTCAATT
TGGTTTGACTTTCTCTTTNT????T???-----RATAAGATTAAGTTAGGTTAGACTAAATGTTTTCAAAA
CATTAAGTGACT--TTAT-TTAGGTCATCTTATGT-AAATGAGTGTTAAATNTTTATTAAGTTGAATATTT
ACTTTAGATCATAAGCGGGTTGGTGTRATTTATACTTTATTRGGTTTNTGNTCAGGTTTTGTAGGTTTAAG
NTTTAGTTTATTAATTCGTGTTAATTTTTTAGAGCCTTATTATAATGTGATTTCTTTGGATTGTTATAAAT
TTTTGRTTACNAATCATGGNATAATAATGATTTTCTTTTTTTTAATGCCTATTTTAATAGGTGGTTTTGGT
AAATATTTNATTCCTTTRGTTGGTGGGTTATCTGATTTNAATTTNCCTCGTTTAAATGCTTTAAGTGCGTG
GTTRTTGRTTCCTTCNATAGCTTTTCTTTTAGTTAGTATGTGTTTAGGTGCTGGTATAGGGTGAACTTTTT
ATCCNCCTTTGTCGTCATCATTATTTTCNAGTAGTAATGGTGTRGATTTTTTAATGTTTTCNTTRCATTTN
GCNGGTGCGTCTAGAATTTTTAGTTCTATTAATTTTATNTGTACTTTNTATAGAATTTTTATGACTAATAT
ATTTTCTCGTACTTCTATANTATTRTGGGCTTATTTATTTACGTCTATTTTATTATTAGTTACTCTTCCTG
TNTTAGCAGCTGCTATNACTATGCTTTTATTTGATCGTAAATTTAGTTCTGCRTTTTTTGATCCRTTAGGT
GGTGGTGATCCTGTTTTATTTCAACATATGTTTTGATTTTTTGGTCATCCNGARGTTTATGTTTTAATTCT
TCCTGGTTTTGGTATAATTAGTCATATATGTTTAAGAATAAGTATGTGTCCAGATGCTTTTGGTTTTTATG
GTTTGTTATTTGCTATGTTTTCAATAGTGTGTTTNGGAAGAAGTGTGTGNGGTCATCATATGTTTACNGTT
GGGTTAGATGTTAAGACTGCTGTATTTTTTAGTTCGGTTACTATGATAATAGGAGTACCNACAGGAATAAA
GGTTTTTACTTGNCTTTATATGCTTTTAAATTCTCGTGTNAATAAGAGTGATCCTATATTNTGNTGNATAG
TTTCTTTTATAGTATTGTTTACTTTTGGTGGTGTRACTGGTATTGTNTTGTCTGCTTGTGTATTGGATAAA
GTTTTNCATGATACTTGATTTGTTGTTGCTCATTTTCATTATGTTATGTCGTTAGGGTCTTATATAAGAAT
AATAATTATGTTTATTTGATGGTGGCCTTTRATTACTGGTTTGAGNTTAAATAAGTGTTTACTTCAATGTC
AATGTATAATATCTAAAATTGGATTTAATTTATGTTTTTTTCCTATGCATTATTTTGGGTTNTGTGGATTA
CCNCGTCGTGTTTGTATTTATGAGTGTGCTTATAATTGRATTAAAATTGTGTGTACTGTRGGTTCTTTTAT
TTCTGCTTTTAGTGGGTGTTTTTTTGTTTTTATACTTTGAGAGTCGATNGTTAATCGTAATGAGGTTTTAG
GTTCNTATGGTTCNTCTAGTTGTTATGTGGATTTTTTTATGAGTCCTGTNGCTTCTCATAATGATTATTTT
TGTTATCCGTATARTATAGATTATACTTATGGTGTNTATTATATGCGTTGGGTNGATGATTGTACNTATGT
GTTTGCTCGT???--GGTTTTTTAGTTTAATTTAAAATNTAGATTTTGTAAATCTATGATGGTTTNT--AT
CATTAACCTTT?-TGATTGATAAATTG-?TNATGNTTTAGTG----TTTTAGGTTTATTTGCCTTTTGCAT
CATGCTTARTGGATTTTTATAAAGTATTTTTNNAATCGAATAGTTTAGATCTTAATANCAGTTGTTTAGAA
TAT?ATTTTGTATTANGTAAATTCAG-ATAAATTGGTTTTATGATGTGATAAGTT-ATTCGTTAAATTTTA
TTTCTATTTAGTTGCTTAANTATTTTNTRTTATATGATANGTTAAAAGATTTRTCATAATATTATACTATA
NATNNAATTTTATTAAGATTAGGTTAGAGGTACCTATTTTTTGTAANTTTGTTATAAAAGATNNT?NGTTN
GNTTT--ATAGTTANNTTGNGTGGCAACTTNN-GTTAATATRTAGTGTGTTATTTATCATTAAATAAGTAA
TTAAATTATACTAATTATTTCTCAGGGTCTTTCCGTCTGTTTATTAAAAACATTTCTAGTTTGAATNTTAA
CTAGTAGTGCCTGCCCAGTGTTGNTTTNNAA--AATAAATGGCCGCAGTATMTTGACTGTGCAAAGGTAGC

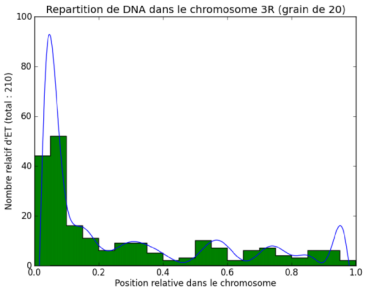
```
ATAATTAATTGCCTTTTAATTGGGGGCTTGTTTGAATGGTTTGACGTGAGAGATGTGAATATTTGGTATTA
NTATGAATTTGATGTTGGGAATACAGAATTTCCANA-GTTTAATAAGACGGAAAGACCCTGGGATCTTT-T
TTTTTTGCTTGGGGCAAGTTTTAGTGTTTNACTATAATTNNGACCCTAGNTAT---AGTTTAGTGGGTTAA
GTTACCCCAGGGATAACAGTGTGATAATTAATAAGAGATCTTATCGAATTAATTGTTTGCCACCTCGATGT
TGACTTAGATTAAAACTTRGGTGCAGTAGTCTARGTTTTTGGTCTGTTCGACCTTATTTATCTTCATGAGT
TGAGTTAAGACCGGCGTGAGCCAGGTCGGTTCTTATCTATTGTAGAAGTTTATCAGTACGAAAGGATAGTA
AGCTTTTTTNTTGAACATGAATTGTGTTAGCTTTGCAAAAGCTAAAGAGAATTA?TTTARTAATTCCATGT
TTT?TATTAATTGTTTAACTGTGGCAAAAGAAAGTTTGTGTTTCATTAACTGCATA-AAATTAGTTTATTT
TTNG?TATGATTTAGTGTTATTTATTTTATTTAGCAGTTAATTTTTTGTTAAAANTA-AAGTTTTANAAGG
GTGANACATTA--RAAGGGGATAGGACACAGTGCCAGCATCTGCGGTTAATCTGTTTTCTTTGCTTTTNTR
TAGATTGTGTGTGTTTATTTATMTTTAAAAATAGGTTAAATTTTTTTATTTAAGTTTTAAATATTCATTTT
TATATAAAATTTATGTTTGTGACAGGGATTAGATACCCCATTAACGTATTTTGTANTATTATCTTAGTTTA
G-TAACTAAAATGGTTTGGCAGTGAGTGATTCTTTTTAGGGGAAGGTGTGGTGTAAAGGATGTTCCGCCTA
TTAATTTACTTTTATTATGTTGGTGTATATCTGGTTTAATATTATTGTTGAATNATATAAGTTTGTGTAGT
TT-NTAGTTAAGCCAAGTCTATGTGCTGCTTATAAAAGTATTCATGCGTTACTTTNATAAAGTTTTAGTTG
TAANTRCTAT--TATATTCAGGACTTAAAAGTAATGTTAAATTAGTTTGTTAATGTGAARTAAGTTTAGCT
CATGTACACACCGCCCGTCACCCTCGATTTTTATTGAGGTAAGTCGTAACAAGGTAACTTTAAATGAATTT
GAAGTTGGTTACTRNGNNNA-NTTNCTYAGTTTTAATGAAACTATCTTTATTNTATTATGATATAGTTTGT
TATATAATTGCTGTGTGTGTATTTATTTTGTGTTTTGTRTATATAATGTTATGTTGGAAANTNTTT?TAGG
TGGTGGTAGAGTTAAATTTGGTGGTGANAAACAGGTTGTTGAATTAACTTGNACTATAGTTCCTACTATGG
TTGTTTTGGTTTTATGTGCATTGAAAGTGAAATTTATAACTAGTGATTTAGATTGTTATTCTAGTGAGACT
ATTAAGATTATAGGACATCAATGNTATTGAACTTATGAGTATYCARAAGGTAGNTATGATTCTTTTNTRAC
TAAGGATTGTTTTTTNGTRGATAAGCCTATGCGTATGATTTATGGTACTCCTTATCATTTAGTNGTGACAT
CAGCTGATGTNATTCATTCGTTTTCTGTNCCATCTTTRAATTTAAAGATGGATGCTATACCKGGTCGGTTA
AATCATTTATTTTTTTGTCCTTCTCAACATGGAGCTTTTATNGGATATTGTGCTGANTTGTGTGGTGTTAA
TCATGGTNTAATGCCTATAATGATTGARGTTGTTGATGTT?T???AT?????T?TNATAGNTTTTGTTTTA
GTATAATNTATTATTTTAGCTTTTCGTGYTAAGGATAGTTTGTNTATGACTAAACAAAAG-------TGGT
ATTAGAGATTTTAGTTACTTYTTATTTTTTTGTGTTGTTNTTATTTTCGTTAAGGAGTCATTGTGTWTATT
ATTGTGTNTTNTTAGTTNTTAAAGCTTTGATTTCNTGTTTAATTTGTTATTTRGTNTATGGTTTTAGGTGA
TATTCATTNGTTTTTTGTTTGGTATATGTNGGTGGAGTTTATATATTNTTTATTTTTGTGTCNGTTTTTAA
ACCTAATGATAGGTTTGCTATTTATCATAAGGTTGGTGAGTCTAATGTTGTTTTATGTTTTGTANTAGGNT
TATTGTGTATGTGTTTATTTTATNGNTTGGTAAAAATTGAGTTTAGTANATTTTTGTGTACNGTNGTNGAA
GGTAGATTTTATGTNTGTTTNTGTTTAACTTTGATATTTGGNTTTGTNGTTTTNAGNTTGTTGGTTAGTTG
RAAGATGAAATTTTATCGTTAGATTTTATNT-???TNNNTCTAGTTTARC--ATATTTTAATGTGRNGGGT
TGTAAACTCTTTRAAGGCTATGGCCAACTAGGAGAANATACCG--------?AAAAAAAAATTTTTTTTTT
ACGGTATNTTCTCCTANTGTATTGAAAAATTTTAAAAT---GTAGAGATGCCAGAAAANTT?NATGGGGTT
AATTTAGGNTTAATTTATGACTTTTANAGTCTCTCTAT------------TAATTTTTCAATACANTAGGA
GAANATACCGTATTTTANANTTATNNGACTTATG????ATATTTGATTTGAAATCAAGTTAA-TTGCTTTT
GNTAGCAACATARGTTNNTT-????TATATANT--ANAGTAGCTATGTCAGAATTTATATGAGTTAGTTTT
AAGCATTAATTATGGAAGTTTTCCTGGCTACTTTATNT???TATN?NTGTTNNTTAGCATATA-TTTGACT
TATGTTACGGCCATAAGATAGGTATTNTATATACCGTATGTTTTATGTTTANTTCTTTGTTTNGTNTTAGT
TTATGTATGTGTTTGTTNATATGATTTTGTTTTTTGTGNTGTNTTAGTTTNTGTGTTAAGTTTAAATTTTT
NTCATTAGGTGGNTATAAATGGGTGATAAAATTTGATTTTGATTTTGTTACTTTTGGTGTGGTTTTAATGT
TRNTTATATGTTTTTTNTTATGTATATTTTTATACACATCATTATTTTRTAGGTGATTTNGCTGGTTTTGAG
TTGAATAAGATTATARTTTTATTTGTNGTTGTTATGGGTRTTTTAGTATCTACTGGTGATTTTTTGACNAC
TTTNATATTTTGAGAATATTTAGGTGTNGTNAGATTTTTTTTRATTTTGTTTTATGATAAATTTTTGAGAT
TACGATCRTCNGTAGTTACTTTAGTTTCTTCTCGATTTGGTGATGTATGTTTGTTTTTTTTAATAGGNTTA
AGTTGTTTTATNTATAGTAAATATTTRTTTTGTATNGTTATATTTTTTTTGATAATATTTACTAAGAGTGC
TAGATTTCCTTTTATAAGTTGGTTATTGGANGCTATGCGTGCTCCNACNCCTGTTAGTTCATTAGTTCATT
```

