HABILITATION À DIRIGER LES RECHERCHES

Spécialité : Informatique Fondamentale

présentée par

Marie-France Sagot

L'espoir est cette chose avec des plumes – Algorithmes combinatoires et biologie moléculaire

Hope is the thing with feathers – Combinatorial algorithms and molecular biology

Soutenue le 11 Octobre 2000 devant le jury composé de :

Pr. Jean Berstel
Pr. Maxime Crochemore
Dr. Pierre Darlu
Pr. Christian Gautier
Pr. Dan Gusfield
Pr. Richard M. Karp
Pr. Eugene W. Myers
Pr. David Sankoff
Dr. Alain Viari

This work is dedicated

to all the persons, real and imaginary,
who helped me along the way,
with deep fondness

a uma pessoa amada

Heureux celui qui arrive à bon port,
Qui laisse derrière lui les mers et les tempêtes,
Dont les rêves sont morts ou jamais ne sont nés,
Et qui s'assied, et boit, à l'auberge de Brême,
Près de l'âtre, et il se sent en paix.
Heureux celui comme une flamme éteinte,
Heureux celui comme le sable des estuaires,
Qui a posé le fardeau, s'est essuyé le front
Et repose au bord du chemin.
Il ne craint rien, ni espère, ni n'attend,
Mais regarde, fixement, le soleil qui se couche.

À bon port.
Primo Levi.
À une heure incertaine. Louis Bonulumi (trad.).
Gallimard, 1997.

Ognuno sta solo sul cuor della terra
trafitto da un raggio di sole:
ed è subito sera.

Salvatore Quasimodo.
Ed è subito sera (Acque e terre 1920-1929).
Tutte le poesie. Gilberto Finzi (cura).
Arnaldo Mondadori Editore, 1995.

Thanks (sorry, there are many) to

La nuit n'est jamais complète
Il y a toujours puisque je le dis
Puisque je l'affirme
Au bout du chagrin une fenêtre ouverte
Une fenêtre éclairée
Il y a toujours un rêve qui veille
Désir à combler faim à satisfaire
Un coeur généreux
Une main tenue une main ouverte
Des yeux attentifs
Une vie la vie à se partager.

Et un sourire. Paul Eluard.
Poètes d'aujourd'hui.
Louis Parrot, Jean Marsenac (Eds.).
Seghers. 1944.

Jean Berstel for his unfailing support of computational biology since taking the responsability for the DEA d'Informatique Fondamentale at Marne, and for his great gentleness. I only wish (I have wished this since his arrival at the IGM) we may still someday be able to collaborate on a problem.

Maxime Crochemore it's been eight years now since we met for the first time. The PhD students I've been supervising are Maxime's students, we've been together in more projects and to more conferences than I can remember (and going to a conference with Maxime is always wonderful because he knows all the good places where to eat and museums to visit !) and all through this he has never once lost his discreet peaceful way and has come to represent a very strong and faithful support.

Pierre Darlu we would never have dared start getting involved with phylogeny and evolution-related problems without a guide; Pierre is the best possible we could have found not just because he knows so much but also because he is so kind and patient.

Christian Gautier incredibly, shyness kept me for years to get to know Christian and the people in his lab, which has been one of the most active in the computational biology community in France in a nicely discreet way. It took me a few weeks to gather the courage to try to invite him to sit in this committee. I am very happy, and very honoured indeed, that he accepted.

Dan Gusfield there are few other persons with whom I'd very dearly like to discuss some of the matters this memoir presents than Dan Gusfield and I'm very grateful that he was willing to read it. His work stood as an inspiration throughout, in particular his book which covers with such luminous clarity most of the basic string-related algorithms used in computational biology, takes time to motivate problems and provides an in-depth yet intuitively so simple to follow description of one of the most powerful data structures in a way that I very much doubt anyone will ever be able to improve upon.

Richard M. Karp my research life started with the first part of a paper by Dr. Karp, and is now coming back to, this time, the third part of the same paper. This is a computer science conference paper dating from 1972 that has been extensively cited in the literature ever since its publication and has inspired a number of other researchers. I feel a particularly strong attachment to it, not only because of the circunstances in which I first met it, but because it, together with Dr. Karp's other papers, contains some of the things – beauty and conciseness of the writing, power and elegance of the ideas – which make me always feel that algorithmics is poetry. To meet both in someone who, despite all his activities, is still willing to answer to mails of "débutants" and report on their work touched me very deeply.

Eugene W. Myers there are so many things one could say about the scientist but it is really about the friend I wish to speak and whom I'd like to thank. I talked to Gene for the first time in South America. For me, it is a nice symbol of what would follow. Gene has a heart that is oversized and never ashamed of his feelings. It was the same huge heart who greeted me at the University of Arizona in Tucson, where he was at the time and where we refreshingly talked about, science yes, but also everything else under the sun. This is quite rare and wonderful. Although I wish we'll be able to collaborate again in the future, it is the friendship that I treasure most and that is most worth of preservation.

David Sankoff whatever area of computational biology you decide to explore, you'll end up by finding a paper of David on the subject; and then you'll find it again and again because it set some landmark and is therefore mentioned in every paper that followed it; so you start feeling a bit of a paralizing awe, until you discover that behind all this, hides a terribly nice person with whom one would very much like to be able to interact.

Alain Viari we meet many persons in our lives, but there are some, because of what they are and also, sometimes, of what we were passing through when we first met them, who move and change us so very deeply that, later, thinking back on that life of ours, we see that it is punctuated by such meetings, that it is these meetings which give it shape and colour and depth. Alain is such a person but perhaps Paul Éluard and Primo Levi can express my feelings better than I "Il y a [...] au bout du chagrin une fenêtre ouverte/ une fenêtre éclairée", "Chacun de nous porte l'empreinte/ De l'ami rencontré en route./ Dans les bons et les mauvais jours,/ Nous les fous ou nous les sages,/ Chacun marqué par chacun."

Dominique Perrin this whole story began with Dominique, that is with the, initially fortuitous, encounter with a person that is so free in his mind that everything becomes possible, and the only thing that matters is the desire someone expresses of fulfilling a dream. That is unique, something I've never met before to this degree in anyone else.

François Rougeon I haven't seen him many times since my arrival at the Pasteur Institute, but each time gave me a great strength that lasted a long time. Once he mentioned to me the possibility of co-tutoring a student with him. I wanted to say that this gave me great happiness and that I haven't forgotten.

Laurent Bloch for his advice, trust and unfailing support of the efforts for developing a research activity in algorithmics and biology at the Institute, and for always greeting the students with generosity.

the students Anne, Eric (M), Estela, Julien, Laurent, Ludovic, Marina, Muriel, Nadia, Philippe, Stéphane, Sylvain, and the post doc, Sébastien, who'll soon join us. For their patience despite difficult conditions, all the times they've tried to lure me into a diner or a cinema (I promise I'll one day go see "Les ailes du désir") and the ease with which they've grown used to an excess of chocolate and Cambronne words vaguely thrown into the air or to the gods in dire moments.

my first home and all the Brazilian friends I still have there, most of which were students like me – Adriana, Alexandre, Diana, Eduardo, Eliane, Eugenio, Flavio, França, Heraldo, (little) Julia, (big) Julio, Kelly, Luciana, Marcelo, Orlando, Reinaldo, Renata, Ricardo, Silvia, Vivian – and the IME professors some of whom have become precious collaborators – Arnaldo, Carlinhos, Coelho, Cristina, Cyro, Imre, José Augusto, Kunio, Nami, Siang and, particularly, Yoshiko. They know how badly "sinto saudades".

the ABI and people around, who can dance, with joyful anarchy, to Polnareff and other quaint french songs well into the middle hours of the next morning: Alain, Anne (M), Antoine (M), Bernard, Christiane, Claudine, Dominique, Eduardo, Eric, Guillaume, Henri, Isabelle, Jacques (H), Joël, Laurent, Peggy, Philippe (B), Pierre, Sophie, Yves and many others I did not have an opportunity to know much. I miss them too, very much.

the IGM its relaxed brazilian-like atmosphere and incessant marvelously unserious and merry "pots" and collective lunches and visits to the Baron Rouge followed by diners which make you so often lose the last metro (as well as the straight line): Anne, Cyril, Dominique (twice), Elsa, Eric, Etienne, Frédérique, Gilles, Giuseppina, Jacques, Jean (twice), Marc (alias big Zip), Marie-Pierre, Mathieu, Maxime, Nicolas, Olivier, PA etc as well as Véronique, Clélia and Antonio when they are in Paris (but they're all very serious theoretical computer scientists).

the computational biology informal group that tried to be Anne, Frédérique, Michaela, Nuch, Valérie, Yves, and Emmanuelle who bore it all with warmth and a nicely wry sense of humor.

the SIS Laurent (see above); the persons with whom I share an office, Michel, Annick and Eric, for making of it a friendly and sheltered place; Marc, Nicolas and Eric for their gentle and sunny temperament and all the patient advice they've given to the students (and for always being hungry at 12); Catherine (see below) and Katja for their passion and friendship; Frédérique for the early friendship; Marie-Christine for her patience; Bernard for his gentleness; Stéphane for having helped me a few times track resistant and hard-to-die bugs, Olivier (alias Pépé), Louis who wakes me up sometimes in the morning :-), Gérard, Daniel and all the "habitués" of the coffee (and chocolate) distributor, Katja, Nicolas, Eric and the students (you'll have noticed that Eric appears three times !).

people from Pasteur in particular, for having greeted me to the Institute and/or for their advice, Pedro Alzari, Marie-Aline Bloch, Henri Buc, Stewart Cole, Antoine Danchin (see below), Bernard Dujon, Nicolas Escriou, Gabriel Gachelin, Alain Israel, Alain Jacquier, André Klier, Franck Kunst, Agnès Labigne, Gordon Langsley, Geneviève Milon, Xavier Montagutelli, Hidde de Reuse, Benoît Robert, Nicole Tandeau de Marsac, Michel Véron, Simon Wain-Hobson and, specially, Maurice Hofnung for his kindness.

the students of the Cours "Biologie pour l'Informatique" of the Pasteur Institute among the most motivated persons I've met; their enthousiasm has been most conforting.

the 1999 computational biology summer school at Lipari its organizers, in particular Raffaele Giancarlo, and the students, for two magical weeks in Italy (next time around, I want to be born an Italian).

the obi not Obi-Wan, Alec Guiness's role in Star Wars, but perhaps destined to the same unhappy fate as he ?

the Pisa algorithmics group Nadia (yes, it is the same Nadia as above) Pisanti, Fabrizio Luccio and all their colleagues, as well as Nadia's friends and boyfriend and family, for an extremely warm welcome and two delightfully relaxed weeks teaching every day but also eating pastaciutto, mozzarela, pommodori, gelatti and that most wondrous thing: pasta con fagiogli (macarrão com feijão ! – next time around, I want to be born an Italian from Napoli).

Nicole Bachala who's always been patient and very kind in explaining to me the customs of a biological research center.

Sophie Brouillet for all the gentle, funny, crazy, orthographically unorthodox (or uncatholic ?) electronic messages, a killing marguerita and an exhuberant way of dancing.

Olivier Carton for his always faithful friendship, strong and constant support that has been so important at various difficult points in these last years; I just wish I'll be able someday to help him realize that life can be beautiful if only one could manage to paint it so.

Antoine Danchin for having followed this work with interest since the beginning and for his nice thursday night discussion group.

Olivier Gascuel for his friendship and the marvelous adventure of Jobim (more on it below).

Christine Gaspin and the team at the INRA of Toulouse for having being able to create a luminous working environment where visitors feel at peace, and for their warm, friendly and very strong support even at a distance.

Jean-Louis Guénet for having greeted me into the Institute with a "meu coração é brasileiro". Even he who knows Brazil so well probably does not realize the effect that had on me.

Jacques Haiech for his interest in the work, his support of the PhD students, and a friendly talk in a train back from a workshop that came in the middle of a huge feeling of helplessness and solitude and that I don't forget.

Catherine Letondal for being able to accept a different point of view and for the hospitality of her heart.

the folk from the Pasteur Library for their constant kindness and efficiency (see Bruno, you weren't forgotten :-). It's always a real pleasure to go "up there" even if only for a stroll.

Frédérique Lisacek for her warm and wonderful complicity, her radiance and her "joie de vivre" which never forgets to be compassionate and mindful of others. She is also one of the only persons I know who can really dance, like Wordsworth's daffodils.

Nadia el Mabrouk I still hope very strongly that, despite the distance separating us now, we'll be able to work together on many subjects.

Laurent Mouchard to the friend with whom so much has been shared these last few years, who has always been present, at the end of a mail or in person, in all the bad and good moments.

Jean-Eric Pin and the people at Rouen for an "audition" which, incredibly, remains one of the best memories in my life (although they barely gave me time to dry my nose; it was, as always – sorry Thierry, cold and raining).

Joël Pothier and Henri Soldano for all those early years, avec beaucoup de tendresse.

François Rechenmann for a phone call and the offer of a friendly ear which moved me very deeply.

Giuseppina Rindone and Marc Zipstein for everything, but specially for that night we went to get Marc drunk before telling him some bad news and we got drunk about three hours before he did (did he ever ? we were certainly not capable anymore of judging) and then he laughed the news away !

Stéphane Robin, Jean-Jacques Daudin and Sophie Schbath for, I hope, a long and fruitful collaboration.

Pierre Rouzé I hope he will not hold it against me if I say that he is one of the "grandes personnalités" of our community, a sometimes fearful but cherished "grande personnalité" with a heart as big (and how we miss him when he cannot come to a conference, everybody starts falling asleep !). Collaborating with him has been a very rich experience, and I do hope we may go on doing so for a very long time. Pierre is also someone who does not fear mixing personal with scientific matters, so that one feels still very human even when talking about impersonal professional subjects.

Claude Thermes and Yves d'Aubenton Carafa it's been many years since we first met, and many memories, including a fond recent one of a diner where the two were feeling down and out and it was somewhat contagious so we spent hours talking about not very sunny subjects (there is an expression: "se sentir au fond du trou"; that night the "trou" probably reached Japan) but then, on leaving, Claude said that he felt good and as I looked at him in bemusement, added that it was great to be able to "talk between us". And so it was, and so nice to see such possibilities grow between persons.

Anne Vanet perhaps this is the only place where I can tell how much fun we had working on that *pylori* paper, from the day we ran at lightning speed to the closest café when the Pasteur server went down (we had just started a run of our algorithm – I reassure you, it happened to be only a coincidence; I reassure you even further, we knew it was not us ;-) it was just a temptation too hard to resist) to the day we laughed ourselves silly when the almost final draft of the paper fell on its own into the rubbish bin, and at many, many other occasions. Incredibly, the fun, and friendship, survived all the difficulties and absurdities and frustations of the outside world, grew stronger with it and, I hope, will continue serenely into the future.

the participants of various projects Capes/Cofecub with Brazil, Remag (INRIA), INRA Génoplante, and those still to come.

the bioinformatics and computational biology community in all french-speaking countries for their friendliness and warmth; it is always a real joy to meet them and discover new people arriving all the time.

all the persons who are participating in the Jobim adventure

- **Olivier** (see above) who had the initial idea
- **the local organizers of the first meeting** (http://www.lirmm.fr/jobim) in particular Gilles Caraux, a "taquineur" if ever there was one, but also Vincent Berry, Isabelle Mougenot, Eric Rivals, Véronique Sals-Vettorel and Philippe Vismara
- **the many persons who came to this first meeting** among them, but certainly not exclusively, those fabulous drum player "boys" (there were talks also)
- **David Sankoff** who enthusiastically, even before the first meeting took place, suggested to organize a next one in Canada
- **the organizers of the second, third, fourth etc meetings** (for the second) Christine Gaspin, Laurent Duret, Thomas Schiex (I hope a piece of beard will remain by the end of it :-) and all the persons at the INRA Toulouse Biométrie Lab
- **those who'll come to the second, third, fourth etc meetings** with fondness.

the guests and participants of the "Séminaire Algorithmique et Biologie" for their faithfulness and for periodically bringing a very much needed and treasured companionship.

my family the french side and the brazilian, in particular Themis; those who are still here and those who left but remain so strong in our hearts; my brother and his children and, specially, minha mãe.

most of all my father who through his shy love and his passion for books gave me a little of the world I could not have reached otherwise. May this find him in a bookstore somewhere, one of those where people can browse at leisure the books that have seduced their eyes sitting at confortable chairs in front of wide windows amid pots of flowers drinking tea or coffee or sipping a glass of wine, where they can come to discuss as much as buy, tell stories and unburden themselves of problems and heartsicknesses, laugh and cry.

Résumé

Ce mémoire présente un travail à un certain moment dans le temps. Une partie couvre les sujets traités depuis le DEA en 1993 et consiste en une revue de résultats obtenus, qu'ils soient publiés ou non, algorithmiques ou biologiques lorsque ces derniers existent. Une autre partie du travail présente une projection vers le futur.

Dans les deux cas, le travail a été fortement influencé par le désir de faire face à des problèmes biologiques d'une manière précisément formulée lors de toutes les étapes : de modélisation, de construction algorithmique et, finalement, d'analyse des résultats et retour en arrière. Les modèles adoptés s'appuyent sur la connaissance biologique que nous avons à l'heure actuelle, souvent limitée mais universelle autant que possible. Ces modèles sont, dans la mesure du faisable et du raisonable, indépendant des données. Ils représentent ainsi des théories *a priori* de processus biologiques que des algorithmes exacts devraient permettre de tester et, éventuellement, de revoir. Ces théories ne peuvent représenter que des visions simplifiées de la réalité mais elles devraient être correctes et la nature et limite de ce que les algorithmes permettent alors de tester bien définies.

Un regard à la plupart des articles en bioinformatique devrait convaincre l'algorithmicien ou le combinatoriste que la biologie pose des problèmes en même temps fort jolis et difficiles, dont quelques-uns étendent considérablement des domaines déjà connus, et d'autres en ouvrent de nouveaux. La phrase de Donald Knuth : "La biologie a facilement 500 ans de problèmes très intéressants à proposer aux informaticiens" n'est donc pas une exagération. Ce qu'il est plus difficile à voir est que ces problèmes deviennent bien plus intéressants, même d'un point de vue purement informatique, lorsqu'ils ne sont pas abstraits de ce qui leur a donné origine : une question biologique. Ce travail essaye donc de toujours commencer avec la biologie. Au début il y a ainsi, autant que cela est possible (ce n'est souvent pas facile à formuler), une question biologique. Parfois, la question est très spécifique; à d'autres moments, elle a une portée plus large. Ce mémoire est également divisé selon des thèmes généraux de recherche en biologie moléculaire et non selon les méthodes algorithmiques qui essayent d'étudier une partie de ces thèmes. Contrairement aux articles, les méthodes peuvent donc apparaître dispersées le long du texte car certaines s'appliquent, avec des modifications, à différents thèmes. Les divisions concernent l'expression et la régulation des gènes, le rôle et la relation de la géométrie et de la topologie à la séquence et à la fonction, la structure d'un génome et, finalement, la dynamique des génomes et l'évolution en général. L'ordre suivi est celui du sujet et de la pensée, même spéculative, pas celle de l'histoire.

Ce travail ne porte ni sur la vitesse ni sur la capacité à gérer d'énormes quantités de données. Bien que les jeux de données aient souvent été très grands et l'espace aussi bien que la vitesse soient des questions importantes qui ont été soigneusement prises en compte, par tempérament, ce travail traite de dentelle et concerne ainsi quelques-uns des détails, délicats et menus, de l'algorithmique et de la biologie.

Abstract

This memoir presents a work at a certain point in time. Part of it covers the subjects treated since its pre-PhD period of 1993. This consists in a review of obtained results, whether published or not, algorithmical or biological when these exist. Another part of the work represents a projection into the future.

In both cases, the work has been strongly influenced by the desire to address biological problems in a precisely formulated way during all its steps: of modelling, of algorithmical construction and, finally, of results analyses and feedback. The models adopted are based on current biological knowledge, often limited, universal whenever possible. They are, as far as feasible and reasonable, data-independent. They represent thus *a priori* theories of a biological process that exact algorithms should enable to test and, possibly, revise. These theories can only be simplified visions of reality but they should be accurate and the nature and limits of what the algorithms then permit to test well-defined.

A look at most computational biology papers should convince the algorithmicist or combinatorist that biology poses extremely nice as well as difficult problems, some of which considerably extend already known areas of research, others that open whole new domains. Donald Knuth's sentence that "Biology easily has 500 years of very interesting problems for the computer scientist" is therefore not an overblown statement. What may be less easy to perceive is that the problems become much more interesting, even from a purely computational point of view, when they are not abstracted from what gave origin to them: a biological question. This work tries therefore to always start with biology. At the beginning there is, as far as possible (this is not always easy to formulate), a biological question. Sometimes the question is a very specific one; at other times, it has a larger scope. The structure of the memoir follows broad molecular biology areas of research, and not the algorithmical methods that try to investigate part of these areas. Contrary to the papers which the memoir summarizes, the methods appear scattered along the text as some apply, with modifications, to different areas. Those covered are divided into gene expression and regulation, the role and relation of geometry and topology to sequence and function, genome structure and, finally, genome dynamics and evolution in general. The order followed is that of subject and of thought, however speculative, not of history.

The concern throughout the work is not with speed and the capacity to handle huge amounts of data. Although the data sets used are often quite big and space as well as speed important issues that were carefully considered, the work is, by temperament, mainly about needlepoint lace and some of the small and delicate details of both algorithmics and biology.

Contents

A Personal Note

"Hope" is the thing with feathers –
That perches in the soul –
And sings the tune without the words –
And never stops – at all –

And sweetest – in the Gale – is heard –
And sore must be the storm –
That could abash the little Bird
That kept so many warm –

I've heard it in the chillest land –
And on the strangest Sea –
Yet, never, in Extremity,
It asked a crumb – of Me.

Poem #254. Emily Dickinson.
The Complete Poems of Emily Dickinson.
Thomas H. Johnson (Ed.).
Faber and Faber, 1975.

It is folklore knowledge that biology offers interesting problems to the computer scientist, in particular to those with a combinatorial and/or algorithmical bend of mind. This is perfectly true but tells only part of the story. A few algorithmicists get interested in biology for the sake of biology itself. This is a less known fact.

This work is that of an algorithmicist at start who fell in deep love with the beautiful intricacies of biology to the point where she finds herself at a crossroads. Although her love for abstract matters and such ethereal subjects as algorithmical complexity and combinatorics has not diminished, on the contrary, she finds her mind increasingly caught by the desire to understand biological processes such as gene expression and regulation, genome structure and dynamics, evolution, metabolism and the role and relation of geometry and topology to sequence and function. So far, she has looked at the world of molecular biology through the lens of mathematical formalisms and algorithmical manipulations but wonders sometimes, in fact increasingly often, how far can that get her. Will the road ahead remain one, or will it have to split in two (and she with the road because she does not want to renounce any of her loves) ? Is the encounter of algorithmics and biology a true marriage (hopefully made in heaven) or just a meeting of sorts ?

Living in two, or more, worlds, as one who was born into two countries intimately knows, can be a difficult, at times frustrating, often exceedingly painful affair. When the worlds represent diverse scientific areas of investigation, the difficulty is related to both science and the intricacies of human relations and egos. In the end, one has to create her own route. Depending on who reads this work, the invented route may appear to lead nowhere. If you feel it is so, come to visit and talk about it, but please, let us, during the parenthesis of our meeting, try to put all differences and strongly held opinions aside; let us meet as persons quietly playing, without *a priori*, in a new world.

Chapter 1

Introduction

This memoir presents a work at a certain point in time. Part of it will cover the subjects treated since its pre-PhD period of 1993. This will consist in a review, hopefully from a reasonably detached point of view, not so much of what has been done, but of what has not been achieved, yet could still be. Obtained results, whether published or not, algorithmical or biological when these exist, will then be succintly presented in a raw, non embellished form. The desire is to progress in understanding. Potential readers are therefore encouraged to question, discuss and suggest.

Another part of the work will represent a projection into the future. It will speak of areas this author has started reading about, or even tentatively working with, and which she deeply wishes to explore in the years ahead. These areas cover a lot of ground, perhaps too much. This reflects not so much the fact that passion is difficult to contain, but more a personal impression that answers to problems within even a small area of investigation require deep understanding of the overall picture. This seems true in biology. It is also true as concerns combinatorial and algorithmical techniques.

In both cases, review of past and immediate present as well as projection into the future, the work has been strongly influenced by the desire to address biological problems in a precisely formulated way during all its steps: of modelling, of algorithmical construction and, finally, of results analyses (where good resolutions sometimes come close to drowning) and feedback. The models adopted are based on current biological knowledge, often limited, universal whenever possible. They are, as far as feasible and reasonable, data-independent. They represent thus *a priori* theories of a biological process that exact algorithms should enable to test and, possibly, revise. These theories can only be simplified visions of reality but they should be accurate and the nature and limits of what the algorithms then permit to test well-defined.

The text will refer details, specially but not only those of a mathematical or technical nature, to the papers, whether published, about to be or just submitted (see the appendix). This is a personal choice, and a risk I wished to take. The hope (that thing with feathers) is that this memoir will then, in almost its entirety, be succint and accessible to all (even the unavoidably more specialized parts that will be clearly indicated and may be safely skipped – a minimum knowledge of biology will, however, be assumed). It will have had any utility if it succeeds in being so, yet remains deep enough to encourage reflection and arouse reactions. Those interested, or needing to go further, can then dip into the papers.

Algorithmicists or mathematicians often come to biology with a method and then ask whether there is a biological problem their method could address. This work tries to always start with biology. At the beginning there is therefore, as far as possible (this is not always easy to formulate), a biological question. Sometimes the question will be a very specific one; at other times, it will have a larger scope. The structure of the memoir will also follow broad molecular biology domains of research, and not the algorithmical methods that try to investigate part of these areas. Contrary to the papers, the methods may then appear scattered along the text as some apply, with modifications, to different domains. The divisions concern gene expression and regulation (Chapter 2), the role and relation of geometry and topology to sequence and function (Chapter 3), genome structure (Chapter 4) and, finally, genome dynamics and evolution in general (Chapter 5). More "esoteric" parts (at current time and for this author) as, for instance, concerning metabolism and the cell are briefly mentioned in the final Perspectives (Chapter 6). The order followed is that of subject and of thought, however speculative, not of history. More developed chapters will therefore alternate with less so. Of course, the divisions adopted are sometimes artificial, as all divisions are. They reflect a state of thought that constantly hesitates and grows. Finally, the title given to a division covers a generally very extensive area. The division contents will be much narrower and center around a few precise initial questions. These will be stated at the appropriate place, often in the introduction to a chapter or at the beginning of a main section. They serve as guides for the investigation, they are not necessarily answered (yet). Questions for the future are distributed all along the chapter, or expressed in conclusion.

Some parts may appear highly speculative or confused, some questions asked ill-structured, useless or wrong. Yet the (sole ?) interest of writing memoirs such as this one, that will be read by a very small number of trusted persons, is to be able to ask even wrong questions, turn public mistakes and shortcomings and display confused thoughts, if all of these are honest. The work itself was almost always the product of collaborations with students and other researchers (their names are indicated at the beginning of each chapter) and is therefore presented in the first person of the plural. Doubts, questions and, specially, mistakes, are exclusively my own.

One thing this work is not about is speed and the capacity to handle huge amounts of data. Although the data sets used are often quite big and space as well as speed important issues that were carefully considered, this work is, by temperament, mainly about needlepoint lace and some of the small and delicate details of both algorithmics and biology.

Finally, I wished to end this introduction by saying that a look at most computational biology papers should convince the algorithmicist or combinatorist that biology poses indeed extremely nice as well as difficult problems, some of which considerably extend already known areas of research, others that open whole new domains. Donald Knuth's sentence that "Biology easily has 500 years of very interesting problems for the computer scientist" is therefore not an overblown statement. What may be less easy to perceive is that the problems become much more interesting, even from a purely computational point of view, when they are not abstracted from what gave origin to them: a biological question.

Chapter 2

Gene Expression and Regulation

Students whose work is presented in this chapter

Laurent Marsan for the part on structured models and for initial thinking on tiling and covering models

Julien Allali for the new version of a suffix tree

Anne Marcel for *Synechocystis* PCC6803

Nadia Pisanti for initial thinking on tiling and covering models

Co-workers and/or collaborators

Vincent Escalier, Joël Pothier, Henry Soldano and Alain Viari for the earlier work on models

Alain Viari for the earlier work on *Bacillus subtilis*

Anne Vanet for all the applications to biology and discussions on models

Agnès Labigne for *Helicobacter pylori*

Nicole Tandeau de Marsac and Hans Geiselmann for the work on *Synechocystis* PCC6803

Jean-Jacques Daudin, Stéphane Robin and Sophie Schbath for part of the work on evaluating models

Mireille Régnier for part of the work on evaluating models

Mathieu Raffinot for the part on better indexes

Yoshiko Wakabayashi for the algorithmical survey on pattern inference

People who gave advice or with whom this work was many times discussed

Maxime Crochemore and the people from the Capes/Cofecub project with Brazil and the Remag project with the Inria

Antoine Danchin and his team

Frédérique Lisacek

Pierre Rouzé, all his team at Gand and his collaborators in Louvain

People from whom we "stole" pieces of code

Eivind Coward (random sequence generator)

Thierry Lecroq (first version of suffix tree)

Alain Viari (library for reading sequences in fasta format and for manipulating stacks)

2.1 Introduction

It is not quite true to say that the interest in gene expression and regulation started with a biological question. It started (close to the beginning of this whole work) with an algorithmical question: how to efficiently and sensitively extract conserved motifs from a set of biologically related sequences ? As the work matured however, the biological questions came back as the driving forces behind everything that has been done so far concerning this subject.

At the current time, these (so far unanswered) biological questions are formulated thus:

1. what elements in sequence and structure are involved in the regulation and expression of genes through their recognition by, or of, a sequence or structure in another macromolecule ?

2. what is recognized and how does the recognition happen ?

3. in what context does that recognition happen ?

4. how do the various elements involved (when there is more than one) influence one another ?

5. what do different types and arrangements of elements tell about the genes they regulate and about cell life ?

This chapter will concentrate on DNA sequence elements and a discussion of just one aspect of DNA structure. Protein and RNA elements, specially those related to structure, will be considered in Chapter 3. Although the current chapter is concerned with "elements involved in gene regulation and expression", that is, basically, promoter sequences, regulatory sites and possibly enhancers/silencers, the algorithms may be applied to the identification of other DNA elements among which are splicing sites and splicing-related enhancers/silencers (section 4.2.2), restriction sites etc.

The model currently built to try to answer some of the above questions as concerns DNA sequence elements is introduced in section 2.3. Before that, the two kinds of problems related to the identification of regulatory DNA elements are briefly recalled. Section 2.4 then succintly presents the main algorithmical ideas and section 2.5 discusses some applications to various organisms. The chapter ends with critical remarks and perspectives on this subject. Although everything has been done so that the non specialist may understand the algorithmical parts (section 2.4 and subsection 2.6.2 – the specialist should read the papers), these may be skipped if desired.

2.2 Two Related Problems

There are basically two questions one must address when trying to search for known or predicted DNA sequence elements possibly involved in gene expression and regulation in whatever type of organism. One is the question of position: where are these sequences located in the genome (location prediction) ? The second question concerns identifying and modelling the sequences *ab initio*: what would be a consensus for them (consensus prediction) ? Both questions beg another one which is, what is the best way of representing or modelling such a consensus ? Is it under the form of a sequence, or a matrix, or yet a (stochastic or deterministic) automaton ? All these questions are of course related. However, one must be aware that the most efficient algorithms for extracting a consensus for sequence elements implicated in gene regulation often do not use the same representation as the most sensitive algorithms for searching for the positions of all the sequence elements belonging to a given family once a consensus for such a family has been identified.

The remaining of this chapter concentrates mainly on the second question: predicting consensi for regulatory related DNA sequence elements although its relation to the first one will be indicated. As an algorithmical problem, the second is known by the name of "motif extraction" or "motif inference".

Approaches for trying to solve the first question are surveyed in a paper co-authored with Anne Vanet and Laurent Marsan [197]. The paper reviews also motif extraction methods for DNA sequences, as does another one [199] more specifically aimed at presenting to the algorithmicist the combinatorial approaches developed by various authors including ourselves. Some of the statistical methods will be recalled below, but only where this serves the purpose of questioning and discussing our own combinatorial algorithms. The reader is otherwise invited to look at the papers.

2.3 The Model

2.3.1 Premises

Before describing the model, the basic premise upon which this model will be based must be discussed.

This premise is that sequence elements that have a biological function are conserved. Where nucleotides at other positions may suffer random and frequent mutations, functional binding sites preserve some properties that should enable to detect them. What properties are preserved and how conserved they must be varies depending on the molecule concerned and the function it is meant to fulfill. It seems to often be a matter of debate even among biologists.

This said, the model can now be introduced. It will be presented in a complete but informal (whenever possible) manner. Readers with a taste for formal definitions are referred to [136], [137] or [199].

2.3.2 Model

2.3.2.1 Terminology

We first establish the terminology that is going to be used in this and other chapters, at least as concerns our work. One of them will be the pair (*model*, *occurrence*). This will be preferred to the term motif which has often been employed in the literature to denote both something that is in a sequence, *i.e.*, a word, and the representative or representation of a set of such words (that possibly satisfy a certain property), *i.e.*, a consensus. We wish to keep both concepts separate. An occurrence will be "something" in the sequence and a model something (which may be more general than a consensus) that "represents" ("models") a set of occurrences. Models will thus serve to locate, as well as describe DNA conserved sites in a set of sequences. They may never be present in the sequences. They may also not correspond to true functional or binding sites, something motifs are sometimes meant to do for some authors [119].