CGTCTACTTTAGTTGCNGCTGGTATTTGGTTTGCTATGCGTTATGATTATNTNCAATTTTTTARARAATCA
ATTTNTTTTAGNGTATTGTTNATTTTRACTATTTTAATTACGGCTTTAAGAAGNTTATTTTTTNTAGATTT
NAAGAAGATTATAGCTTTATCTACATGTAAAAATATTGCRTGNTGTRTTTTATANTTGATATATGGTGATT
TAGNTCTTTCATTATTTCAATTATTAAGACATGGNGTATCTAAGTGTATNTTATTTATGTTGATAGGTGAT
GTAATGAGNGGTAGTGGTGGTTCTCANGGTAGAAATTGTGTTTATAGTACTAATTTATATGGTAAATGAAA
ATTATTTAGNTCAATTTTAGTARTTCTTGGTTTAGCAGGAGTWCCTTTTATAGGTGTNTTTTTTACTAAGC
ATTTTTTATTNTCAATGTTTGTTAAANTTGTTAANNTRGTTGTTTGTTTGATAATTTGTTTATGTATGTTT
ATGTCTTATTTATATTCTTTTCGTTTATGTGCTATTTTATTTAAARTTAAGAGTAGAATTAGATTNGGNGT
TTTNTTTTTTTTTAAATCTGGTTTGATGGTRTTTTTTTGGTTATTTATNAAATTTTATGTTTTTTTTATRT
TAGATGAAAYTGTATATCTTAAYAGATTTATNAGTTTTANTTTAATTGTNTTTCAATTRTTGTCAATT?TA
ATNAYATA-TATGTTTTATGATAGTAATTTGATAAGTAAATGAAGTAGTAGTTTRTTTGGNTGTGATAANT
TAGTNGAATTNTGTTATNNNAAGTTTTATCNTATNTTGAAAAGTGTNAGRTTATTTTTTTTTCGTTGAGAT
AAGNTTATGATTGAATTATTTACTGGTATAGGGTTATANAAYGTAGGNYATTTATTTATTTGAGTTTTATT
AAAATTGTTNATGNTT?GTGGTTTTGC?TTAATTATNTATTTNTTAATTTGGTAAA-AAAAAAATAATRTT
AATATATTATATGAATATATATATATACGGGGG-------CCCCCGTATATATA----------
---------TGTATATATACAATA------ATGGTAGGT??ACTACCAT-----TATTGTATATATACATA
TAT-----GTTARGTAT?ATATANNAGTAG-------TATATNTNAYA???TAYATATNNNTNTT-

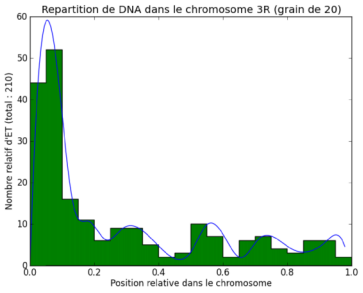
G

PROBABILITY LAWS OF JUMPS AND INITIAL CONDITIONS FOR TRANSPOSABLE ELEMENTS

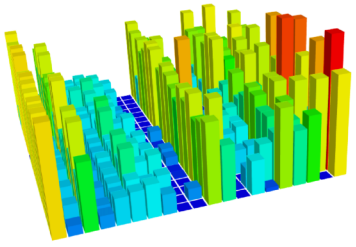
We have systematically obtained all the probability laws of jumps $p(x, y)$ and all initial conditions for the four chromosomes and the four types of transposable elements. A few of them have already been given in Chapter 9. A few other cases, sufficiently specific, are provided in this appendix.



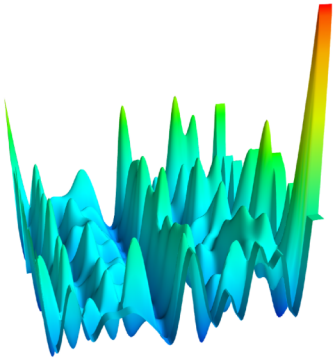
(a) Initial condition, polynomial version



(b) Initial condition, splines version

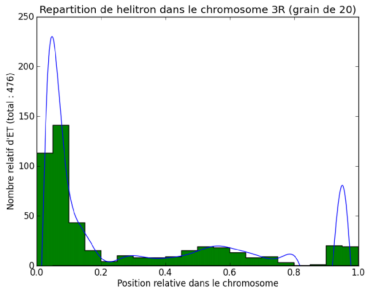


(c) Histograms of DNA jumps

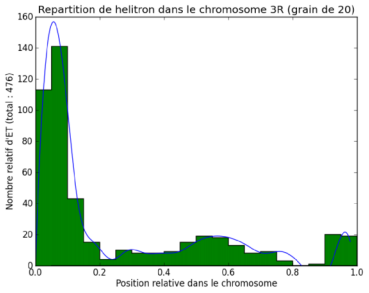


(d) Splines of DNA jumps

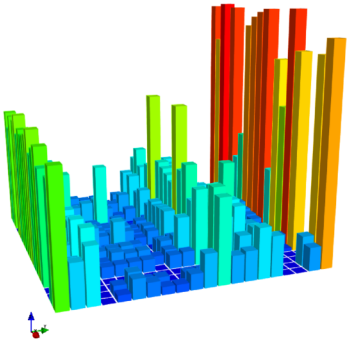
Figure G.1: Chromosome 3R, DNA elements



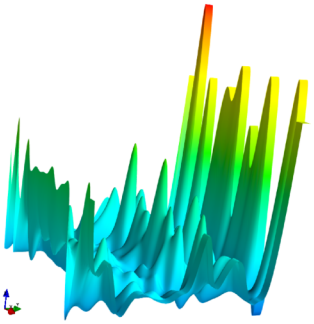
(a) Initial condition, polynomial version



(b) Initial condition, splines version

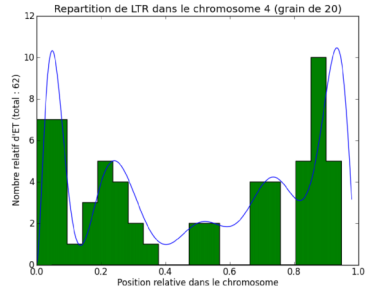


(c) Histograms of Helitron jumps

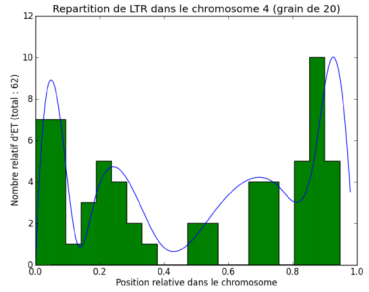


(d) Splines of Helitron jumps

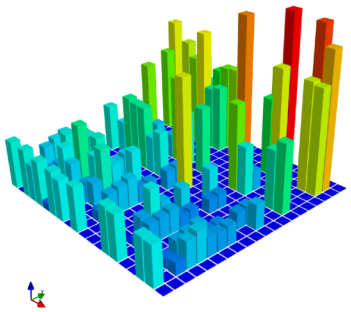
Figure G.2: Chromosome 3R, Helitron elements



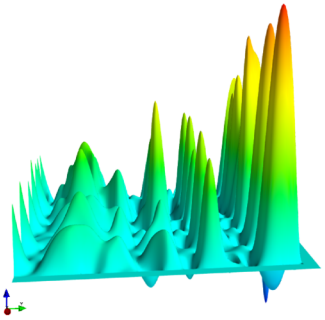
(a) Initial condition, polynomial version



(b) Initial condition, splines version

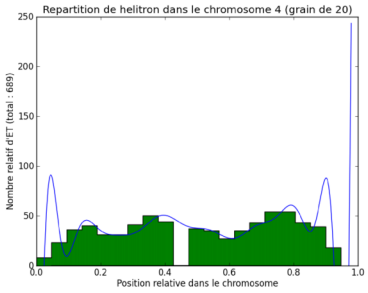


(c) Histograms of LTR jumps

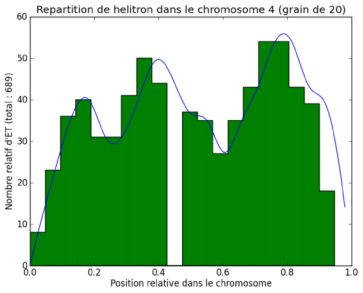


(d) Splines of LTR jumps

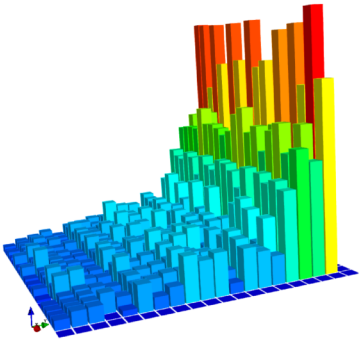
Figure G.3: Chromosome 4, LTR elements



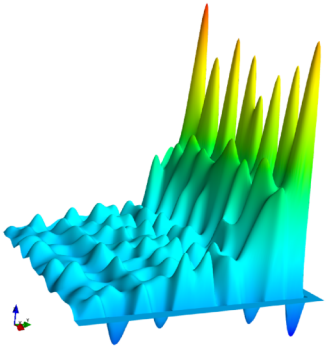
(a) Initial condition, polynomial version



(b) Initial condition, splines version

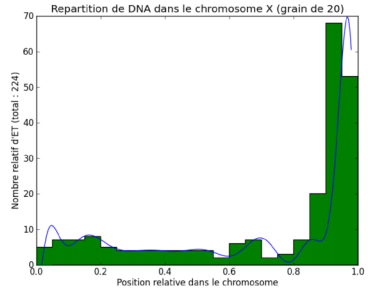


(c) Histograms of Helitron jumps

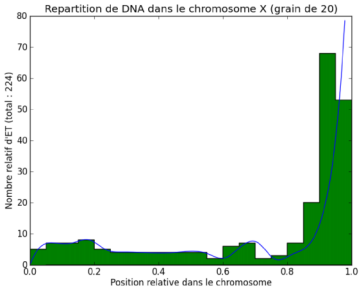


(d) Splines of Helitron jumps

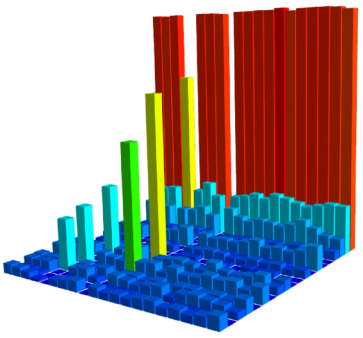
Figure G.4: Chromosome 4, Helitron elements



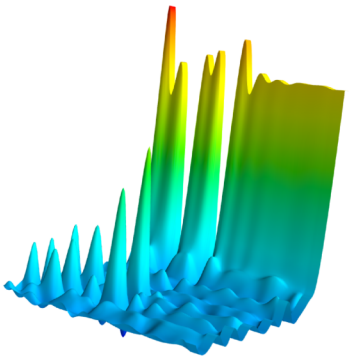
(a) Initial condition, polynomial version



(b) Initial condition, splines version

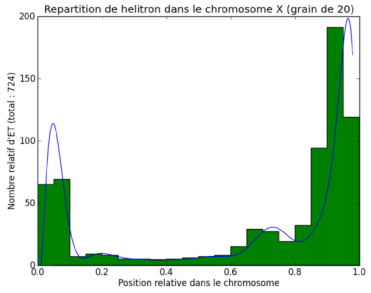


(c) Histograms of DNA jumps

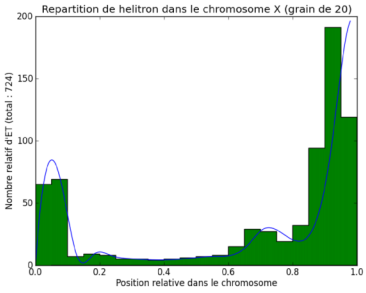


(d) Splines of DNA jumps

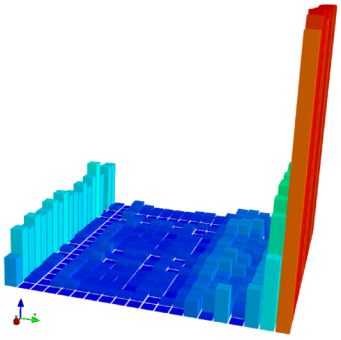
Figure G.5: Chromosome X, DNA elements



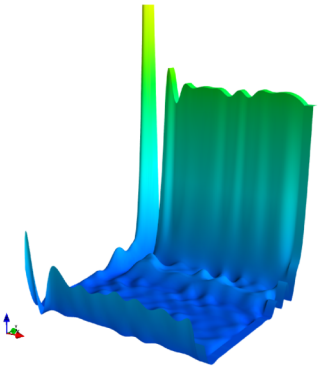
(a) Initial condition, polynomial version



(b) Initial condition, splines version



(c) Histograms of Helitron jumps



(d) Splines of Helitron jumps

Figure G.6: Chromosome X, Helitron elements

List of Figures

List of Tables

BIBLIOGRAPHY

[Abu64] K.I. Abuladze. *Principles of cestodology. Taeniata of Animals and Man and Diseases Caused by them*, volume 4. Izdatel'stvo Nauka, Moskow, 1964.

[AFM98] D. G. Arquès, J. P. Fallot, and C. J. Michel. An evolutionary analytical model of a complementary circular code simulating the protein coding genes, the 5' and 3' regions. *Bull Math Biol*, 60(1):163–194, Jan 1998.

[AKM65] R. L. Adler, A. G. Konheim, and M. H. McAndrew. Topological entropy. *Trans. Amer. Math. Soc.*, 114:309–319, 1965.

[AN04] K.B. Athreya and PE Ney. *Branching processes*. Dover Publications, 2004.

[Anf73] Christian B. Anfinsen. Principles that govern the folding of protein chains. *Science*, 181(4096):223–230, 1973.

[BA08] J. Barbier and S. Alt. Practical insecurity for effective steganalysis. In *Proc. of Information Hiding, 10th International Workshop, IH 2008*, volume 5284 of *Lecture Notes in Computer Science*, pages 195–2008, Santa Barbara, (CA) USA, May 2008. Springer.

[BAM09] J. Barbier, S. Alt, and E. Mayer. Modèles de sécurité en stéganographie. In *Proc. of Workshop Interdisciplinaire sur la Sécurité Globale, WISG'09*, Troyes, France, January 2009.

[BB01] Richard Bonneau and David Baker. Ab initio protein structure prediction: Progress and prospects. *Annual Review of Biophysics and Biomolecular Structure*, 30(1):173–189, 2001.

[BBM11] Axel Bacher and Mireille Bousquet-Mélou. Weakly directed self-avoiding walks. *J. Comb. Theory Ser. A*, 118(8):2365–2391, November 2011.

[BBS86] Lenore Blum, Manuel Blum, and Michael Shub. A simple unpredictable pseudo-random number generator. *SIAM Journal on Computing*, 15:364–383, 1986.

[BCF+13] Jacques Bahi, Jean-François Couchot, Nicolas Friot, Christophe Guyeux, and Kamel Mazouzi. Quality studies of an invisible chaos-based watermarking scheme with message extraction. In *IIHMSP'13, 9th Int. Conf. on Intelligent Information Hiding and Multimedia Signal Processing*, pages ***–***, Beijing, China, October 2013. To appear.

[BCFG12a] Jacques Bahi, Jean-François Couchot, Nicolas Friot, and Christophe Guyeux. Application of steganography for anonymity through the internet. In *IHTIAP'2012, 1-st Workshop on Information Hiding Techniques for Internet Anonymity and Privacy*, pages 96–101, Venice, Italy, June 2012.

[BCFG12b] Jacques Bahi, Jean-François Couchot, Nicolas Friot, and Christophe Guyeux. A robust data hiding process contributing to the development of a semantic web. In *INTERNET'2012, 4-th Int. Conf. on Evolving Internet*, pages 71–76, Venice, Italy, June 2012.

[BCG11a] Jacques Bahi, Nathalie Côté, and Christophe Guyeux. Chaos of protein folding. In *IJCNN 2011, Int. Joint Conf. on Neural Networks*, pages 1948–1954, San Jose, California, United States, July 2011.

[BCG11b] Jacques Bahi, Jean-François Couchot, and Christophe Guyeux. Performance analysis of a keyed hash function based on discrete and chaotic proven iterations. In *INTERNET 2011, the 3-rd Int. Conf. on Evolving Internet*, pages 52–57, Luxembourg, Luxembourg, June 2011. Best paper award.

[BCG11c] Jacques Bahi, Jean-François Couchot, and Christophe Guyeux. Steganography: a class of algorithms having secure properties. In *IIH-MSP-2011, 7-th Int. Conf. on Intelligent Information Hiding and Multimedia Signal Processing*, pages 109–112, Dalian, China, October 2011.

[BCG12a] Jacques Bahi, Jean-François Couchot, and Christophe Guyeux. Quality analysis of a chaotic proven keyed hash function. *International Journal On Advances in Internet Technology*, 5(1):26–33, 2012.

[BCG12b] Jacques Bahi, Jean-François Couchot, and Christophe Guyeux. Steganography: a class of secure and robust algorithms. *The Computer Journal*, 55(6):653–666, 2012.

[BCGG10] Jacques Bahi, Jean-François Couchot, Olivier Grasset, and Christophe Guyeux. Discrete Dynamical Systems: Necessary Divergence Conditions for Synchronous Iterations. Research Report RR2010-04, LIFC - Laboratoire d'Informatique de l'Université de Franche Comté, September 2010.

[BCGH11] Jacques M. Bahi, Raphaël Couturier, Christophe Guyeux, and Pierre-Cyrille Héam. Efficient and cryptographically secure generation of chaotic pseudorandom numbers on gpu. *CoRR*, abs/1112.5239, 2011.

[BCGR11] Jacques Bahi, Jean-François Couchot, Christophe Guyeux, and Adrien Richard. On the link between strongly connected iteration graphs and chaotic boolean discrete-time dynamical systems. In *FCT'11, 18th Int. Symp. on Fundamentals of Computation Theory*, volume 6914 of *LNCS*, pages 126–137, Oslo, Norway, August 2011.

[BCGS12a] Jacques Bahi, Nathalie Côté, Christophe Guyeux, and Michel Salomon. Protein folding in the 2D hydrophobic-hydrophilic (HP) square lattice model is chaotic. *Cognitive Computation*, 4(1):98–114, 2012.

[BCGS12b] Jacques Bahi, Jean-François Couchot, Christophe Guyeux, and Michel Salomon. Neural networks and chaos: Construction, evaluation of chaotic networks, and prediction of chaos with multilayer feedforward network. *Chaos, An Interdisciplinary Journal of Nonlinear Science*, 22(1):013122–1 – 013122–9, March 2012. 9 pages.

[BCGW11] Jacques Bahi, Jean-François Couchot, Christophe Guyeux, and Qianxue Wang. Class of trustworthy pseudo random number generators. In *INTERNET 2011, the 3-rd Int. Conf. on Evolving Internet*, pages 72–77, Luxembourg, Luxembourg, June 2011.