If s is a sequence of length $|s|$ over an alphabet Σ, its individual elements (the letters composing it) will be denoted by s_i for $1 \leq i \leq |s|$. A non-empty word u in a sequence s is a segment (also called a factor by some authors [48]) $s_i s_{i+1} \ldots s_j$ for a given pair (i, j) such that $1 \leq i \leq j \leq |s|$. The empty word trivially is a factor of all strings. It is denoted by λ.

2.3.2.2 Single Models

For didactic reasons, let us start by introducing single models. A single model, let us call it m, is a word over the alphabet of nucleotides that is said to be present in a DNA sequence s with at most e errors if there exists at least one word u in s such that the minimum number of substitutions, insertions and deletions necessary to transform u into m is no more than e. This number is called the Levenshtein or edit distance [130] in the computer science community. When substitutions are the only errors allowed, which will be the case in the examples treated below, the same number corresponds to the Hamming distance. A word u at a Levenshtein or Hamming distance at most e from m is said to be an e-occurrence of m, or simply an occurrence when no ambiguity is possible.

For instance, the word of length 5 located at position 2 in sequence TACGACGACTGACG (positions in sequences start at 1) is at a distance 1 of models ACGAT (substitution of the last letter), AGAC (deletion of the second letter), ACGATC (insertion of letter T between the 4th and 5th letters) and of many others.

More flexible definitions of models and of similarity between models and words in the sequences may be used. A model can thus be defined instead over part or the whole of the degenerate nucleotide alphabet [176], including the wild card [172] (*i.e.*, the symbol corresponding to $\{\Sigma\}$ itself). Given such models, the definition of similarity between them and words in the sequences are then a special set-Hamming or set-Levenshtein distance where the distance between a word u in s and a model m that is defined over the degenerate nucleotide alphabet is the minimum distance between u and any word v that can be obtained from the set of oligonucleotides represented by m [172] [176]. Instead of a classical Levenshtein distance, one may also work with a weighted Levenshtein distance. This would allow to penalize mismatches differently. The meaning of e would then be as a threshold in lieu of a maximum number of errors.

For instance, the same word as in the previous example is at a distance of 0 from model AC[AG]AC where [AG] denotes A or G.

Models of whatever type are said to be *valid* if, given a set of N sequences, they have at least one occurrence in at least q sequences of the set, for $1 \leq q \leq N$ a fixed value. This value is called the *quorum* a model must satisfy to be of interest for our purposes.

2.3.2.3 Structured Models

Although the objects defined in the previous section can be reasonable, algorithmically tractable models for DNA binding sites, they do not incorporate any information concerning the relative positions of such sites when more than one participates in a biological process. It is common knowledge that these relative positions are often not random. For instance, the most frequently observed prokaryotic promoter sequences are in general composed of two parts, or "boxes", that come approximately 10 and 35 bases respectively upstream from the start of transcription. The two boxes, whose core sequences are six bases long, are therefore frequently situated 15 to 19 bases apart. The reason for this strict distance is that the boxes are recognized by the same protein, the RNA polymerase, in fact, a factor of the polymerase, the σ^{70}. RNA polymerases may have attached to them other σ factors. Prokaryotic promoters recognized by these factors are, for the same reason, commonly composed of two boxes although the distance between them may be different [90] and the content of the boxes may also vary. There may be other regulatory sites, whose distance from the promoter sequence is more variable but often not random. Activator binding sites are thus frequently located upstream from the promoter at a distance allowing for the activator to interact with the RNA polymerase. Repressor binding sites on the other hand may overlap the promoter sequences or come just downstream from it.

The case of eukaryotic transcription regulation is more complicated [201]. Promoter sequences contain, as for prokaryotes, one or two boxes recognized by a same protein, but there may be more regulatory sites, sometimes repeated, that are recognized by distinct proteins which interact with one another. The relative positions of such sites along a DNA sequence remains thus not always indifferent [201]. Finally, enhancer and silencer sequences are an important feature of eukaryotic gene regulation. These sequences may be located at very long distances from the gene start or downstream from it.

There is therefore a need for defining promoter, regulatory site or enhancer/silencer models as objects that take the possible cooperative characteristics of such sites into account (this may apply as well to some of the elements involved in splicing). This has the biological motivation just mentioned but presents also interesting algorithmical aspects: exploiting such characteristics could lead to algorithms that are both more sensitive and more efficient. Models that incorporate such characteristics are called *structured models*.

A structured model is an ordered suite of p "boxes", each box representing a single model which satisfies the quorum constraint as described in the previous section, while the p boxes together further verify the following: let $m_1, m_2, \ldots, m_p$ be the p boxes; given any occurrence u_i of m_i (for $1 \leq i < p$) in a sequence s of the set, there exists at least one occurrence u_{i+1} of m_{i+1} also in s such that u_i comes before u_{i+1} and the distance between the end position of u_i and the start position of u_{i+1} is within an interval $[d_{min_i}, d_{max_i}]$ for d_{min_i} and d_{max_i} non negative integers fixed by the user. An equivalent, symmetrical, property must be true also for all occurrences u_{i+1} of m_{i+1} in relation to the preceding box in the suite.

Put more simply, a structured model is an ordered suite of p boxes separated by strict distances. The intervals of distances allowed between boxes may all be the same, or they may be different between any two successive boxes in the model. An interval may be expressed by giving its minimum and maximum limits, or by saying that it centers around a value of d with an allowed margin of δ for δ a fixed number that may be as small or large as one wishes. The value of d is free to roam over a big(ger) interval. Of course, there must exist at least one d that is the same for all occurrences of a valid model. An example for the second case is given in Figure 2.1.

The value of e determining whether a model has an occurrence in a sequence may be fixed at different levels for the different boxes in a model. This allows to possibly ask that one box be more conserved than the others it is associated with. An overall maximum number of errors may also be established [136] [137] [197]. Such a global error rate permits to consider in a limited way possible correlations between boxes in a model.

Other possible local and/or global constraints one may wish to add to the models for some applications refer to the composition of the boxes. One may, for instance, determine that the frequency of one or more nucleotide in a box (or among all boxes) be below or above a certain threshold. This constraint should be used with great discernement as it introduces an arbitrariness into the search that has to be justified (although one could argue here that it is no greater than that introduced by the other parameters). For structured models composed of more than 2 boxes, one may also establish that a box i is palindromic in relation to a box j, or a mirror image of it, or identical to it (this may be useful for identifying some eukaryotic regulatory sites), for $1 \leq i < j \leq p$.

What is important to observe is that, given all these possible parameters and constraints, the algorithms (described below) then find **all valid models**. This is the essential point one must keep

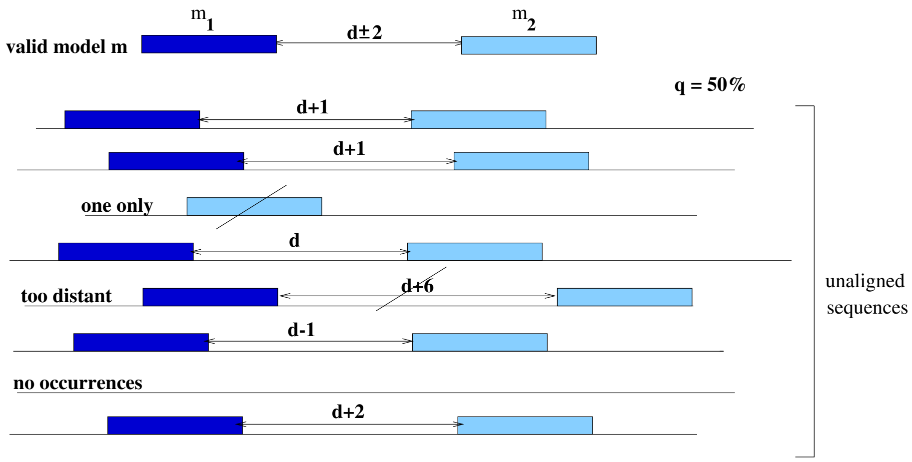

Figure 2.1: Example of a valid model with two boxes ($p = 2$).

in mind if one wishes to skip the next section and directly go to the applications (section 2.5).

2.4 Algorithm – Main Ideas

We developed, extended and refined various combinatorial algorithms over the years [136] [137] [167] [172] [197] [199] (the last two are surveys). One way of succintly describing them refers to a well-known problem in computational biology which is the comparison of two sequences, and to an equally well-known technique for solving it (as well as a whole range of other problems): dynamic programming [180].

Suppose we wish to calculate the Levenshtein distance between two sequences r and s defined over a certain alphabet Σ. This is the same as finding the score of an optimal (minimal in this case) alignment between the sequences using a system of score where a match costs 0 and a mismatch, insertion or deletion costs 1. This is easy to do once we observe that:

$$A(i,j) = \min \begin{cases} A(i-1,j-1) + score(r_i, s_j) \\ A(i-1,j) + score(r_i, \lambda) \\ A(i,j-1) + score(\lambda, s_j) \end{cases}$$

where $A(i,j)$ is the score of an optimal alignment of two sequences up to positions i and j respectively, λ denotes the empty character (an indel) and $score(x,y)$ is the score of an aligned pair (either 0 if $x = y$ or 1 otherwise).

Calculating the distance between r and s implies therefore filling in the cells of a matrix A of dimension $|r| \times |s|$. An additional column and line are needed to measure the alignment of r and s respectively with nothing in the other sequence. This represents the basis of the recurrence formula given above. For calculating a global distance, this basis is given by:

$$\begin{aligned}
&A(0,0) = 0; \\
&A(i,0) = A(i-1,0) + score(r_i, \lambda) \text{ for } 1 \le i \le |r| \\
&A(0,j) = A(0,j-1) + score(\lambda, s_j) \text{ for } 1 \le j \le |s|
\end{aligned}$$

When one of the sequences, let us say r, is much shorter than the other one, one may be interested instead in locating all positions in s where r may be found with less than a maximum number of errors. In this case, one is unconcerned with the presence of gaps at the start or end of s once r is aligned with it. The first line is then initialized with zero (indicating an alignment of r with s may start at any position), that is, $A(0, j) = 0$ for $0 \leq j \leq |s|$. Everything else in the recurrence remains the same, or almost the same as one may observe that not all cells in the matrix A need to be kept. Indeed, all those having a value above the maximum number of errors allowed cannot contribute to any future cell leading to an occurrence. They may be safely forgotten. Doing that implies performing what has been called *sparse dynamic programming* [57].

In our case, s will represent a concatenation of the sequences in a set from which valid models must be extracted. On the other hand, r is not known. It represents precisely the models m we wish to infer. The same technique may nevertheless be employed, except this time r may be described as a virtual trie $\mathcal{M}$ of all possible valid models that is exhaustively but efficiently traversed to determine all models m satisfying the parameters initially set (basically maximum error rate and quorum).

Finally, one may observe that long sequences over a small alphabet may contain many exact repetitions. One does not want to compare such repeated parts more than once with the potentially valid models. One way of doing that is using a representation of the string s that allows to group together some of the repetitions, that is, using an index of s. One such good index is given by a suffix tree $\mathcal{T}$ of s (see [48] and [93] for descriptions of suffix trees and of methods for building them).

From the initial, classical problem of comparing two sequences using dynamic programming, we have evolved to a problem of comparing two trees, one virtual $\mathcal{M}$, the other truly built $\mathcal{T}$, using the same technique of sparse dynamic programming. Occurrences of models are now nodes of $\mathcal{T}$ and no longer positions in s. An illustration is given in Figure 2.2.

This is the main idea. Other ideas are needed to do that as efficiently as possible. These are described in [136] and [137]. Earlier approaches that did not use suffix trees but could present other advantages are described in [167] [172]. Many more ideas will be required as increasingly more sophisticated models are needed to confront some of the problems met when applying the algorithms to real-life data. These are discussed in the Perspectives part of this chapter.

Let us observe that, among the constraints a model must satisfy, all are not equivalent in algorithmical terms. Length of models, maximum number of errors allowed, whether local to a box or global, quorum and existence of palindromic, mirror or repeated boxes in a model will result in a, sometimes substantial, pruning of the virtual trie of models. Box composition on the other hand, either local or global, can, in general (when it is given as a percentage), be verified only *a posteriori*.

All algorithms, whether using a suffix tree or not, have a time complexity that scales linearly with the total length of the sequences and exponentially with the number of errors allowed. In the case of structured models, the exponential factor is $k_{total}^{e_{global}}$ where k_{total} is the total length of all boxes in a model and e_{global} is the global error rate over all boxes.

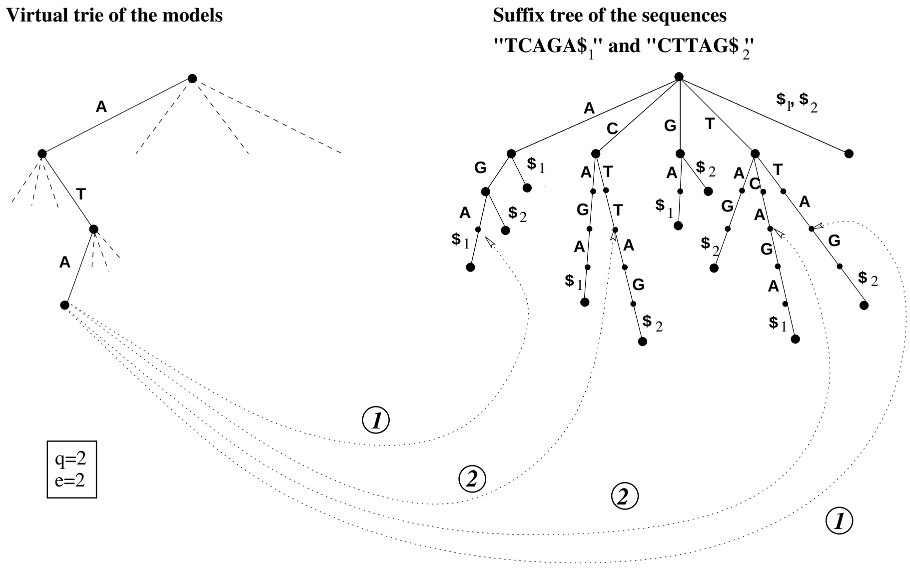

Figure 2.2: Extracting single models with a suffix tree – An example. Single models are extracted by simultaneously and recursively traversing (in a depth-first way) the (virtual) lexicographic trie $\mathcal{M}$ of all possible models of length k (stopping the descent down an arc and pruning the subtree rooted at the end node of that arc as soon as the quorum is no longer verified) and the (actually built) suffix tree $\mathcal{T}$ of the sequences. While traversing $\mathcal{T}$, up to e mispellings are allowed of the labels of the arcs in $\mathcal{M}$ corresponding to potentially valid models. In the Figure, model ATA has been spelled. It has suffixes AGA, CTT, TCA and TTA as occurrences with up to two substitutions. It is valid since 2 occurrences belong to sequence 1 and 2 to the other sequence (the quorum is 2).

2.5 Applications

2.5.1 Motivation

The aim of the present section is to illustrate and compare the use of the two main methods available for extracting promoter or regulatory site consensi or models from a set of sequences, one using a stochastic approach and the other a combinatorial one. The combinatorial approach is our own sketched above, while the stochastic method chosen is MEME [11] [12] [13] [16], currently one of the best publicly available programs in that category[1]. MEME's functioning is replaced in the context of other statistical approaches that are briefly recalled in section 2.5.2. Our own approach requires at the current time a further step for being of practical use. This concerns evaluating the statistical significance of the models found, *i.e.*, determining how "unexpected" each is given a certain context. The methods used for doing such evaluation are described in section 2.5.3.

Both algorithms are applied to the identification in bacteria of promoter sequences (mainly those recognized by the RNA polymerase σ^{70} factor) and some regulatory sites (recognized by

[1]We are currently implementing an improved version of Lawrence's earlier algorithm [127] (see section 2.5.2) using basically the same idea. This will allow us to have a better control of the algorithm's behaviour and will be used for our own purposes (see 2.6.2.4 and 3.4.2.1).

dimeric proteins). We use two kinds of data. One represents "clean" data in the sense of data containing experimentally determined transcription starts, sometimes promoters. The other kind of data employed is genomic data. In this memoir, we concentrate attention in both cases on data coming from bacteria about which a lot is already known, including what concerns regulation (*Escherichia coli* and *Bacillus subtilis*). The reader interested in applications to data coming from a less well-studied bacterium (*Helicobacter pylori*) may have a look at [196]. Work is also in progress on *Synechocystis* PCC 6803, *Mycobacterium tuberculosis* and *Mycobacterium leprae*.

2.5.2 Statistical Approaches – A Quick Reminder of Expectation-Maximization and Gibbs Sampling for Motif Extraction

Expectation-Maximization (hereafter abbreviated into EM) is a two-step iterative procedure for obtaining the global maximum likelihood parameter estimates for a model of observed (in this case, regulatory) data: a collection of binding sites of length k whose positions are not known in a set of unaligned sequences, one site per sequence [39] [104] [128]. The collection corresponds to a matrix presenting four lines, one for each nucleotide, and as many columns as there are positions in the site the matrix is modelling. Each cell $M(a,i)$ of the matrix (where a is a line number and i a column) corresponds to the weight assigned to nucleotide a at position i of a putative binding site (represented by a word in a sequence). The algorithm involves calculating expected parameter values that describe the data (*i.e.*, the matrix of putative sites), then maximizing the likelihood of obtaining these values (*i.e.*, refining the matrix by maximizing its relative entropy). The two steps are repeated until the parameters that best explain the data are obtained, or until a fixed maximum number of iterations is reached. Cardon's algorithm as well as a more recent one by Frech *et al.* [71] [72] are further able to handle variable spacers between parts (two only) of a promoter or regulatory sequence. However, the process can be tedious (as there are too many parameters to estimate), unless *a priori* information about the second element and/or spacer is available.

Uncarefully chosen initial data sets may lead EM to converge inappropriately to a local maximum. This problem is in part solved by using a Gibbs sampling approach [127]. The maximization step is replaced by one that increases the likelihood of observing the data with a certain probability only. The chances of reaching the maximum augment with the number of iterations. Lawrence *et al.* state that the algorithm provedly converges, although again not always to the global maximum.

Stormo's and Lawrence's approaches have the advantage of being able to define a binding site from scratch: the first iteration usually starts with randomly chosen positions for the sites. Roth's method [163] is an extension of Lawrence's Gibbs sampling algorithm [127] that, among the many possible DNA site alignments, considers only those that match less than 1% positions inside ORFs. Motifs are therefore required to be selective. Another modification of Lawrence's method is introduced by Frech *et al.* [71] [72]. In order to account for the fact that DNA/protein interactions are influenced by nearest neighbour contacts either directly or via the DNA conformation, Frech *et al.* average the consensus matrix column indices over a window of three positions (the "best" size of the window was chosen based on simulations).

These methods assume, for computational as well as statistical reasons, that each sequence in the data set contains one, and only one site whose length is known. The statistical reasons include the fact that calculation of the likelihood of a site (given by its relative entropy) does not take into account the absolute number of occurrences but only the relative frequency of occurrence of each of the nucleotides, at each position in the site. Under a uniform distribution model, a perfectly conserved motif occurring in only a few sequences would, for instance, have a greater likelihood than an imperfectly conserved motif occurring in nearly all sequences.

Both assumptions, of existence and unicity of the site in each sequence of a set, are removed in more recent versions of Lawrence's algorithm (not publicly available) and in a statistical method called MEME [11] [12] [13] [16]. Although more frequently used for extracting conserved motifs from a set of protein sequences, MEME may also be applied to DNA sequences. The algorithm is basically an EM method with more parameters to estimate, among them, the number and length of sites as well as the number of sequences where each site is present. The time requirement of the method is higher than Stormo's or Lawrence's. The latter have an expected linear behaviour with the size of the input data whereas MEME execution time scales quadratically with that size. Possibly because of that, it cannot at current time handle data sets containing more than 100.000 characters (http://meme.sdsc.edu/meme/website/).

2.5.3 A Currently Necessary Added Feature to Combinatorial Approaches – Model Statistical Evaluation

2.5.3.1 Motivation

The properties verified by the models identified by the combinatorial algorithms sketched in section 2.4 will often be not selective enough to discriminate true sites from false positives. The statistical significance of a model has therefore to be evaluated. It is important to observe that this evaluation will not *per se* eliminate models (as may be the case with statistical approaches when extracting motifs), but will serve the sole purpose of classifying models. One may then choose to print the whole list of valid models, or present only those with a "high enough" probability of not being due to chance alone.

Various ways of performing such evaluation are possible (see Brazma *et al.* [31] for a review). These are recalled in [197]. The two principal methods for assessing the statistical significance of models having occurrences in a set of sequences are those based on a theoretical model of the sequences and those corresponding to a data shuffling (*i.e.*, a simulation) approach.

Among the methods from the first category, none appeared completely satisfactory when we started our work. Indeed, as errors are authorized between models and their occurrences (even if only substitutions), there is a need for approaches that are able to either deal with such errors, or handle multiple exact motifs statistics. Those available [157] [158] [195] appeared too computationally intensive. A further complication comes from the fact that our models may be composed of an ordered suite of more than one element separated from its neighbours by intervals of distances. Finally, our interest is in assessing the statistical significance of the number of occurrences of a model per sequence, *i.e.*, of distinct sequences having at least one occurrence, and not of the total number of occurrences. Only Tompa [195] seemed to deal with this situation when we started, but his method is applicable exclusively to small single motifs (typically of length 5).

2.5.3.2 Data Shuffling Approaches

Evaluating the significance of a model by using the data shuffling approach [121] appeared therefore initially more appropriate. This can be done in the following two different ways. The first consists in performing a χ^2 test (with one degree of freedom) on two contingency tables, one corresponding to the number of occurrences observed per sequence, the other to the number expected under the null hypothesis (random sequences having same k-mer composition) [152]. Alternatively, one may assume that the number of occurrences in random sequences follows a normal distribution and estimate the significance of the Z-score corresponding to the number observed [197]. The random sequences are currently being generated using an algorithm due to Eivind Coward [47] based on an idea by Kandel [120]. The model for random is a Markov model of order k obtained from the initial data set. Since the sequences in the input data are generally short, k is at most 4 or 5. In the examples below, k is equal to 1, that is, we preserve the composition in dinucleotides.

2.5.3.3 Approaches Based on a Theoretical Model of the Sequences

As the work progressed however, we started collaborating with a number of statisticians. This led them to elaborate two new different ways of estimating the statistical significance of models such as ours. Both are based on a theoretical model of the sequences. The mathematical formulae are described in [155] [156] and in a paper in preparation by Stéphane Robin *et al.* (see the appendix). They are being tested and only partial results are known at the current time. Preliminary ones will be discussed for one of the approaches in the paper by Robin *et al.* Roughly, they show that the order obtained for models (based on their statistical significance) is approximately preserved whether one uses simulation or a theoretical model. The theoretical model seems to provide a finer estimation of that significance even though this is for now calculated using an approximation. Further analyses will be done to determine how much faster it is in relation to simulation (this last can be very slow, typically on the order of a day for two to three thousand models; this is due not to the shuffling but to the model matching step of the calculation and might therefore be improved with the elaboration of smarter multiple pattern approximate matching algorithms).

2.5.4 Testing on Experimentally Known Data

2.5.4.1 Data

The data consisted in two test sets of well established *Escherichia coli* and *Bacillus subtilis* sequences containing an experimentally determined transcription start or, sometimes, promoter. These sets (called Sets Ec and Es respectively, E denoting "Experimental") were obtained from Ozoline [145] and Helmann [102]. In both cases, the sequences are aligned on the start of transcription. Sequences in Set Ec stop at that point, sequences in Set Es contain 20 more bases downstream from the transcription start. The *E. coli* experimental set contains 441 sequences of average length 79, for a total of 35115 nucleotides. The *B. subtilis* set contains 131 sequences of average length 99, for a total of 13099 nucleotides. The GC content is 41% for *E. coli* and 32% for *B. subtilis.*

2.5.4.2 Results for the Stochastic Algorithm

The MEME algorithm [11] [12] [13] [16] used was the version provided by the authors at their laboratory's web site (http://www.sdsc.edu/MEME/meme/website/). There are few parameters to set. Since MEME does not permit spacers, we asked for short motifs (asking for long ones produces the same results - not shown) present zero or one time in each sequence of the set. Three motifs only were asked for, asking for more identifies no significant new ones (not shown). When more than one motif is sought, MEME identifies each in turn, usually starting with the most significant whose occurrences cannot then participate in another motif.

The results are presented in Figure 2.3. This shows, for each set and each motif, the latter's width and expected number of occurrences in the training set (top), the information content diagram

```
Escherichia coli Ec

MOTIF 1  width = 46  sites = 185.2

         bits 2.2
              2.0
              1.7                                      *
              1.5                                      *
Information   1.3                                  *   *
content       1.1                                  *   *
(10.0 bits)   0.9                                 **   *
              0.7           *                     **   *
              0.4           *                     **   *
              0.2          *** *                  ** ***   ** *
              0.0 ----------------------------------------------

Multilevel        AAATAAAAGTTGACATTTTTTGGAGTAAATGGTATAATGCGCCCCC
consensus         CTTATTTCT  TGACAACGCGCCCAATTTGTT  A C T CGGGGA
sequence             C  CTA   C   CACGAATGTC CGCC       A A    T
                      GGC         A C  T    C

MOTIF 2  width = 43  sites = 11.0

         bits 2.2
              2.0
              1.7
              1.5                              *
Information   1.3                  *         * **        *
content       1.1     *           **         * **        **
(32.9 bits)   0.9     *       *   **   *    * * ***  *  **** *
              0.7  *  *       *** **  ** *** * ***  ******* *
              0.4 ** *******  *** ** ******* * *** ******** *
              0.2 ***************************************** *
              0.0 -------------------------------------------

Multilevel        TAATTTATTCACAAACTTTCATTGAAATGGTGGATTAAATCAC
consensus         CCGC G GCACTCCGA  GA  GT  C T  ATGCA     GA
sequence                         T  C             A

MOTIF 3  width = 52  sites = 27.4

         bits 2.2
              2.0
              1.7
              1.5
Information   1.3
content       1.1
(19.0 bits)   0.9 *                                        *
              0.7 *              *         *               *       *
              0.4 *   * *      * * *      **   * * *   *   *   *  **
              0.2 *  ******* *** * **  *  ***  ****** *** ******* ***
              0.0 ----------------------------------------------------

Multilevel        CGGTGCGCCATTTTTCTTGAAATTATGCACCTTCTATGGATAAACTAGAAGA
consensus         TTTACACAAGCCACGGAATTCGACT CACAT ATGCA TTATCTTATTCTAG
sequence           C CA  GGTGA  C C  GT GA  TG  A GGAT  CC      GC   C
```

```
Bacillus subtilis Es

MOTIF 1  width = 30  sites = 121.0

         bits 2.2
              2.0
              1.7
              1.5
Information   1.3 *                           *
content       1.1 *                       *   *
(11.6 bits)   0.9 **                     **   *
              0.7 ***                    ** * *
              0.4 ***                    ** ***
              0.2 ******              ** *******
              0.0 ------------------------------

Multilevel        TTGACATTATTTAAAAATATGATATAATA
consensus           TTATAATAAAATTT TGT  G  A   G
sequence             C        CC    AG  T

MOTIF 2  width = 30  sites = 21.8

         bits 2.2
              2.0
              1.7
              1.5
Information   1.3                           *
content       1.1                *          *
(16.1 bits)   0.9                *         ****
              0.7                *  * ** ******
              0.4   *   * *    *** ***** ******
              0.2   ** **** ** ********* *******
              0.0 ------------------------------

Multilevel        GCTATAATTCATTAAAAATTTTACAAAAAG
consensus         TTCTACCGT   G T C CC CTAC    C
sequence          A   G  A G                   A
                  A

MOTIF 3  width = 22  sites = 10.8

         bits 2.2
              2.0
              1.7
              1.5         *
Information   1.3   *     *
content       1.1   ** *  *
(16.0 bits)   0.9   ** *  *
              0.7 **** * ** *    * *   *
              0.4 **** * *****  ********
              0.2 **********************
              0.0 ----------------------

Multilevel        AAAAGGTTGCTTTAACAAAGTG
consensus         TT  A CC TC AT  GC  C
sequence              C
```

Figure 2.3: Motifs found by MEME in Sets Ec, Es. Short motifs were asked. Each was requested to have zero or one occurrence per sequence.

and the multilevel consensus sequence (see the on-line manual at http://www.sdsc.edu/MEME/meme/website/ for details).

The two motifs identified first by MEME in Sets Ec and Es correspond in most cases to the promoter sequences experimentally determined for both organisms. In *E. coli*, 185 sites are expected occurrences of the motif, of which most are accurate. In *B. subtilis*, the number of expected occurrences is 121 (covering 92% of the 131 sequences as against 42% for *E. coli*), the majority correct.

The multilevel consensus sequences are similar in both cases to the consensus given in the literature [154], TTGACAx(15-19)TATAAT, with few positions strongly conserved in the site at -35 in *E. coli*. On the other hand, the three most significant positions in the TATA-box for *E. coli* (two for *B. subtilis*) are those known to be best conserved (first T and A and last T) [154]. The motif found in *B. subtilis* further presents two conserved letters just upstream from the TATA-box. These letters (a T followed by a G) had been previously considered as forming part of an extended TATA-box [154]. The distance between the two contiguous sets of highest significant positions identified by the motifs is of 17. This corresponds also to what was already known about promoters in these two organisms. Since MEME looks for single motifs and is unable to deal with motifs composed of two parts separated by a spacer, TATA-boxes for which the length of the spacer is different are either not identified or are misaligned on one or the other box.

The second next "best" motif found in *B. subtilis* has expected occurrences in only 16% of the sequences and although a fuzzy TATAAT forms part of the motif, this seems to correspond to the TATA-box in 13 cases out of a possible 20 only. The following two motifs extracted from Sets Ec and the last one from Es have few expected occurrences (covering less than 6% and 7% of the sequences respectively) and are unaligned in relation to the start of transcription. It is therefore unlikely that the motifs represent a biological site, although the information content in one case (second motif for *E. coli*) is high (32.9 bits).

2.5.4.3 Results for the Combinatorial Algorithm

The combinatorial algorithm was run at least twice on each set (Ec and Es), looking first for single models, then for models comprising two parts separated from one another by a distance whose value was allowed to vary inside a narrow interval. When it was run more than once in either case, this was because no statistically significant models were found at the higher quorums initially chosen. In the case of structured models, a range of intervals was tried, from (4-6) to (24-26) by increments of one.

The results concerning single models as well as parameter values (length of models, quorum and maximum number of substitutions allowed) are presented in Table 2.1. Results about structured models are given in graphical form. Figure 2.4 thus shows two graphics where the intervals of distance allowed between the two parts of a structured model are plotted against the statistical value obtained by the most significant model. Parameter values are indicated in the legend.

Whether one searches for single or structured models, the only ones identified as statistically significant in Sets Ec and Es (using either a χ^2 test or a Z-score (result not shown)) are the TATA-box and the site at -35 (separately – Families 1 and 2 in Table 2.1 – or together). Figure 2.4 further shows that the only intervals d plus or minus one for which significant structured models are found correspond to d equal to 17 for *E. coli* and between 16 and 21 for *B. subtilis* with the highest significant one (TTGACAx$(d-1, d+1)$TATAATAA) at d equal to 16.

Among the single models, it is worth observing that one of those found in *E. coli*, TCACATT, may either refer to an approximate form of the consensus for the site at -35 found in the literature (TTGACA), or to the second part of the CRP binding site (TCACA) [23] [45] [128] [182]. The first part would be missing because the sequences are too short (average length 80 bases). This motif is not identified by the MEME algorithm, probably because it contains only a few bases.

2.5.5 Testing on Genomic DNA

2.5.5.1 Foreword

Obtaining reasonably good results on "clean" datasets may be misleading as it is not indicative of the true limits of an algorithm. Indeed, one of the potential limits of any method is related to its (in)capacity to handle noise above a certain level, where by noise we mean various things:

Escherichia coli

Family 1	ATAATGCGG	34	3.90	24
	TATAATGCGC	23	1.60	19
	ATAATGCGC	30	5.75	17
	TGTGTATA	47	15.85	16
	ACAATGCGC	24	3.85	15
Family 2	GTTGACAC	36	10.80	14
	TCACACTT	36	11.10	13
	TGACACTT	38	12.35	13
	GCTGACA	64	31.55	12
	ACACTTAT	41	14.95	12
	TTGACACT	37	13.75	11
Family 3	TTACGCTG	39	12.80	14
	TGTTACGC	39	14.45	12
	TTTACGCT	44	17.85	11
Family 4	TTTTTTTTC	23	5.40	11
Family 5	GCGCCCC	44	18.85	10

Bacillus subtilis

Family 1	TATAATA	94	48.06	32
	GTATAAT	74	34.34	24
	TGTTATA	66	34.96	15
	TTTTACA	76	45.96	13
	ATAATAT	82	52.52	13
Family 2	GTGACA	68	39.76	12
	TTTACAA	75	48.56	10
	GTTGAC	66	40.10	10
	TTGACA	92	66.34	10
	ATGATA	10	80.26	10

Table 2.1: Simple models found with the combinatorial algorithm in Sets Ec and Es. Column 1 gives the names of models of length greater than 6 found. At most one substitution is allowed. The quorums used each time are indicated beside each Set. Column 2 corresponds to the number of sequences where at least one occurrence of the model is found, column 3 the same for shuffled versions of the sets (1000 shuffles were performed) and, finally, column 4 to the χ^2. Only models with a probability of happening by chance below 10^{-2} are shown. Models are (manually) grouped by family and only the most statistically significant representatives in each family are shown.

- sequences in the set from which a motif is to be extracted that do not contain the motif (coupled with an ignorance of how many such sequences there are);
- presence of more than one family of conserved motifs in the set of sequences (the diverse families partitioning the set in an exact or approximate way);
- sequences that represent a context that is too large and/or wrong (examples of such are sequences which incorporate both non coding and coding regions, or contain non coding regions that are not "seen" by the transcriptional machinery – more on this important point in section 2.6.1.2.5);
- (this is for combinatorial algorithms only) composition of the sequences, *e.g.*, if the sequences are very AT-rich, then a lot of models containing As and Ts will be identified at high quorums even though they may have no biological meaning whatsoever (alternatively, if the bias in composition is very strong – as, for instance, in *Falciparum plasmodium* whose chromosome II is over 80% AT-rich – everything may be eliminated during evaluation and, indeed, this may pose a problem even for statistical approaches).

We therefore show below preliminary results obtained on "unclean", genomic data, that is, on data extracted from a whole genome (corresponding approximately to the non coding regions upstream from a gene). This concerns data for which at least some of the motifs that should have been found were already known (*B. subtilis* and *E. coli* again).

2.5.5.2 Re-Exploring "Known" Territory

2.5.5.2.1 Data

The data was extracted from the whole genomes of *B. subtilis* (ftp://ncbi.nlm.nih.gov/genbank/ genomes/bacteria/ Bsub/) and *E. coli* (http://mol.genes.nig.ac.jp/ecoli/) and consisted in two sets (called Gs and Gc respectively, G denoting "Genomic") of non coding regions located between two divergent genes, that is, between genes transcribed in divergent directions (one on each strand). Each non coding region is therefore non coding on both strands and appears twice in the set, once as a sequence read from the genomic one as publicly released, the other as the same sequence reversed and complemented. Sequences having less than 40 bases were eliminated and only up to 330 nucleotides before the start of translation (as annotated) were initially kept. The first and last 15 bases were then discarded from sequences in both sets. This eliminated the Shine-Dalgarno sequence as a potential motif. Gs contains 1,062 sequences for a total of 196,736 nucleotides while Gc has 1,148 sequences and 226,928 nucleotides. The GC content is 40% for *E. coli* and 34%

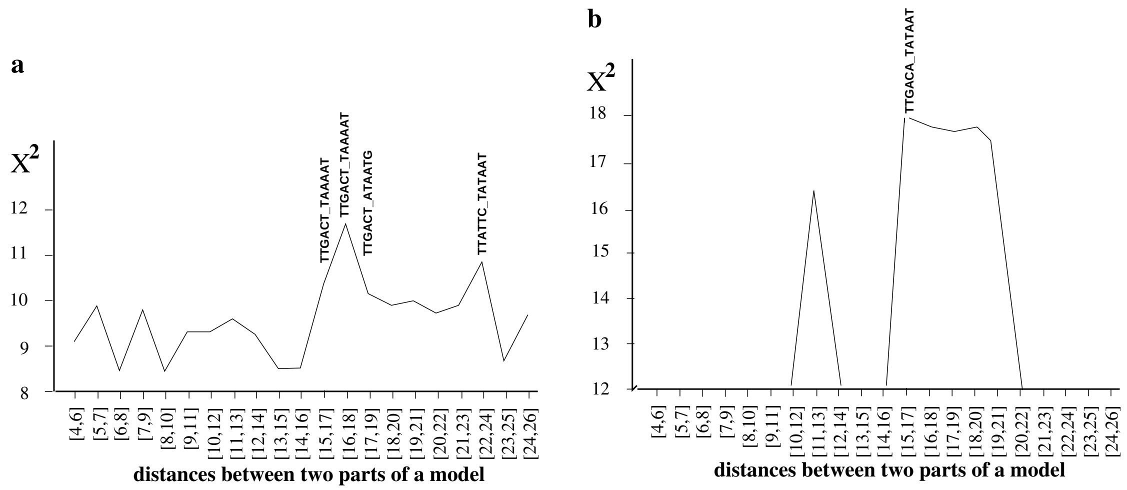

Figure 2.4: Intervals of distance allowed between the two parts of a structured model plotted against the statistical value obtained by the most significant model found using the combinatorial algorithm in Sets Ec (Fig. 2.4a) and Es (Fig 2.4b) with up to 1 substitution permitted and a quorum of 2% for the *E. coli* sets of sequences and 4% for *B. subtilis*. For this, a χ^2 and a Z-score were calculated between the number of occurrences observed (allowing for the same maximum number of errors) in both sets against that observed on average in shuffled versions of each. The most pertinent model identified for each interval is shown above the curves. This is done only for the intervals located at or near a peak.

for *B. subtilis*. Since MEME can not handle sequences having total length greater than 100.000 nucleotides, for comparative purposes, we created from Sets Gc and Gs two smaller sets that represent one sequence in three, randomly chosen among those in Gs and Gc. We called these sets GRs and GRc respectively (R standing for "Random").

2.5.5.2.2 Results for the Statistical Algorithm

The motifs found by MEME in either Set GRs or Set GRc have low information content and/or very few strongly conserved positions and are almost certainly not significant (Figure 2.5). The only exception is the first motif identified in *E. coli* which corresponds well to what we know about the CRP binding site [23] [45] [128] [182]. The first motif in *B. subtilis* is not, despite appearances, related to the Shine-Dalgarno sequence (whose position from the start of translation is known).

Observe that another of the motifs identified in *B. subtilis* (the third) is rich in Thymines. Even if such a motif were biologically pertinent, it would be impossible to say whether it corresponds to a well-defined signal or simply to a region rich in Thymines that could help in the recognition of a binding site somewhere in the neighbourhood.

2.5.5.2.3 Results for the Combinatorial Algorithm

Working with sets extracted from a genome introduces a lot more noise. The models found, in particular those structured, verify therefore lower quorums which may make analysing their statistical significance more difficult. The results presented in Table 2.2 (Set Gs) and the graphical plots in Figure 2.6 show however that it is possible, in the case of *B. subtilis* at least, to identify the same models (basically the known sites at -10 and -35 – this corresponds, in Table 2.2, to Family 1 and Families 2 and 3 respectively) as in the cleaner, experimentally characterized sets. As is the case for MEME, single significant models rich in Thymines (Family 4 in Table 2.2) are also identified by the combinatorial algorithm, as well as one composed of a few As followed by, or intermingled with a few Gs (Family 6). The same observation as at the end of the previous section could be made as concerns these models except that this time, the two together reappear in the structured model GAAAAAx(6-8)CTTTTT which corresponds to another peak in the Figure 2.6b. This model is also identified in GRs (not shown); however, the promoter itself appears less clearly in GRs than in Gs (see [197]).

A few models are identified in Set Gc, among them what could perhaps be the site at -35 (Family 1 in Table 2.2) with a C in place of the first T of the consensus given in the literature. The TATA-box is not found and the possible biological meaning of the other models must be further investigated. Surprisingly, no structured model may confidently be put forward as being significant, even at a very low quorum (2% – Figure 2.6), except for intervals of distance with a peak around 6 where the motif TGTGAx(6)TCTCA for the CRP binding site is identified as the only, highly significant model (χ^2 of 31 – slightly less for TGTGATx(5-7)TCTCAT). The results obtained for set GRc are the same (not shown).

```
Escherichia coli GRc

MOTIF 1  width = 30  sites = 111.

         bits 2.2
              2.0
              1.7
              1.5
Information   1.3              *          *
content       1.1            * *        * *
(12.3 bits)   0.9            * *        * *
              0.7            ***       ****
              0.4        **  ****      *****
              0.2       *** *****      ***** **
              0.0 ------------------------------

Multilevel        AATTTTAAATTGTGATCTAAATCACATATT
consensus         CGAAGATTTA  C   AGTGT  G  ATAA
sequence          G G             TAGT         G


MOTIF 2  width = 39  sites =128.7

         bits 2.2
              2.0
              1.7
              1.5
Information   1.3
content       1.1
(12.0 bits)   0.9                                  *
              0.7  *                    *          *
              0.4  ** * * *           * *      **  * *
              0.2  **** ****    *  *  ***    ****  * ***
              0.0 ---------------------------------------

Multilevel        TAATTAATATACACAATTTTTTTTTTATTTTCATGATTT
consensus         AC AATTATCTAGT TAAAACAAGAATAAA AT TCAAA
sequence          C    C    CGTA  C GG G A  C    T  C


MOTIF 3  width = 12  sites = 181.4

         bits 2.2
              2.0
              1.7   *
              1.5   *
Information   1.3  **
content       1.1  **
(6.2 bits)    0.9  **
              0.7  **
              0.4  **   *
              0.2 ********  **
              0.0 ------------

Multilevel        CGCCCTGTTTGC
consensus         T  GACTCCGTG
sequence             AGG   ACT
```

```
Bacillus subtilis GRs

MOTIF 1  width = 12  sites = 308.7

         bits 2.2
              2.0
              1.7
              1.5
Information   1.3
content       1.1      **
(5.6 bits)    0.9      **
              0.7      **
              0.4     ***** *
              0.2    ********
              0.0 ------------

Multilevel        AAAAAAAGGAGG
consensus         TGG    ACGAA
sequence          CT     T   T


MOTIF 2  width = 22  sites = 54.4

         bits 2.2
              2.0
              1.7
              1.5
Information   1.3
content       1.1
(8.3 bits)    0.9
              0.7  **    *             *
              0.4 ***  * **       *    *
              0.2 ****** *****  * **  **
              0.0 ----------------------

Multilevel        GGCAGCAGCCCGTGCAGAGCGA
consensus          C T  C AAA GAATACCGAG
sequence             G  T     A   CTAAC


MOTIF 3  width = 43  sites = 173.2

         bits 2.2
              2.0
              1.7
              1.5
Information   1.3
content       1.1
(11.3 bits)   0.9
              0.7  *                         *
              0.4 **  *           **         *  *
              0.2 *****     *   *****  ******** ** * * *  *
              0.0 -------------------------------------------

Multilevel        TTTTTTCATAATTTTTTTTTTTTTCTTTTTTTATTTAATATTT
consensus         CCCCCAACACCAACCACACACC CTCA  ACCTCAACTATAGA
sequence               CTT TT CA         G     C    G T C   C
```

Figure 2.5: Motifs found by MEME in Sets GRc and GRs. Short motifs were asked. Each was requested to have zero or one occurrence per sequence.

2.6 Critique and Perspectives

2.6.1 Critique and Perspectives on the Model

2.6.1.1 Modelling "Recognition"

One may wonder why a model such as described in section 2.3 was established instead of a number of others one could have imagined. Although the model chosen presents obvious relations to some known biological facts (*e.g.*, that promoter sequences are often composed of two distinct elements and regulatory sites tend to come in clusters [2]), some of its other characteristics may seem more questionable.

One of these concerns how conservation, *i.e.*, similarity between a model and its occurrences,

	Escherichia coli				Bacillus subtilis			
	CCTGAC	573	424.60	39	TATGATA	627	407.05	91
	CTGACG	587	439.70	38	TATCATA	615	403.00	84
Family 1	CTGACA	701	557.00	36	TATAATAA	445	277.95	58
	TCCTGA	671	538.70	30	TTATTATA	439	273.85	57
	CCCTGA	575	446.80	28	TACTATA	491	325.70	54
	GTCAGG	576	412.10	47	ATGATAA	617	477.10	36
	TGTCAG	702	555.00	37	ATGAGAA	500	377.15	29
Family 2	CATCAG	711	574.60	32	TGAGAAA	520	417.85	19
	CGTCAG	580	443.60	32				
	ATCAGG	689	553.30	32				
	TTTTCTG	553	419.20	31	TGACAAA	510	405.20	21
	CTCTTTT	464	348.50	25				
Family 3	TTTCTGT	469	357.05	23				
	TTTTCAG	531	416.40	23				
	CTGATTT	498	384.60	23				
	CAGAAAA	539	410.55	29	CCTTTTT	638	413.05	95
	CTGAAAA	525	407.75	24	CCTTTTC	496	291.10	84
Family 4	GAGAAAA	460	359.50	19	CTCTTTT	600	391.80	81
	AGATAAA	512	415.60	16	CTTTTCT	613	410.90	77
	GTGAAAA	509	414.75	16	CTTTTTC	652	451.20	76
	CCGGCA	604	480.80	26	CCTGCT	478	339.45	38
	CAGGAA	717	595.55	26	CCCCTT	488	349.35	37
Family 5	CAGGCT	609	490.45	24	CTCCTT	644	512.70	32
	CAGCAG	589	471.00	24	TCTCCT	609	477.65	32
	GCCAGG	519	405.80	23	CCGCTT	484	359.95	30
					CTCCCT	457	336.45	29
					AAAAAGG	629	418.70	83
					AAAAGAG	599	393.85	79
Family 6					GAAAAGG	496	301.75	75
					GAAAAAG	651	450.70	75
					GAAAAGA	604	430.55	56

Table 2.2: Simple models found with the combinatorial algorithm in Sets Gc and Gs. Column 1 gives the names of models of length greater than 6 found. At most one substitution is allowed. The quorums used each time are indicated beside each Set. Column 2 corresponds to the number of sequences where at least one occurrence of the model is found, column 3 the same for shuffled versions of the sets (1000 shuffles were performed) and, finally, column 4 to the χ^2. Only models with a probability of happening by chance below 10^{-2} are shown. Models are (manually) grouped by family and only the most statistically significant representatives in each family are shown.

is defined. Among the many definitions that were tried ([165] [199]), none appears completely satisfying. Others were adopted by different authors. As we saw, the statistical approaches for instance (with some variants between them – see section 2.5.2) use a definition of "model" which makes it correspond to a set of occurrences (with no reference to an external object as is the case with our models) such that the likelihood (or relative entropy) of the matrix formed by an alignment (without gap, or allowing for gaps in a recent development due to Rocke and Tompa [161] [162]) of the occurrences is maximal. Such a definition reflects how "surprising" is the model (in the sense of the set of occurrences it corresponds to) in relation to its background (the remaining of the sequences).

This begs the question: "what backgroung should one consider ?", which leads to two other, more important ones: "how does the recognition happen ?" and "what is recognized ?".

In a certain sense, statistical approaches, such as MEME, do not even address such questions. They try to model instead what, on average, is "seen", once this has been seen. They present the further inconvenient of being sometimes too sensitive to even a small change of context. We report in [197] a case where by adding to each sequence in a set under examination (average length: 300 bases) the 15 bases coming just downstream in the genome was enough to lead MEME to identify slightly different motifs in a different order. Since no new motif was found in the 15 bases added, the only explanation possible for such behaviour is that the algorithm was mislead by a change in context, even though the change was quite small. All approaches are sensitive to the choice of background. Combinatorial ones will in particular be sensitive to it when evaluating models, an almost always compulsory step for now. Statistical approaches are more sensitive in that they make decisions, sometimes very early in the search for motifs, that combinatorial algorithms do not have to make. Current statistical approaches employ also only a very simple probability model (Markov of order 0). Given all these factors, it is disturbing how well they can work, perhaps not so much for nucleic acid sequences but certainly for proteins (see the next chapter).

Current combinatorial approaches on the other hand work with a model of how recognition

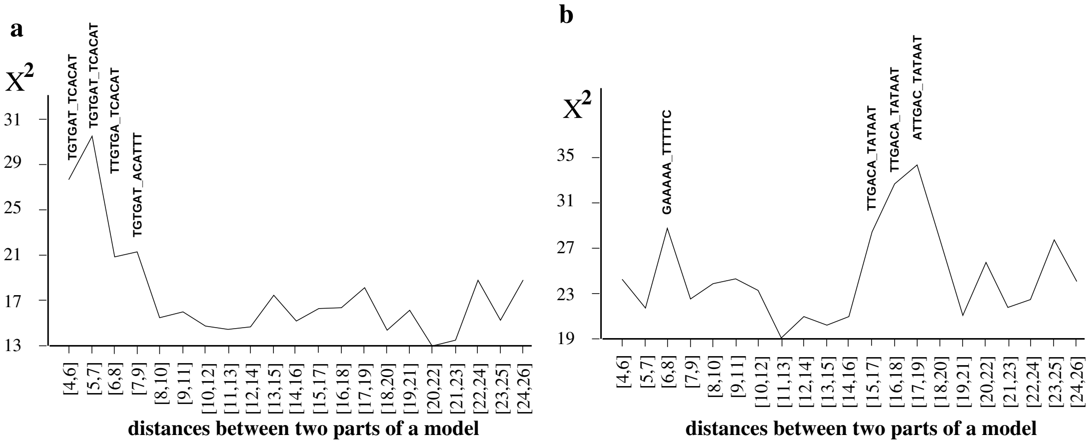

Figure 2.6: Intervals of distance allowed between the two parts of a structured model plotted against the statistical value obtained by the most significant model found using the combinatorial algorithm in Sets Gc (Fig 2.6a) and Gs (Fig 2.6b) with up to 1 substitution permitted and a quorum of 2% for the *E. coli* sets of sequences and 4% for *B. subtilis*. For this, a χ^2 and a Z-score were calculated between the number of occurrences observed (allowing for the same maximum number of errors) in both sets against that observed on average in shuffled versions of each. The most pertinent model identified for each interval is shown above the curves. This is done only for the intervals located at or near a peak.

happens that most certainly does not take into account all the elements that are involved in it. Some may be difficult to model (for instance, elements of the DNA structure if these intervene as is believed, although probably no one knows yet for sure how). Other elements may be impossible to consider when all we have as information is the sequence. Included among such elements is the cell within which all these processes (transcription but also translation etc) are happening. Attention has also been called many times (see [50] to mention just one case) to the fact that a DNA sequence forms a compact object. Geometry considerations that are not related to the DNA structure but to the way DNA must be squeezed into very small places may have important consequences for the recognition process. Besides being a tight place, a cell is also a crowdy environment whose participants are for the most part invisible at the sequence level. Often they are not even known or are only very partially understood.

The models adopted by current combinatorial approaches are therefore probably a huge simplification of reality. Even without incorporating information that is not at present time, or may ever be obtainable, such models may be improved. This is worth the effort, at least in this author's (strong) belief, because of two main advantages combinatorial approaches present. The first one comes from the fact that the algorithms are, by definition, exact. They may thus be powerful exploratory tools even though they simplify greatly biological processes. In particular, the algorithms allow to "say something", derive some conclusions even from what is **not** identified because we know that, modulo some parameters, they exhaustively enumerate all objects they were built to find. Whether they indeed find something (for instance, a motif initially thought to exist with a certain characteristic) or not, we may thus, by understanding their inner workings, emit biological hypotheses on "how things function". This is often possible also with non exact approaches such as most of the statistical ones, but is more difficult to realize. The second reason why it is worth trying to improve the models used by the combinatorial approaches is that, though limited, they mimic better how recognition really happens. At least, they are meant to do so.

We therefore quickly examine next diverse ways in which the models adopted by combinatorial approaches such as ours may be ameliorated. Not all of them introduce new biological considerations but all seem important to us. Some have started being explored. We then say a few words of the improvements they call for at the algorithmical level, as well as other betterments that are possible even for current models. Perhaps it is worth mentioning once and for all that the need for such betterments is not just to satisfy the algorithmicist from a purely theoretical point of view. The improvements obtained may also have an important influence on the type and number of "questions" that can be asked.

2.6.1.2 Future Models Worth Exploring ?

2.6.1.2.1 Tiling Models

When conservation is not strong (as will often be the case), unless it is possible to add generic biological information to the model used for promoter, regulatory site or enhancer/silencer sequence, one is obliged to use weakly constraining parameters in the search. This has the consequence that an often great number of models are found. It may make the interpretation of the results almost as hard to do as was solving the initial problem.

In a recent paper, Parida *et al.* defined what they called a *basis* of motifs. This is defined as a set of motifs that are both maximal in a certain sense and irredundant [147]. The motifs she considers roughly correspond to models defined over the alphabet of nucleotides (or amino acids

– the method is aimed in fact for inferring motifs in proteins) plus the wild card (the latter is forbidden at the first and last positions of a motif). The definition of a motif is not exactly that of our model but for our purposes here it is enough to consider that this is the case. In particular, motifs must also satisfy a quorum.

A maximal motif m in that case is such that no other motif, possibly shorter, having exactly the same set of occurrences (modulo a possibly non null shift which applies to all occurrences) may be obtained from m by substituting the letters in m corresponding to nucleotides with the wild card.

A motif m is redundant if it is maximal and there exist maximal motifs $m_1, \ldots, m_p$ such that the set of occurrences of m is exactly the union of the sets of occurrences of $m_1, \ldots, m_p$.

Parida *et al.* then propose an algorithm in $O(n^3 \log n)$ [2] for inferring all irredundant motifs from a set of sequences of a total length n. Notice that the length of the motifs needs no longer be a parameter of the algorithm.

This is an elegant and very useful idea that we believe can be improved in terms of the algorithm complexity and may be extended to more complex models such as the ones with which we are currently able to deal (including with errors allowed). The definition of a basis may have to be modified in some cases to obtain more pertinent models, we therefore decided to give such models a new name. We called them "tiling models" because in some sense they summarize (tile over) the space of all valid models. A preliminary form of tiling models was considered in [172], but we believe the results obtained in that paper can be improved (work in progress).

2.6.1.2.2 Covering Models

Contrary to tiling models, the ones we define in this section have a strong biological motivation. As mentioned in section 2.5.5.1, at least two potential sources of "noise" may greatly increase the difficulty in extracting a consensus for all promoter and regulatory elements from a set of sequences. One of these sources comes from the fact that some sequences in the set may contain no such element, the second from the presence of more than one family of conserved elements in the set.

We are currently able to extract only the most frequent or best conserved elements present in a set of sequences (this is true in some measure of all approaches). We are thus still a long way from being able to identify **all** the elements involved in the regulation of the genes present in the set under consideration.

One idea that we wish to explore consists in extending the concept of a conserved model to that of a family of conserved models, isolated or cooperating with others, which "completely explain" a set of sequences. This last could represent all the non coding sequences upstream from the genes in a genome. The family of possibly cooperative conserved models on the other hand could stand for all the elements involved, at the level of the sequence and, perhaps, of a predicted structure, in the regulation of the gene expression of the considered organism.

In a slightly more precise way, the problem could be expressed in the following (informal) way. Given a set of sequences, one must find a set of $r \geq 1$ structured models (the value of r is unknown at start) such that:

- each model has an occurrence in at least one sequence of the set;

[2] The notation $O(n^3 \log n)$ or, in a more general way, $O(f(n))$, where $f(n)$ is a function of n, roughly indicates that the number of operations needed will be a constant times $f(n)$.

- distinct models among the r may have occurrences in the same sequence but the number of times this may happen is smaller than a threshold value t (possibly t may be 0: there is no "sequence overlap" of the models);
- all sequences are covered by (at least) one model (strictly one in case the threshold t is 0).

The notion of "cooperating" models must also be developed further. It will require the introduction of a concept which is briefly exposed in the next section.

2.6.1.2.3 Meta Errors

This is the concept of a meta-error, that is of an error that represents no longer a point mutation inside a site, but the insertion, deletion or replacement of a whole site, that is of a whole box in a structured model.

2.6.1.2.4 Correlated Errors

The possibility of cooperation, or correlation, inside one box of a model for a promoter or regulatory site must also be examined and possibly exploited, either *a posteriori* or during the search for models. It is important to become able to detect whether some positions mutate always together, or alternatively, whether two positions never mutate at the same time. This is true in other situations such as, for instance, in models for sites or structurally conserved elements in proteins.

2.6.1.2.5 Contextual Models

As we argued in section 2.5.5.1, another limitation of our models and of those used by others is that we do not take into account the context in which the recognition of a binding site by a macromolecular complex happens. Yet a context is implicitely present in all methods that evaluate the statistical significance of a model and, *a fortiori*, in all statistical approaches for inferring motifs. Models or motifs are thus ordered, and sometimes eliminated, according to how "surprising" they are against a certain background. As we saw in section 2.5.2, some authors (as, for instance, Frech [71] [72]) define the background to be represented by the ORFs in a genome. The results we have obtained so far seem to indicate that this is not a correct assumption. Another possibility is that the "surprising" factor would be in relation to a window representing what the macromolecular complex is capable of "seeing" as it slides along the sequence (or the sequence along it) or looks for the place where it must land on the DNA sequence. Such a window could, for instance, be associated with the DNA spatial conformation, in particular but not exclusively, with the regions of the DNA that are more susceptible of being opened, and therefore of containing the sites important for the functioning of the organism. Studies concerning these hypotheses have been initiated by some authors [74]. We wish to pursue them.

If the window hypothesis is correct, we might consider introducing the following constraint in our search for structured models: a valid model would now be one for which there exists a window of a fixed length such that the model has one, *and only one* occurrence inside the window in at least q of the sequences in the set.

2.6.2 Critique and Perspectives on the Algorithms

2.6.2.1 Building Better Indexes

We are currently working on obtaining more compact indexes. A lot of work has been done already in this area, in particular by Kurtz [81] [125], Farach [61] and Munro [140]. This will become crucial when dealing with proteins, but is also an important issue where DNA sequences are concerned, specially the very long ones we are often led to treat.

Already, as indicated in [136] [137], we need to put in the tree only suffixes that are long enough to contain the models being sought, and no longer. Tree $\mathcal{T}$ therefore needs to contain only the factors (prefixes of suffixes) of the text that have a length between the minimum and maximum a model may have. Furthermore, when the sequences that must be indexed into a suffix tree are over a 4 letter alphabet and are very long, the tree will be full up to a certain level k. We are therefore exploiting the idea of a mixed structure composed of a simple index or hash table for prefixes of length k of the factors in the sequences. Each cell in the table points then to the root of a tree that represents the remaining part of the factors having a same prefix (the one stored in the cell) [59]. Preliminary results have shown a gain in space of a factor of 3 to 10 to an already sparse implementation of the suffix tree for a set of sequences (this set is typically quite big, containing up to 2000 sequences for a total of 3 to 4 million nucleotides). These results must however be more finely analysed and can, quite possibly, be still improved upon.

2.6.2.2 More Efficiently Exploring the Solution Space

We wish also to explore other indexes, such as compact DAWGs [49] that might allow for a more efficient exploration of the space of models, especially of structured models with two or more boxes. The adaptation to our problem (where spacers of variable lengths are allowed between boxes and, indeed, insertions and deletions might, theoretically, be permitted inside a box) is not straighforward.

It seems obvious also that we are still doing too many times a same, or almost the same operation. Short of being able to build indexes for approximate suffixes, or any other type of factors, of a text (a possibly non solvable problem), we may be able to obtain more efficient ways of exploring the solution space without unduly increasing the space required. Particularly frustrating is also the fact that indexes are good for putting things together, but terrible for keeping them ordered in a way useful to our purposes. The problem of recovering this "useful order" has been faced by some authors for a few specific problems (*e.g.*, exact tandem repeats [94] [190] or exact repeats separated by a bounded gap [32]), but not for those we have been concerned with. The results obtained might nevertheless be exploited for our cases (structured models but also models for some RNA structural elements we sketch in the next chapter).

2.6.2.3 Developing Algorithms that Explore Part of the Parameter Space in a Single Run

Statistical approaches have the (big) advantage over combinatorial ones that they do not require a user to fix the quorum a model must satisfy before the search can start. Indeed, they often require no parameters at all. Of course, the estimation may then be wrongly made, but when exploring completely unknown territory, this may be a great help. The alternative is to run many times the (combinatorial) algorithm with different parameters and to do that often enough that an intuition may develop on how to set them (this may take a long time and possibly have to be often repeated as surprising new organisms come along one's path).

Another approach we are therefore exploring concerns obtaining a way of trying various sets of distinct parameters in a single run of the algorithm. This problem has started been tackled in the case of the distances separating the diverse boxes inside a same model [136] [137]. The most troublesome parameter to fix is, however, the quorum. Of course, one may always fix it at very low levels but this then creates another problem, which is, an exponential increase of the number of valid solutions most of which are biologically worthless.

2.6.2.4 Combining Combinatorial and Statistical Approaches

At the current time, there is no going around the need to evaluate the models extracted from a set of sequences in terms of their probability of being due to chance only. As is developed in [31] and recalled in [197], there are many ways of doing that evaluation.

One we have not yet explored consists in using the results of the model extraction algorithm as seeds to an EM or Gibbs sampling algorithm such as developed by Lawrence [127] [128]. This had been suggested to us by a referee of one of our papers ([175]). We are feeling strongly encouraged in that direction by more recent versions of the EM or Gibbs sampling approach [161] [162] [194] [198], coupled with a different method that uses more complex models of sequences (Markov models of higher order) [4]. There is in particular one example we have met in the course of our work where it would be interesting to see whether using higher order models changes the results obtained. This concerns a motif, the TATA-box, that should be present[3] in the non coding sequences of a genome, *Synechocystis* PCC6803, and is indeed found to be quite frequent yet has an extremely low statistical value despite the fact that the sequences are less AT-rich than, for instance, the same in *B. subtilis*.

2.6.3 Regulation and Pathogenicity

There are a number of things we wish to do in the next two years in terms of applications of our currently existing or yet to be elaborated methods. For now, we shall almost certainly focus on prokaryotic regulation mostly, although an eye will be kept to potential applications to eukaryotes.

One first question that has appeared increasingly fascinating as the algorithms have been applied to different genomes concerns why even organisms within a same cellular domain, such as bacteria, have kept part of their regulatory system conserved between species while other parts have diverged ? Is this solely explainable by differences in the environment where such bacteria live and to which they have to respond ? Or does conservation reflect what is essential in the system and diversity what are just left-overs from previous, possibly redundant systems ? Is there such a thing as, given an environment, an uniquely optimal regulatory system ?

A biologist[4] has also recently mentioned to us the case of a bacterium, *Mycobaterium leprae*, which apparently has a shrunk genome in comparison to that of a very closely related organism, *Mycobacterium tuberculosis*. The "discarded" parts seem to include whole families of promoters.

[3]The reason this motif should be present is because the σ^{70} factor region that is believed to recognize the box is very well conserved in relation to that of *B. subtilis* and *E. coli* where, as we saw, a model corresponding almost exactly to the TATA-box motif is found among the most statistically significant.

[4]Stewart Cole, from the Pasteur Institute.

Have the genes that such promoters regulated also disappeared ? And why have promoter families disappeared ?

Such examples seem to point to the importance of defining the "identity card" of each bacterium (habitat, environment, GC content, whether the bacterium is a pathogen etc) as this is probably essential for understanding its regulation system, and vice versa. For instance, *Helicobacter pylori* lives in the stomach, has a weak GC content and possesses a σ^{80} that is different from the σ^{70} of *Escherichia coli*. One or many among these factors may explain why the consensus for the promoter sequences associated with the σ^{80} we have suggested [196] does not resemble that for *Escherichia coli*. On the other hand, the rather strict conservation observed in the promoters for the same σ factor in *Bacillus subtilis* could be explained by the bacterium's way of life, and by the fact that it is gram-positive. What factor is really influential could be, for instance, tested on bacteria such as *Shigella flexneri*, *Vibrio cholerae* and *Escherichia coli* which belong to different families but have the same pathogenic function. The ideal would be to become able to define a correlation network between the regulatory sequences found by our algorithms and the different factors intervening in the "identity card" of a bacterium, in particular of a pathogen. By comparison of the sequences that appear conserved upstream from genes in pathogenic and non pathogenic strains of a same bacterium, we may also, perhaps, be able to reveal what (if anything) distinguishes the regulation system of the two strains.

2.6.4 Using Other Types of Information

Finally, it appears important to start considering two types of data we have so far not tried to integrate into our methods of analysis of regulation systems. The data are related to what is beginning to be known about, on the one hand, gene synteny between related species or strains, and, on the other hand, expression levels, metabolism and regulation networks. Concerning this last, we thus wish to study the relations between the function of a gene and its form of regulation, and to better exploit the ways of using what may be discovered or hypothesized about the one to better characterize and understand the other.

Chapter 3

Role and Relation of Geometry and Topology to Sequence and Function

Students whose work is presented in this chapter

Sylvain Lhullier for the later work on RNA

Ludovic Le Botlan for a three months implementation work on trying to improve an earlier version of the algorithm for inferring small RNA structural motifs

Estela Maris Rodrigues for initial ideas on RNA structure comparison/prediction and tree algorithms

Co-workers and/or collaborators

Alain Viari for the earlier work on small RNA structural motifs, and for all the work on proteins

Christine Gaspin (RNAs – just started)

Nadia El Mabrouk for RNA structural matching (just started, in progress)

Maurice Hofnung and his team for the work on repeated palindromic units

Henry Soldano for the earlier work on proteins

People who gave advice or with whom this work was many times discussed

Maxime Crochemore

Joël Pothier (proteins)

People from whom we "stole" pieces of code

Alain Viari (all the model evaluation part in the algorithm for proteins plus the algorithm for the alignment of protein profiles with sequences - this last co-written with Vincent Escalier)

3.1 Introduction

It is an accepted fact that biological functions are supported more by structure than by sequence, although the extent to which this is true remains unknown and is a matter of controversy. Relations between objects may also be at least as important in determining function as structure. Structures however, and how they are connected to sequence and, possibly, function, are what will interest us in this chapter. By structure is meant the spatial arrangement, local or global, of the atoms composing a macromolecule. We shall be concerned mostly with RNA and protein structures.