[BFG12a] Jacques Bahi, Xiaole Fang, and Christophe Guyeux. An optimization technique on pseudorandom generators based on chaotic iterations. In *INTERNET'2012, 4-th Int. Conf. on Evolving Internet*, pages 31–36, Venice, Italy, June 2012.

[BFG12b] Jacques Bahi, Nicolas Friot, and Christophe Guyeux. Lyapunov exponent evaluation of a digital watermarking scheme proven to be secure. In *IIH-MSP'2012, 8-th Int. Conf. on Intelligent Information Hiding and Multimedia Signal Processing*, pages 359–362, Piraeus-Athens, Greece, July 2012. IEEE Computer Society.

[BFG13] Jacques Bahi, Nicolas Friot, and Christophe Guyeux. Topological study and lyapunov exponent of a secure steganographic scheme. In Javier Lopez and Pierangela Samarati, editors, *SECRYPT'2013, Int. Conf. on Security and Cryptography. SECRYPT is part of ICETE - The International Joint Conference on e-Business and Telecommunications*, pages ***–***, Reykjavik, Iceland, July 2013. SciTePress. 8 pages. To appear.

[BFGG12] Nicholas R. Beaton, Philippe Flajolet, Timothy M. Garoni, and Anthony J. Guttmann. Some new self-avoiding walk and polygon models. *Fundam. Inf.*, 117(1-4):19–33, January 2012.

[BFGW11] Jacques Bahi, Xiaole Fang, Christophe Guyeux, and Qianxue Wang. On the design of a family of CI pseudo-random number generators. In *WICOM'11, 7th Int. IEEE Conf. on Wireless Communications, Networking and Mobile Computing*, pages 1–4, Wuhan, China, September 2011.

[BFGW13] Jacques Bahi, Xiaole Fang, Christophe Guyeux, and Qianxue Wang. Suitability of chaotic iterations schemes using XORshift for security applications. *JNCA, Journal of Network and Computer Applications*, *(*):***–***, 2013. Accepted manuscript. To appear.

[BFM06] J. Barbier, É. Filiol, and K. Mayoura. New features for specific JPEG steganalysis. 2(3):119–124, 2006.

[BFP11] P. Bas, T. Filler, and T. Pevný. Break our steganographic system — the ins and outs of organizing boss. In T. Filler, editor, *Information Hiding, 13th International Workshop*, Lecture Notes in Computer Science, Prague, Czech Republic, May 18–20, 2011. Springer-Verlag, New York.

[BG85] Manuel Blum and Shafi Goldwasser. An efficient probabilistic public key encryption scheme which hides all partial information. In *Proceedings of CRYPTO 84 on Advances in cryptology*, pages 289–302, New York, NY, USA, 1985. Springer-Verlag New York, Inc.

[BG08] Jacques M. Bahi and Christophe Guyeux. Chaotic iterations and topological chaos. 2008.

[BG10a] Jacques Bahi and Christophe Guyeux. Hash functions using chaotic iterations. *Journal of Algorithms & Computational Technology*, 4(2):167–181, 2010.

[BG10b] Jacques Bahi and Christophe Guyeux. A new chaos-based watermarking algorithm. In *SECRYPT'10, Int. conf. on security and cryptography*, pages 455–458, Athens, Greece, July 2010. SciTePress.

[BG10c] Jacques Bahi and Christophe Guyeux. A new chaos-based watermarking algorithm. In *SECRYPT'10, Int. conf. on security and cryptography*, pages 455–458, Athens, Greece, July 2010. SciTePress.

[BG10d] Jacques Bahi and Christophe Guyeux. Topological chaos and chaotic iterations, application to hash functions. In *IJCNN'10, Int. Joint Conf. on Neural Networks, joint to WCCI'10, IEEE World Congress on Computational Intelligence*, pages 1–7, Barcelona, Spain, July 2010. Best paper award.

[BG13] Jacques Bahi and Christophe Guyeux. *Discrete Dynamical Systems and Chaotic Machines: Theory and Applications*. Chapman & Hall, CRC Press, June 2013. 212 pages.

[BGG13] Jacques M. Bahi, Alain Giorgetti, and Christophe Guyeux. Unfoldable self-avoiding walks are infinite. Consequences for the protein structure prediction problem. *arXiv.org*, 2013.

[BGH13] Jacques Bahi, Christophe Guyeux, and Pierre-Cyrille Héam. A cryptographic approach for steganography. In *IIHMSP'13, 9th Int. Conf. on Intelligent Information Hiding and Multimedia Signal Processing*, pages ***–***, Beijing, China, October 2013. To appear.

[BGM] Jacques M. Bahi, Christophe Guyeux, and Abdallah Makhoul. A security framework for wireless sensor networks: Theory and practice. Submitted to The Scientific World Journal.

[BGM10a] Jacques Bahi, Christophe Guyeux, and Abdallah Makhoul. Efficient and robust secure aggregation of encrypted data in sensor networks. In *SENSORCOMM'10, 4-th Int. Conf. on Sensor Technologies and Applications*, pages 472–477, Venice-Mestre, Italy, July 2010.

[BGM10b] Jacques Bahi, Christophe Guyeux, and Abdallah Makhoul. Secure data aggregation in wireless sensor networks. homomorphism versus watermarking approach. In *ADHOCNETS 2010, 2nd Int. Conf. on Ad Hoc Networks*, volume 49 of *Lecture Notes in ICST*, pages 344–358, Victoria, Canada, August 2010.

[BGM14] Jacques Bahi, Christophe Guyeux, and Abdallah Makhoul. Two security layers for hierarchical data aggregation in sensor networks. *IJAACS, International Journal of Autonomous and Adaptive Communications Systems*, 7(3):***–***, 2014. Accepted manuscript. To appear.

[BGMP11] Jacques Bahi, Christophe Guyeux, Abdallah Makhoul, and Congduc Pham. Secure scheduling of wireless video sensor nodes for surveillance

applications. In *ADHOCNETS 11, 3rd Int. ICST Conference on Ad Hoc Networks*, volume 89 of *LNICST*, pages 1–15, Paris, France, September 2011. Springer.

[BGMP12] Jacques Bahi, Christophe Guyeux, Abdallah Makhoul, and Congduc Pham. Low cost monitoring and intruders detection using wireless video sensor networks. *International Journal of Distributed Sensor Networks*, 2012, 2012. 11 pages.

[BGMP13] Jacques M. Bahi, Christophe Guyeux, Kamel Mazouzi, and Laurent Philippe. Computational investigations of folded self-avoiding walks related to protein folding. *Computational Biology and Chemistry (Elsevier)*, 2013. Accepted paper, to appear.

[BGNP13] Jacques M. Bahi, Christophe Guyeux, Jean-Marc Nicod, and Laurent Philippe. Protein structure prediction software generate two different sets of conformations. Or the study of unfolded self-avoiding walks. *arXiv:1306.1439*, 2013. Submitted to Computational Biology and Chemistry (Elsevier).

[BGPa] Jacques M. Bahi, Christophe Guyeux, and Antoine Perasso. Chaos in dna evolution. Submitted to International Journal of Biomathematics.

[BGPb] Jacques M. Bahi, Christophe Guyeux, and Antoine Perasso. Relaxing the symmetric hypothesis in genome evolutionary models. Journal of Biological Systems.

[BGP12a] Jacques Bahi, Christophe Guyeux, and Antoine Perasso. Predicting the evolution of gene ura3 in the yeast saccharomyces cerevisiae. In *CSBio 2012: 3rd Int. Conf. on Computational Systems-Biology and Bioinformatics*, pages ***–***, Bangkok, Thailand, October 2012. To appear.

[BGP12b] Jacques M. Bahi, Christophe Guyeux, and Antoine Perasso. Predicting the evolution of two genes in the yeast saccharomyces cerevisiae. *Procedia CS*, 11:4–16, 2012.

[BGS11] Jacques Bahi, Christophe Guyeux, and Michel Salomon. Building a chaotic proven neural network. In *ICCANS 2011, IEEE Int. Conf. on Computer Applications and Network Security*, pages ***–***, Maldives, Maldives, May 2011.

[BGW09] Jacques Bahi, Christophe Guyeux, and Qianxue Wang. A novel pseudo-random generator based on discrete chaotic iterations. In *INTERNET'09, 1-st Int. Conf. on Evolving Internet*, pages 71–76, Cannes, France, August 2009.

[BGW10a] Jacques Bahi, Christophe Guyeux, and Qianxue Wang. A pseudo random numbers generator based on chaotic iterations. application to watermarking. In *WISM 2010, Int. Conf. on Web Information Systems and Mining*, volume 6318 of *LNCS*, pages 202–211, Sanya, China, October 2010.

[BGW10b] Jacques M. Bahi, Christophe Guyeux, and Qianxue Wang. Improving random number generators by chaotic iterations. Application in data hiding. In

ICCASM 2010, Int. Conf. on Computer Application and System Modeling, pages V13–643–V13–647, Taiyuan, China, October 2010.

[BHS+12] Yann Blouin, Yolande Hauck, Charles Soler, Michel Fabre, Rithy Vong, Céline Dehan, Géraldine Cazajous, Pierre-Laurent Massoure, Philippe Kraemer, Akinbowale Jenkins, Eric Garnotel, Christine Pourcel, and Gilles Vergnaud. Significance of the identification in the horn of africa of an exceptionally deep branching <italic>mycobacterium tuberculosis</italic> clade. *PLoS ONE*, 7(12):e52841, 12 2012.

[biq11] Biqi page, 2011. http://live.ece.utexas.edu/research/quality/BIQI_release.zip.

[BL98] Bonnie Berger and Tom Leighton. Protein folding in the hydrophobic-hydrophilic (hp) is np-complete. In *Proceedings of the second annual international conference on Computational molecular biology*, RECOMB '98, pages 30–39, New York, NY, USA, 1998. ACM.

[BM04] Jacques M Bahi and Christian J Michel. A stochastic gene evolution model with time dependent mutations. *Bull Math Biol*, 66(4):763–778, Jul 2004.

[BM08a] Jacques M Bahi and Christian J Michel. A stochastic model of gene evolution with chaotic mutations. *J Theor Biol*, 255(1):53–63, Nov 2008.

[BM08b] J. Barbier and E. Mayer. Non-malleable schemes resisting adaptive adversaries. In *Proc. of Digital Watermarking, 7th International Workshop, IWDW 2008*, Lecture Notes in Computer Science, Busan, Korea, November 2008. Springer.

[BMG10] Jacques Bahi, Abdallah Makhoul, and Christophe Guyeux. Efficient and robust secure aggregation of encrypted data in sensor networks for critical applications. In *RESSACS, Journée thématique PHC/ResCom sur RESeaux de capteurS et Applications Critiques de Surveillance*, Bayonne, France, June 2010. Communication orale.