The biological questions asked at the current time mostly bear on the relation between sequence and structure, and between "deduced" structure and function. We indicate in parenthesis whether the question refers to RNAs or proteins. Observe that the first follows on a preoccupation of the previous chapter. This and the other main questions are:

1. (RNAs) what (small) elements of structure are involved in the large scale function of a genome, or the expression and regulation of genes, and how to identify such elements ?
2. (RNAs) how to infer a structure from a sequence ?
3. (RNAs) how to compare two (or more) RNAs using both the sequence and (deduced, or predicted) RNA structure ?
4. (proteins) can we identify conserved structural elements by looking at physico-chemical conservation at the level of sequence elements ? Stated in other terms, how far can homology takes us in elucidating a structure from a sequence ?

A further question may be asked in the case where we possess known, possibly partial, structural information (experimentally determined contacts or atom coordinates) concerning one or more of the molecules (RNAs or proteins) we wish to study. The question concerns how to incorporate such information into the search for conserved parts, the comparison of two or more macromolecules or the determination of the structure from one or more sequences.

The questions in this chapter have only just started being addressed and the reader should therefore be aware that what follows indicates more work in progress or perspectives for the future than any real accomplishment yet. In particular, we have so far considered RNA structures as static objects only and have not yet tried to deal with its dynamic and energetic aspects (see [206] for a recent general survey).

Section 3.2 treats the case of RNAs while section 3.3 deals with work done so far on proteins. The chapter ends with perspectives and future work (section 3.4.2).

3.2 RNAs

3.2.1 Foreword

Although a number of methods have been elaborated in recent years to, either describe and then search for an RNA structure, or compare two RNA structures, or, finally, predict such structures at their secondary or, less frequently, tertiary levels, none is completely satisfying for various reasons.

In the case of the search for a structure from a given, high-level description of it, few approaches have tried to take both the sequence and the structure information into consideration. Among those which interweave spatial and sequence elements, the two most successful either cannot deal with some of the more complex correlations between the elements potentially forming a structure [76] (although this seems to be less true of later versions of the algorithm [135]), or may sometimes be slow [25]. The question of which level of detail to allow in the description of a structure, and how to describe it externally (as input to the algorithm) and internally (so as to allow for a most efficient search) is already not trivial.

It is this problem, we call *RNA structure matching*, that we have started addressing. We are doing it in two steps. The initial one is concerned with identifying as efficiently as possible simple RNA structural elements. These include palindromes and pseudo-knots (in the original paper, this was extended to allow finding mirror images as well as triple helices in DNA sequences [113]). It is an interesting problem in itself and has important applications a few of which are briefly discussed. The second step concerns trying to solve the general structural matching problem. The first step is well advanced but not exhausted. It is surveyed in the next section. The second step is in progress and the main ideas are just sketched.

The other two kinds of RNA structural problems, comparison and prediction, are not unrelated to matching, at least in some of the formulations we may give to them. This is sketched in the Perspectives part of this chapter where we indicate the road we wish to follow in future.

3.2.2 Identifying Simple RNA Structural Elements – Palindromes and Pseudo-Knots

3.2.2.1 Main Ideas

Before presenting the main ideas behind the algorithm introduced in [173] and being extended (in preparation), let us recall that the best programs developed so far (RnaMot [76] and Palingol [25]) for solving the more general RNA structure matching problem proceed by first identifying the atomic elements of a structure, that is, the palindromes composing it. RnaMot does an "on the fly" search for such atomic elements while Palingol finds and stores all (exact and therefore small) palindromes in an initial step. The approach used by RnaMot is more efficient as palindromes having no chance of belonging to a structure of the requested form are never generated. As we saw, Palingol, though slower in general, is nevertheless the most interesting of the two programs because it allows to search for more complex structures (for instance, pseudo-knots).

One of the main ideas of our algorithm [173] is somewhat equivalent to the "on the fly" identification of palindromes adopted by RnaMot: the core of the structure whose description is given by the biologist is searched as a whole and only the palindromes having a chance of belonging to such a structure are generated. By "core of the structure", we mean the set of potential palindromes together with a few characteristics we check as we go such as the palindromes lengths, the distances separating the two parts of a palindrome and the relative position along the sequence of each palindrome in the set. Finer elements of the structure such as the stem-forming potential of the palindromes (the free energy each loop – base pairings; hairpin, internal and multiple loops; bulges – may contribute to or substract from the energy of the overall structure) as well as possible correlations within or between palindromes are checked in a second phase only.

A second idea is to enable the search for longer palindromes by allowing for errors. Ideally, approximate palindromes (corresponding to stems with unusual pairings, or with bulges, internal or multiple loops) should therefore not need to be patched together from small exact palindromes. Although this is not required for either efficiency or conceptual reasons, such approximate palindromes are identified using the same idea of a model as employed for motif extraction. This enables,

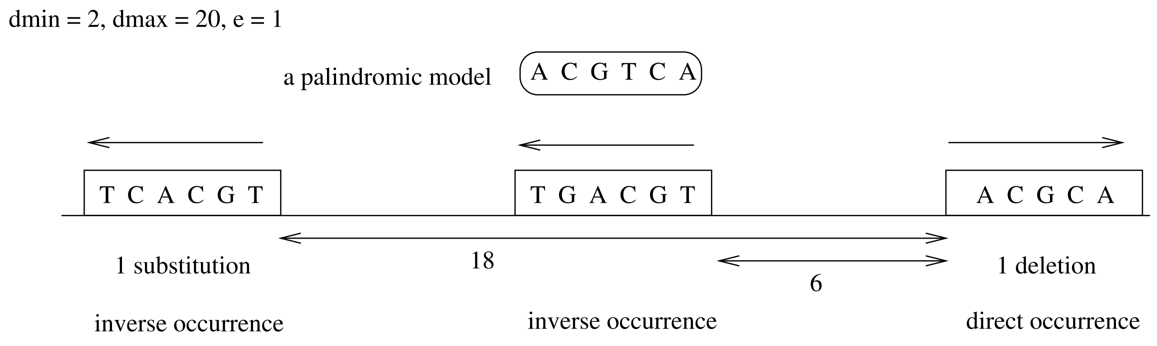

Figure 3.1: Example of a model for an approximate palindrome.

if needed, to look for RNA structural motifs which are conserved at least partially in terms of their sequence also and that appear repeated in one or more RNA molecules (see an example of such objects in section 3.2.2.4).

3.2.2.2 Models

3.2.2.2.1 Palindrome Models

Given a word u over the alphabet of nucleotides, we denote by $\overline{u}$ the inverse of u and $\overline{u_c}$ the complementary inverse of u.

If the pairing (G, T or U) is also considered besides the classical Watson-Crick (A, T) and (C, G), a word may have more than one complementary inverse.

Definition 3.2.1 *Given non-negative integers e, d_{min} and d_{max}, a word m defined over the alphabet Σ of nucleotides is a model for an approximate palindrome in s if, and only if, there exists two words u and v in s such that:*

- *the minimum distance between m and u on the one hand, and m and $\overline{v_c}$ on the other hand, is no more than e;*
- *$d_{min} \leq d \leq d_{max}$ where d is the distance between the end position of v and the start position of u in s.*

An example is given in Figure 3.1.

We say that u is a *direct* occurrence of m in s and v a complementary inverse, or *indirect*, occurrence (for those wishing to read the more formal definition in the corresponding paper [173], observe that this has been changed: direct occurrences in [173] had to be exact. The definition in this memoir is more general).

We may allow for a different maximum number of errors for direct and complementary inverse occurrences. If $e = 0$, we have of course an exact palindrome. Substitutions in a palindrome or repeat may correspond to unusual pairings or interior loops, and deletions and insertions to bulges. The sequence between the two halves of a palindrome represent the hairpin loops.

3.2.2.2.2 Pseudo-Knot Models

Let us first recall that a peudo-knot is composed of two interlaced palindromes. The definition of a pseudo-knot model is thus the following:

Definition 3.2.2 *Given a sequence s, non-negative integers e, $d[i]_{min}$ and $d[i]_{max}$ for $0 \leq i \leq 3$, we say that a pair of words (m_1, m_2), of lengths k_1 and k_2 respectively, each defined over the alphabet Σ of nucleotides, is an approximate pseudo-knot model present in s and satisfying the constraints (e, $d[i]_{min}$, $d[i]_{max}$) if, and only if:*

- *m_1 and m_2 are models for approximate palindromes as defined in 3.2.1;*
- *for each pair of indirect/direct occurrences (u_1, v_1) of m_1 there is (at least) one pair of indirect/direct occurrences (u_2, v_2) of m_2 such that:*
  - *u_1, u_2, v_1 and v_2 appear in that order in s;*
  - *$d[i]_{min} \leq d[i] \leq d[i]_{max}$ where $d[i]$ is the distance between:*
    for i=1 *the end position of u_1 and the start position of u_2;*
    for i=2 *the end position of u_2 and the start position of v_1;*
    for i=3 *the end position of v_1 and the start position of v_2.*

Observe that the lengths of models m_1 and m_2 may be different although the algorithm described in [173] and just briefly sketched below starts by finding pseudo-knot models composed of two interleaved palindromes of equal lengths.

3.2.2.2.3 Valid Models

A valid palindrome or pseudo-knot model is one satisfying definition 3.2.1 or 3.2.2 respectively.

This condition differs from the one given in the previous chapter where the notion of a model "validity" was associated with a quorum. One may add such a quorum constraint to the definition of a valid RNA structural model. This is required if we wish to find RNA structural elements that present a conservation in terms of sequence also. We speak then of "repeated palindromes" (more generally called in the biological literature [7] [9] [82] [83] "repeated palindromic units") or "repeated pseudo-knots".

3.2.2.3 Algorithm

To facilitate, we informally present the main ideas of the algorithm for palindromes and then explain what changes in the case of pseudo-knots.

Each step of the algorithm is composed of two parts. The first one consists in, from a given model, let us call it m, obtained at a certain point, extend m to $m\alpha$ to see whether $m\alpha$ still has any occurrences in the sequence being considered. This is done in much the same way as was described for single models in the previous chapter, except that we have two types of occurrences this time: direct ones that are extended left to right, and indirect ones that are extended right to left and their letters complemented. Both extensions are done in a completely independent way. Model $m\alpha$ must have at least one occurrence of each.

This is, however, not enough since these two occurrences must be at an allowed distance from one another to form a valid RNA palindrome. The second part of each step of the algorithm consists therefore in verifying whether we still have such a valid palindrome. This requires checking the distance separating an indirect from a direct occurrence for all pairs of occurrences possible. The main idea is that, for a direct (same for indirect) occurrence to be "valid", it needs to have just one companion at the right distance. This is all we must check during the whole run of the algorithm. We do not, and indeed, we must not attempt to find all possible companions for a given occurrence, and even less actually store such pairs together. This is done at the end only.

The important point to observe is that the sets of direct and indirect occurrences are kept permanently ordered (this follows naturally from the way models are constructed if stacks are used [167] instead of suffix trees [166]). Checking all possible companions for a given occurrence would already require at most $O(d_{max} - d_{min} + 1)$ time. We can, however, do better than that. The idea is to go down the list of (for instance) increasing positions of direct and indirect occurrences flagging any two at the right distance. Suppose that at some point we are considering indirect occurrence number p in the list and direct occurrence number r. Two things may happen:

1. p and r are at the right distance, we flag them both and go on to considering occurrence $r+1$;
2. p and r are not at a right distance from one another. This may be because:
   (a) they are too close. One must then continue considering p and advance to the next direct occurrence $r+1$ (r will remain unflagged if it was not flagged in a previous step, with a predecessor of p);
   (b) they are too far away. We must consider (for the first time) $p+1$ (p will remain unflagged if it was not flagged in a previous step) and step back to $r-1$ (if it exists). We never need to go back to occurrence $r-2$ because if $r-1$ is too far away from $p+1$, it is even farther away from p and we would have advanced to $p+1$ earlier.

Each direct occurrence is considered at most twice for each indirect one.

By postponing putting together the diverse parts of a structural object until the end, we are therefore able, during the whole step of searching for the structures (palindromes or pseudo-knots), to depend not on the number of such objects there is in a sequence, but instead on the number of parts that compose them. In the case of palindromes for instance, an upper bound for the number of objects is $O(n^2)$ where n is the length of the sequence, while the number of parts composing such objects is bounded over by $O((2e+1)n)$. In fact, this may be reduced to $O(n)$ if, for any position i in s we consider just one representative occurrence starting (or ending) at that position (for instance, the occurrence with least error against the model). This requires making a "softer" check of the distance separating an indirect from a direct occurrence. It may therefore lead to a slightly larger number of occurrences than will be contained in the final solution but this can be easily corrected at the end (further details may be found in [173]).

The main differences one needs to introduce to the basic algorithm just sketched for identifying pseudo-knots instead of palindromes is that two models must be built, one for each palindrome in the pseudo-knot. In [173], the two models are built at the same time (and have therefore same length – other constructions could be considered, see section 3.2.3). Each step consists in trying all possible combinations of extensions of one letter for each model, and then in verifying whether we still have a pseudo-knot. Checking the pseudo-knot property is done in much the same way as for palindromes, except that we have this time to simultaneously go down four lists of occurrences, two for each palindrome. Details may be found in [173].

3.2.2.4 A (Very) Preliminary Application – Repeated Palindromic Units

Repeated palindromic units form part of a number of more complex structures which may be present in the genomes of some organisms.

Such units are for instance often found at the borders of inserted elements (IS) and are the main players in so-called "BIMEs" ("Bacterial Interspersed Microbial Elements") [7] [9] [82] [83] which have been identified in *Escherichia coli*, *Salmonella typhimurium* and *Klebsiella pneumoniae.* The global function of such BIMEs has not yet been fully elucidated, some palindromic units have been shown to act as bidirectional transcription terminators. Other repeated palindromic units may form part of RNA structures as an element that would be conserved in terms of sequence as well as structure.

Preliminary unpublished analyses have been made in all sequenced bacterial genomes. The idea was to extract all occurrences of repeated palindromic units in a systematic way. The only hypotheses considered were that the motif should correspond to a palindrome of a minimum length l with a hairpin loop of a length between two values d_{min} and d_{max}. Initially, since BIMEs were the objects that interested us, the three parameters l, d_{min} and d_{max} were fixed at 9, 3 and 15 respectively. The repeated units were extracted with the algorithm described just above. They were identified using models. One substitution was allowed between a model and its occurrences (both direct and indirect), and a valid model further had to satisfy a quorum (typically, of 10).

As expected, repeated palindromic units in abundant numbers were found in *E. coli.* They could be grouped into at least three classes corresponding to those, Y, Z_1 et Z_2, identified by S. Bachellier *et al.* [7] [9] [82] [83].

Other genomes considered were:

Aquifex aeolicus;
Aeropyrum pernix K1;
Archaeoglobus fulgidus;
Bacillus subtilis;
Borellia burgdorferi;
Chlamydia trachomatis;
Chlamydia pneumoniae;
Deinococcus radiodurans;
Haemophilus influenzae;
Helicobacter pylori;
Methanobacterium thermoautotrophicum;
Methanococcus jannaschii;
Mycobacterium tuberculosis;
Mycoplasma genitalium;
Mycoplasma pneumoniae;
Neisseria meningitidis Z2491;
Pyrococcus abyssi;
Pyrococcus horikoshii;
Rickettsia prowazekii;
Synechocystis sp. PCC6803;
Thermotoga maritima;
Treponema pallidum;
Ureaplasma urealyticum;
Vibrio cholerae (incomplete);
Yersinia pestis (incomplete).

We are still in the process of analysing these initial results. A preliminary look at them has permitted to observe that genomes *Archaeoglobus fulgidus*, *Deinococcus radiodurans*, *Haemophilus influenzae*, *Methanobacterium thermoautotrophicum*, *Neisseria meningitidis* Z2491, *Pyrococcus abyssi*, *Pyrococcus horikoshii*, *Treponema pallidum*, *Synechocystis sp.* PCC6803 and *Yersinia pestis (incomplete)* present interesting regularities. Some of these may have gone unnoticed and be functional. For instance, units forming part of longer palindromic regions [8] were discovered by running the algorithm on the still incomplete genome of *Yersinia pestis.* Units found in *Archaeoglobus fulgidus* (TTGAAATCAGACCAAAATGGGATTGAAA - EMBL id: A80405) were already known and are patented. Results concerning other genomes need to be more carefully investigated. The parameters have also to be varied (work in progress).

It will not have escaped the algorithmical reader that the use of suffix trees as for the models in the previous chapter are worth exploring when a structural element is conserved in terms of sequence as in the case of the present application. This is already possible for palindromes (see section 2.3.2.3) but perhaps may be improved and even extended to more complex structures. This is currently been examined.

There remains a lot of work to be done to properly analyse and evaluate the above results, in terms of both their statistics and biology. Concerning statistical evaluation, the best thing we seem to be able to do for now is simulation: compare the palindromic units (number, distribution etc) found in the "real" genome against what is observed in the same genome after shuffling, perhaps inside regions with a distinct compositional landscape (*e.g.*, coding and non coding, or some finer decomposition), according to a Markov model of order p.

3.2.3 Towards RNA Structure Matching

The work on palindromes and pseudo-knots may be generalized to more complex structures. Let us illustrate this first with an example which concerns a piece of structure found in DNA instead of RNA. This is called *triple helices*.

Triple helices were initially described to us[1] by J. Maher III. These are composed of three sequence elements, one of which interacts with the other two. In one case, the liaison established is that of a classical Watson-Crick pairing, in the other we have a Hoogsteen pairing (same as Watson-Crick but with no sequence inversion) or reverse Hoogsteen (A pairs with A or T; C, G and T with themselves). The element that pairs with the other two is the middle one in the sequence (case of a reverse Hoogsteen pairing), or the first/last one (case of a Hoogsteen pairing). It is at least 80% AG-rich. Each element is, in general, no less than 10 bases long and the distances between any two successive elements is believed to be rather strict (between 0 and 10). Finally, unusual pairings may be present in a limited number and type.

The algorithm described in [173] was capable of identifying triple-helix-like objects. The definition we had given of such objects at the time was in fact wrong but required only a small change to the algorithm to be corrected. That change referred to the relation between the model for the triple helix and its three elements.

A possible immediate generalization to more complex structures therefore defines a meta-model for an RNA structure as being composed of the following basic elements:

- p models for $p \geq 1$;
- p sets of q_m relations between a model m and the q_m elements composing it;
- p sets of e_{q_m} error rates between a model m and each of its q_m elements;
- an order on the q_m elements for all m and possible distance constraints between all or some of the adjacent pairs of elements in the order;
- a meta-model error rate;
- a meta-error rate indicating the number of models that may be missing from the meta-model.

[1] Nadia el Mabrouk who then passed the information to this author.

Parts composing a structure	Properties pertaining to an element

loops
  hairpin
  internal
  bulge
  multiple loop
internal unpaired bases
external unpaired bases
	length
	conserved bases
	composition (plus possible correlations)
	error rate on conserved bases
	error rate on paired bases
	energy

Relation between the parts	**Properties of the overall structure**
relative arrangement of the elements possible correlation between elements	global error rate on conserved bases error rate on the elements (*i.e.*, possible missing elements) global error rate on paired bases global energy total length of a structure

Table 3.1: Elements of an RNA structure to be considered in the search for such a structure from a description of it.

In fact, more constraints can be considered. They are summarized in Table 3.1. Those mentioned above are the ones we believe should drive the search. The other properties of a structure may then be checked *a posteriori*.

The question now is how best to search for meta-models for RNA structures as defined above ? There are at least three possible ways of going about it. We can:

- identify all valid models in the meta-model separately and then piece them together (they will all have to be stored for final processing – this is approximately the Palingol approach);
- iteratively enumerate for each possible first model all second models that are correctly arranged in relation to it, and for each of those, all third models and so on until the p-th models;
- enumerate all possible p models simultaneously, that is, identify all p-tuple of models by extending them all together one letter at a time (for all combinations of letter possible).

The last approach profits the most from all constraints the meta-model contains and might therefore be the most efficient in terms of both space and time. This is not without reminding the approach we used to tackle a different problem (not described in this memoir but which may be found in [164] [165] [174]): avoiding small repeated contiguous structural motifs in proteins (corresponding to small helices). The idea there was to build the motifs by initially jumping to and fro in the string coding for the protein structure. This is what constrained "that" problem in the sense that bigger helices are less frequent than smaller ones and smaller ones are uninteresting in general. In the case of the third possible way described above for RNA structures, we also jump along the sequence, the idea being that p initial models simultaneously satisfying all the pairing and distance constraints of an RNA structure should be harder to find than a complete first model, then a second at the right distance from the first and so on. One can describe such a procedure as the walk of a p-legged millipede along the sequence.

3.3 Identifying Conserved Protein Structural Elements at the Sequence Level – Where Are We ?

3.3.1 Foreword

What follows is a first attempt (dating from 1996/1997) on our part to examine how far we could go with the tools we had towards trying to identify, from the sequence alone, elements that appear conserved in terms of the protein structure, and may therefore be involved in its enzymatic function or final adopted topology.

The motivation at the time was a work being done at the laboratory by Joël Pothier and Pascale Jean [115]. Their problem was, in a sense, a dual of the one we wished to address. Their objective was to use an algorithm elaborated by Henry Soldano, Joël Pothier and Alain Viari (see [164] and [174] for a very partial description of it) to identify contiguous structural motifs conserved among a set of three cytochromes P450 protein structures which were known by crystallography (Swiss-Prot [19] id: CPXA_PSEPU, CPXB_BACME, CPXL_PSESP). The sequence alignments corresponding to such conserved structural blocks were then used as profiles for predicting parts of the structure, unknown at the time, of another protein from the same family [115].

Thirteen conserved structural blocks were identified, each one corresponding to one or more elements of the secondary structure (α helices and β sheets). One remarkable fact that called our attention was that, of the 255 positions covered by all the structural blocks, only 27 present the same amino acid, and among these 27, 9 are located inside just one block. This is the heme region, also called the cys-pocket, which is one of the only well conserved part in terms of both sequence and structure. Seeing this, we wondered how much of such conserved structural elements we would be able to identify by looking at the sequences only.

We used an approach that resembles the one described in the previous chapter concerning single models and a way of extracting such models from DNA sequences. There is, however, a twist this time that comes from the observation that amino acids are more complex biological objects than nucleotides. They have, for instance, various physico-chemical properties, like, for instance, electrical charge, polarity, size, different levels of acidity etc. Some, but seldom all, of these properties may be shared by two or more amino acids. It is therefore unrealistic to work with an identity relation between the letters of the protein alphabet as we most often do for the DNA (or RNA) alphabet. A more realistic relation will typically be reflexive, symmetric but non transitive [187]. We start by briefly explaining what this changes in the definition of the models, their relation to the occurrences and, finally, the algorithm itself (sections 3.3.2 and 3.3.3.1, details may be found in [176]). Once models have been extracted, a second part of the work must consist in evaluating their statistical significance. The question here is not only to determine which models found are statistically significant, but more precisely to establish which ones have a chance of corresponding to structurally conserved elements. The approach must therefore be different from the one utilized for evaluating the statistical significance of DNA conserved elements. The one we initially adopted is presented in section 3.3.3.2. Since this work and its preliminary applications have not yet been published, they are described in somewhat more detail. A preliminary version of the unsubmitted paper is available upon request.

3.3.2 Model

3.3.2.1 Symbol-Based Distance

A way of modelling a non transitive relation between amino acids is by means of an extended alphabet composed of a subset of the set of all parts of Σ (denoted by $\mathcal{P}(\Sigma)$) where Σ is the alphabet of proteins. A mathematical way of formulating this uses an hypergraph $\mathcal{H}$ whose nodes are the amino acids and edges the "symbols" of the extended alphabet. If Γ is the set of edges of $\mathcal{H}$, then a model M is an element over Γ^{*}[2].

An extended alphabet may be combined with the possibility of errors (substitutions only, or substitutions, insertions and deletions). If up to e are allowed, an occurrence u in a sequence s of a model M defined over the extended alphabet Γ is a word in s such that the Hamming or Levenshtein distance between u and any peptide $v \in M$ is no more than e.

For instance, the word of length 5 located at position 2 in sequence IMGHPFNTRDCSSA (positions in sequences start at 1) is at a distance at most 1 of models [ILMV][AG]HP[FYW] (exact match – letters between brackets mean any one may appear at that position in the occurrences of the model), [ILMV][AST]HP[FYW] (substitution of the second letter), [ILMV]HP[FYW] (deletion at the second position), [ILMV][AG]P[FYW] (insertion of letter H between the 2nd and 3rd letters) and of many others.

Among the subsets allowed in Γ may be $\{\Sigma\}$ itself (the *wild card*). It is obvious that allowing the wild card as symbol may lead to trivial models. The alphabet Γ may then come with a non negative integer value attached to each of its elements (we say that hypergraph $\mathcal{H}$ is weighted [172]) indicating how many times (possibly infinite) it may appear in a model of interest to us.

3.3.2.2 Word-Based Distance

Especially where proteins are concerned, other definitions of similarity between words and the model of which they are occurrences could be considered. For instance, instead of counting errors, we could attribute a score to the similarity. The main idea behind scoring a resemblance between two objects is that it allows to average the differences that may exist between them. It may thus provide a more flexible similarity function. A simple example illustrates this point.

Example 3.3.1 *Let* $\Sigma = \{A, B, C\}$ *and:*

$$\begin{array}{lll} \textit{score(i,i)} & = \textit{1}, \forall i \in \Sigma; & \\ \textit{score(A,B)} & = \textit{score(B,A)} & = \textit{-1}; \\ \textit{score(A,C)} & = \textit{score(C,A)} & = \textit{-1}; \\ \textit{score(B,C)} & = \textit{score(C,B)} & = \textit{-1}. \end{array}$$

If we say that 2 words are similar either if:

- *the number of substitutions between them is no more than 1;*
- *their score is at least 1;*

then by the first criterion the words AABAB and AACCB are not similar, while by the second criterion they are, the second substitution being allowed because the two words on average share enough resemblance.

[2]The notation Γ^* indicates all words, of any finite length, that may be written over the alphabet Γ.

We consider first models defined over the alphabet Σ of amino acids instead of over an extended alphabet.

Let a numerical matrix $\mathcal{M}$ of size $\mid \Sigma \mid \times \mid \Sigma \mid$ be given such that:

$$\mathcal{M}(a, b) = \text{score between } a \text{ and } b \text{ for all } a,\ b \in \Sigma.$$

If this score measures a similarity between a and b, we talk of a similarity matrix (two well-known examples of which in biology are PAM250 [52] and BLOSUM62 [103]), while if the score measures a dissimilarity between a and b we talk of a dissimilarity matrix. A special case of this latter matrix is when the dissimilarity measure is a metric, that is when the scores obey, among other conditions, the triangular inequality. In such a situation, we talk of a distance matrix (an example of which is the matrix proposed by J.-L. Risler [159]).

Given $\mathcal{M}$ a similarity matrix, we say a word u of length k in s is an occurrence of a model m of same length defined over Σ if all subwords of length w at equivalent positions in u and m have a score at least t, for w a positive integer such that $w \leq k$ and t a threshold value. We speak of a *word-based* relation of similarity in contrast to the *symbol-based* one used so far.

Example 3.3.2 *Let $\Sigma = \{A, B, C\}$, $w = 3$ and $t = 6$. Let $\mathcal{M}$ be the following matrix:*

	A	*B*	*C*
A	*3*	*1*	*0*
B	*1*	*2*	*1*
C	*0*	*1*	*3*

Given the three sequences:

$$s1 = ABCBBABB\underline{BACABACB}BBAB$$
$$s2 = \underline{CABACAAC}BACCABCACCACCC$$
$$s3 = BBBACACCABA\underline{BACABACA}BA$$

then the longest model that is present in all sequences is $CACACACC$ (at positions 9, 1 and 12 respectively).

In the examples 3.3.1 and 3.3.2 and in the definition given above, gaps are not considered, only substitutions. This was done essentially to facilitate exposition. Gaps may, however, be authorized, the reader is referred to [175] for this and further details (in particular, concerning the algorithm itself). Extension to models defined over an extended alphabet are not discussed. It is straighforward to imagine what such an extension could be.

From now on, we shall consider similarity definitions based on counting errors only.

3.3.3 Algorithm

3.3.3.1 Model Extraction (also called Model Combinatorial Sampling)

The algorithm consists of two parts. The first extracts models from the set of sequences being considered (we say we *sample* the space of all possible models), the second part evaluates such models. In order to diminish the risk of missing even very weak similarity, the parameters are set in such a way (low quorums, very degenerate alphabet, many wild cards allowed, high error rate authorized) that often a great number of models is identified.

The core ideas for the sampling step are basically the same as very briefly described in section 2.4. The corresponding algorithms may [166] [171] or not [172] [176] make use of a suffix tree. When dealing with models over an extended alphabet while further allowing for errors, it becomes even more crucial to acquire the ability to eliminate redundancy. This may come, for instance but not exclusively, under the form of two models that, though different, correspond to the same (or almost the same) sets of occurrences.

This problem was addressed in a preliminary fashion in [172] but is far from being solved. Perspectives as concern proteins more particularly are discussed in section 3.4.2.2.

3.3.3.2 Evaluating Models Possibly Related to Structure and/or Function (also called Model Sieving)

3.3.3.2.1 Main Idea

As we saw in the previous chapter, there are many ways to evaluate the statistical significance of a model (or motif). Since our objective in the case we are treating here is to identify structure (and therefore possibly function) related elements that appear conserved in proteins, a different approach than the one used for DNA seemed appropriate.

Currently, the evaluation step starts by effecting a change of representation. While models drove the sampling search in the first step, emphasis is now placed on their occurrences (which were extracted together with the models). To each model M is associated the set of its occurrences in the sequences under consideration (denoted by E) and to each set of occurrences may be associated a matrix, also called a profile, P_M. Matrices very similar to these, augmented sometimes to consider the possibility of gaps, are often used for locating other sequences, for instance in a database, that contain the motifs the matrices refer to [88] [89]. In order to do this, the matrix is aligned with each sequence in the database. The score of the alignment with sequence s determines whether s contains the motif associated with the matrix. In our case, we wish to calculate the best alignment obtained between a matrix P_M of a model M and a well-chosen set T of sequences **in order to score the model itself** [170] [171]. Set T may have one, or more than one sequence. It will be phylogenetically or functionally related to the sequences in E. The idea behind all this is the following. If the model is "explainable" by chance, it will have a poor score when aligned with the sequences in set T. If, on the other hand, the model corresponds to a conserved structural element (possibly, a functional site), then its score once aligned with T will be significantly high.

Since in the case of the problem that interested us initially, it is unnecessary to consider gaps, we decided to use a profile scoring method due to Goldstein and Waterman [84] instead of Gribskov's one [88] [89]. This has also the advantage of providing an *a priori* way of calculating the p-value of an obtained score. Models are then evaluated and sorted by order of increasing p-values. Only those having significant p-values among all the models sampled in the first step of the algorithm are deemed potentially interesting.

We call the first set of sequences (from which models are sampled) the *example set* and the second the *test set* (hence the E and T).

3.3.3.2.2 Basic Evaluation Procedure

The basic procedure consists in, first, establishing the profile P_M associated with each model M. This is done by using Goldstein and Waterman's formula:

$$P_M(i,j) \;=\; \sum_{a=1}^{20} f(a,j) \times \mathcal{M}(i,a)$$

where $f(a,j)$ is the observed frequency of monomer a at position j in the aligned (in an obvious fashion) set of occurrences of M and $\mathcal{M}$ a substitution matrix (in the case of proteins for instance, $\mathcal{M}$ can be Dayhoff's PAM250 [52] or Henikoff's BLOSUM62 [103] matrix).

The best positioning x of profile P_M in a sequence s of length n, is the one having best score S_0 where S_0 is given by formula:

$$S_0 = \sum_{j=1}^{k} P_M(s_{x+j-1}, j)$$

with k referring to the width of the profile (and also, in our case, to the length of model M). The estimated probability that a score greater or equal to S_0 may be observed in a random sequence of same length and monomer composition, its p-value, is then easy to derive. For details see [84]. Regularizers [123] may be introduced in the calculation of the profile to deal with the fact that the example set is often small.

These are the bare essentials upon which rests the evaluation procedure. There are now several difficulties we have to deal with.

3.3.3.2.3 Multiple Occurrences

The first difficulty concerns the fact that a model M may have more than one occurrence in a given sequence of the example set. When this happens, it is often the case that only one of these occurrences, or none, will correspond to a given structural or functional characteristic of the macromolecules under study. We may therefore wish to eliminate the possibly irrelevant ones before attributing a score and p-value to the profile P_M associated with M.

There are various possible manners of doing that. The important thing here is to select one that gives results independent of the order in which sequences containing multi-occurrences are considered.

We chose to do it the following way. Let s be a sequence containing multiple occurrences of M. Let $p_1, p_2, \cdots, p_m$ be the positions of these occurrences in s. We align P_M with s at every position p_i and select the one corresponding to the smallest p-value as the only occurrence of M in s that will remain in the profile at the end. Once this operation is done for all sequences containing multiple occurrences of M, a new profile P'_M is built using a single occurrence for each sequence s in the example set (either the only one present in s to begin with, or the one chosen by the process just described). The score and p-value of P'_M are then calculated as in the basic procedure given in section 3.3.3.2.2.

3.3.3.2.4 Missing Occurrences

Another difficulty that may present itself when evaluating models comes from the fact that, when the quorum q is fixed to a value less than the number N of sequences in the example set, a model may miss an occurrence in a sequence. Although this is an "acceptable" behaviour, we may wish to use the profile P_M obtained at the sampling step to find occurrences for M in those sequences where none was found that verified the initial definition of similarity between models and occurrences.

In the procedure we adopt we take the precaution, as with the operation described in the previous section, of proceeding in a way that is independent of the order in which sequences that miss an occurrence are considered.