[BPS+] Caroline Bréchet, Julie Plantin, Marlène Sauget, Michelle Thouverez, Pascal Cholley, Christophe Guyeux, Didier Hocquet, and Xavier Bertrand. The wastewater treatment plant's outflow and sludge release esbl-producing escherichia coli in the environment. Submitted to Clinical infectious disease (30/09/13) 27 pages.

[BR10] E. Barker and A. Roginsky. Draft NIST special publication 800-131 recommendation for the transitioning of cryptographic algorithms and key sizes, 2010.

[BSNP96] S. Bakhtiari, R. Safavi-Naini, and J. Pieprzyk. Keyed hash functions. In *Cryptography: Policy and Algorithms*, volume 1029, pages 201–214. 1996.

[BUAM97] Michael Braxenthaler, R. Ron Unger, Ditza Auerbach, and John Moult. Chaos in protein dynamics. *Proteins-structure Function and Bioinformatics*, 29:417–425, 1997.

[BWC99] R. Backofen, S. Will, and P. Clote. Algorithmic approach to quantifying the hydrophobic force contribution in protein folding, 1999.

[Bö91] Gerald Böhm. Protein folding and deterministic chaos: Limits of protein folding simulations and calculations. *Chaos, Solitons & Fractals*, 1(4):375 – 382, 1991.

[Cac98] Christian Cachin. An information-theoretic model for steganography. In *Information Hiding*, volume 1525 of *Lecture Notes in Computer Science*, pages 306–318. Springer Berlin / Heidelberg, 1998.

[CB08a] F. Cayre and P. Bas. Kerckhoffs-based embedding security classes for woa data hiding. *IEEE Transactions on Information Forensics and Security*, 3(1):1–15, 2008.

[CB08b] François Cayre and Patrick Bas. Kerckhoffs-based embedding security classes for woa data hiding. *Information Forensics and Security, IEEE Transactions on*, 3(1):1–15, 2008.

[CEG93] A. R. Conway, I. G. Enting, and A. J. Guttmann. Algebraic techniques for enumerating self-avoiding walks on the square lattice. *Journal of Physics A Mathematical General*, 26:1519–1534, April 1993.

[CFS08] E Chrysochos, V Fotopoulos, and A N Skodras. Robust watermarking of digital images based on chaotic mapping and dct. In *16th European Signal Processing Conference (EUSIPCO 2008), Lausanne, Switzerland,*, pages 17–21. EURASIP, August 2008.

[CG13] Raphaël Couturier and Christophe Guyeux. *Pseudorandom number generator on GPU*, chapter 19, pages ***–***. CRC press, 2013. To appear.

[CGH07] Nigel Crook, Wee Jin Goh, and Mohammad Hawarat. Pattern recall in networks of chaotic neurons. *Biosystems*, 87(2-3):267 – 274, 2007.

[CGP+98] Pierluigi Crescenzi, Deborah Goldman, Christos Papadimitriou, Antonio Piccolboni, and Mihalis Yannakakis. On the complexity of protein folding (extended abstract). In *Proceedings of the thirtieth annual ACM symposium on Theory of computing*, STOC '98, pages 597–603, New York, NY, USA, 1998. ACM.

[cit03] *Analysis of energy consumption of RC4 and AES algorithms in wireless LANs*, volume 3, 2003.

[CKM+05] Dylan Chivian, David E. Kim, Lars Malmström, Jack Schonbrun, Carol A. Rohl, and David Baker. Prediction of casp6 structures using automated robetta protocols. *Proteins*, 61(S7):157–166, 2005.

[CLBGB] Nathalie M.-L Côté, Matthieu Le Bailly, Christophe Guyeux, and Jacques M. Bahi. A molecular phylogeny of 33 eucestoda species based on complete mitochondrial genomes. Submitted to...

[CMK+97] Ingemar J. Cox, Senior Member, Joe Kilian, F. Thomson Leighton, and Talal Shamoon. Secure spread spectrum watermarking for multimedia. *IEEE Transactions on Image Processing*, 6:1673–1687, 1997.

[com88] Efficient and portable combined random number generators. *Communications of the ACM*, 31(6):742–749, 1988.

[Cop55] W. A. Coppel. The solution of equations by iteration. *Mathematical Proceedings of the Cambridge Philosophical Society*, 51(01):41–43, 1955.

[CPFPG05] Pedro Comesaña, Luis Pérez-Freire, and Fernando Pérez-González. Fundamentals of data hiding security and their application to spread-spectrum analysis. In Mauro Barni, Jordi Herrera-Joancomartí, Stefan Katzenbeisser, and Fernando Pérez-González, editors, *IH'05: Information Hiding Workshop*, volume 3727 of *Lecture Notes in Computer Science*, pages 146–160. Lectures Notes in Computer Science, Springer-Verlag, 2005.

[Cyb89] George Cybenko. Approximation by superpositions of a sigmoidal function. *Mathematics of Control, Signals and Systems*, 2:303–314, 1989.

[DBL10] *Proceedings of the IEEE Congress on Evolutionary Computation, CEC 2010, Barcelona, Spain, 18-23 July 2010*. IEEE, 2010.

[DD10] Ilker Dalkiran and Kenan Danisman. Artificial neural network based chaotic generator for cryptology. *Turk. J.Elec. Eng. & Comp. Sci.*, 18(2):225–240, 2010.

[Dev89] Robert L. Devaney. *An Introduction to Chaotic Dynamical Systems*. Addison-Wesley, Redwood City, CA, 2nd edition, 1989.

[dG72] P. G. de Gennes. Exponents for the excluded volume problem as derived by the Wilson method. *Physics Letters A*, 38(5):339–340, February 1972.

[DG02] Richard Desper and Olivier Gascuel. Fast and accurate phylogeny reconstruction algorithms based on the minimum-evolution principle. In Roderic Guigó and Dan Gusfield, editors, *Algorithms in Bioinformatics*, volume 2452 of *Lecture Notes in Computer Science*, pages 357–374. Springer Berlin Heidelberg, 2002.

[Dil85] KA Dill. Theory for the folding and stability of globular proteins. *Biochemistry*, 24(6):1501–9–, March 1985.

[DMBB05] Chuong B. Do, Mahathi S. P. Mahabhashyam, Michael Brudno, and Serafim Batzoglou. Probcons: Probabilistic consistency-based multiple sequence alignment. *Genome Res*, 15:330–340, 2005.

[DMHK95] I. Dubchak, I. Muchnik, S. R. Holbrook, and S. H. Kim. Prediction of protein folding class using global description of amino acid sequence. *Proc Natl Acad Sci U S A*, 92(19):8700–8704, Sep 1995.

[DPMS+08] Roberto Di Pietro, Luigi V. Mancini, Claudio Soriente, Angelo Spognardi, and Gene Tsudik. Catch me (if you can): Data survival in unattended sensor networks. In *Proceedings of the 2008 Sixth Annual IEEE International Conference on Pervasive Computing and Communications*, PERCOM '08, pages 185–194, Washington, DC, USA, 2008. IEEE Computer Society.

[DPV11] Roberto Di Pietro and Nino Vincenzo Verde. Epidemic data survivability in unattended wireless sensor networks. In *Proceedings of the fourth ACM conference on Wireless network security*, WiSec '11, pages 11–22, New York, NY, USA, 2011. ACM.

[EAP+06] Karen Egiazarian, Jaakko Astola, Vladimir Ponomarenko, Nikolayand Lukin, Frederica Battisti, and Marco Carli. New full-reference quality metrics based on hvs. In Baoxin Li, editor, *CD-ROM Proceedings of the Second International Workshop on Video Processing and Quality Metrics, Scottsdale, USA*, January 2006.

[Edg04] R. C. Edgar. Muscle: a multiple sequence alignment method with reduced time and space complexity. *BMC Bioinformatics*, 5(1), August 2004.

[ETOSD05] G. S. El-Taweel, H. M. Onsi, M. Samy, and M. G. Darwish. Secure and non-blind watermarking scheme for color images based on dwt. *ICGST International Journal on Graphics, Vision and Image Processing*, 05:1–5, April 2005.

[Fan13] Xiaole Fang. *Utilization of chaotic dynamics for generating pseudorandom numbers in various contexts*. PhD thesis, Université de Franche-Comté, 2013.

[Fel80] J. Felsenstein. A view of population genetics. *Science*, 208(4449):1253, Jun 1980.

[FGB11] Nicolas Friot, Christophe Guyeux, and Jacques Bahi. Chaotic iterations for steganography - stego-security and chaos-security. In Javier Lopez and Pierangela Samarati, editors, *SECRYPT'2011, Int. Conf. on Security and Cryptography. SECRYPT is part of ICETE - The International Joint Conference on e-Business and Telecommunications*, pages 218–227, Sevilla, Spain, July 2011. SciTePress.

[Flo49] Paul J. Flory. The Configuration of Real Polymer Chains. *The Journal of Chemical Physics*, 17(3):303–310, 1949.

[For98] Enrico Formenti. *Automates cellulaires et chaos : de la vision topologique à la vision algorithmique*. PhD thesis, École Normale Supérieure de Lyon, 1998.

[FPK07] Jessica J. Fridrich, Tomás Pevný, and Jan Kodovský. Statistically undetectable jpeg steganography: dead ends challenges, and opportunities. In Deepa Kundur, Balakrishnan Prabhakaran, Jana Dittmann, and Jessica J. Fridrich, editors, *MM&Sec*, pages 3–14. ACM, 2007.

[Fri09] Jessica Fridrich. *Steganography in Digital Media: Principles, Algorithms, and Applications*. Cambridge University Press, New York, NY, USA, 1st edition, 2009.

[FS97] R. Fischlin and C. P. Schnorr. Stronger security proofs for rsa and rabin bits. In *Proceedings of the 16th annual international conference on Theory and application of cryptographic techniques*, EUROCRYPT'97, pages 267–279, Berlin, Heidelberg, 1997. Springer-Verlag.

[Gas97] O. Gascuel. BIONJ: an improved version of the NJ algorithm based on a simple model of sequence data. *Molecular biology and evolution*, 14(7):685–695, July 1997.

[GB12] Christophe Guyeux and Jacques Bahi. A topological study of chaotic iterations. application to hash functions. In *CIPS, Computational Intelligence for Privacy and Security*, volume 394 of *Studies in Computational Intelligence*, pages 51–73. Springer, 2012. Revised and extended journal version of an IJCNN best paper.

[GBC] Christophe Guyeux, Jacques M. Bahi, and Raphaël Couturier. Introducing the truly chaotic finite state machines and theirs applications in security field. Submitted to the International Journal of Unconventional Computing.