Let s be a sequence missing an occurrence of M. We align P_M with every position in s. Let p_i be the one for which the alignement score corresponds to the smallest p-value. If this p-value is lower than 10^{-3}, the occurrence starting at position p_i in s and having length $|M|$ will be the one selected to be added to the set of occurrences of M. Each possible sequence missing an occurrence is thus tested against the profile as initially constructed from the model. Missing occurrences, when accepted, are then globally added at the end only in order to calculate a new profile P'_M for M as indicated in section 3.3.3.2.2.

Observe also that, although the two procedures described here and in the preceding section act on the sets of occurrences of the models in a way that is independent of the order in which the sequences in the example set are considered, such sets of occurrences are changed in consequence of each of these operations. The order in which the operations themselves are performed may therefore modify what is obtained in the end. We choose an order that:

- starts by eliminating objects actually present in the sets of occurrences of the models at the end of the sampling step (that is, we deal first with the problem of multiple occurrences);

- then possibly adds objects missing from these sets.

The justification for this is that is seems more sensible to first eliminate things from the set of occurrences corresponding to a profile, thus "stabilizing" it in some way, before adding objects to it that may be assumed to be more distant from the ones already in it.

3.3.3.2.5 Overlapping Profiles

A third difficulty we have to face is that profiles deemed pertinent may overlap. Different cases have to be taken into account.

If the relative shift observed for any two occurrences of models M_1 and M_2 respectively is always the same and is different from zero (if it is zero, P_{M_1} and P_{M_2} are identical and the second profile need not even be calculated), then both profiles are kept if the p-value of each is below 10^{-3}.

If the overlaps differ at least once for any two such pairs of occurrences, we may consider the following approaches:

- eliminate the profile having higher p-value;

- keep both profiles if their p-values are below 10^{-3} and the number of overlapping occurrences is less than a certain percentage.

How this percentage is fixed may be left at the decision of the user. For instance, it could be established at $t\%$ of the number of occurrences common to M_1 and M_2 where t is a threshold value.

3.3.3.2.6 Refinement

Once multi-occurrences have been eliminated and missing ones possibly added, and before overlaps are dealt with, we may want to question the pertinence of each occurrence originally considered as belonging to the model. For each sequence s, we may thus re-align with s the newer profile obtained after the operations described in sections 3.3.3.2.3 and 3.3.3.2.4 have been performed. This is not strictly necessary, and we may choose to forgo this operation. Should we wish to employ it, then our approach could be compared to that of an EM algorithm or Gibbs sampler applied, not to an initially random set of occurrences, but to sets of occurrences obtained by a first combinatorial search of the space of sequences.

3.3.4 Preliminary Application

The example concerns the cytochromes P450 (abbreviated thereafter into P450s) mentioned above (section 3.3.1). The example set E was therefore composed of the three P450s whose structure was already known in 1995. To facilitate, we shall call them P450_{Cam} (Swiss-Prot id: CPXA_PSEPU), P450_{BM3} (Swiss-Prot id: CPXB_BACME) and P450_{Terp} (Swiss-Prot id: CPXL_PSESP). The test set T contained just one sequence, that of the P450 whose structure was about to be obtained by crystallography at about the same time. This is the P450_{Eryf} (Swiss-Prot id: CPXJ_SACER).

The alphabet used for the models in our algorithm is inspired on a diagram established by Taylor [191]. It is: $\Gamma = \{\{I, L, M, V\}, \{A, G\}, \{A, S, T\}, \{C\}, \{F, Y, W\}, \{K, H, R\}, \{E, D\}, \{N, Q\}, \{G, P\}, \{\Sigma\}\}$, with the wild card allowed at most five times in each model. The maximum number of mismatches authorized between a model and each of its occurrences was one. For this preliminary test, only models of a fixed length of 12 were sampled and then sieved.

The models we found (given as sets of occurrences) and those found by the Gibbs sampler (we used the version given in the PROTOMAT program that can be run at http://www.blocks.fhcrc.org/blockmkr) [127], MEME (http://www.sdsc.edu/MEME/meme.2.0/website, we asked for the maximum number of motifs which is 10) [13] [14] [15] [16] and Pratt (this is another combinatorial approach that works with a degenerate alphabet for the motifs but does not allow for errors, see http://www.ii.uib.no/ inge/Pratt.html; the number of motifs reported is the one by default, that is 50) [117] [118] are shown in Table 3.2. Sets of occurrences that are just shifted in relation to one another are put together in the Table which explains why some motifs may be longer than 12. Not shown in the Table is the fact that just one false positive is reported by our algorithm (with or without refinement). We define false positive a set of occurrences whose p-value is below 10^{-3} and yet is not one of the conserved structural blocks identified by [115].

Let us observe that this example is quite unfair to statistical approaches: it contains too few sequences from which motifs must then be inferred. It was chosen because it was a difficult example and illustrated the fact that the use of combinatorial approaches is appropriate over a possibly wider range of cases than statistical methods (although no method can really be said to be very good in difficult situations). Examples with more sequences may be found in the thesis dissertation [165] and in two manuscripts [170] [171] describing the algorithm sketched above (available upon request). For instance, on the example used by Lawrence *et al.* in their 1993 paper on the Gibbs sampler and concerning over 30 sequences containing an HTH (Helix Turn Helix) motif, our algorithm performs almost as well as Lawrence's: it misplaces only one occurrence of the motif in one of the sequences. All other occurrences are accurately identified.

3.3.4.1 Have Motifs Gaps ?

3.3.4.1.1 Foreword

One may wonder why indels, or gaps, are seldom ever considered when looking for motifs corresponding to DNA or protein binding sites. Biologically, this would seem to be correct since binding sites are geometrically constrained, making an insertion or deletion in such a place impossible. However, one reason why indels in a site have never been reported may be because they were never sought. This is a problem that we meet repeatedly and is not specific to biology: we only find what we look for and we only look for what we believe is possible. Algorithmically, we can now, in a reasonably efficient way, search for models whose occurrences are at a maximum Levenshtein instead of Hamming distance from the model [176]. We show an (unpublished) example in the case of proteins where performing such a search produced an intriguing result that we do not know how to interpret yet which appears interesting.

This result concerns a model indicated in the Prosite database [18] as corresponding to one of the active sites of a family of proteins, which is found also in a different family, with one deletion, when models are sought in the set obtained by joining representatives from the two families. These represent subfamilies of the nitrogenases and the hydrogenases.

Nitrogenases are enzymes responsible, among other things, for fixing nitrogen in a reduction operation. The reduction frees hydrogen which may then be oxydated by a hydrogenase. Nitrogenases contain different clusters of iron and sulfur which are necessary for their function. They may also need a metallic ion (which fixes itself on the iron-sulfur cluster). The metallic ion can be either molybdenum (subfamilies with mnenomic names nifK and nifD) or, more rarely, vanadium (subfamilies with mnenomic names vnfD, vnfK et vnfG). Some nitrogenases seem to be able to fix iron only; they are named anfD, anfK et anfG. Hydrogenases on the other hand need nickel. These metals are brought to the nitrogenases and hydrogenases by proteins that belong to the same sub-families (nitrogenases) or that are involved in their regulation (hydrogenases) [65] [107] [160] respectively. It is the sequence of such proteins which constitute our data set. Their mnenomic names are nifE and hypD. They represent a set of 5 + 6 sequences whose names in Swiss-Prot are: NIFE_AZOVI,

Table 3.2: (See table on next page) Motifs/models (indicated by the start and end positions of their occurrences) found in the P450s by the Gibbs sampler, MEME, Pratt and by our algorithm (called CoSamp), without and with refinement respectively. For all algorithms except ours, parameters used were those given by default. In our case, the maximum number of mismatches authorized between a model and each of its occurrences was one. The alphabet used for models is $\Gamma = \{\{I, L, M, V\}, \{A, G\}, \{A, S, T\}, \{C\}, \{F, Y, W\}, \{K, H, R\}, \{E, D\}, \{N, Q\}, \{G, P\}, \{\Sigma\}\}$. The occurrences of each motif belong, from top to bottom, to P450$_{BM3}$, P450$_{Cam}$ and P450$_{Terp}$. Occurrences in italics and placed between parenthesis are wrong, however when the corresponding structural elements are helices and the shift of the wrong occurrence relative to the right one is of less than 10 positions (that may represent up to 3 helix turns), this is indicated beside by a $+x$ or $-x$. Occurrences between square brackets correspond to profiles whose p-value is between 10^{-3} and $10^{-2.5}$ with no false positives between the last one at 10^{-3} and these. The execution times for CoSamp (non optimized version) were of 211 and 289 s. respectively on a Dec Alpha 4000 5/466.

Conserved Structural Blocks (elements of sec. structure they belong to)	Gibbs	MEME	Pratt	CoSamp without refine	CoSamp with refine
Strand β_{1-2} and helix B					
Helix C		— 101-108 103-110		(*1-15*) 101-115 107-121	(*171-182*) 104-115 111-122
Helix D	120-129 (*112-121*) 134-143				[(*26-37*)] [124-135] [126-137]
Helix E	(*150-168*) 142-160 145-163	(*154-168*) 146-153 149-156	150-156 142-166 145-169	(*217-228*) 160-171 163-174	(*431-442*) 159-170 161-173
Helix F			(*169-191*) 167-183 170-186		
Helix G					203-215 191-203 210-222
Helix H		213-224 201-212 220-231		(*346-359*) 212-225 231-244	224-243 210-229 229-248
Strand β_{5-2} and helices I and J				262-274 246-258 (*52-63*)	257-74 241-258 260-277
Helix K and strand β_{6-1}			270-288 (*35-53*) 273-291	(*284-295*) 278-289 297-308	(*284-296*) 278-290 297-309
Strands β_{1-4} and β_{2-1}					332-352 298-309 318-329
Strands β_{2-2} and β_{1-3} and helix K'		(*97-104*) 313-320 333-340	347-352 312-317 (*292-297*)	363-367 328-332 348-352	[346-362]-367 [311-327]-332 [331-347]-352
Meander				(*375-379*) +5 334-338 354-358	(*361-366*) -9 334-339 354-359
Cys-pocket	393-398 350-355 370-375	393-398 350-355 370-375	393-398 350-355 370-375	389-398 346-355 366-375	387-398 344-355 364-375
Helix L	399-419 356-376 376-396	399-403 356-360 376-380	399-404 356-361 376-381	399-413 356-370 376-390	399-414 356-371 376-391
Strands β_{4-2} + β_{3-2}				(*112-123*) 396-407 415-426	[(*11-22*)] [396-407] [415-426]

Table 3.2: Legend on previous page.

HYPD_ALCEU	57	RMIHGP-GC
HYPD_AZOVI	57	RLIHGP-GC
HYPD_BRAJA	24	RMIHGP-GC
HYPD_RHILV	57	RMIHGP-GC
NIFE_AZOVI	54	HIVHGPIAC
NIFE_BRAJA	62	HLVHGPIAC
NIFE_CLOPA	71	HLVHGPIGC
NIFE_KLEPN	54	HLVHGPIGC
NIFE_RHOCA	63	HLVHAPAAC
		* * *

HYPD_ALCEU	220	YGKPVVIAGFEPL
HYPD_AZOVI	219	YGTPLVIAGFEPL
HYPD_BRAJA	187	FGKPVVIAGFEPL
HYPD_ECOLI	218	FHRPLVVAGFEPL
HYPD_RHILV	220	FRKPVVVAGFEPL
HYPD_RHOCA	218	YGKPLVVAGFEPT
NIFE_AZOVI	265	YGTPWFEGSFYGI
NIFE_BRAJA	263	YGIPFFEGSFYGI
NIFE_CLOPA	270	FGIPYIKVSFYGV
NIFE_KLEPN	265	YGTPWFEGSFYGI
NIFE_RHOCA	265	YGIPYFEGSFYGI
		* *

Figure 3.2: Blocks found by the algorithm described in [176], with up to one error, respectively 0 and 5 wild cards and quorums of 100% and 80%. A star below a column indicates perfectly conserved positions.

NIFE_BRAJA, NIFE_CLOPA, NIFE_KLEPN, NIFE_RHOCA, HYPD_ALCEU, HYPD_AZOVI, HYPD_BRAJA, HYPD_ECOLI, HYPD_RHILV and HYPD_RHOCA.

3.3.4.1.2 Results

The algorithm described at length in [176] (which allows for insertions and deletions as well as substitutions between a model and its occurrences) was run twice searching for the longest models with at most one error. The alphabet used for the models was the following: $\{\{I, L, M, V\}, \{A, G\}, \{A, S, T\}$, $\{C\}, \{F, Y, W\}, \{K, H, R\}, \{E, D\}, \{N, Q\}, \{G, P\}\}$ In one case, wild cards were also permitted. The quorum was 80% (no wild cards) and 100% (up to five wild cards).

After some cleaning (explained in [165]), we obtained the two blocks shown in Figure 3.2 with the names of the sequences where they are present as well as their positions. The first block is the one which, as mentioned, corresponds to part of the active site of the proteins in the nitrogenase family ([LIVMFYH][LIVMFST]H[AG][AGSP][LIVMQA][AG]C as indicated in Prosite). The blocks may be summarized into the patterns [KHR][ILMV][ILMV]H[AG]P[ILMV][AG]C and [FY]GXPxxx[AG]xFx[GP][ILVM] where x denotes the wild card. Observe that the patterns contain respectively 3 and 2 perfectly conserved positions (indicated by a star in the Figure). We then searched in the Swiss-Prot database (release 33 – this was done in early 1996) for sequences having each of the patterns (separately) allowing for up to one error (substitution or indel). Figures 3.3 and 3.4 show the results obtained. Besides the sequences from which it was extracted, the first pattern is found present also in 12 nifDs, 1 nitrogenase which fixes iron (anfD), 2 from different species which fix vanadium (vnfD) and 1 *E. coli* hypothetical protein (YJGB_ECOLI). When forbidding an error to occur in any one of the three perfectly conserved positions, the hypothetical protein disappears from the list. The second pattern seems to be less selective although, again, by disallowing errors in the two perfectly conserved positions of the pattern, the number of potential false positives diminishes (the sequences still containing the pattern are indicated by a plus sign in Figure 3.4, the eliminated sequences are flagged with a minus sign).

We then decided to search for the blocks themselves (instead of the patterns) and to do it simultaneously (the first block must appear before the second in the sequences as this is what happens in the initial data set). We used for this purpose an algorithm for aligning an ordered sequence of profiles allowing for gaps inside each profile in a way similar to what Gribskov does [89]

[88]. This is a homemade program due to Alain Viari [198]. The weight given to a gap is equal to the highest value observed in the profile (with a minus sign). Figure 3.5 shows the distribution of the scores that were obtained.

The tail of the distribution, corresponding to sequences with a score higher than 47, are listed in Figure 3.6. The sequence with highest score is a protein of the same type as the hypDs but from another organism (a plant instead of a bacterium). Among the remaining sequences, we find all the hypDs and nifEs, 13 among the 17 nifDs, 6 nitrogenases (1 nifK, 2 nifN, 2 vndF et 1 anfD) and a single hypothetical protein of unknown function.

3.4 Critique and Perspectives

3.4.1 Critique and Perspectives on RNAs

3.4.1.1 The Problem of Redundancy

One of the main problems we shall have to face in the future concerning the RNA structure matching problem in particular but not only, is that of the redundancy observed in terms of the models (and their occurrences) identified as satisfying the input constraints. Consider the case of a search for palindromes. Specially when insertions and deletions as well as substitutions are allowed, a single

```
+ sp|P16266|ANFD_AZOVI NITROGENASE IRON-IRON PROTEIN ALPHA CHAIN
+ sp|P31903|HYPD_ALCEU HYDROGENASE EXPRESSION/FORMATION PROTEIN H
+ sp|P31882|HYPD_AZOVI HYDROGENASE EXPRESSION/FORMATION PROTEIN H
+ sp|P31904|HYPD_BRAJA HYDROGENASE EXPRESSION/FORMATION PROTEIN H
+ sp|P40598|HYPD_RHILV HYDROGENASE EXPRESSION/FORMATION PROTEIN H
+ sp|P00464|NIFD_ANASP NITROGENASE MOLYBDENUM-IRON PROTEIN ALPHA
+ sp|P25313|NIFD_AZOBR NITROGENASE MOLYBDENUM-IRON PROTEIN ALPHA
+ sp|P07328|NIFD_AZOVI NITROGENASE MOLYBDENUM-IRON PROTEIN ALPHA
+ sp|P06121|NIFD_BRAJA NITROGENASE MOLYBDENUM-IRON PROTEIN ALPHA
+ sp|P06120|NIFD_BRASP NITROGENASE MOLYBDENUM-IRON PROTEIN ALPHA
+ sp|P00467|NIFD_CLOPA NITROGENASE MOLYBDENUM-IRON PROTEIN ALPHA
+ sp|P20620|NIFD_METTL NITROGENASE MOLYBDENUM-IRON PROTEIN ALPHA
+ sp|Q00239|NIFD_PLEBO NITROGENASE MOLYBDENUM-IRON PROTEIN ALPHA
+ sp|P06769|NIFD_RHICP NITROGENASE MOLYBDENUM-IRON PROTEIN ALPHA
+ sp|P00465|NIFD_RHILT NITROGENASE MOLYBDENUM-IRON PROTEIN ALPHA
+ sp|P08717|NIFD_RHOCA NITROGENASE MOLYBDENUM-IRON PROTEIN ALPHA
+ sp|P06662|NIFD_THIFE NITROGENASE MOLYBDENUM-IRON PROTEIN ALPHA
+ sp|P08293|NIFE_AZOVI NITROGENASE IRON-MOLYBDENUM COFACTOR BIOSY
+ sp|P26506|NIFE_BRAJA NITROGENASE IRON-MOLYBDENUM COFACTOR BIOSY
+ sp|P10996|NIFE_CLOPA NITROGENASE IRON-MOLYBDENUM COFACTOR BIOSY
+ sp|P08737|NIFE_KLEPN NITROGENASE IRON-MOLYBDENUM COFACTOR BIOSY
+ sp|P19055|NIFE_RHOCA NITROGENASE IRON-MOLYBDENUM COFACTOR BIOSY
+ sp|P15332|VNFD_AZOCH NITROGENASE VANADIUM-IRON PROTEIN ALPHA CH
+ sp|P16855|VNFD_AZOVI NITROGENASE VANADIUM-IRON PROTEIN ALPHA CH
- sp|P27250|YJGB_ECOLI HYPOTHETICAL 36.5 KD PROTEIN IN GNTV-LEUX
```

Figure 3.3: Database sequences where the pattern [KHR][ILMV][ILMV]H[AG]P[ILMV][AG]C (corresponding to the first block at the left of Figure 3.2) is found allowing for up to 1 error. The block is no longer present in the sequences flagged with a minus sign when no error is permitted at the positions in the block where a strict conservation of a single amino acid is observed (those indicated with a star in Figure 3.2).

position may represent the start of various direct (or indirect) occurrences of a same model (having same "companion" occurrence or not). Two direct occurrences of a same model may also be shifted of just one position in relation to one another.

We have so far been postponing confronting this problem for various reasons. The main, and for us, most disturbing one, is that there does not seem to be (or at least, we do not see it) a fool-proof way of formalizing what is redundant and may thus be safely eliminated. Some cases do not pose a dilemma. If, for instance, two direct occurrences have exactly the same indirect occurrence, and

```
+ sp|Q00831|AMPD_CITFR AMPD PROTEIN.
+ sp|P13016|AMPD_ECOLI AMPD PROTEIN.
+ sp|P30013|AMPD_SALTY AMPD PROTEIN (FRAGMENT).
+ sp|P20073|ANX7_HUMAN ANNEXIN VII (SYNEXIN).
+ sp|Q07076|ANX7_MOUSE ANNEXIN VII (SYNEXIN).
- sp|Q03025|APRE_PSEAE ALKALINE PROTEASE SECRETION PROTEIN APRE.
+ sp|Q04656|AT7A_HUMAN COPPER-TRANSPORTING ATPASE 1 (EC 3.6.1.36)
+ sp|P35670|AT7B_HUMAN COPPER-TRANSPORTING ATPASE 2 (EC 3.6.1.36)
- sp|P29705|ATPZ_PROMO ATP SYNTHASE PROTEIN I, SODIUM ION SPECIFI
- sp|P08125|CA1A_CHICK COLLAGEN 1(X) CHAIN PRECURSOR.
- sp|Q04539|CXEP_ALCEU RIBULOSE-PHOSPHATE 3-EPIMERASE, PLASMID (E
+ sp|P28639|DHN1_PEA DEHYDRIN DHN1.
+ sp|P07918|DPOL_HSV21 DNA POLYMERASE (EC 2.7.7.7).
+ sp|Q03763|DSG1_BOVIN DESMOGLEIN 1 PRECURSOR (DESMOSOMAL GLYCOPR
+ sp|Q02413|DSG1_HUMAN DESMOGLEIN 1 PRECURSOR (DESMOSOMAL GLYCOPR
+ sp|P17486|EGL5_CAEEL HOMEOBOX PROTEIN EGL-5 (HOMEOBOX PROTEIN C
+ sp|P06131|FDHA_METFO FORMATE DEHYDROGENASE ALPHA CHAIN (EC 1.2.
+ sp|P14324|FPPS_HUMAN FARNESYL PYROPHOSPHATE SYNTHETASE (FPP SYN
+ sp|P40223|GCSR_MOUSE GRANULOCYTE COLONY STIMULATING FACTOR RECE
+ sp|P30870|GLNE_ECOLI GLUTAMATE-AMMONIA-LIGASE ADENYLYLTRANSFERA
- sp|P38631|GLS1_YEAST 1,3-BETA-GLUCAN SYNTHASE COMPONENT GLS1 (E
- sp|Q03460|GLSN_MEDSA GLUTAMATE SYNTHASE (NADH) PRECURSOR (EC 1.
- sp|P24944|GLTT_BACCA PROTON/SODIUM-GLUTAMATE SYMPORT PROTEIN (G
- sp|P05373|HEM2_YEAST DELTA-AMINOLEVULINIC ACID DEHYDRATASE (EC
+ sp|P31903|HYPD_ALCEU HYDROGENASE EXPRESSION/FORMATION PROTEIN H
+ sp|P31882|HYPD_AZOVI HYDROGENASE EXPRESSION/FORMATION PROTEIN H
+ sp|P31904|HYPD_BRAJA HYDROGENASE EXPRESSION/FORMATION PROTEIN H
+ sp|P24192|HYPD_ECOLI HYDROGENASE ISOENZYMES FORMATION PROTEIN H
+ sp|P40598|HYPD_RHILV HYDROGENASE EXPRESSION/FORMATION PROTEIN H
+ sp|P26411|HYPD_RHOCA HYDROGENASE EXPRESSION/FORMATION PROTEIN H
+ sp|P13129|LUCI_LUCCR LUCIFERIN 4-MONOOXYGENASE (EC 1.13.12.7) (
+ sp|Q01158|LUCI_LUCLA LUCIFERIN 4-MONOOXYGENASE (EC 1.13.12.7) (
+ sp|P08659|LUCI_PHOPY LUCIFERIN 4-MONOOXYGENASE (EC 1.13.12.7) (
+ sp|P00467|NIFD_CLOPA NITROGENASE MOLYBDENUM-IRON PROTEIN ALPHA
+ sp|P20620|NIFD_METTL NITROGENASE MOLYBDENUM-IRON PROTEIN ALPHA
+ sp|P08293|NIFE_AZOVI NITROGENASE IRON-MOLYBDENUM COFACTOR BIOSY
+ sp|P26506|NIFE_BRAJA NITROGENASE IRON-MOLYBDENUM COFACTOR BIOSY
+ sp|P10996|NIFE_CLOPA NITROGENASE IRON-MOLYBDENUM COFACTOR BIOSY
+ sp|P08737|NIFE_KLEPN NITROGENASE IRON-MOLYBDENUM COFACTOR BIOSY
+ sp|P19055|NIFE_RHOCA NITROGENASE IRON-MOLYBDENUM COFACTOR BIOSY
```

Figure 3.4: Database sequences where the pattern [FY]GXPxxx[AG]xFx[GP][ILVM] corresponding to the second block at the right of Figure 3.2) is found allowing for up to 1 error. The block is no longer present in the sequences flagged with a minus sign when no error is permitted at the positions in the block where a strict conservation of a single amino acid is observed (those indicated with a star in Figure 3.2). Continues next page.

```
- sp|P34755|PHYB_ASPAW 3-PHYTASE B PRECURSOR (EC 3.1.3.8) (MYO-IN
- sp|P34754|PHYB_ASPNG 3-PHYTASE B PRECURSOR (EC 3.1.3.8) (MYO-IN
+ sp|P22221|PODK_FLATR PYRUVATE,ORTHOPHOSPHATE DIKINASE PRECURSOR
- sp|Q05057|POLG_PYFV1 GENOME POLYPROTEIN (CONTAINS: 22.5 KD PROT
+ sp|P06186|PRGR_RABIT PROGESTERONE RECEPTOR (PR).
+ sp|P05168|PSBE_TOBAC CYTOCHROME B559 ALPHA CHAIN.
- sp|P37189|PTKC_ECOLI PTS SYSTEM, GALACTICOL-SPECIFIC IIC COMPON
+ sp|P38972|PUR4_YEAST PHOSPHORIBOSYLFORMYLGLYCINAMIDINE SYNTHASE
+ sp|P21750|SALA_DROME PROTEIN SPALT-ACCESSORY.
+ sp|P21748|SALA_DROOR PROTEIN SPALT-ACCESSORY.
+ sp|P21749|SALA_DROSI PROTEIN SPALT-ACCESSORY.
+ sp|P07819|SCRB_BACSU SUCROSE-6-PHOSPHATE HYDROLASE (EC 3.2.1.26
+ sp|P33388|SEFC_SALEN SEFC PROTEIN PRECURSOR.
- sp|P04802|SYDC_YEAST ASPARTYL-TRNA SYNTHETASE, CYTOPLASMIC (EC
- sp|Q03577|SYD_CAEEL PROBABLE ASPARTYL-TRNA SYNTHETASE (EC 6.1.1
- sp|P21889|SYD_ECOLI ASPARTYL-TRNA SYNTHETASE (EC 6.1.1.12) (ASP
- sp|P14868|SYD_HUMAN ASPARTYL-TRNA SYNTHETASE (EC 6.1.1.12) (ASP
- sp|P15178|SYD_RAT ASPARTYL-TRNA SYNTHETASE (EC 6.1.1.12) (ASPAR
- sp|P36419|SYD_THETH ASPARTYL-TRNA SYNTHETASE (EC 6.1.1.12) (ASP
- sp|P30053|SYH_STREQ HISTIDYL-TRNA SYNTHETASE (EC 6.1.1.21) (HIS
- sp|P14825|SYK2_ECOLI LYSYL-TRNA SYNTHETASE, HEAT INDUCIBLE (EC
- sp|P15180|SYKC_YEAST LYSYL-TRNA SYNTHETASE, CYTOPLASMIC (EC 6.1
- sp|P32048|SYKM_YEAST LYSYL-TRNA SYNTHETASE, MITOCHONDRIAL PRECU
- sp|P37477|SYK_BACSU LYSYL-TRNA SYNTHETASE (EC 6.1.1.6) (LYSINE-
- sp|P41258|SYK_CAMJE LYSYL-TRNA SYNTHETASE (EC 6.1.1.6) (LYSINE-
- sp|P37879|SYK_CRILO LYSYL-TRNA SYNTHETASE (EC 6.1.1.6) (LYSINE-
- sp|P41255|SYK_THETH LYSYL-TRNA SYNTHETASE (EC 6.1.1.6) (LYSINE-
+ sp|P28176|TYSY_MYCTU PUTATIVE THYMIDYLATE SYNTHASE (EC 2.1.1.45
- sp|P09842|UGST_HORVU GLYCOGEN (STARCH) SYNTHASE PRECURSOR (EC 2
- sp|P04713|UGST_MAIZE GLYCOGEN (STARCH) SYNTHASE PRECURSOR (EC 2
- sp|P19395|UGST_ORYSA GLYCOGEN (STARCH) SYNTHASE PRECURSOR (EC 2
- sp|P27736|UGST_WHEAT GLYCOGEN (STARCH) SYNTHASE PRECURSOR (EC 2
+ sp|P27870|VAV_MOUSE VAV PROTO-ONCOGENE.
+ sp|P03224|VGP8_EBV PROBABLE MEMBRANE ANTIGEN GP85.
- sp|P03234|VP40_EBV CAPSID PROTEIN P40 (VIRION STRUCTURAL PROTEI
- sp|P37340|YDHE_ECOLI HYPOTHETICAL PROTEIN IN RIBC 5'REGION (FRA
- sp|P40468|YIM9_YEAST HYPOTHETICAL 269.9 KD PROTEIN IN RPL13-STH
- sp|P40893|YJ67_YEAST HYPOTHETICAL 22.0 KD PROTEIN IN CRT1 5'REG
- sp|P34428|YL37_CAEEL HYPOTHETICAL 45.5 KD PROTEIN F44B9.7 IN CH
```

Figure 3.4: (Continuation from previous page) Database sequences where the pattern [FY]GXPxxx[AG]xFx[GP][ILVM] corresponding to the second block at the right of Figure 3.2) is found allowing for up to 1 error. The block is no longer present in the sequences flagged with a minus sign when no error is permitted at the positions in the block where a strict conservation of a single amino acid is observed (those indicated with a star in Figure 3.2).

one of the direct occurrences is included in the other (this may happen when indels are authorized), then it seems obvious that we should keep the direct occurrence presenting least errors against the model. Another easy case to treat is that of two pairs of (indirect, direct) occurrences, one of which is completely included in the other pair (this may happen when models of more than one length are sought). However, such clear-cut cases are rare. A direct occurrence may also pair with two indirect ones. If the two are distinct, both should be kept (for instance as candidates for forming part of a more complex structure). What decision should we make however if the two indirect occurrences overlap: keep them both, keep only one ? By how much should they overlap for us to

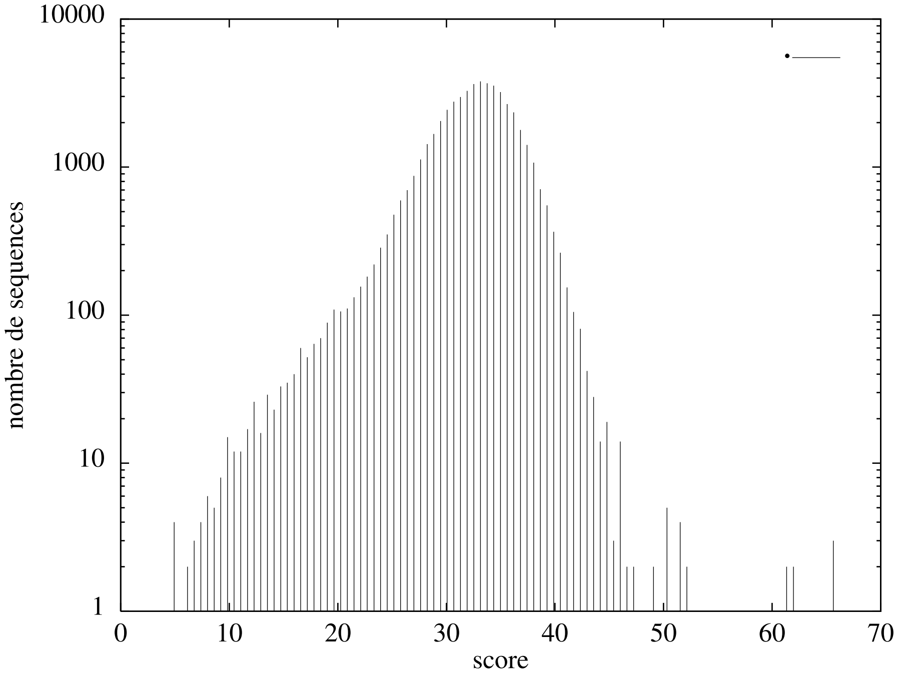

Figure 3.5: Distribution of the scores obtained by searching in the database for the two blocks shown in Figure 3.2 simultaneously.

consider that they are "distinct" ?

Once we arrive at a clear formalization of the problem, if such were possible, there will remain the task of obtaining efficient algorithms for eliminating the identified redundant elements. We are still for now working at formalizing the issue although this is not, and can not be completely divorced from algorithmical concerns.

3.4.1.2 RNA Structure Comparison

Algorithms for comparing RNA structures are surpringly rare [46] [60] [178] [205] when one remembers that this is such an important problem, whether one is comparing RNAs whose structures are known or just predicted. It is also a difficult problem. Indeed, Evans's work shows that most formulations of it lead to algorithmically hard problems [60] [116].