[GCBB] Christophe Guyeux, Nathalie Côté, Wojciech Bienia, and Jacques M. Bahi. Is protein folding problem really a np-complete one? first investigations. *Journal of bioinformatics and computational biology (JBCB)*, (*):***–***, *. Accepted paper, to appear.

[GCR+] C. Guyeux, J.-F. Couchot, J.-Y. Roland, B Al Kindy, and J.M. Bahi. Reconstructing the last common ancestor of the mycobacterium tuberculosis complex: a position paper. Submitted to Bioinformatics, the 5th international conference on bioinformatics models, methods and algorithms.

[GDL+10] S. Guindon, JF Dufayard, V. Lefort, M. Anisimova, W. Hordijk, and O. Gascuel. New algorithms and methods to estimate maximum-likelihood phylogenies: assessing the performance of phyml 3.0. *Systematic Biology*, 59(3):307–321, 2010.

[GFB10] Christophe Guyeux, Nicolas Friot, and Jacques Bahi. Chaotic iterations versus spread-spectrum: chaos and stego security. In *IIH-MSP'10, 6-th Int. Conf. on Intelligent Information Hiding and Multimedia Signal Processing*, pages 208–211, Darmstadt, Germany, October 2010.

[GGM86] Oded Goldreich, Shafi Goldwasser, and Silvio Micali. How to construct random functions. *J. ACM*, 33:792–807, August 1986.

[GMB] Christophe Guyeux, Abdallah Makhoul, and Jacques M. Bahi. Epidemiological approaches for data survivability in unattended wireless sensor networks: considering the sensors lifetime. Submitted to New Generation Computing (Springer).

[Gol07] Oded Goldreich. *Foundations of Cryptography: Basic Tools*. Cambridge University Press, 2007.

[Gra53] F. Gray. Pulse code communication, 1953. US Patent 2,632,058, March 17 1953,(filed November 13 1947).

[Guy10] Christophe Guyeux. *Le désordre des itérations chaotiques et leur utilité en sécurité informatique*. PhD thesis, Université de Franche-Comté, 2010.

[Guy12] Christophe Guyeux. *Le désordre des itérations chaotiques - Applications aux réseaux de capteurs, à la dissimulation d'information, et aux fonctions de hachage*. Éditions Universitaires Européennes, 2012. ISBN 978-3-8417-9417-8. 362 pages. Publication de la thèse de doctorat.

[GWHC09] Wei Guo, Xiaoming Wang, Dake He, and Yang Cao. Cryptanalysis on a parallel keyed hash function based on chaotic maps. *Physics Letters A*, 373(36):3201 – 3206, 2009.

[GY94] N. Goldman and Z. Yang. A codon-based model of nucleotide substitution for protein-coding DNA sequences. *Molecular biology and evolution*, 11(5):725–736, September 1994.

[HAQJ01] Eric P. Hoberg, Nancy L. Alkire, Alan D. Queiroz, and Arlene Jones. Out of Africa : origins of the Taenia tapeworms in humans. (September 2000), 2001.

[HC10] Dragos Horvath and Camelia Chira. Simplified chain folding models as metaheuristic benchmark for tuning real protein folding algorithms? In *IEEE Congress on Evolutionary Computation* [DBL10], pages 1–8.

[HCS09] Md. Hoque, Madhu Chetty, and Abdul Sattar. Genetic algorithm in ab initio protein structure prediction using low resolution model: A review. In Amandeep Sidhu and Tharam Dillon, editors, *Biomedical Data and Applications*, volume 224 of *Studies in Computational Intelligence*, pages 317–342. Springer Berlin Heidelberg, 2009.

[Het00] Herbert W. Hethcote. The mathematics of infectious diseases. *SIAM Review*, 42:599–653, 2000.

[HF10] H. C. Huang and W. C. Fang. Metadata-based image watermarking for copyright protection. *Simulation Modelling Practice and Theory*, 18(4):436–445, 2010.

[HJR+00] E.P. Hoberg, A. Jones, R.L. Rausch, K.S. Eom, and S.L. Gardner. A phylogenetic hypothesis for species of the genus taenia (eucestoda : Taeniidae). *J Parasitol*, 86(1):89–98, 2000.

[HKB09] A. Houmansadr, N. Kiyavash, and N. Borisov. Rainbow: A robust and invisible non-blind watermark for network flows. In *NDSS'09: 16th Annual Network and Distributed System Security Symposium*, 2009.

[HKY85] M. Hasegawa, H. Kishino, and T. Yano. Dating of the human-ape splitting by a molecular clock of mitochondrial dna. *J Mol Evol*, 22(2):160–174, 1985.

[HNW+07] M. Hüttner, M. Nakao, T. Wassermann, L. Siefert, J.D.F. Boomker, A. Dinkel, Y. Sako, U. Mackenstedt, T. Romig, and A. Ito. Genetic characterization and phylogenetic position of echinococcus felidis (cestoda: Taeniidae) from the african lion. *Int J Parasitol*, 2007.

[Hob06] E.P. Hoberg. Phylogeny of taenia: Species definitions and origins of human parasites. *Parasitol Int*, 55 Suppl, 2006.

[HOZS10] Jiri Holoska, Zuzana Oplatkova, Ivan Zelinka, and Roman Senkerik. Comparison between neural network steganalysis and linear classification method stegdetect. *Computational Intelligence, Modelling and Simulation, International Conference on*, 0:15–20, 2010.

[HR01] John P. Huelsenbeck and Fredrik Ronquist. MRBAYES: bayesian inference of phylogenetic trees. *Bioinformatics*, 17(8):754–755, August 2001.

[HSHS10] Trent Higgs, Bela Stantic, Tamjidul Hoque, and Abdul Sattar. Genetic algorithm feature-based resampling for protein structure prediction. In *IEEE Congress on Evolutionary Computation* [DBL10], pages 1–8.

[HSW89] Kurt Hornik, Maxwell B. Stinchcombe, and Halbert White. Multilayer feedforward networks are universal approximators. *Neural Networks*, 2(5):359–366, 1989.

[IC09] Md. Kamrul Islam and Madhu Chetty. Novel memetic algorithm for protein structure prediction. In *Proceedings of the 22nd Australasian Joint Conference on Advances in Artificial Intelligence*, AI '09, pages 412–421, Berlin, Heidelberg, 2009. Springer-Verlag.

[IC10] Md. Kamrul Islam and Madhu Chetty. Clustered memetic algorithm for protein structure prediction. In *IEEE Congress on Evolutionary Computation* [DBL10], pages 1–8.

[JC69] T. H. Jukes and C. R. Cantor. *Evolution of Protein Molecules*. Academy Press, 1969.

[Jen96] R. J. Jenkins. ISAAC. *Fast Software Encryption*, pages 41–49, 1996.

[Jen04a] Iwan Jensen. Enumeration of self-avoiding walks on the square lattice. *J. Phys. A*, pages 5503–5524, 2004.

[Jen04b] Iwan Jensen. Enumeration of self-avoiding walks on the square lattice. *J. Phys. A*, pages 5503–5524, 2004.

[JKP+05] J. Jurka, V. V. Kapitonov, A. Pavlicek, P. Klonowski, O. Kohany, and J. Walichiewicz. Repbase Update, a database of eukaryotic repetitive elements. *Cytogenetic and genome research*, 110(1-4):462–467, 2005.

[Jun99] P. Junod. *Cryptographic secure pseudo-random bits generation: The Blum-Blum-Shub generator*. August, 1999.

[JYL+12] Wanzhong Jia, Hongbin Yan, Zhongzi Lou, Xingwei Ni, Viktor Dyachenko, Hongmin Li, and D. Timothy J. Littlewood. Mitochondrial genes and genomes support a cryptic species of tapeworm within taenia taeniaeformis. *Acta Tropica*, 123(3):154 – 163, 2012.

[Kal01] T. Kalker. Considerations on watermarking security. In *Multimedia Signal Processing, 2001 IEEE Fourth Workshop on*, pages 201–206, 2001.

[Kap82] N. Kaplan. A note on the branching random walk. *J. Appl. Probab.*, 19(2):421–424, 1982.

[KBB+13] Andrew Ker, Patrick Bas, Rainer Böhme, Rémi Cogranne, Scott Craver, Tomáš Filler, Jessica Fridrich, and Tomas Pevny. Moving Steganography and Steganalysis from the Laboratory into the Real World. In *ACM IH-MMSEC 2013*, pages ACM 978–1–4503–2081–8/13/06, Montpellier, France, June 2013.

[KDR06] Younhee Kim, Zoran Duric, and Dana Richards. Modified matrix encoding technique for minimal distortion steganography. In Jan Camenisch, Christian S. Collberg, Neil F. Johnson 0001, and Phil Sallee, editors, *Information Hiding*, volume 4437 of *Lecture Notes in Computer Science*, pages 314–327. Springer, 2006.

[KF11] J. Kodovský and J. Fridrich. Steganalysis in high dimensions: fusing classifiers built on random subspaces. In *Proc. SPIE, Electronic Imaging, Media Watermarking, Security, and Forensics XIII, San Francisco, CA,*, January 2011.

[KFGT07] D. Koller, N. Friedman, L. Getoor, and B. Taskar. Graphical models in a nutshell. In L. Getoor and B. Taskar, editors, *Introduction to Statistical Relational Learning*. MIT Press, 2007.

[KFH11] J. Kodovský, J. Fridrich, and V. Holub. Ensemble classifiers for steganalysis of digital media. *IEEE Transactions on Information Forensics and Security*, PP Issue:99:1 – 1, 2011. To appear.

[KiKTM05] Kazutaka Katoh, Kei ichi Kuma, Hiroyuki Toh, and Takashi Miyata. Mafft version 5: improvement in accuracy of multiple sequence alignment. *Nucleic Acids Research*, (2):511–518, 2005.

[Kim80] Motoo Kimura. A simple method for estimating evolutionary rates of base substitutions through comparative studies of nucleotide sequences. *Journal of Molecular Evolution*, 16:111–120, 1980. 10.1007/BF01731581.

[KM27] W. O. Kermack and Ag McKendrick. A Contribution to the Mathematical Theory of Epidemics. *Proceedings of the Royal Society of London. Series A, Containing Papers of a Mathematical and Physical Character*, 115(772):700–721, August 1927.

[Knu94a] Knudsen. Chaos without nonperiodicity. *Amer. Math. Monthly*, 101, 1994.

[Knu94b] C. Knudsen. *Aspects of noninvertible dynamics and chaos*. PhD thesis, Technical University of Denmark, 1994.

[Knu97] D. E. Knuth. *Seminumerical Algorithms*, volume 3. Addison-Wesley, Reading, MA, USA, third edition edition, 1997.