The subject has also in general been tackled by using simplified representations of an RNA structure. One such possible representation is, for instance, as a tree [183] [184]. Only the secondary structure is considered in such case but this should lead to better performing algorithms. This is the model with which we wish to start. We were led to it by the work we have been doing on a different subject: the detection of the presence and exact location of recombination events in a set of aligned sequences. As will be briefly sketched in Chapter 5, one combinatorial approach to such detection has led us into some tree problems, among which are the search for maximal

```
sp|P42033|HUPD_AZOCH HYDROGENASE score : 66.25
[1]      57       70.11     RLIHGP-GC
[2]      219      62.38     YGTPLVIAGFEPL
sp|P31882|HYPD_AZOVI HYDROGENASE score :  66.25
[1]      57       70.11     RLIHGP-GC
[2]      219      62.38     YGTPLVIAGFEPL
sp|P08737|NIFE_KLEPN NITROGENASE score : 66.05
[1]      54       75.56     HLVHGPIGC
[2]      265      56.54     YGTPWFEGSFYGI
sp|P31903|HYPD_ALCEU HYDROGENASE score : 65.52
[1]      57       68.11     RMIHGP-GC
[2]      220      62.92     YGKPVVIAGFEPL
sp|P31904|HYPD_BRAJA HYDROGENASE score : 64.82
[1]      24       68.11     RMIHGP-GC
[2]      187      61.54     FGKPVVIAGFEPL
sp|P26506|NIFE_BRAJA NITROGENASE score : 64.19
[1]      62       72.00     HLVHGPIAC
[2]      263      56.38     YGIPFFEGSFYGI
sp|P40598|HYPD_RHILV HYDROGENASE score :  62.21
[1]      57       68.11     RMIHGP-GC
[2]      220      56.31     FRKPVVVAGFEPL
sp|P08293|NIFE_AZOVI NITROGENASE score : 62.16
[1]      54       67.78     HIVHGPIAC
[2]      265      56.54     YGTPWFEGSFYGI
sp|P10996|NIFE_CLOPA NITROGENASE score :  61.89
[1]      71       75.56     HLVHGPIGC
[2]      270      48.23     FGIPYIKVSFYGV
sp|P26411|HYPD_RHOCA HYDROGENASE score : 61.39
[1]      62       62.56     EFIHGP-GC
[2]      218      60.23     YGKPLVVAGFEPT
sp|P19055|NIFE_RHOCA NITROGENASE score : 59.25
[1]      63       61.89     HLVHAPAAC
[2]      265      56.62     YGIPYFEGSFYGI
sp|P24192|HYPD_ECOLI HYDROGENASE score : 58.59
[1]      62       62.56     EFIHGP-GC
[2]      218      54.62     FHRPLVVAGFEPL
sp|P20620|NIFD_METTL NITROGENASE score :  57.12
[1]      64       73.56     HMVHGPIGC
[2]      275      40.69     FDIPYTRATFFGL
sp|P00467|NIFD_CLOPA NITROGENASE score : 53.60
[1]      70       65.67     HITHGPIGC
[2]      276      41.54     YGIPWIKCNFIGV
sp|P06662|NIFD_THIFE NITROGENASE score : 52.38
[1]      89       62.00     HISHGPVGC
[2]      299      42.77     YGIPYIEFNFFGP
sp|P08738|NIFN_KLEPN NITROGENASE score:  52.24
[1]      36       56.78     PLVHGAQGC
[2]      391      47.69     FALPLVRAGF-PL
sp|Q00239|NIFD_PLEBO NITROGENASE score :  52.04
[1]      88       62.00     HISHGPVGC
[2]      303      42.08     YGIPWLEYNFFGP

sp|P07328|NIFD_AZOVI NITROGENASE score : 51.62
[1]      79       62.00     HISHGPVGC
[2]      289      41.23     YGIPWMEYNFFGP
sp|P15332|VNFD_AZOCH NITROGENASE score : 51.61
[1]      65       65.22     QMIHGPLGC
[2]      425      38.00     NG-PYM--GFEGF
sp|P16855|VNFD_AZOVI NITROGENASE score : 51.61
[1]      66       65.22     QMIHGPLGC
[2]      426      38.00     NG-PYM--GFEGF
sp|P25313|NIFD_AZOBR NITROGENASE score : 50.92
[1]      80       62.00     HISHGPVGC
[2]      291      39.85     FGIPWMEYNFFGP
sp|P06769|NIFD_RHICP NITROGENASE score : 50.77
[1]      85       62.00     HISHGPVGC
[2]      298      39.54     FGIPWCEYNFFGP
sp|P06120|NIFD_BRASP NITROGENASE score :  50.77
[1]      85       62.00     HISHGPVGC
[2]      298      39.54     FGIPWCEYNFFGP
sp|P06121|NIFD_BRAJA NITROGENASE score : 50.77
[1]      85       62.00     HISHGPVGC
[2]      298      39.54     FGIPWCEYNFFGP
sp|P00464|NIFD_ANASP NITROGENASE score : 50.69
[1]      82       62.00     HISHGPVGC
[2]      297      39.38     YGMPWMEFNFFGP
sp|P16266|ANFD_AZOVI NITROGENASE score : 50.04
[1]      67       62.00     HISHGPVGC
[2]      272      38.08     YGIPrLDIDGF-GF
sp|Q07933|ANFD_RHOCA NITROGENASE score : 49.65
[1]      67       61.22     HISHGPNGC
[2]      272      38.08     YGIPrLDIDGF-GF
sp|P00466|NIFD_KLEPN NITROGENASE score : 49.26
[1]      81       62.44     HISHGPAGC
[2]      300      36.08     FG-PTKIA--ESL
sp|Q02452|NIFD_FRAAL NITROGENASE scrore : 48.41
[1]      82       56.67     TISHGPVGC
[2]      289      40.15     FGTPWTEFNFFGP
sp|P06122|NIFK_BRASP NITROGENASE score : 47.48
[1]      85       52.11     PFVHGSQGC
[2]      452      42.85     TGTPLIRIGF-PI
sp|P30268|YKAB_BACFI HYPOTHETICAL score : 47.36
[1]      74       59.33     HLTHGPIGV
[2]      127      35.38     TKRPYVV-SFTGC
sp|P08717|NIFD_RHOCA NITROGENASE score : 47.23
[1]      95       62.00     HISHGPVGC
[2]      344      32.46     Y-RPLV-DGI--L
sp|P19077|NIFN_RHOCA NITROGENASE score : 47.10
[1]      36       50.89     PLFHGAQGC
[2]      392      43.31     LHLPLVRAGF-PI
```

Figure 3.6: Database sequences where the two blocks shown in Figure 3.2 are found with a score above 47 when aligned, as profiles, using a dynamic programming algorithm developed by Alain Viari. The sequences appear by decreasing order of score.

homeomorphic subtrees common to a set of trees[3] [3] [62] [63]. We believe this, and a previous algorithm on isomorphic subtrees[4] due to Karp [122], could be adapted and extended to the problem of RNA structure comparison in a way that is more pertinent than the method used by Shapiro in [184]. In his case, the trees corresponding to RNA structures are compared in a global way. It has been suggested, by Bouthinon *et al.* [30] and Gorodkin *et al.* [85] among others, that this is biologically unrealistic. We agree with this. A model based on homeomorphic subtrees, could provide a better solution. Shapiro's tree model (where stems are represented by unweighted arcs

[3]Two trees are said to be homeomorphic if one can be obtained from the other by the creation of arcs in series (*i.e.* by the insertion of nodes of degree two) or by the merger of arcs in series.

[4]Two trees are said to be isomorphic if there is a one-to-one correspondence between their nodes and arcs such that the incidence relationship is preserved.

[183] or nodes [184] and loops by unweighted nodes) may also be enriched to take other constraints of the structure into account (length of a loop or stem, elements of sequence etc).

3.4.1.3 Towards RNA Structure Prediction

RNA structure prediction is a very hard problem. The approach we want to follow would use the information provided by a set of sequences instead of a single one like in Zuker's Mfold algorithm [208] [207]. The sequences in the set are presumed to adopt a same RNA fold. This is the approach used also in [30], [85] and a number of other algorithms. Bouthinon's method [30] in particular is interesting in the sense that the sequences do not need to be aligned and that his method allows to predict some tertiary interactions (basically, pseudo-knots) as well as secondary. It has, however, the limitation that the common substructures identified for a set of RNAs must be composed of palindromic elements that are contiguous along the sequence.

Very briefly, the idea behind his algorithm is to first determine the potential structures for each sequence in the set (such structures correspond to a maximal subset of compatible palindromes with “low enough” energy). The predicted potential structures are then encoded in such a way that the problem is reduced to one of finding the largest repeated word suffixes in a dictionary.

One of the paths we wish to explore is based on these ideas but would represent the predicted potential structures under the form of trees and then adapt (and possibly improve) the existing algorithms for identifying maximal homeomorphic subtrees in a set of trees in order to infer the maximal substructure common to the set of sequences (see previous section).

Another, different approach we want to examine rests on the observation that an RNA structure is often conserved both in terms of its topology and of some of its sequence elements. These elements may represent very short motifs. Such motifs present however the further characteristic of being embedded in an environment with “high pairing potentiality”. In other terms, such motifs should be surrounded, or even be part of, potential stems. The environment with “high pairing potentiality” could be “big” or “small”. An example of the latter are terminators [38]. The motifs could represent something more flexible than the kinds of models we have introduced so far. Some could be conserved in terms of their overall composition rather than sequence. Pragmatism could go to the point of taking into account in the model pairings experimentally known to exist [124].

3.4.2 Critique and Perspectives on Proteins

3.4.2.1 Conjugating EM/Gibbs Sampling Approaches with Models

It is possible to choose the set of sequences T that permits to evaluate the models equal to the set E of example sequences from which the models were inferred. In this case, our algorithm behaves in a way even more akin, but still not identical, to an EM algorithm or Gibbs sampler [13] [14] [15] [16] [127] [128] applied to a combinatorially obtained set of starting hypotheses (the models) than by using the refinement described in section 3.3.3.2.6. As suggested there, and in a way similar to what was proposed for the structured models in DNA sequences of the previous chapter, we could consider using the models identified in the sampling step of the algorithm described in section 3.3.3.1 as seeds to an EM or Gibbs sampling method.

Whether one choses the above approach or another one, it is important to produce better methods for evaluating models. It has indeed been shown by Bailey and Gribskov [17] that Goldstein and Waterman's formula errs on the optimistic side. The reason is that their calculation is valid only if one assumes infinitely long profiles. The analysis we have made so far shows us also that, although a valid set of occurrences may find itself amid the models identified by the sampling step, it is not always well classified in the final evaluating scheme. There can be various reasons for that, not all of which are related to the statistical methods used to ascertain the p-value of a profile score, but it is clear that there is room for improvement as concerns these methods.

3.4.2.2 Tiling Models

As for nucleic acid sequences (whether one is searching for sequence or structural motifs such as palindromes), a flexible search for models in a set of proteins (that is, a search allowing for a physico-chemical alphabet, a number of wild cards – corresponding to positions not conserved in a site – and for errors – imperfect conservation of some positions) may produce a lot of models. For instance, in the case of the example given in section 3.3.4, 116607 models are obtained at the end of the sampling step, 77732 of which are included in or identical to another model (in terms of theirs sets of occurrences). Such redundancy may appear even more absurd than in the case of models for regulatory sequences in the sense that, at least in the current algorithm for proteins, it is only the sets of occurrences that interest us. Models serve the sole purpose of enabling to identify such sets in a well formalized and more efficient way. The included or identical sets of occurrences are for now eliminated using a quasi-naive algorithm that may be very inefficient (far more than the sampling step itself). We have therefore started considering an extension and improvement of the algorithm developed by Parida *et al.* for obtaining a basis of models for proteins as well as for DNA sequences (the case of RNAs is more complex).

3.4.2.3 Towards Introducing Spatial Information

The algorithm presented in section 3.3 was aimed, among other things, at identifying, from the sequence alone, elements in a protein that are conserved in terms of the structure. However, there are cases where structural information concerning a protein is at least partially known. As it stands now, our algorithm could not make use of such information. This is a problem we have been wanting to address for quite some time now: how to integrate partial structural information (possibly predicted) with sequence data to improve motif detection ?

A different problem concerns analysing protein structures *per se*. As very briefly mentioned in sections 3.2.3 and 3.3.1, this problem was addressed by Joël Pothier, Henry Soldano and Alain Viari [115] [164] [174] in a way that led them to encode such structures by strings of discretized pairs of angles to which they could then apply regular string algorithms.

The idea of trying to encode protein structures in the form of strings is a powerful one. In general, this restricts the motifs which may be identified to represent contiguous structural motifs but it was suggested at the end of Escalier's thesis [58] that this might not always be the case. Indeed, recently, Jonassen *et al.* established an original approach were the spatial neighbourhood of each residue along a protein chain is represented by a string of the spatially close residues (*i.e.*, having spatial distance below some threshold). The strings in this case do not contain information about the exact geometry of the neighbours, but they do give the sequence order along the backbone of the neighbouring residues. Sequence pattern extracting methods, such as the one previously developed by Jonassen *et al.* (Pratt), can then be applied to the sets of strings obtained. The approach has been shown to be capable of identifying local packing motifs that correspond well to known sites or motifs in sets of very disparate proteins [119].

This is one path we wish to explore further, and we are convinced it is not the only one possible.

Chapter 4

Genome Structure

Students whose work is presented in this chapter

Philippe Blayo for the work on orphan gene finding

Eric Poiret for a three months implementation work on trying to improve the interface for the satellites problem

Laurent Marsan for initial thinking on finding long repeats

Nadia Pisanti for initial thinking on finding long repeats

Co-workers and/or collaborators

Vincent Escalier, Joël Pothier, Henry Soldano and Alain Viari for the earlier work on general (possibly dispersed) repeats

Pierre Rouzé for the work on orphan gene finding

Thomas Schiex, Christine Gaspin and the computational biology team at the INRA Toulouse (gene finding)

Alain Viari and François Rechenmann (gene finding)

Catherine Mathé (gene finding)

Claude Thermes, Yves d'Aubenton Carafa, Alain Viari and Joël Pothier for the earlier (and future) work on splicing

Gene Myers for all the work on satellites and initial ideas on finding long repeats

Maxime Crochemore and Laurent Mouchard for initial thinking on finding long repeats

People who gave advice or with whom this work was many times discussed

Pierre Rouzé's team at Gand, in particular Sébastien Aubourg and Dimitri Holsteens (gene finding)

Laurent Duret (gene finding)

Alain Viari (satellites)

People from whom we "stole" pieces of code

Lars Arverstad (small bits of code for orphan gene finding algorithm)

Gene Myers for the filter in the satellites algorithm

4.1 Introduction

This chapter is something of a rattle bag. It gathers together two subjects which may be seen as whole domains of research in themselves. Both refer in some way to "genome structure", that is, to a division of the genome into well identified units. The units we shall be concerned with are genes and repeated elements (dispersed or in organized fashion).

The first domain of research covered will therefore be gene finding. This an important but difficult problem both in terms of biology and algorithmics. In eukaryotes, in contrast to prokaryotes, the problem is compounded by the presence of introns that interrupt the coding regions of genes, the exons. Finding a gene in the case of eukaryotes, as will be our exclusive concern in this memoir, implies therefore not only finding the beginning and end of the gene in a genomic sequence, but also determining the internal structure of the gene, that is, the beginning and end of each exon. What we present below, and the question we have addressed so far, is a very specific one concerning finding the structure of a gene that has no known homolog in the currently existing databases. Genes like these have been called in the literature *orphan genes* [53]. They are orphan in the sense that they can not, based on sequence homology alone, be connected to any known gene family. This question is treated in section 4.2 where we also refer to two papers submitted [29] or in preparation on the subject.

The question is not really, as currently formulated, biological *stricto sensu*. The biological question that would correspond to the gene finding problem would rather concern understanding how the limits, external and internal, of genes are "identified", not by a human with or without the help of a computer, but by the eukaryotic cellular machinery. In other terms, the biological question would refer to how some signals are recognized that allow for a gene to be correctly transcribed, spliced and translated. Such questions, in particular as regards the recognition of splicing sites and splicing enhancers, have been addressed in a preliminary way. The work planned for the future is briefly discussed (section 4.2.2).

The second domain of research we present in this chapter (section 4.3) concerns repeats of various types. The biological purposes served by these repeats, when they are known, appear to be so diverse that one can hardly talk of one specific biological question in this case. The initial question is rather to try to identify the presence of such repeats, in particular in a genomic sequence, and then see what the observed may tell us about genome structure and, possibly, evolution. The observed will in general vary greatly from one organism to another even inside a same cellular domain. Some of the biological questions we can ask may then be, why is there such variety, and what does this variety tell us about possible function, evolution, environment pressure and selection.

4.2 Genes

4.2.1 (Orphan) Gene Finding by Humans

4.2.1.1 Foreword

Many algorithms and software for gene finding in eukaryotes have been elaborated, particularly as sequencing projects for whole genomes have proliferated in the last few years. One may consult the web page whose URL is http://linkage.rockfeller.edu/wli/gene/right.html for pointers to an extensive bibliography and references on the subject. Currently published algorithms call a gene what represents to the biologist the "protein coding region of genes". What is denoted by an exon corresponds therefore to a "coding exon" as opposed to a "non-coding exon". This last may be found in a gene due to the possible presence of introns in the 5' and 3' non-coding parts of a gene. This is a simplification of reality that we also adopt for ease of exposition. We therefore henceforward use the terms gene and exon to denote what are in fact, respectively, the protein coding region of genes and a coding exon.

Current algorithms commonly tackle the problem of gene finding by using one or, more generally, a combination of three basic approaches. The first two may be called intrinsic, they use information contained in the sequence only, while the last is extrinsic, it is based on information coming from outside the sequence as well as inside.

The first approach consists in identifying genes and their exon/intron structure based on an observed difference of composition between intergenic, interexonic and exonic regions ([68] [185] [188] were the pioneer work in this area). This is most often not a fine enough criterion to accurately establish the exact frontiers of genes, and of exons inside genes, at least in eukaryotes. The method may also be strongly organism-dependent.

The second approach tries to improve the localization of the gene and exons frontiers by further identifying some signals that occur at, or near the limits between gene/intergene on the one hand, and exon/intron on the other [33] [35] [112] [193] [203]. The signals are in general not exact; indeed, they may often be quite degenerate in relation to a consensus. Those typically considered are transcriptional start points, splicing sites and poly-A gene tails. The splicing sites taken into account are mostly donors and acceptors. Branch points may also be searched for as they may help to find acceptors. These are all signals occurring in introns, donors at the beginning and acceptors at the end whereas branch points are located near the end of introns. Promoters have been suggested as important signals to help identify gene beginnings but they are currently considered too difficult to model [33] [35]. Chris Burge is the only one at the present time who incorporates a simplified promoter model (for the TATA-box) in his program (GenScan [33] [35]). Like the first approach, this second may also be organism-dependent. It is in a likewise manner based on information about the organism that has to be learned before it can be used. Both approaches are statistical in nature.

The third and, historically, the last approach[1] to be used for gene finding has been homology with cDNAs (sometimes ESTs) or with previously determined genes, often the protein products of the genes that one may find annotated (not always accurately) and stored in databases. Algorithms using such an approach proceed in general in two successive steps: in a first step they find candidate exons in a DNA sequence believed to contain a gene and then, in a second step, they select the best among all subsets of candidates that may be assembled to form a potential gene. Selection of the best is done by comparison of the potential gene, as a DNA sequence [70] or after translation to a protein [77] [78] [79], to known cDNAs or database proteins. This second step is called exon assembly [92] or spliced alignment [78] [79]. Candidate exons may be found by any method, including either predictive like the two mentioned above (based on a difference of composition and/or the presence of sometimes weak signals) or exhaustive. The possibility of exhaustive enumeration comes from the known fact that (i) eukaryotic genes start with a particular codon (most often, but not exclusively, this is ATG) and end with one of three possible stop codons (TAA, TAG or TGA) and (ii) donors start with one among a few specific dinucleotides, usually this is GT (for introns of U2 type), or, much less frequently, GC, and acceptors also end with a specific dinucleotide which in general is AG. An exon is therefore always something framed by two known dinucleotides, or starting/ending

[1]In terms of algorithmics because in terms of manually performed analyses, this represents the first approach.

with a given triplet. Gelfand *et al.* [78] [79] call "site mode" the exhaustive method for producing a best spliced alignment, and "exon mode" or "exon/intron mode" methods based on a preliminary filtering of potential exons by using a score that combines the strengh of predicted splicing sites with the coding potential of the exon. Depending on the method used, a large number of candidate exons may be produced.

It appears [36] [44] [69] [148] that the homology approaches, alone or in conjunction with the other two (prediction based on content or presence of signals), produce the best results for gene finding. However, as currently formulated, homology considerations may only be used when an homolog of the gene one is trying to predict is already present in the database and, incidentally, when the database gene has itself been accurately annotated. Both assumptions are often not true. In the case of *Saccharomyces cerevisiae* (yeast) for instance, approximately 40% of predicted genes have no known homologs in the databases [53], that is, are orphans. This rate varies between organisms but is often as high as for yeast. Some statistics exist for the number of inaccurately annotated genes (see [192] for eukaryotes – the case of prokaryotes is a little less dramatic). They point to the poor status of current sequence databases and to the great number of annotation errors they contain. This number is bound to grow as whole genome sequences are spurt out and often summarily analysed at ever increasing speeds. Very few algorithms take into account the possibility of such errors, that may happen anywhere, including within exons. The result is that what is often detected (by programs such as tblastx [1] or Procrustes [78] [79] for instance) are incomplete pieces of exons or exons that do not contain any frameshifts.

One may still use homology to predict in a more reliable way the internal structure of orphan genes (and one may include here genes with very weak homologs or with homologs possibly inaccurately annotated) if one considers that an orphan may have brothers that are orphans like itself. Orphan brothers may correspond to ortholog or to paralog genes, that is, genes that are the result of a duplication event in a same organism. Although the gene structure of such homologous genes is often similar, *i.e.*, they have the same number of exons in the same order, this structure may neverthess differ (*e.g.*, by addition or loss of introns and/or of domains). Comparing homologous orphan genes to better determine their internal structure is the main problem we address.

The objects we consider are, for now, two DNA sequences, each potentially containing one gene, plus possibly some intergenic parts both upstream and downstream from the gene. Moreover, it is assumed that the gene in one sequence is homologous with the gene in the other. Although we aim at finding a gene structure using two homologous genomic sequences, we can treat as a particular case the problem of gene finding using a cDNA, possibly translated, as one of the homologous genes [70]. The case of alternative genes (*i.e.*, of one DNA sequence being read in various ways, differing in the exon structure due to the use of alternative splice sites, or alternative sites of transcription initiation or termination) is more complex to handle and has only started to be considered. In both biological and algorithmical terms, one could approach the problem we are contemplating in two basic ways. One way corresponds to an exclusively exonic model of a gene, that is to a model that considers a gene as a set of independent coding exons, each exon corresponding to a "well-conserved" fragment of two homologous genes. The second approach sees a gene as a set of coding exons that are pieces of an assembly and are, therefore, not independent. Each internal piece is thus related to the exons coming just before and after, as well as to the neighbouring introns. We have considered both models, but have concentrated so far more on the second one, that is, on a coding exon assembly approach to orphan gene finding for which we introduced a first algorithm [29].

If the statistical properties of exons are still poorly understood in the case of one organism as mentioned by Gelfand *et al.* [78] [79], the use of statistics becomes even trickier when potentially two organisms are concerned, as will often be the case for our orphans. The approach adopted rests therefore on a model by "pure" homology. This means in particular that all potential left and right exon frames are considered. Although not using other assessable biological information such as content and presence of signals may weaken the predictive power of our algorithm, this last is completely organism-independent. This is important for our purposes. Our aim is not to replace or even compete with currently developed gene finding packages, but to try to complement them where these packages behave poorly or fail to provide a solution (among others, finding the structure of orphan genes).

The models adopted (for an eukaryotic gene and for comparing DNA sequences that contain both coding – the exons – and non coding parts – mainly the introns but also the intergenic regions) are introduced in section 4.2.1.2. The two algorithms elaborated so far for finding one best assembly are informally and succintly sketched in section 4.2.1.3. Readers wishing for the full details, including the data structures used, recurrence formulae and complexity analyses, are invited to read [29]. A few initial applications are presented in section 4.2.1.3.3. Currently on-going work, in particular concerning finding all assemblies within a given maximum distance of the optimal and improving the "gene as a set of well-conserved coding exons" approach, are briefly discussed in sections 4.2.1.3.4 and 4.2.2.

4.2.1.2 Models

4.2.1.2.1 Coding DNA Comparison Model

Up to now, there have been essentially three models for comparing DNA coding sequences. All are based on codon (*i.e.*, triplet) alignments in the sense that, for each triplet of nucleotides in both sequences that are matched as amino acids:

- the number of gaps that may be present at codon boundaries is a multiple of three;
- the number of gaps internal to the codon that are allowed is either zero or may be positive (in this case, the nucleotides in the triplet considered will not appear consecutive in the sequence) but is a multiple of three.

The models vary therefore according to the number and type of gaps permitted, and also whether frameshifts are considered, and how.

Hein suggested in [100] [101] that there are 11 possible types of codon alignments (see Figure 4.1) but in fact, more may be considered [149].

The most sophisticated model established so far [100] [101] [108] [150] for comparing two coding DNA sequences allows for all the 11 codon alignment types suggested by Hein. Observe that in this case, the order of the events that lead to a particular codon alignment (nucleotide substitution or gap first) may not be indifferent and has to be taken into account. For complexity reasons, these algorithms do not consider the possibility of sequencing errors, that is, they do not allow for frameshifts, although incorporating this within the model is not impossible as is indicated at the end of both Pedersen's and Hua's papers. However, although this is an interesting model (once it is revised and improved upon) for exploring gene evolution, it is not the most appropriate for our purposes.

The simplest coding DNA comparison model allows for codon alignments of types 1, 2 and 3 only (see Figure 4.1). They therefore perform the comparison entirely at the protein level.

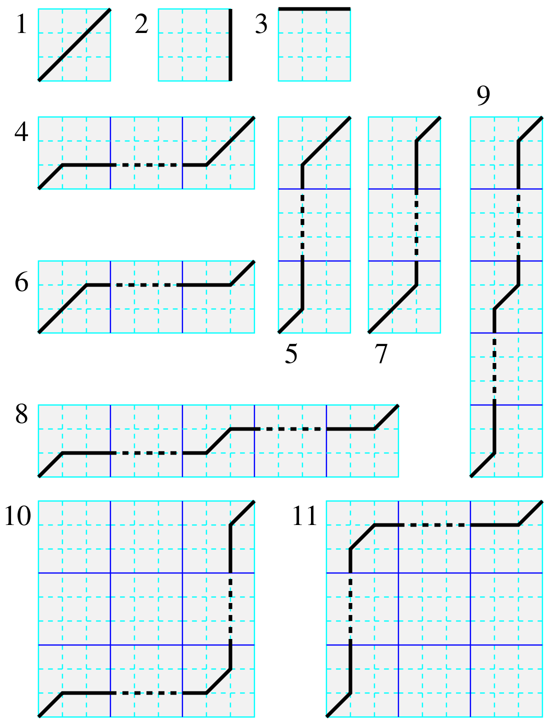

Figure 4.1: Codon alignments.

The third and last currently established class of coding DNA models also treat codon alignments of types 1, 2 and 3 only, as for the protein/protein model. However, they further consider the possibility of frameshifts [5] [91]. An extended alphabet of codons $\Sigma = N^0 \cup N^1 \cup \ldots \cup N^c = N^{\leq c}$ [2] may thus be used for such purpose, where c is a positive integer, $N = \{A, C, G, T\}$ and $N^0 = \{\lambda\}$ (λ denoting the empty word). In Arvestad [5], $\Sigma = (N^{\leq 5} - N^0)$. Frameshifts due to sequencing errors may therefore correspond to the insertion or deletion of one or two nucleotides anywhere in a codon. The score of a pair of codons from the extended alphabet $(\tilde{x}, \tilde{y})$ is then the maximum of the scores of all pairs of triplets (x, y) with $x, y \in \overline{N}^3$ that may be obtained from $\tilde{x}$ and $\tilde{y}$ respectively where $\overline{N} = N \cup \{-\}$ ($-$ denoting an indel).

For our gene finding purpose, we adopt a slightly different version of Arvestad's model. In our case, the extended alphabet of codons Σ is equal to $N^{\leq 3}$. Our aim is to recover as gracefully as possible from a frameshift event due to a sequencing error, not to try to make a guess (based on an optimisation criterion that may seem arbitrary) of what that error was. For this reason also, we use a simpler scoring system for codons from the extended alphabet that is as follows. Given $\tilde{x}, \tilde{y} \in \Sigma = N^{\leq 3}$, we have:

[2] The notation $N^{\leq c}$ indicates all words of length between 0 and c that can be written over the alphabet N, in this case, the alphabet of nucleotides.

$$w(\tilde{x},\tilde{y}) = \begin{cases} \mathcal{M}(\tilde{x},\tilde{y}) & \text{if } |\tilde{x}|,|\tilde{y}| = 3 \\ \gamma \times \min\{(|\tilde{x}|+|\tilde{y}|),(6-|\tilde{x}|-|\tilde{y}|)\} & \text{in all other cases} \\ & \text{except if } |\tilde{x}| = 0 \text{ and } |\tilde{y}| = 3 \\ & \text{or if } |\tilde{x}| = 3 \text{ and } |\tilde{y}| = 0 \\ & \text{both of which correspond to a gap} \end{cases} \tag{4.1}$$

where

- w is the weighting function for the extended alphabet of codons; it is defined from $\Sigma \times \Sigma$ to $\mathbb{Z}$ (the set of (positive and negative) integers);
- $\mathcal{M}$ is a protein scoring matrix such as PAM [52] or Blosum [103];
- $\gamma \in \mathbb{Z}$ is a penalty for frameshifts.

Frameshifts have therefore a constant cost instead of an affine cost as will be the case for gaps. This is motivated by algorithmical considerations (both treatment and exposition become easier) but also by the fact that we do not know whether the same reasoning as for gaps could be applied to frameshifts, that is, whether an error committed in "reading" a letter at a given position in a sequence makes it more probable that a second error will be committed just afterwards during sequencing.

4.2.1.2.2 Gene Model

We distinguish the gene model that is considered from the model that we use for finding a gene. The second is an algorithmical concept while the first is a biological one. To simplify exposition, we assume that the first codon of a gene is ATG, a donor always starts with a GT and an acceptor always ends with an AG. Considering other codon starts or donors (resp. acceptors) starts (resp. ends) does not increase the asymptotic time complexity of the algorithm and can easily be implemented as a parameter.

The biological model of gene we adopt is thus the following:

- a gene begins with an ATG;
- a gene ends with a stop codon (TAA, TAG or TGA);
- a gene contains any positive number of compatible exons, possibly only one, where:
  - an exon starts with an ATG (first exon) or is framed upstream by an AG (internal exon);
  - an exon ends with a stop codon (last exon) or is framed downstream by a GT (internal exon);
  - an exon codes for part of a protein;
  - an exon is compatible with the exon that comes just before or after (if any) if it does not overlap it and the two are in phase (assuming there is no frameshift).

Two exons, $exon_1$ and $exon_2$ (considered in that order), are in phase if:

- either both end, respectively start, with a codon;
- or, in the case where a codon is interrupted by an intron, the number of nucleotides (1 or 2) at the end of $exon_1$ and the number of nucleotides (1 or 2) at the beginning of $exon_2$ add up to three (see Figure 4.2).

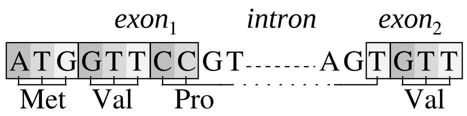

Figure 4.2: An example of codon in phase.

We call *dangling* end (or start) of an exon the nucleotides in a codon cut by an intron, and the nucleotides themselves, *dangling nucleotides*.

The algorithmical model we use to predict genes sees a gene as an ordered set of exons that correspond to the pieces of an assembly. Since exons code for proteins, they are scored as such. The scoring function used is the one given in equation 4.1 above. The score of an ordered set of exons assembled into a gene is the score of the assembly where gaps and frameshifts are allowed and exons frontiers are taken into account[3]. The cost of a gap is an affine function of its length l. It is thus $g_{open} + lg_{extend}$ where g_{open} is the cost for opening a gap and g_{extend} is the cost for extending the gap.

Finally, it is implicit in both the biological and algorithmical models that two homologous genes may have a different number of exons. In particular, one of the genes could have only one exon. It could even correspond to a cDNA. The cDNA/DNA comparison problem is thus a special case of the one we treat. In the case of different gene structures, possibly corresponding to the deletion of an intron in one or the other sequence, a penalty for such mutational event may be considered and easily incorporated into the algorithm.

4.2.1.3 Algorithms

4.2.1.3.1 Foreword for the Non-Specialized or Hurried Reader

Given the models introduced in the previous section, there are two problems we have addressed so far in terms of algorithms. One is finding a best assembly and the other is finding all assemblies having a score within a maximum distance of the optimal. For the reader not wanting to lose time and effort with even a sketchy description of the algorithms, we state below what these algorithms provedly identify and the complexity of the first one. After reading that, the reader may, if he wish, skip sections 4.2.1.3.2 and 4.2.1.3.4.

In the first case, given two sequences a and b of lengths m and n respectively, each one containing a gene that is homologous with the gene present in the other, a protein matrix function, an affine gap cost function and a scoring function for an extended alphabet of codons (that incorporates a frameshift penalty), the algorithm gives as result the set of exons that may be assembled into a gene verifying the biological model defined above, and has highest score among all sets.

In the second case, in the same circumstances, the algorithm gives as result all sets of exons that may be assembled into a gene verifying the biological model defined above, and having a score within a maximum distance of the optimal obtained among all possible sets.

[3]Whenever a gap crosses an intron, it is considered that two mutational events have taken place. The reason for this choice is that, although the gap appears continuous at the protein level, these events take place at different, often distant locations in at least one of the DNA sequences.

The technique used in both cases is dynamic programming, as in Procrustes [78] [79]. Procrustes has a time complexity in $O(mn)$ and a space complexity in $O(\min\{m,n\})$ (using the "site mode" approach) for comparing two DNA sequences of lengths m and n at the DNA level only, one of them being of known structure and without allowing for frameshifts (the complexity for comparing one error-free DNA sequence and a known protein is not given by the authors). The challenge was to obtain the same time and space complexities although treating the considerably more complex problem of comparing two DNA sequences presenting both an unknown structure and possibly containing sequencing errors. The algorithm for finding a best assembly for instance has a time complexity in $O(mn)$ and a space complexity in $O(\min\{m,n\})$. Both complexities are asymptotically independent of the number of potential exon candidates one wishes to consider. The constant factors in both cases, although bigger than for usual sequence comparison algorithms, are reasonable enough to make the algorithm efficient in practice. Details may be found in [29].

Although extreme cases have been observed, introns have in general a minimum length which may vary slightly between organisms, and which is often considered to be 40 nucleotides. It is not difficult to add this as a constraint to the algorithm. This requires however more space. The complexity will be multiplied by a term that is proportional to half of this minimum length.

4.2.1.3.2 Main Ideas for Finding a Best Assembly

The algorithm we propose for finding a best assembly consists in:

- comparing the sequences taking into account the fact that they are made up of an alternation of coding and non coding regions and using for this purpose a dynamic programming matrix (in fact three because we are using an affine gap cost function [86]);
- locating at the same time a best scoring set of exons verifying the biological gene model.

We present below a sketch of the algorithm's main ideas. Let a and b be the two sequences to be compared.

As we go comparing sequences a and b by filling up the dynamic programming matrices almost the usual way (from left to right, top to bottom but in our case, considering an extended alphabet of codons instead of nucleotides), every time an ATG is met in both a and b, we make the hypothesis that we are inside the first exon of the homologous genes in the two sequences.

In the same way, when an AG is encountered later on in any of the two sequences (possibly both), we could be entering a new exon of the homologous genes. All possible joins with the best assemblies that are compatible and end before it in both sequences have to be contemplated. We therefore need to store all information relative to such best assemblies.

Finally, when a GT is crossed, again in any of the two sequences (possibly both), we have to consider that, just before it, we reached the potential end of an exon, not necessarily the first one, in the sequence(s) where the GT was met. Let i, j denote the positions we had reached in a and b respectively when the GT was encountered. Since we have been keeping somewhere the information concerning the best assemblies ending before line i and column j (see above), it is easy to verify whether our exon could be compatible with some of them. If it is, we need to update the information concerning these best assemblies.

The information about the best assemblies reached so far in our traversal of the dynamic programming matrices can be grouped into three main types: two concerning best assemblies interrupted by an intron in one sequence only, and one concerning the best assemblies interrupted by an intron in both sequences. Each piece of information is then subdivided for both biological and algorithmical purposes into as many parts as there are possible last events for an exon (substitution, insertion or deletion) and, in the case of substitution, quantity of information (corresponding to the number of dangling nucleotides) that has to be memorized. Depending on that number, we choose to recall known nucleotides or anticipated ones (*i.e.*, the combination of nucleotides that may come at the start of a next exon).

The information just described is all we need to decide at the end what is the score of a best assembly. There are then three sets of recurrence formulae, one for each potential end of an exon in at least one sequence, one for each potential start of an exon in at least one sequence and, finally, one when neither of the above two cases applies. The first set is for updating the arrays that will keep the information concerning the last exon in the best assembly ending at a given column j for all j. The second set uses such arrays to calculate the values of a cell in any of the three matrices $\mathcal{S}$ (spliced alignment ending with a match or mismatch), $\mathcal{D}$ (spliced alignment ending with a deletion) and $\mathcal{I}$ (spliced alignment ending with an insertion). The values of the cell in all three cases will correspond to the best between a value obtained from the matrices at the appropriate place and a value obtained from the appropriate array. Finally, the last set of equations uses only what is already in the matrices to calculate the value of a new cell in each.

So far, we have a time complexity that is in $O(mn)$ and a space complexity that we can make linear in the length of the smallest sequence as for a "normal" pairwise sequence comparison.

The last problem we have to face consists in recovering a best assembly. The way we choose to go about this allows us to leave unchanged the asymptotic behaviour of both the time and space complexity of the algorithm. The idea here is reminiscent of the method developed by Hirschberg [105] for calculating the alignment (as opposed to just the score) of two sequences but resembles more closely the formulation given by Durbin [54]. Simply put, the core of this method uses the principle of divide and conquer to find the row (or column) where an optimal alignment crosses the middle column (resp. row) in the dynamic programming matrix, and then to iterate on the two separate smaller submatrices thus obtained. In our case, we can not proceed exactly like that both because we do not know whether any exon passes through the middle row (or column) of the matrices and because all exons are not equal (they end with different reading phases, with or without gaps or frameshift errors). However, we can adapt the idea to our purposes. The interested person is referred to [29] for details.

4.2.1.3.3 Initial Applications

Initial applications were done on a set of sequences of various plants, both dicotyledons (*Arabis stelleri*, *Arabidopsis thaliana*, *Leavenworthia stylosa*, coton, petunia, *Vitis vinifera*, *Trifolium repens*; id numbers: AB015498, D84240, AF037564, U53701, X54106, U36586, Z24755, U63931) and monocotyledons (maize, rice, pearl millet, *Washingtonia robusta*, barley; id numbers: X04050, X02915, M59082, M36469, U65972, X12733), each containing in its middle a gene coding for alcool deshydrogenase (ADH). The number of exons varied between 5 and 10 among the 14 sequences. They were all successively compared to a contig of *Arabidopsis thaliana* (id number: AB006703) – the target sequence – that was assumed to contain an ADH whose structure we initially ignored (but it was known to our collaborator and co-author of the paper [29], Pierre Rouzé). The result obtained is presented for all sequences in Figure 4.3. Since the gene does not start at the same place in the sequences, the Figure allows us to compare the sets of exons predicted to form part of the gene in each sequence on the one hand, and in the target sequence on the other. It is not

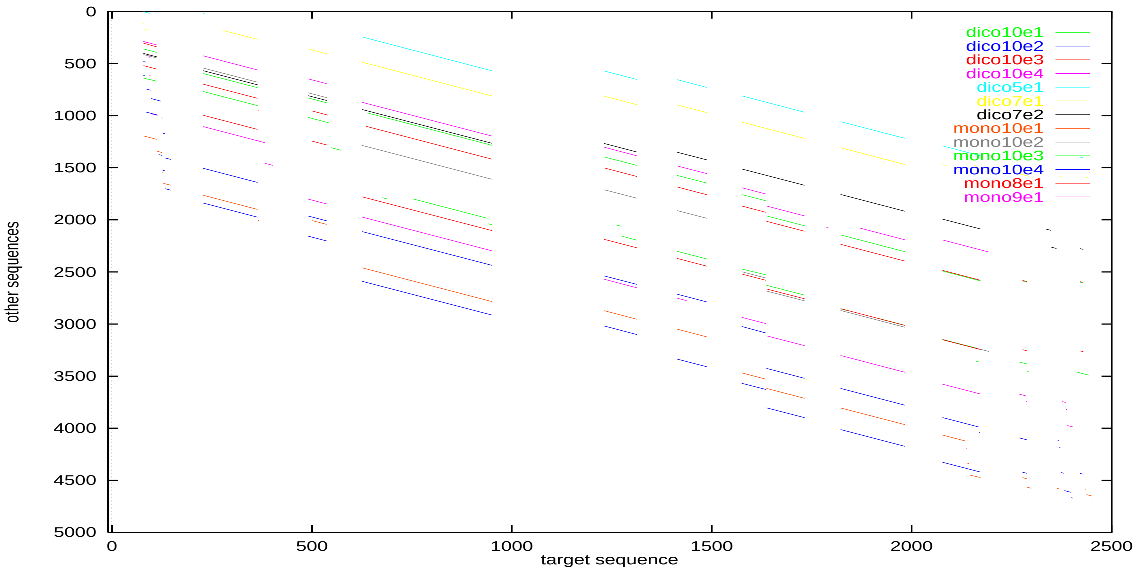

Figure 4.3: (Best seen in color) Best assembly for each of the sequences of a gene coding for alcool deshydrogenase in various plants (EMBL id numbers: AB015498, D84240, AF037564, U53701, X54106, U36586, Z24755, U63931, X04050, X02915 , M59082, M36469, U65972, X12733) against a contig of *Arabidopsis thaliana* (id number: AB006703). The matrix $\mathcal{M}$ chosen is PAM100, the frameshift penalty is 500, the cost for opening a gap is 10 and the cost for extending it also 10.

important to our purposes in this memoir to indicate the exact starts and stops of each exon in the ADH gene present in the *Arabidopsis thaliana* contig. It suffices to say that 9 exons are (correctly) predicted but that the frontier of some exons, and, specially, of the gene itself (its beginning and end) are the hardest to identify. In particular, all the small speckles at the right-hand side of the figure are false positives: the gene stop is not correctly identified.

These were relatively “clean” datasets in the sense that the sequences contained little more than the genes whose structure we had to determine. The genes themselves are well-conserved (better than the intronic parts).

Applications to “less clean” sequences (of *Arabidopsis thaliana* as well as from man and mouse [55]) have shown the necessity to extend the algorithm so that it may identify not just one best assembly, but all best assemblies or, even better, all assemblies having a score within a given distance of the optimal.

4.2.1.3.4 Initial Ideas for Finding All Assemblies Having a Score Within a Distance of the Optimal

The initial ideas we have been exploring for finding all assemblies having a score within a distance of the optimal, that is all suboptimal assemblies, are extensions of an algorithm elaborated by Chao [40] for solving a simpler problem. This consisted in finding all suboptimal global alignments (not suboptimal gene assemblies) between two sequences using an affine score gap function. Of course, the important point for him was to be able to do it in space linear with the length of the smallest sequence[4]. That is also our concern.

In a sketchy way, Chao mixes Naor and Brutlag's algorithm [143] for solving the same problem in quadratic space, with the divide and conquer approach of Hirschberg [105] (more precisely, Miller and Myers's extension to the case of an affine gap cost function [141]) for obtaining one best alignment in linear space. Simplifying a little, Naor and Brutlag determine all cells in the dynamic programming matrix that belong to at least one suboptimal alignment. In order to do that in linear space, a pass is made twice over the matrix, the first time to calculate the optimal score, and the second to identify the cells along the middle column and middle line which belong to at least one suboptimal alignment[5]. This delimits four subrectangles where the same method is recursively applied. A few further tricks lead to an algorithm whose time complexity is in $O(mn + f \log\log w)$ where f is the area of the matrix bounded by the suboptimal alignments and w the maximum width of that region.

Extending this algorithm to solve our own problem, and possibly improving its time complexity, requires careful attention to some details. These are related to the fact that we are calculating suboptimal assemblies of genes instead of suboptimal global alignments, and therefore also comparing sequences that contain coding parts. This is work currently in progress.

Once it is finished, there remains deciding what to do with the best suboptimal assemblies found. Various ideas seem worth exploring. We thus may:

- use additional finer critera to decide which of the suboptimal assemblies have a chance of being correct. One possible criterion would consist in checking whether the pattern of conservation in the first, second and third positions inside the gene predicted by each assembly is significantly different from the same pattern for the surrounding sequence [55]. If the gene has been correctly "guessed", we know the reading phase of each nucleotide (even in the presence of frameshifts). Third positions inside a gene should be far less conserved than the second and first, whilst all positions outside genes should present the same rate of conservation.

- use the suboptimal assemblies to "vote" for each corresponding nucleotides in the sequences [181].

- use an idea akin to the one introduced by Bouthinon for predicting an RNA structure from a set of sequences assumed to adopt a common fold (section 3.4.1.3). Suppose we had a family of p orphan genes that are brothers. From each pair of genes, we derive a set of suboptimal assemblies. This gives us p sets of assemblies from which, by performing a multiple comparison (on the assemblies only instead of the whole sequences), we could derive a maximum assembly having one representative in each of the p sets and which could represent the "correct" gene.

4.2.1.3.5 Exploring the "Gene as Ensemble of (Initially Independent) Conserved Parts" Model

Although we have so far put more stress on the "gene as an assembly" model, the other model, of a gene as an "ensemble of (initially independent) conserved parts", has not been ignored. In fact, we started our work on orphan gene finding with it. The difficulty is the same as faced with the "gene

[4] The reason why this is important is not just aesthetic. An algorithm needing space quadratic with the average length of the sequences can often not be used in practice.

[5] In fact, the algorithm works with a graph representation of all possible alignments of two sequences, and identifies the nodes and edges of the subgraph containing all suboptimal alignments.

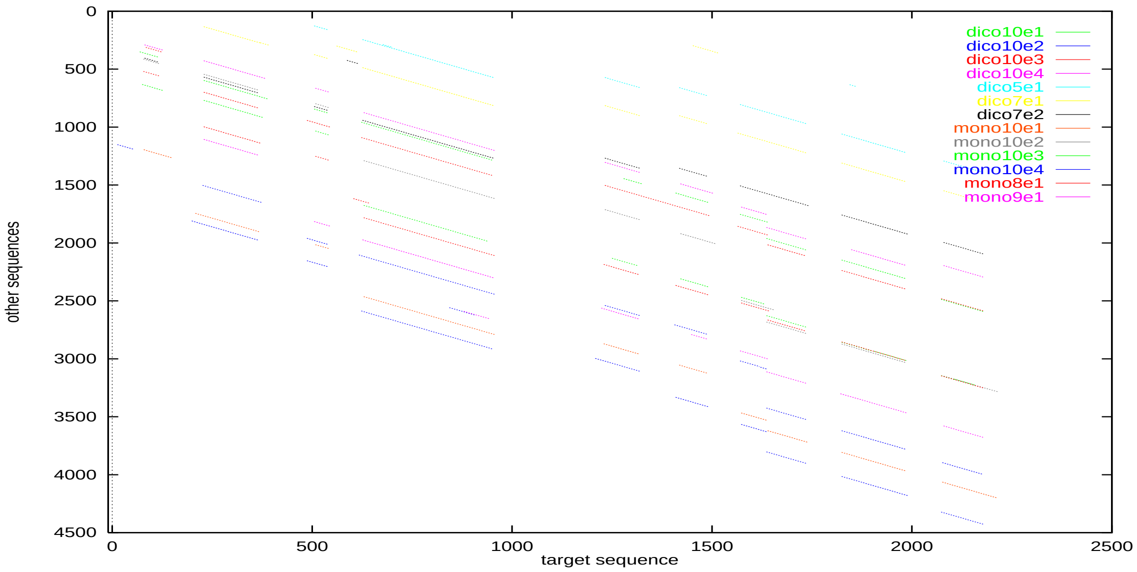

Figure 4.4: (Best seen in color) Best local alignments for each of the sequences of a gene coding for alcool deshydrogenase in various plants (EMBL id numbers: AB015498, D84240, AF037564, U53701, X54106, U36586, Z24755, U63931, X04050, X02915, M59082, M36469, U65972, X12733) against a contig of *Arabidopsis thaliana* (id number: AB006703). The matrix $\mathcal{M}$ chosen is PAM100, the frameshift penalty is 500, the cost for opening a gap is 10 and the cost for extending it also 10.

as an assembly" model: it is in being capable of recovering not just a best but all "best conserved" parts (that is, all local alignements – corresponding to potential exons – which have a score above a certain value, or all k best local alignments[6]). A further difficulty in this case is in managing to elaborate an algorithm that has a space complexity below quadratic. Huang, Hardison and Miller's algorithm [110] was initially extended (on paper and implemented) to handle the model of coding DNA we adopted. The time complexity of this first algorithm is in $O(kmn)$ where k is the number of best local alignments retained. We then extended (on paper) another algorithm due to Huang and Miller [109] for solving the same problem also in linear space. The second algorithm should, on average, perform better than the first, although the time complexity may still in the worst case be $O(kmn)$ (we believe this may be improved by considering better envelops than Huang and Miller's rectangles – work in progress). Preliminary application of the first algorithm to the same data set as in section 4.2.1.3.3 produced the results shown in Figure 4.4.

Compared with the results given in Figure 4.3, one notices at first few differences. The most immediate ones concern the limits of exons: these are fuzzier in the "gene as an ensemble of (initially independent) conserved parts" approach than in the "gene as an assembly of exons" model. This is not surprising. The first model does not take into account the fact that exons are bordered by specific dinucleotides and, provided there are no frameshifts, should be in phase with preceding and following exons in the assembly. On the other hand, the speckles at the right-hand side of Figure 4.3 disappear. This is not, however, because this approach is better at locating gene ends,

[6]Observe, however, that local alignments are identified by order of decreasing scores and the p^{th} alignment for $p > 1$ must not share a pair with any local alignment identified before p [200].

but simply because the speckles correspond to sequence fragments too small to be identified among the best conserved local regions. For this same reason, the first exon is missed in some of the sequences.

At least two problems are common to both approaches. This concerns:

- a single exon that finds itself split in two in one sequence or misses an inner (not well enough conserved) segment in both sequences;
- a reasonably well conserved intron which is erroneously identified as an exon and merged with the true exons bordering it.

Starts and ends of genes remain equally troublesome to identify in the two cases. Finally, as with many other problems in computational biology, there remains the (open) question of what score to affect to mismatches, gaps and frameshifts.

4.2.2 Gene "Building" by the Cell – Identifying Enhancer Sequences

Homology is of course not the way used by the translational and splicing machinery to identify the start and end of genes as well as the exact frontiers of exons/introns. Among the elements involved in the correct recognition and assembly of the exons forming a gene by the cellular machinery are a number of small sequences, in particular splicing sites and activator sequences which may strengthen the signal carried by a weakly conserved splicing site. Other sequences may also be important.

At the end of his 1996 paper with Burset [36], Guigó wrote that "An interesting line of research would, therefore, appear to be the exploration of the possibilities of predicting gene structure in genomic DNA sequences by relying only on the identification of sequence signals and by using some model – if only rudimentary – of how such signals are recognized and processed".

This is the, probably utopian but so beautiful idea behind a work that has been going on and off and is back on again with Claude Thermes, Yves d'Aubenton Carafa and others. The algorithm for inferring structured models, possibly with some adaptation to the particular case of identifying sequences involved in the splicing of introns, will be one of the methods driving the search. Cooperativeness between sites may be a bigger issue in this case than with bacterial regulatory sequences (but not with eukaryotic regulatory sequences where such phenomenom seems to be important). It will also be important to take possible correlations inside a site into consideration as has been done by Burge [34].

4.3 Repetitions

4.3.1 Satellites

4.3.1.1 Foreword

Tandem repeats, more accurately called tandem arrays [93] when there are more than two units, are divided into three categories depending on the length of the repeated element, the span of the repeat region and location within the chromosome [41]. Repeats occurring in or near the centromeres and telomeres are called simply *satellites*. Their span is large, up to a million bases, and the length of the repeated element varies greatly, anywhere from five to a few hundreds of base pairs. In the remaining, euchromatic region, of the chromosome tandem repeats are classified as either *micro* or *mini satellites*, according to the length of the repeated element. Micro satellites are composed of short units, usually of 2 to 5 base pairs, in copy numbers in general around 100. Mini satellites on the other hand involve slightly longer repeats, typically around 15 base pairs, in clusters of variable sizes, comprising between 30 and 2000 elements. The functional role of satellites is not currently understood, but they tend to be highly polymorphic, and thus at a minimum are very useful as genetic markers.

4.3.1.2 Model

Let s be the DNA sequence in which satellites are being sought. A *prefix model* of a satellite is a word m defined over the alphabet of nucleotides (or over a degenerate alphabet) that approximately matches a train of wagons. A *wagon* of a satellite model m is a word in s having no more than e errors (substitutions, insertions and deletions) when compared to m. A *train* of m is a sequence of wagons $u_1, u_2, \ldots, u_p$ ordered by their starting positions in s and such that:

Property 1: $p \geq min\_repeat$ where min_repeat is a parameter chosen by the user that indicates the minimum number of elements a repeating region must contain.

Property 2: $left_{u_{i+1}} - left_{u_i} \in JUMP$ where $left_u$ is the position of the left-end of wagon u in s and:

$$JUMP = \{y : y \in \cup_{x \in [1, max\_jump]}\ x \times [min\_range, max\_range]\}$$

with the three parameters min_range, max_range and max_jump chosen by the user.

In plain english, Property 2 means that the distance between the starting positions of two successive wagons in a valid train must lie within an interval $[min\_range, max\_range]$, or within an interval that is equal to $[min\_range \times x, max\_range \times x]$ for x a positive integer that is equal to at most max_jump. The purpose of parameter max_jump is to allow us to deal with very badly conserved elements inside a satellite (by actually not counting them) while we require that the satellite be relatively well conserved overall. Such badly conserved elements are called *bad wagons*.

A prefix-model m is said to be valid if there is at least one train of m in the sequence s. Similarly, a train, when viewed simply as a sequence of substrings of s, is valid if it is the train for some model m. A prefix-model represents the invariant that must be true as we progressively search for our final goal, which is to arrive at a *consensus-model*. This is a prefix model that further satisfies the following:

Property 3: $left_{u_{i+1}} - right_{u_i} \in GAP$ where $right_u$ is the position of the right-end of wagon u, and $GAP = \{y : y \in \cup_{x \in [0, max\_jump-1]}\ x \times [min\_range, max\_range]\}$. Note that $0 \in GAP$. This guarantees that, when jumps are not authorized, the repeats found are effectively tandem.

Again, in plain english, if bad wagons are not allowed, then two successive wagons in a train must be contiguous in the sequence, otherwise (bad wagons are permitted) the distance separating the end of one and the start of the next wagon in a train must lie within a multiple x of the interval $[min\_range, max\_range]$ where x is at most $max\_jump - 1$.

Observe that $[min\_range, max\_range]$ correspond also to the minimum and maximum repeat period a run of the algorithm will be able to consider. Models satisfying Properties 1 and 2 only correspond to approximately periodic repeats.

When allowing for the presence of a number of bad wagons, since mutations affecting a unit concern indels as well as substitutions, it may be interesting to work with a variant of the above properties where $JUMP$ and GAP are defined as follows:

$$JUMP = \{y : \begin{array}{l} y \in [min\_range, max\_range] \text{ or} \\ y \in \cup_{x \in [2, max\_jump]} \, x \times [min\_range - g, max\_range + g] \end{array} \}$$

$$GAP = \{y : \begin{array}{l} y \in [min\_range, max\_range] \text{ or} \\ y \in \cup_{x \in [1, max\_jump-1]} \, x \times [min\_range - g, max\_range + g] \end{array} \}$$

where $g \geq e$ is a value chosen by the user. The idea is to allow the length of the badly conserved elements to vary more than is permitted for the detected wagons.

The hurried reader may skip the next section and content himself with being aware that, given all parameters (maximum number of errors allowed e, max_repeat, min_range, max_range and max_jump), the algorithm identifies **all** satellite models which satisfy such parameters.

4.3.1.3 Algorithm

As for simple or structured models, satellite models are constructed by increasing lengths, that is, a valid prefix-model of length $k+1$ is obtained from its prefix of length k. In order to determine if a model is valid, we must have some representation of the train or wagons that make it so. There are two possibilities:

- we can keep track of each validating train and its associated wagons, or
- we can keep track of individual wagons, and, on the fly, determine if they can be combined into validating trains.

The first possibility is appealing because model extension is straightforward. We would just have to check, for each wagon of each train, whether it can be extended in relation to the extended model, and then count how many wagons remain to see whether the train it belonged to remains a valid one. However, there are generally many overlapping trains involving many of the same wagons for a given model. Common wagons may therefore be present more than once in the list of occurrences of m if this is kept as a list of trains. This approach entails therefore redundancies that lead to an inefficient algorithm. We thus take the second approach, of keeping track of wagons and determining if they can be assembled into trains as needed. This is somewhat reminiscent of the method used for detecting RNA structural motifs such as palindromes (section 3.2.2.3).

For each prefix-model m, we keep a list of the wagons of m that are in at least one train validating m. We describe such wagons as being valid with respect to m. When we extend a model to $m' = m\alpha$, we perform two tasks:

- first, determine which valid wagons of m can be extended to become wagons of m' (the basic rules of prefix-model extension are the same as for previously described algorithms, see section 2.4).
- second, of these newly determined wagons of m', keep only those that are valid with respect to m'. This requires effectively assemblying wagons into trains, something that is not needed in an approach that would keep track of trains directly.

Note that we need not actually enumerate the trains in the second step, we simply must determine if a wagon is part of one. This will allow us to perform an extension step in time linear with the length of the sequence.

As an insight on how the second step is performed, consider the directed graph $\mathcal{G} = (V, E)$ where V is the set of all valid wagons and there is an edge from wagon u to v if $left_v - left_u \in JUMP$. Then a wagon u is valid if it is part of a path of length min_repeat or more in $\mathcal{G}$. Determining this is quite simple as the graph is clearly acyclic. Furthermore, the edges of $\mathcal{G}$ can be calculated on the fly, thus requiring no effective building of $\mathcal{G}$.

The preceding operations allow us to identify all valid prefix models. Our final goal is however to arrive at consensus models, that is, at prefix models that further verify Property 3.

Consider for this the directed bipartite graph $\mathcal{G}_m = (L_m \cup R_m, E)$ whose vertices are the positions at which the left- and right-ends of valid wagons occur. Let there be a *wagon edge* $i \rightarrow j$ if and only if $j \in Right(i)$ where $Right(i)$ is the set of all possible right-end positions for a wagon starting at position i. Further, let there be a *gap edge* $j \rightarrow k$ if and only if $k - j \in GAP$. Thus the edge sequence $i \rightarrow j \rightarrow k$ occurs in $\mathcal{G}_m$ if and only if there are valid wagons u and v such that $u = s_{i+1}s_{i+2}\cdots s_j$, $left_v = k$, and $left_v - right_u$ satisfies Property 3. It follows that a position/vertex which is on a path of length $2 \times min\_repeat$ or more is part of a valid train satisfying all three properties, and is called a *final position*. Let $\mathcal{G}'_m$ be the graph induced by the set, F_m, of all final vertices. If G'_m is non-empty then m is a consensus model.

Each weakly-connected component of $\mathcal{G}'_m$ corresponds to a collection of overlapping trains. One should note that such a set of overlapping wagons/trains for a given model does indeed frequently occur. As an example if $s =$ AGAAAAAAAGAATTTAGAAGAACAA, and the search parameters are $e = 1$, $min\_repeat = 3$, $min\_range = 2$, $max\_range = 4$, and $max\_jump = 1$, then $\mathcal{G}'$ for the model $m =$ AGA consists of two components whose trains span the intervals $[1, 12]$ and $[16, 25]$. For instance, the left-end position sequences (1,4,7), (1,4,7,10), (4,7,10), (1,4,6,8,10), (1,3,6), (1,3,6,9), (3,6,9), (16,19,22), (16,18,20,22) among others, are all valid train-occurrences.

Let $[I, J]$ be the interval spanned by a weakly-connected component, $C \subseteq F_m$, where $I = min_{i \in C}\ i$ and $J = max_{j \in C}\ j$. This interval, called a *satellite span* of m, clearly contains every substring corresponding to a valid train within the component.

The graph $\mathcal{G}_m$ has at most $O(n)$ vertices and $O(n(e + |GAP|))$ edges. Thus we could explicitly build the graph $\mathcal{G}_m$ and then compute, with conventional linear graph algorithms, (1) the set of final positions F_m, (2) the subgraph $\mathcal{G}'_m$ it induces, and (3) the satellite spans for m, in $O(n(e + |GAP|))$ time and space. As before, we can improve space to $O(n)$ by determining the edges of $\mathcal{G}_m$ on the fly (further details in [168]).

4.3.1.4 Filtering

There may be an initial step to the algorithm we have not mentioned yet. When an exhaustive search through progressively longer prefix models fails to end at a final consensus model, then we can be guaranteed that a repeat meeting the constraints of the problem parameters does not occur. While we show that such searches can be performed surprisingly efficiently, it is still desirable in most cases (but not compulsory) to first attempt to filter out all portions of s from further consideration with a certain empirical guarantee that little may be missed by doing so.

The basic idea behind the filtering step is that, if there is a satellite m^r, then in the dynamic programming matrix of s against itself, there should be a local alignment of length $p \cdot (r - i)$ centered on diagonal $i \cdot p$ for $i \in [1, r-1]$ and $p = |m|$. The similarity in these local alignments and the extent to which they lie on the given diagonal or at least near it, depends on the fidelity of the repeat units in the satellite. Generally, these off-diagonal alignments, corresponding to offset copies of the satellite, are strong enough to detect, provided the difference between the offset copies is at most 30-40%, which translates to a 15-20% difference between the repeat prototype and the instances in the satellite.

The specific heuristic we arrived at is to incrementally compute the local dynamic programming matrix of s versus itself in a scan of s, where after scanning s_i we have the matrix in the triangle $(i - band, i - band)$ to $(i - band, i)$ to (i, i). The parameter $band$ is chosen so that it is at least $\max\{3.5 \cdot max\_range, 50\}$, thus guaranteeing that at least three of the off-diagonal alignments of the satellite will be in the triangular region. Clearly, the total time taken is $O(n \cdot max\_range)$ as the algorithm computes an $O(max\_range)$ band about the diagonal.

For each potential period p of interest, we compute the sum, $S(p)$, of the values in all the cells in diagonals $\{p \cdot i + \delta : i \in [1, \lfloor band/p \rfloor] \text{ and } \delta \in [-w(p), w(p)]\}$ where $w(p) = \max\{2, \epsilon \cdot p\}$. Effectively, we are taking the sum of the values in a $2 \cdot w(p)$ strip about each off-diagonal at period p, within the triangle. We keep this sum for each $p \in [min\_range, max\_range]$ at an additional factor of $O(w(max\_range) - w(min\_range))$ over the cost of dynamic programming in the band. What one expects is that in regions occupied by a satellite of period p, $S(p)$ is significantly larger than it is when the sequence is random, to the point where one can distinguish between the two.

A thousand trials were run on either a random sequence, or one containing a length 4, 8, or 16 repeat at various fidelities ϵ. For the local alignment objective, we used a scoring scheme that scored matches 1, and differences -2. In all cases, it is only till one gets to $\epsilon = 20\%$ that the distribution of random scores and those of the satellite filter scores begin to overlap, and only then by .2% of the 1000 trials with the indicated separating values. On several experiments with 10,000 trials, we were able to get only 1 overlap with ϵ at 10%, supporting our claim that the filter misses only 1 in 10^4 satellites at that fidelity. It allows to eliminate, in general, for a fidelity of 15-20%, 80% of the sequence from further consideration.

4.3.1.5 Selecting Models to Report: Model Fitness

It often happens that the same, or approximately the same span corresponds to more than one consensus model, *e.g.*, models ACA, ATA, and AAA are also valid for the two satellite spans in the example of section 4.3.1.3. It is also often the case that the models are permuted versions of one another, *e.g.*, models AGA, GAA and AAG. Furthermore, the models covering a given span can have different lengths, *e.g.*, AGAA also covers the span $[1, 12]$ above. Finally, it may happen that one model is a *power* of the other: if, for instance, the minimum number of repeats in the example is lowered to 2, then both AGA and AGAAGA = AGA^2 are valid models, and we say that AGA is a *root* of AGAAGA. Frequently, models having overlapping satellite spans will differ in terms of a few wagons at the beginning or end of trains, and possibly of a few wagons in the middle when jumps are allowed.

In light of these observations, we have chosen to combine all overlapping spans into one large span and to associate with it all the models contributing satellite spans to it. This merging of the intervals takes place as the spans for individual models are found so that at the completion of the algorithm, we have a list of disjoint intervals of s called *satellite regions* and for each region R, the set, $\mathcal{M}_R$, of models having a satellite span within R. Further details may be found in [168].

Given that $\mathcal{M}_R$ frequently contains more than a single model, we rank each $m \in \mathcal{M}_R$ according to the *fitness* of its match to R, which we define as the best scoring alignment between $(m)^\star$ (that is,

m repeated any number of times) and a substring of R under a user-specified scoring scheme δ. The efficient comparison of the regular expression $(m)^\star$ (matching zero or more repetitions of m) was introduced by Myers and Miller [142] for any regular expression, and a later reincarnation of their work for the case of this special pattern type became popularly known as *wrap around dynamic programming* [126]. A ranking of models according to such a fitness score for the region relegates secondary model matches to the back of the list. Moreover, we further avoid reporting models that are permutations or powers or roots of one another, by always picking the fitter of any two such models and removing the other. The complexity of this post-processing phase is dominated by the search for consensus models.

4.3.1.6 Initial Applications

We develop in fuller detail an example of application and use the results obtained to discuss some points of the algorithm. Other brief applications are given in [169].

The example is drawn from chromosome IX of yeast *Saccharomyces cerivisiae.* Figure 4.5 shows a tandem repeat starting at position 391 131 (in the sequence as recovered from the Web address ftp://ftp.mips.embnet.org/pub/yeast/) of the chromosome. This repeat is composed of 41 full units, 16 of which present a deletion of 9 bp. as against the other elements. Apart from this, the repeat is well conserved overall (on average one mutated base per element), except for the first six units and for the last one. The repeat is located inside a coding region (in the other strand) and Figure 4.6 shows the DNA repeat (inversed and complemented) together with its corresponding translation. The translated repeat is part of a glucoamylase s1/s2 precursor protein (SwissProt id: AMYH_YEAST).

Table 4.2 gives a summary of the consensus for the DNA repeat (as determined by the majority rule) and of the fittest satellite models found by us concerning this repeat under a scoring scheme that is as follows: matches, $+1$, mismatches, -1, and indels, -2 (it is thus more permissive of mismatches than indels). The other parameters of the algorithm were set as indicated in Table 4.1.

The results in Table 4.2 follow the order in which we obtained them as we progressively refined the parameters to get to the final, possibly correct, model for the repeat. Observe that, although the intervals of lengths we worked with in our first runs of the algorithm were too short to capture the consensus model for the satellite, we were nevertheless able, thanks to the fact that bad wagons were allowed to be jumped, to notice something of interest was present in the region. We managed even to get the satellite prefix model and span approximately right.