[KNY+11] Jenny Knapp, Minoru Nakao, Tetsuya Yanagida, Munehiro Okamoto, Urmas Saarma, Antti Lavikainen, and Akira Ito. Phylogenetic relationships within echinococcus and taenia tapeworms (cestoda: Taeniidae): An inference from nuclear protein-coding genes. *Mol Phylogenet Evol*, 2011.

[KPD+04] Stefan Kurtz, Adam Phillippy, Arthur L. Delcher, Michael Smoot, Martin Shumway, Corina Antonescu, and Steven L. Salzberg. Versatile and open software for comparing large genomes. *Genome biology*, 5(2):R12+, 2004.

[LBB+07] M.A. Larkin, G. Blackshields, N.P. Brown, R. Chenna, P.A. McGettigan, H. McWilliam, F. Valentin, I.M. Wallace, A. Wilm, R. Lopez, J.D. Thompson, T.J. Gibson, and D.G. Higgins. Clustal w and clustal x version 2.0. *Bioinformatics*, 23(21):2947–2948, November 2007.

[LDX10] Yantao Li, Shaojiang Deng, and Di Xiao. A novel hash algorithm construction based on chaotic neural network. *Neural Computing and Applications*, pages 1–9, 2010.

[LGS02] Christopher Lee, Catherine Grasso, and Mark F Sharlow. Multiple sequence alignment using partial order graphs. *Bioinformatics*, 18(3):452–64, March 2002.

[LHL+08] A. Lavikainen, V. Haukisalmi, M. J. Lehtinen, H. Henttonen, A. Oksanen, and S. Meri. A phylogeny of members of the family Taeniidae based on the mitochondrial cox1 and nad1 gene data. *Parasitology*, 135(12):1457–1467, October 2008.

[LHL+10] Antti Lavikainen, Voitto Haukisalmi, Markus J. Lehtinen, Sauli Laaksonen, Sauli Holmström, Marja Isomursu, Antti Oksanen, and Seppo Meri. Mitochondrial {DNA} data reveal cryptic species within taenia krabbei. *Parasitology International*, 59(2):290 – 293, 2010.

[Lia09] Shiguo Lian. A block cipher based on chaotic neural networks. *Neurocomputing*, 72(4-6):1296 – 1301, 2009.

[LLB09] Nicolas Lartillot, Thomas Lepage, and Samuel Blanquart. PhyloBayes 3: a Bayesian software package for phylogenetic reconstruction and molecular dating. *Bioinformatics*, 25(17):2286–2288, September 2009.

[LM08] Gregory I. Lang and Andrew W. Murray. Estimating the per-base-pair mutation rate in the yeast saccharomyces cerevisiae. *Genetics*, 178(1):67–82, January 2008.

[LY75] T. Y. Li and J. A. Yorke. Period three implies chaos. *Amer. Math. Monthly*, 82(10):985–992, 1975.

[Mar96] G. Marsaglia. Diehard: a battery of tests of randomness. *http://stat.fsu.edu/ geo/diehard.html*, 1996.

[Mar03] G. Marsaglia. Xorshift rngs. *Journal of Statistical Software*, 8(14):1–6, 2003.

[MB08] Saraju P. Mohanty and Bharat K. Bhargava. Invisible watermarking based on creation and robust insertion-extraction of image adaptive watermarks. *ACM Trans. Multimedia Comput. Commun. Appl.*, 5:12:1–12:22, November 2008.

[MB10] Anush K. Moorthy Moorthy and Alan Conrad Bovik. A two-step framework for constructing blind image quality indices. *IEEE Signal Processing Letters*, 17(5):513–516, May 2010.

[MCBE10] Michele Muzzarelli, Marco Carli, Giulia Boato, and Karen Egiazarian. Reversible watermarking via histogram shifting and least square optimization. In *Proceedings of the 12th ACM workshop on Multimedia and security, MM&Sec'10,Roma, Italy*, pages 147–152, New York, NY, USA, 2010. ACM.

[MG94] S V Muse and B S Gaut. A likelihood approach for comparing synonymous and nonsynonymous nucleotide substitution rates, with application to the chloroplast genome. *Molecular Biology and Evolution*, 11(5):715–724, 1994.

[Miy11] Sanzo Miyazawa. Selective constraints on amino acids estimated by a mechanistic codon substitution model with multiple nucleotide changes. *PLoS ONE*, 6(3):e17244, 03 2011.

[MK08] B.Chandra Mohan and S.Srinivas Kumar. Robust digital watermarking scheme using contourlet transform. *IJCSNS International Journal of Computer Science and Network Security*, 8(2), 2008.

[MLS+13] Steven J. Marygold, Paul C. Leyland, Ruth L. Seal, Joshua L. Goodman, Jim Thurmond, Victor B. Strelets, and Robert J. Wilson. Flybase: improvements to the bibliography. *Nucleic Acids Research*, 41(Database-Issue):751–757, 2013.

[MP09] Abdallah Makhoul and Congduc Pham. Dynamic scheduling of cover-sets in randomly deployed wireless video sensor networks for surveillance applications. *2nd IFIP Wireless Days Conference, WD'09*, pages 73–78, December 2009.

[MS88] Neal Madras and Alan D. Sokal. The pivot algorithm: A highly efficient monte carlo method for the self-avoiding walk. *Journal of Statistical Physics*, 50:109–186, 1988.

[MS93] Neal Noah Madras and Gordon Slade. *The self-avoiding walk*. Probability and its applications. Birkhäuser, Boston, 1993.

[MT08] Di Ma and Gene Tsudik. Dish: Distributed self-healing. In *Proceedings of the 10th International Symposium on Stabilization, Safety, and Security of Distributed Systems*, SSS '08, pages 47–62, Berlin, Heidelberg, 2008. Springer-Verlag.

[NHH00] C. Notredame, D. G. Higgins, and J. Heringa. T-Coffee: A novel method for fast and accurate multiple sequence alignment. *Journal of molecular biology*, 302(1):205–217, September 2000.

[NLI+] Minoru Nakao, Antti Lavikainen, Takashi Iwaki, Voitto Haukisalmi, Sergey Konyaev, Yuzaburo Oku, Munehiro Okamoto, and Akira Ito. Molecular phylogeny of the genus taenia (cestoda: Taeniidae): proposals for the resurrection of hydatigera lamarck, 1816 and the creation of a new genus versteria. *Int J Parasitol*.

[NMS+07] M. Nakao, D.P. McManus, P.M. Schantz, P.S. Craig, and A. Ito. A molecular phylogeny of the genus echinococcus inferred from complete mitochondrial genomes. *Parasitology*, 134:713–722, 4 2007.

[NYO+10] Minoru Nakao, Tetsuya Yanagida, Munehiro Okamoto, Jenny Knapp, Agathe Nkouawa, Yasuhito Sako, and Akira Ito. State-of-the-art echinococcus and taenia: Phylogenetic taxonomy of human-pathogenic tapeworms and its application to molecular diagnosis. *Infection, Genetics and Evolution*, 10(4):444 – 452, 2010.

[PBF10] Tomás Pevný, Patrick Bas, and Jessica J. Fridrich. Steganalysis by subtractive pixel adjacency matrix. *IEEE Transactions on Information Forensics and Security*, 5(2):215–224, 2010.

[PFB10a] Tomás Pevný, Tomás Filler, and Patrick Bas. Using high-dimensional image models to perform highly undetectable steganography. In Rainer Böhme, Philip W. L. Fong, and Reihaneh Safavi-Naini, editors, *Information Hiding*, volume 6387 of *Lecture Notes in Computer Science*, pages 161–177. Springer, 2010.

[PFB10b] Tomáš Pevný, Tomáš Filler, and Patrick Bas. Break our steganographic system, 2010. available at http://www.agents.cz/boss/.

[PFCTPPG06] Luis Perez-Freire, Pedro Comesana, Juan Ramon Troncoso-Pastoriza, and Fernando Perez-Gonzalez. Watermarking security: a survey. In *LNCS Transactions on Data Hiding and Multimedia Security*, 2006.

[PHRVGJ10] Luis Germán Pérez-Hernández, Katya Rodríguez-Vázquez, and Ramón Garduño-Juárez. Estimation of 3d protein structure by means of parallel particle swarm optimization. In *IEEE Congress on Evolutionary Computation* [DBL10], pages 1–8.

[PLHP01] M. J. Phillips, Y. H. Lin, G. L. Harrison, and D. Penny. Mitochondrial genomes of a bandicoot and a brushtail possum confirm the monophyly of australidelphian marsupials. *Proc Biol Sci*, 268(1475):1533–1538, July 2001.

[PMS11] Congduc Pham, Abdallah Makhoul, and Rachid Saadi. Risk-based adaptive scheduling in randomly deployed video sensor networks for critical surveillance applications. *Journal of Network and Computer Applications*, 34(2):783–795, 2011.

[PSE+07] Nikolay Ponomarenko, Flavia Silvestri, Karen Egiazarian, Marco Carli, Jaakko Astola, and Vladimir Lukin. On between-coefficient contrast masking of dct basis functions. In Baoxin Li, editor, *CD-ROM Proceedings of the Third International Workshop on Video Processing and Quality Metrics for Consumer Electronics VPQM-07,Scottsdale, Arizona, USA*, January 2007.

[PSG03] Jimin Pei, Ruslan Sadreyev, and Nick V. Grishin. Pcma: fast and accurate multiple sequence alignment based on profile consistency. *Bioinformatics*, 19(3):427–428, 2003.

[psn11] Psnr-hvs-m page, 2011. http://www.ponomarenko.info/psnrhvsm.htm.

[PV13] Roberto Di Pietro and Nino Vincenzo Verde. Epidemic theory and data survivability in unattended wireless sensor networks: Models and gaps. *Pervasive and Mobile Computing*, 9(4):588 – 597, 2013.

[PX11] Jian Peng and Jinbo Xu. Raptorx: Exploiting structure information for protein alignment by statistical inference. *Proteins*, 79(S10):161–171, 2011.

[QBCG11] W. Qianxue, J. Bahi, J.-F. Couchot, and C. Guyeux. Class of trustworthy pseudo-random number generators. In *INTERNET'2011. The 3rd Int. Conf. on Evolving Internet*, 2011.

[QSL09] Mengyu Qiao, Andrew H. Sung, and Qingzhong Liu. Steganalysis of mp3stego. In *Proceedings of the 2009 international joint conference on Neural Networks*, IJCNN'09, pages 2723–2728, Piscataway, NJ, USA, 2009. IEEE Press.

[Ran11] Aaron Randall. A novel semi-fragile watermarking scheme with iterative restoration. Available at http://www.aaronrandall.com/Files/WatermarkingPaperLight.pdf, 2011.

[RDBM03] Maria Rifqi, Marcin Detyniecki, and Bernadette Bouchon-Meunier. Discrimination power of measures of resemblance. In *IFSA'03*, 2003.

[Rob86] F. Robert. *Discrete Iterations: A Metric Study*, volume 6 of *Springer Series in Computational Mathematics*. 1986.

[Rue01] Sylvie Ruette. *Chaos en dynamique topologique, en particulier sur l'intervalle, mesures d'entropie maximale*. PhD thesis, Université d'Aix-Marseille II, 2001.

[SB06] Hamid R. Sheikh and Alan Conrad Bovik. Image information and visual quality. *IEEE Transactions on Image Processing*, 15(2):430–444, 2006.

[SHeb] Alena Shmygelska and Holger H Hoos. An ant colony optimisation algorithm for the 2d and 3d hydrophobic polar protein folding problem, 2005 Feb.

[SH05] Alena Shmygelska and Holger Hoos. An ant colony optimisation algorithm for the 2d and 3d hydrophobic polar protein folding problem. *BMC Bioinformatics*, 6(1):30, 2005.

[Sha49] Claude E. Shannon. Communication theory of secrecy systems. *Bell Systems Technical Journal*, 28:656–715, 1949.

[SHW03] Liu Shaohui, Yao Hongxun, and Gao Wen. Neural network based steganalysis in still images. *Multimedia and Expo, IEEE International Conference on*, 2:509–512, 2003.

[Sim84] Gustavus J. Simmons. The prisoners' problem and the subliminal channel. In *Advances in Cryptology, Proc. CRYPTO'83*, pages 51–67, 1984.

[SJM+09] U. SAARMA, I. JÕGISALU, E. MOKS, A. VARCASIA, A. LAVIKAINEN, A. OKSANEN, S. SIMSEK, V. ANDRESIUK, G. DENEGRI, L. M. GONZÁLEZ, E. FERRER, T. GÁRATE, L. RINALDI, and P. MARAVILLA. A novel phylogeny for the genus echinococcus, based on nuclear data, challenges relationships based on mitochondrial evidence. *Parasitology*, 136:317–328, 2 2009.

[SJT+04] K. Satish, T. Jayakar, C. Tobin, K. Madhavi, and K. Murali. Chaos based spread spectrum image steganography. *Consumer Electronics, IEEE Transactions on*, 50(2):587 – 590, May 2004.

[SL05] Ros Stamatakis and Thomas Ludwig. Raxml-omp: An efficient program for phylogenetic inference on smps. In *In Proc. of PaCT05*, pages 288–302, 2005.

[Sla11] Gordon Slade. The self-avoiding walk: a brief survey. Blath, Jochen (ed.) et al., Surveys in stochastic processes. Selected papers based on the presentations at the 33rd conference on stochastic processes and their applications, Berlin, Germany, July 27–31, 2009. Zürich: European Mathematical Society (EMS). EMS Series of Congress Reports, 181-199 (2011)., 2011.

[SM02] Richard Simard and Université De Montréal. Testu01: A software library in ansi c for empirical testing of random number generators., 2002.

[SM07] Richard Simard and Université De Montréal. Testu01: A c library for empirical testing of random number generators. *ACM Transactions on Mathematical Software*, page 2007, 2007.

[SMCM06] Kenneth Sullivan, Upamanyu Madhow, Shivkumar Ch, and B. S. Manjunath. Steganalysis for markov cover data with applications to images. *IEEE Trans. Inform. Forensics and Security*, 1:275–287, 2006.

[SN87] N. Saitou and M. Nei. The neighbor-joining method: a new method for reconstructing phylogenetic trees. *Molecular Biology and Evolution*, 4(4):406–425, July 1987.

[SQW+01] Li Shujun, Li Qi, Li Wenmin, Mou Xuanqin, and Cai Yuanlong. Statistical properties of digital piecewise linear chaotic maps and their roles in cryptography and pseudo-random coding. *Proceedings of the 8th IMA International Conference on Cryptography and Coding*, 1:205–221, 2001.

[SSM] Anindya Sarkar, Kaushal Solanki, and B. S. Manjunath. Further study on yass: Steganography based on randomized embedding to resist blind steganalysis.

[SSM07] Kaushal Solanki, Anindya Sarkar, and B. S. Manjunath. Yass: Yet another steganographic scheme that resists blind steganalysis. In Teddy Furon, François Cayre, Gwenaël J. Doërr, and Patrick Bas, editors, *Information Hiding*, volume 4567 of *Lecture Notes in Computer Science*, pages 16–31. Springer, 2007.

[sZlC97] Chang song Zhou and Tian lun Chen. Extracting information masked by chaos and contaminated with noise: Some considerations on the security of communication approaches using chaos. *Physics Letters A*, 234(6):429 – 435, 1997.

[Tam92] K Tamura. Estimation of the number of nucleotide substitutions when there are strong transition-transversion and g+c-content biases. *Molecular Biology and Evolution*, 9(4):678–687, 1992.

[TCL05] C. Temi, S. Choomchuay, and A. Lasakul. A robust image watermarking using multiresolution analysis of wavelet. In *ISCIT 2005. IEEE International Symposium on Communications and Information Technology*, pages 623–626, Washington, DC, October 2005. IEEE Computer Society.

[TN93] K Tamura and M Nei. Estimation of the number of nucleotide substitutions in the control region of mitochondrial dna in humans and chimpanzees. *Molecular Biology and Evolution*, 10(3):512–526, 1993.

[UM93] Ron Unger and John Moult. Genetic algorithm for 3d protein folding simulations. In *Proceedings of the 5th International Conference on Genetic Algorithms*, pages 581–588, San Francisco, CA, USA, 1993. Morgan Kaufmann Publishers Inc.

[UN47] S. M. Ulam and J. V. Neumann. On combination of stochastic and deterministic processes. *Amer. Math. Soc.*, 53:1120, 1947.

[VDB10] Nagaraj V. Dharwadkar and Amberker B.B. Watermarking scheme for color images using wavelet transform based texture properties and secret sharing. *International Journal of Information and Communication Engineering*, 6(2):93–100, 2010.

[VV85] Umesh Vazirani and Vijay Vazirani. Efficient and secure pseudo-random number generation (extended abstract). In George Blakley and David Chaum, editors, *Advances in Cryptology*, volume 196 of *Lecture Notes in Computer Science*, pages 193–202. Springer Berlin / Heidelberg, 1985. 10.1007/3-540-39568-7_17.

[Wan12] Qianxue Wang. *Generating pseudo-random numbers. Applications in cryptology*. PhD thesis, Université de Franche-Comté, 2012.

[WBGF10] Qianxue Wang, Jacques Bahi, Christophe Guyeux, and Xiaole Fang. Randomness quality of CI chaotic generators. application to internet security. In *INTERNET'2010. The 2nd Int. Conf. on Evolving Internet*, pages 125–130, Valencia, Spain, September 2010. IEEE Computer Society Press. Best Paper award.

[Wes01] Andreas Westfeld. F5-a steganographic algorithm. In Ira S. Moskowitz, editor, *Information Hiding*, volume 2137 of *Lecture Notes in Computer Science*, pages 289–302. Springer, 2001.

[WZZ03] X. M. Wang, J. S. Zhang, and W. F. Zhang. One-way hash function construction based on the extended chaotic maps switch. *Acta Phys. Sinici.*, 52, No. 11:2737–2742, 2003.

[XLW09a] Di Xiao, Xiaofeng Liao, and Yong Wang. Improving the security of a parallel keyed hash function based on chaotic maps. *Physics Letters A*, 373(47):4346 – 4353, 2009.

[XLW09b] Di Xiao, Xiaofeng Liao, and Yong Wang. Parallel keyed hash function construction based on chaotic neural network. *Neurocomputing*, 72(10-12):2288 – 2296, 2009. Lattice Computing and Natural Computing (JCIS 2007) / Neural Networks in Intelligent Systems Designn (ISDA 2007).

[XSL10] Di Xiao, Frank Y. Shih, and Xiaofeng Liao. A chaos-based hash function with both modification detection and localization capabilities. *Communications in Nonlinear Science and Numerical Simulation*, 15(9):2254 – 2261, 2010.

[Yam59] S. Yamaguti. *Systema helminthum. The Cestodes of Vertebrates*, volume 2. Interscience Publishers Inc., New York., 1959.

[Yan94] Z. Yang. Estimating the pattern of nucleotide substitution. *Journal of Molecular Evolution*, 10:105–111, 1994.

[Yao82] Andrew C. Yao. Theory and application of trapdoor functions. In *Proceedings of the 23rd Annual Symposium on Foundations of Computer Science*, SFCS '82, pages 80–91, Washington, DC, USA, 1982. IEEE Computer Society.

[YNH98] Z Yang, R Nielsen, and M Hasegawa. Models of amino acid substitution and applications to mitochondrial protein evolution. *Molecular Biology and Evolution*, 15(12):1600–1611, 1998.

[ZAS05] Yang Zhang, Adrian K. Arakaki, and Jeffrey Skolnick. Tasser: An automated method for the prediction of protein tertiary structures in casp6. *Proteins*, 61(S7):91–98, 2005.

[ZLW05] Linhua Zhang, Xiaofeng Liao, and Xuebing Wang. An image encryption approach based on chaotic maps. *Chaos, Solitons & Fractals*, 24(3):759 – 765, 2005.

[ZW96] Huai-bei Zhou and Lu Wang. Chaos in biomolecular dynamics. *The Journal of Physical Chemistry*, 100(20):8101–8105, 1996.

Abstract:

Our research has focused on the study of complex dynamics and on their use in both information security and bioinformatics. Our first work has been on chaotic discrete dynamical systems, and links have been established between these dynamics on the one hand, and either random or complex behaviors. Applications on information security are on the pseudorandom numbers generation, hash functions, information hiding, and on security aspects on wireless sensor networks. On the bioinformatics level, we have applied our studies of complex systems to the evolution of genomes and to protein folding.

Résumé :

Nous nous sommes intéressés à l'étude des dynamiques complexes et à leur utilisation en sécurité informatique et en bio-informatique. Nos premiers travaux ont porté sur les systèmes dynamiques discrets chaotiques, et des liens ont été établis entre ces dynamiques d'une part, et entre des comportements aléatoires ou complexes au sens de la théorie du même nom. Les applications en sécurité informatique concernent la génération de nombres pseudo-aléatoires, les fonctions de hachage, l'information dissimulée, et divers aspects de sécurité des réseaux de capteurs sans fil. Niveau bioinformatique, nous avons appliqué notre étude des systèmes complexes à l'évolution des génomes et au repliement des protéines.

SPIM

■ École doctorale SPIM 16 route de Gray F - 25030 Besançon cedex

■ tél. +33 (0)3 81 66 66 02 ■ ed-spim@univ-fcomte.fr ■ www.ed-spim.univ-fcomte.fr

UFC UNIVERSITÉ DE FRANCHE-COMTÉ