That we were getting various slightly different models with close enough scores may, on the other hand, have hinted to the fact that we were not accurately constraining our search, in particular concerning the period of the repeat. However, it will often be the case that there will be more than one model for a satellite region not only because we are setting the parameters wrong for the identification of a given tandem repeat, but also because it is not clear what template or prototype sequence the repeated units came from. This could suggest working with a Levenshtein distance, even a weighted one, as a similarity measure between repeats may not be the best way of capturing the whole flavor of such repeats.

Locating the repeat after translation is, in this case, a slightly easier affair as some of the substitutions at the DNA level occur at silent positions. Table 4.3 gives a summary of the consensus for the protein repeat (again, as determined by the majority rule) and of the fittest satellite model found by us concerning this repeat under the same scoring scheme as above (other schemes, in particular using substitution weight matrices, may also be employed). The parameters used for the

```
GTTGCTAGAGGAAGATGGGGTTGGTACTGGTGCTACAGAGCTTTC
AGTGGTGGAGCTGGAT         ACTGGAGCAGAAGAGCTTTC
AGTAGTAGAGCTTGATGGGGTTGGTACTGGAACAGAAGAGCTTTC
AGTGCTAGAGCTGAATGGGGTTGAAGATGGAGCGGAGGAAGTGAT
GTTGCTAGAGGAAGATGGGGTTGGTACTGGTGCTACAGAGCTTTC
AGTAGTAGAGCTTGATGGGGTTGGTACTGGAGCAGAAGAGCTTTC
GGTAGTAGAGCTGGATGGAGTTGGCACTGGAGCAGAAGAGCTTTC
AGTAGTAGAGCTGGATGGAGTTGGTACTGGAGCAGAAGAGCTTTC
AGTGGTAGAGCTGG      TT   ACTGGAGCAGAAGAGCTTTC
AGTAGTAGAGCTGGATGGAGTTGGTACTGGAGCAGAAGAGCTTTC
AGTAGTAGAGCTGGATGGAGTTGGTACTGGAGCAGAAGAGCTTTC
AGTAGTAGAGCTGGATGGAGTTGGTACTGGAGCAGAAGAGCTTTC
AGTGGTAGAGCTGG      TT   ACTGGAGCAGAAGAGCTTTC
AGTGGTAGAGCTGG      TT   ACTGGAGCAGAAGAGCTTTC
AGTAGTAGAGCTGGATGGAGTTGGTACTGGAGCAGAAGAGCTTTC
AGTAGTAGAGCTGGATGGGGTTGGTACTGGAGCAGAAGAGCTTTC
AGTAGTAGAGCTGGATGGAGTTGGTACTGGAGCAGAAGAGCTTTC
AGTGGTAGAGCTGG      TT   ACTGGAGCAGAAGAGCTTTC
AGTAGTAGAGCTTGATGGGGTTGGAGCTGGAGCAGAAGAGCTTTC
AGTAGTAGAGCTTGATGGAGTTGGTACTGGAGCAGAAGAGCTTTC
AGTAGTAGAGCTTGATGGAGTTGGCACTGGAGCAGAAGAGCTTTC
AGTGGTGGAGCTGG      TT   ACTGGAGTAGAAGAGCTTTC
AGTAGTAGAGCTGGATGGAGTTGGTACTGGAGCAGAAGAGCTTTC
AGTGGTAGAGCTGG      TT   ACTGGAGCAGAAGAGCTTTC
AGTGGTGGAGCTTGATGGGGTTGGAGCTGGAGCAGAAGAGCTTTC
AGTAGTAGAGCTGGATGGAGTTGGTACTGGAGCAGAAGAGCTTTC
AGTGGTAGAGCTGG      TT   ACTGGAGCAGAAGAGCTTTC
AGTAGTAGAGCTGGATGGAGTTGGTACTGGAGCAGAAGAGCTTTC
AGTGGTAGAGCTGG      TT   ACTGGAGCAGAAGAGCTTTC
AGTAGTAGAGCTGGATGGAGTTGGTACTGGAGCAGAAGAGCTTTC
AGTGGTAGAGCTGG      TT   ACTGGAGCAGAAGAGCTTTC
AGTGGTAGAGCTGG      TT   ACTGGAGCAGAAGAGCTTTC
AGTGGTAGAGCTGG      TT   ACTGGAGCAGAAGAGCTTTC
AGTGGTAGAGCTGG      TT   ACTGGAGCAGAAGAGCTTTC
AGTAGTAGAGCTGGATGGAGTTGGTACTGGAGCAGAAGAGCTTTC
AGTGGTAGAGCTGG      TT   ACTGGAGCAGAAGAGCTTTC
AGTGGTAGAGCTGG      TT   ACTGGAGCAGAAGAGCTTTC
AGTAGTAGAGCTTGATGGGGTTGGTACTGGAGCAGAAGAGCTTTC
AGTGGTAGAGCTGG      TT   ACTGGAGCAGAAGAGCTTTC
AGTAGTAGAGCTGGATGGAGTTGGTACTGGAGCAGAAGAACTTTC
AGTAGTAGAGCTTGATGGGGTTGGTACTGGAGTAGTAGTCTTCTT
```

Figure 4.5: An example of a tandem repeat in chromosome IX of yeast *Saccharomyces cerevisiae*, starting at position 391 131 (in the sequence as recovered from the Web address given in the text).

Parameter Set	e	g	min_repeat	min_range	max_range	max_jump
1	1	0	10	20	30	3
2	1	4	20	20	30	3
3	2(subs. only)	4	15	30	40	4
4	2(subs. only)	5	15	40	50	4

Table 4.1: Parameter sets for the results shown in Table 4.2 concerning the 45 bp satellite in Chromosome IX of Yeast.

algorithm in this case were set as follows: $e = 1$ (subs. only), $g = 1$, $min\_repeat = 15$, $min\_range = 13$, $max\_range = 18$, $max\_jump = 4$.

```
AAGAAGACTACTACTCCAGTACCAACCCCATCAAGCTCTACTACT             KKTTTPVPTPSSSTT
GAAAGTTCTTCTGCTCCAGTACCAACTCCATCCAGCTCTACTACT             ESSSAPVPTPSSSTT
GAAAGCTCTTCTGCTCCAGT   AA      CCAGCTCTACCACT             ESSSAPV T SS TT
GAAAGCTCTTCTGCTCCAGTACCAACCCCATCAAGCTCTACTACT             ESSSAPVPTPSSSTT
GAAAGCTCTTCTGCTCCAGT   AA      CCAGCTCTACCACT             ESSSAPV T SS TT
GAAAGCTCTTCTGCTCCAGT   AA      CCAGCTCTACCACT             ESSSAPV T SS TT
GAAAGCTCTTCTGCTCCAGTACCAACTCCATCCAGCTCTACTACT             ESSSAPVPTPSSSTT
GAAAGCTCTTCTGCTCCAGT   AA      CCAGCTCTACCACT             ESSSAPV T SS TT
GAAAGCTCTTCTGCTCCAGT   AA      CCAGCTCTACCACT             ESSSAPV T SS TT
GAAAGCTCTTCTGCTCCAGT   AA      CCAGCTCTACCACT             ESSSAPV T SS TT
GAAAGCTCTTCTGCTCCAGT   AA      CCAGCTCTACCACT             ESSSAPV T SS TT
GAAAGCTCTTCTGCTCCAGTACCAACTCCATCCAGCTCTACTACT             ESSSAPVPTPSSSTT
GAAAGCTCTTCTGCTCCAGT   AA      CCAGCTCTACCACT             ESSSAPV T SS TT
GAAAGCTCTTCTGCTCCAGTACCAACTCCATCCAGCTCTACTACT             ESSSAPVPTPSSSTT
GAAAGCTCTTCTGCTCCAGT   AA      CCAGCTCTACCACT             ESSSAPV T SS TT
GAAAGCTCTTCTGCTCCAGTACCAACTCCATCCAGCTCTACTACT             ESSSAPVPTPSSSTT
GAAAGCTCTTCTGCTCCAGCTCCAACCCCATCAAGCTCCACCACT             ESSSAPAPTPSSSTT
GAAAGCTCTTCTGCTCCAGT   AA      CCAGCTCTACCACT             ESSSAPV T SS TT
GAAAGCTCTTCTGCTCCAGTACCAACTCCATCCAGCTCTACTACT             ESSSAPVPTPSSSTT
GAAAGCTCTTCTACTCCAGT   AA      CCAGCTCCACCACT             ESSSTPV T SS TT
GAAAGCTCTTCTGCTCCAGTGCCAACTCCATCAAGCTCTACTACT             ESSSAPVPTPSSSTT
GAAAGCTCTTCTGCTCCAGTACCAACTCCATCAAGCTCTACTACT             ESSSAPVPTPSSSTT
GAAAGCTCTTCTGCTCCAGCTCCAACCCCATCAAGCTCTACTACT             ESSSAPAPTPSSSTT
GAAAGCTCTTCTGCTCCAGT   AA      CCAGCTCTACCACT             ESSSAPV T SS TT
GAAAGCTCTTCTGCTCCAGTACCAACTCCATCCAGCTCTACTACT             ESSSAPVPTPSSSTT
GAAAGCTCTTCTGCTCCAGTACCAACCCCATCCAGCTCTACTACT             ESSSAPVPTPSSSTT
GAAAGCTCTTCTGCTCCAGTACCAACTCCATCCAGCTCTACTACT             ESSSAPVPTPSSSTT
GAAAGCTCTTCTGCTCCAGT   AA      CCAGCTCTACCACT             ESSSAPV T SS TT
GAAAGCTCTTCTGCTCCAGT   AA      CCAGCTCTACCACT             ESSSAPV T SS TT
GAAAGCTCTTCTGCTCCAGTACCAACTCCATCCAGCTCTACTACT             ESSSAPVPTPSSSTT
GAAAGCTCTTCTGCTCCAGTACCAACTCCATCCAGCTCTACTACT             ESSSAPVPTPSSSTT
GAAAGCTCTTCTGCTCCAGTACCAACTCCATCCAGCTCTACTACT             ESSSAPVPTPSSSTT
GAAAGCTCTTCTGCTCCAGT   AA      CCAGCTCTACCACT             ESSSAPV T SS TT
GAAAGCTCTTCTGCTCCAGTACCAACTCCATCCAGCTCTACTACT             ESSSAPVPTPSSSTT
GAAAGCTCTTCTGCTCCAGTGCCAACTCCATCCAGCTCTACTACC             ESSSAPVPTPSSSTT
GAAAGCTCTTCTGCTCCAGTACCAACCCCATCAAGCTCTACTACT             ESSSAPVPTPSSSTT
GAAAGCTCTGTAGCACCAGTACCAACCCCATCTTCCTCTAGCAAC             ESSVAPVPTPSSSSN
ATCACTTCCTCCGCTCCATCTTCAACCCCATTCAGCTCTAGCACT             ITSSAPSSTPFSSST
GAAAGCTCTTCTGTTCCAGTACCAACCCCATCAAGCTCTACTACT             ESSSVPVPTPSSSTT
GAAAGCTCTTCTGCTCCAGT         ATCCAGCTCCACCACT             ESSSAPV   SSSTT
GAAAGCTCTGTAGCACCAGTACCAACCCCATCTTCCTCTAGCAAC             ESSVAPVPTPSSSSN
```

Figure 4.6: The same tandem repeat as in Figure 4.5, inversed and complemented (left) together with its translation (right). The translated sequence is part of a glucoamylase s1/s2 precursor protein.

	start	end	score	consensus or model	time
consensus	391 131	392 832		AGTAGTAGAGCTGGATGGAGTTGGTACTGGAGCAGAAGAGCTTTC	
model for param. 1	391 337	392 718	606	AGTAGTAGAGCAGAAGAGCTTTC	66s
	391 349	392 718	582	AGTAGTAGAGCTAGAAGAGCTTTC	
	391 349	392 717	573	AGTAGTGAGCAGAAGAGCTTTC	
model for param. 2	391 337	392 717	606	AGTAGTAGAGCAGAAGAGCTTTC	90s
	391 349	392 718	594	TGGTACTGGACAGAAGAGCT	
	391 349	392 717	586	GTACTGGAGCAGAAGAGCTTTC	
model for param. 3	391 379	392 675	871	AGTAGTAGAGCTGGTGGGTACTGGAGCAGAAGAGCTTTC	310s
	391 376	392 720	864	AGTAGTAGAGCTTGGTTACTGGAGCAGAAGAGCTTTC	
	391 378	392 675	858	AGTGGTAGAGCTGGTGGTACTGGAGCAGAAGAGCTTTC	
	391 338	392 718	855	AGTAGTAGAGCTGGTTACTGGAGCAGAAGAGCTTTC	
model for param. 4	391 338	392 741	1060	AGTAGTAGAGCTGGATGGAGTTGGTACTGGAGCAGAAGAGCTTTC	970s
	391 350	392 741	1045	AGTAGTAGAGCTGATGGAGTTGGTACTGGAGCAGAAGAGCTTTC	

Table 4.2: Consensus obtained by majority rule for the 45 bp satellite in Chromosome IX of Yeast and satellite consensus models obtained by our algorithm for the various parameters indicated in Table 4.1.

	start	end	score	consensus or model	time
consensus	308	873		ESSSAPVPTPSSST	
model	313	872	436	ESSSAPVPTPSSST	107 s.

Table 4.3: Consensus obtained by majority rule for the 15 bp satellite in AMYH_YEAST and satellite consensus model obtained by our algorithm for the parameters as indicated in text.

4.3.1.7 Dispersed Repeats

The reader interested by our work on repeats in general, that is, possibly dispersed, are referred to, in particular but not exclusively, [167] [166]. This concerns so far algorithms and not yet applications.

4.4 Critique and Perspectives

4.4.1 Critique and Perspectives on Gene Finding

It probably did not escape the reader who looked at Figures 4.3 and 4.4 that it would be interesting to introduce a multiple aspect into the orphan gene finding problem. A first way of doing that was suggested in section 4.2.1.3.4. This concerned applying the multiple criterion to a set of previously predicted assemblies. It is also possible to consider applying such criterion at the initial sequence comparison level whatever the model used, "gene as an assembly of exons" or "gene as ensemble of (initially independent) conserved parts". Indeed, it seems absurd to be missing all the extra information contained in those two figures. Of course, one easy way of doing it is suggested by them: use a pivot sequence, do pairwise comparison and summarize the results in relation to the pivot. This, however, begs the question of which sequence to choose as pivot. This problem is not new, it has been faced by every researcher who has tried to solve the (simpler ?) multiple (global or local) alignment problem. Our way of going into this subject will then follow the normal route: it will consist in exploring the various solutions that were adopted for the multiple alignment of sequences and examine how they might adapt or be extended to solve our problem. This should not discard completely new ideas.

The multiple sequence alignment problem has one more link with gene finding: this is when what needs to be aligned represent DNA sequences which are coding or, worse, contain both coding and non coding parts. This is not unusual [139]. It is also something that is not considered by most multiple alignment programs. At the current time, the biologist gets around the problem by using his favorite algorithm (in general, this will be Clustal) and then "revising" the alignment, locally, by hand. The revision most often concerns correcting indels in regions they know to be coding when these change the reading frame.

4.4.2 Critique and Perspectives on Repetitions

As in the case of the algorithm for inferring structured models (section 2.4), our satellite identifier requires the user to fix the values of a number of parameters. This problem is not encountered when one uses instead algorithms based on a statistical approach. There is one very nice satellite detector elaborated by Benson [20] [21] [22] whose input could be just the sequence one wishes to analise. The algorithm is fast (apparently faster than our own) and seems quite sensitive. It is, however, a heuristic. It is based on the principle of finding first the exact parts of an approximately periodic repeat. We are exploring this idea for an algorithm that would be exhaustive. Such an approach is also probably required in order to identify much longer repeats that the ones we can currently deal with (typically up to 50 bases long for a 10 to 15% fidelity against a model).

Another fascinating question concerns more finely analysing the mutational structure of a repeat. Answers to questions such as: is there a regularity in the distribution of mutated positions of a tandem array, or in the nature of the mutations observed, may prove interesting biologically. These are challenging problems, as it is often difficult even to decide which model among those found best represents the possible originator of a repeat. As mentioned, working with a Levenshtein distance may also not be the best way to get at this originator and more phylogenetically appropriate similarity metrics may be required. These issues are biologically important but we do not know yet how to treat them algorithmically (or even, sometimes, formalize them in terms of biology). Neither do we know yet how to reconstruct the evolution of a repeat. The statistical significance of a repeat is a further problem for which we do not have any good solution (see also section 3.2.2.4). Finally, we observe for satellites and repeats in general the same phenomenom we mentioned already for, among others, regulatory sequences and palindromic units: what can explain the variety found among different genomes even inside a same cellular domain ? And what can such variety tells us about the function of repeats ?

Chapter 5

Genome Dynamics and Evolution

Students whose work is presented in this chapter

Marina Zelwer for the work on a statistical approach to identifying the presence and exact location of recombinations

Estela Maris Rodrigues for the work on a combinatorial approach to identifying the presence and exact location of recombinations and on tree algorithms

Nadia Pisanti for the work on rearrangements

Co-workers and/or collaborators

Pierre Darlu for the work on identifying the presence and exact location of recombinations (both approaches)

Yoshiko Wakabayashi for the work on identifying the presence and exact location of recombinations (combinatorial approach)

Anne Vanet for initial thinking on the "ancestral operon" problem and phylogeny of non coding regions

People who gave advice or with whom this work was discussed

Henri Buc (recombinations)

Eric Coissac and Guillaume Achaz (duplications)

Gordon Langsley (rearrangements)

Bernard Dujon and his team (rearrangements)

People from whom we used algorithms

Jotun Hein (RecPars, to test ideas)

Joe Felsenstein (DnaPars)

John Farris (ILD test)

5.1 Introduction

This chapter tentatively deals with the very broad biological question of genome evolution and dynamics from initially a few points of view only. The reason for this is simple. Evolutionary mechanisms are perhaps the hardest ones to understand, specially for a non biologist. They are also among the hardest (and most fascinating) to analyse. Probably more than any of the other areas of research inside computational biology, they require a deep knowledge of a variety of subjects such as classificatory techniques and stochastic processes besides combinatorics and algorithmics. It is sometimes difficult to determine even what questions and methods are pertinent and how to interpret results once these are obtained by whatever approach. We have therefore tried to address the question in ways that would enable us to get familiar with the issues, although such ways represent a simplified, and thus possibly at first not very realistic view of evolution.

We have focused our attention initially on large-scale evolution, that is, on evolutionary processes which involve rearrangements of whole pieces of a genome. There are many questions one may ask concerning such processes, among which are:

1. how to identify with a fair degree of confidence and accuracy the rearrangements that took place ?
2. how to use such rearrangement events to:
   (a) calculate a distance between organisms ?
   (b) infer a phylogeny (and possibly an ancestral arrangement) for the organisms ?
3. what purpose, and scale, to attribute to rearrangement events ?

The problems we have tried to treat so far concern, first, the identification of recombination events along a set of aligned and phylogenetically well-characterized sequences and, second, the computation of a distance between genomes based on rearrangement events. The first problem was treated in two different ways, one that is quite pragmatic and the other theoretical. The (preliminary) theoretical work has shown potential applications to the manipulation of RNA structures as well as phylogenies. The second problem is, at the current time, treated in a completely theoretical manner.

The first problem is discussed in section 5.2 while the second is introduced in section 5.3.

There are many more questions we would like to address in future, a few of these are presented in the Perspectives section of this chapter.

The reader will notice that this is the chapter where he will probably find the greatest amount of confusion and, perhaps, the greatest number of "theoretical divagations" that may have little bearing on reality. May he be patient and willing to discuss.

5.2 Detecting the Presence and Exact Location of Recombinations

5.2.1 Foreword

Recombinations are a driving force for the creation of diversity among individuals within a species. They may happen:

- in eukaryotes, between alleles of a same gene during meiosis;

- in prokaryotes, through the process of conjugation (direct transfer of genetic information from one cell to another while the two are in physical contact), cell fusion, transformation (uptaking of naked chromosomal or plasmid DNA from the surrounding medium) or transduction (transfer, mediated by a virus particle, of chromosomal or plasmid DNA into a cell);
- in viruses, when many of a same species find themselves together in an unique cell.

Recombinations influence also evolution by enabling the exchange of material between genes of a same multigenic family (such genes are the result of duplication from a common ancestor) as was first evidenced in mammals in 1980 [186].

Exchange of genetic material between different species on the other hand (also called *horizontal transfer*) has been shown to exist in viruses [80], prokaryotes [24] and eukaryotes [6].

The sequencing of whole genomes is starting to reveal that such evolutionary mechanisms, whether intra or inter-species, may occur far more frequently than previously thought (see [144] for a discussion concerning *Thermogota maritima*, perhaps the most important example among currently sequenced genomes of the frequency of such phenomena).

Recombination mechanisms contradict the classical hypothesis of a phylogenetic tree as a good model for representing the history of the species. Indeed, such mechanisms modify a given tree either by changing the length of the branches while preserving the topology, or by changing the topology. However, since the presence of recombinations make phylogenetic reconstructions far more complex, they are seldom considered in currently existing methods. In fact, recombination poses a problem at three levels of increasing difficulty. At the first level, one is concerned only with determining whether an event such as recombination or horizontal transfer has occurred among the sequences of a set of strains or species. The second level addresses the question of locating all recombination points along the sequences. Finally, the third and last level treats the problem of reconstructing the phylogeny of the sequences taking such recombination events into account (and possibly obtaining something that it is no longer possible to represent under the form of a tree).

We have started dealing with the first and second levels mentioned above and have focused our attention on the case of related organisms only. We are not for now concerned with the detection of horizontal transfer between evolutionary distant organisms for which hidden Markov model (or even simpler) approaches seem more appropriate. Our objective is to elaborate a method for detecting and precisely locating recombination events between close sequences. Such recombinations may be observed between genes of different strains of a same bacterium [129], between viruses of a same species but of different types [106] or between genes of a same multigenic family [114].

Various methods have been elaborated in the last ten years to address this problem. Three surveys were recently written on the subject, one due to Hein *et al.* [202], another to Drouin *et al.* [87] and the third to Maynard-Smith [138]. The methods may be classified into two main categories: those that can be applied directly to the sequences under consideration and those that can be applied only to aligned and phylogenetically well characterized sequences (that is, to sequences whose phylogeny has been reconstructed). A critical review of the most interesting algorithms in both categories (as examined by the biologist with whom we have been collaborating on this subject, Pierre Darlu) may be found in [204]. The principal limitations of approaches in both classes are that they often miss many true recombinations (high level of false negatives), handle in general poorly the presence of multiple recombinations and do not always provide statistical criteria for evaluating whether a detected recombination is significant.

We have concentrated for now on the problem of detecting the presence and exact location of recombination events in the case of aligned and phylogenetically reconstructed DNA sequences.

In a very abstract way, the problem we wanted to treat may be expressed in the following way. Given a set of phylogenetically "correctly" aligned sequences, find a partition by column of the alignement such that the sum of the weights (distances or number of evolutionary events) of the branches of the trees obtained for each partition plus a penalty proportional to the number of recombination events detected is minimum.

We have tried to come as close to this problem as possible in two different ways, one using a statistical approach and the other a combinatorial. They are briefly described in sections 5.2.2 and 5.2.3. The reader may find further details on the statistical approach in [204]. The combinatorial approach is in progress. It has led us to some interesting subproblems often a little distant from the original one but with a possibly much larger domain of application than the one addressed in this chapter.

5.2.2 Statistical Approach

5.2.2.1 Algorithm

Let us first recall that our input data comes under the form of an alignment of DNA sequences we assume to be "correct". Not all columns of the alignment will be used by the algorithm. The only interesting ones are columns representing *informative sites*. These are columns where at least two letters of the nucleotide alphabet occur at least twice each.

The main idea of the algorithm as it stands at the present time is the following.

We start by locating the positions of potential recombination events. These will be defined as the local maxima[1] of an index (called "Incongruence Length Difference" and due to Farris *et al.*[64]) applied to a pair of small adjacent windows that is slided along the alignment[2]. The index (henceforward abbreviated into "ILD") measures the difference in lengths of the most parsimonious trees calculated for the matrices corresponding to each window in the pair separately and for the combined matrix (which includes columns in both windows). The program used for determining the most parsimonious tree is Felsentein's DnaPars available in the Phylip package [66]. If X and Y are the matrices, L_Z the length of the most parcimonious tree calculated from a matrix Z, we have:

$$D_{XY} = L_{(X+Y)} - (L_X + L_Y)$$

where $X + Y$ is the matrix obtained by concatenation of X and Y and D_{XY} is the index for the pair of matrices (windows) X, Y. The value of this index is 0 when there is at least one tree on which the two matrices agree. It is large when minimizing the number of mutation steps in one set of characters increases this number in the other set.

The potential recombination points determined in this way, let us say there are p of them, partition the alignment into $p + 1$ regions. These regions are then pairwise compared using Farris's ILD index with, this time, a test to establish whether the incongruence length obtained is significative (we use the implementation of Farris's test available in the last version of the Paup package). The principle of such a test is the same as a Mann-Whitney test. The null hypothesis represents the case

[1] A local maximum is for now defined simply either as a position with a higher value than those obtained for its two neighbours in the alignment or as a position situated in the middle of a plateau of higher values in relation to the positions just before and after the plateau.

[2] Various sizes of windows were tried, between 3 and 7; this may change slightly the results obtained and a compromise must be made between sizes two small for observing significant differences between adjacent windows and too big for establishing recombination points with enough precision. A final choice was made on 5.

of no recombination (the two regions are congruent). The test is performed by applying a random partitioning technique. Let us suppose that the regions being compared are denoted by R and S. They represent matrices of m and n columns respectively. Matrix $R+S$ is randomly repartitioned a great number of times (typically 1000) into matrices P and Q with m and n columns. The null hypothesis will be rejected if only 5% of the indices obtained for P and Q are above that given by X and Y. Observe that this test is conservative. It allows to avoid a type I error only (rejection of the null hypothesis when it is true). If the null hypothesis is not rejected, we cannot guarantee that a recombination has not occurred.

At the end of such a comparison operation, we get a matrix $\mathcal{M}$ indexed on both lines and columns by the regions obtained in the first step. The value of a cell (i,j) in the matrix is a boolean indicating whether the hypothesis of a congruence between region i and region j is rejected (there is a recombination between the two regions of the alignment – this is denoted by a 0) or not (regions i and j are not significatively different in terms of their phylogeny – this is denoted by a 1).

From matrix $\mathcal{M}$, we then derive a matrix $\mathcal{N}$ of same dimension where the value of a cell (i,j) measures this time the distance between regions i and j. This will be given by the Hamming distance between lines i and j in $\mathcal{M}$. Such lines indicate whether regions i and j respectively present the same phylogenetic relation to all the other regions, that is, whether they "behave" the same in that sense. If the two regions have exactly the same behaviour in relation to the others, the distance between them is zero (they are phylogenetically the same). Since Farris's test is conservative, the meaning of a 1 is not the same as that of a 0 in matrix $\mathcal{M}$. When calculating the Hamming distance between two regions, we can therefore use a ponderation which depends on whether two aligned values from lines i and j in $\mathcal{M}$ are both 0, both 1, or a 0 and a 1. We could also utilize the original probabilities provided by the Farris test.

Matrix $\mathcal{N}$ is then fed into a clustering algorithm (for instance, Saitou's neighbour-joining [177]). This results in a tree that we can root using a same procedure as in Clustal (the root is chosen as the place such that the sum of the lengths in the left subtree is the same as that in the right subtree). Farris's test extended to n matrices [64] can then be recursively applied to left and right subtrees until the null hypothesis is rejected for one of these. This allows us to obtain a clustering of the regions such that, inside a given cluster, the regions represent the same phylogeny with a statistical chance of error below 5%. Furthermore, each obtained cluster should be maximal under such property (given the tree obtained by neighbour-joining).

This last part of the algorithm (derivation of $\mathcal{N}$ from $\mathcal{M}$, clustering and application of Farris's test to the clusters) has not yet been implemented. In what follows, we therefore discuss only the initial results obtained up to the construction of $\mathcal{M}$.

5.2.2.2 Preliminary applications

Before doing that, it is perhaps worth mentioning that the ILD index has the properties required for our purposes. This was checked by applying it to 8 artificial sequences of a 1000 nucleotides each whose evolution was simulated along a tree containing three speciation events using the Seq-Gen program of Rambaut [153]. To simulate a recent recombination, 200 nucleotides were manually exchanged between 2 of the 8 sequences. The ILD index was calculated for pairs of adjacent matrices of variable size. It is indeed maximal when the junction between the two matrices in a pair sits upon a recombination point although when the length of the matrices does not correspond to that of the recombinant region, the position where the maximal value of the ILD is obtained may be shifted of a few positions in relation to that point.

Various alignments of "real" sequences were used for an initial application of the algorithm. One is the alignment chosen by Jakobsen and Easteal [114] (corresponding to γ-globin gene sequences) for testing their method. This gene is located in a duplicated region and is therefore present in two copies in most primates. The six sequences in the alignment represent the two copies of this gene in the chimpanzee and the two copies of haplotypes A and B in man. Two gene conversions are present in the alignment: segment 381:1716 and segment 1302:1388 (positions here refer to those in the full alignment, not those in the alignment expurgated of its non informative sites). The two gene conversions are imbricated in one another, thus representing a good test for the algorithm. The phylogenetic relations inside each region are unambiguous.

The other data used represent some of the alignments of genes from different strains of *Escherichia coli* studied by Lecointre *et al.* [129]. The Farris test was used by them just to determine the presence of a recombination event in a gene. The exact location of the recombination points was then established by "eye" by various researchers in an independent fashion and the results confronted. Automatic methods could not be used because the alignments often contained various recombination events which they could not detect. The alignments used by us are those of the chromosomal genes implied in the metabolism of tryptophan *trpB* and DNA reparation *mutU*. The number of strains and their identification is different for the two alignments, but at least one sequence from each main strain type (A, B1, B2, D et UG (no group)) is present in the alignments. In this case, the phylogenetic relations may be ambiguous, even inside a same region, but we knew the positions of each recombination because the sequences had been extensively studied by experts.

regions	1	2	3	4	5	6	7
1	1	1	0	0	0	0	1
2	1	1	0	0	0	0	1
3	0	0	1	1	0	1	0
4	0	0	1	1	1	1	0
5	0	0	0	1	1	0	0
6	0	0	1	1	0	1	0
7	1	1	0	0	0	0	1

Table 5.1: Results obtained for the γ-globin genes. A 0 indicates that the ILD between two regions was not statistically significative, a 1 indicates the opposite.

The results obtained for the γ-globin genes are quite good (Table 5.1. The lines corresponding to regions 1, 2 and 7 are identical which means these regions will be clustered, as they should. They correspond to the parts coming just before and after the two imbricated gene conversions. Regions 3 and 6 may also be clustered. Region 4 presents a single difference (a 1) with regions 3 and 6 and possibly will be put together with them. Region 5 presents 2 differences (the two concerning a 0) and probably will stand apart as it should since it represents the region of the inner conversion. The results for the gene *mutU* are also interesting (Table 5.2). There is only one recombination in this case, covering regions 3, 4 and 5 (for a total of 25 informative sites; the first step slighly misplaces the end location of the recombination by 3 informative sites). Regions 3 and 4 are, as expected, identical. Region 5 is not and differs significantly with 4. Further analysis are required to determine whether this results from a shortcoming of the current method or there is indeed a

regions	1	2	3	4	5
1	1	1	1	1	0
2	1	1	1	0	0
3	1	1	1	0	0
4	1	0	0	1	0
5	1	0	0	0	1

Table 5.2: Results obtained for the alignment of the *mutU* gene. A 0 indicates that the ILD between two regions was not statistically significative, a 1 indicates the opposite.

regions	1	2	3	4	5	6	7	8	9	10	11
1	1	0	1	1	1	1	1	1	0	1	1
2	0	1	1	1	0	1	0	0	0	1	0
3	1	1	1	1	1	1	1	1	1	1	1
4	1	1	1	1	1	0	0	1	0	1	1
5	1	0	1	1	1	1	1	1	0	1	0
6	1	1	1	0	1	1	0	1	0	1	0
7	1	0	1	0	1	0	1	1	1	1	1
8	1	0	1	1	1	1	1	1	1	1	1
9	0	0	1	0	0	0	1	1	1	1	0
10	1	1	1	1	1	1	1	1	1	1	1
11	1	0	1	1	0	0	1	1	0	1	1

Table 5.3: Results obtained for the alignment of the *trpB* gene. A 0 indicates that the ILD between two regions was not statistically significative, a 1 indicates the opposite.

statistically significant recombination point between regions 3/4 and 5 that was missed by a simple visual inspection of the sequences. Finally, the results concerning the *trpB* gene (Table 5.3) are more difficult to interpret without further processing. What is worth observing nevertheless is that the recombinant segments are among the regions observed, that is, the frontier of such segments is approximately identified by the first step of the algorithm. Indeed, region 1 corresponds to the first segment, region 2 to the second and regions 3 to 11 to the last. As with the gene *mutU*, there is only a slight error in locating the exact location of the end of each recombinant segment. Both regions 1 and 2 cover 2 more informative sites than determined by eye.

5.2.3 Combinatorial Approach

5.2.3.1 Foreword

Our initial idea for trying to solve the problem of detecting the presence and exact location of recombinations as stated at the end of section 5.2.1 was to find a combinatorial algorithm. We based ourselves for this on an earlier work by Hein [98] and [99]. His paper of 1990 [98] describes an exhaustive method which corresponds to trying all possible partitions of the alignment. Of course, such an exhaustive exploration requires exponential time. This is, furthermore, probably not even a pertinent way of posing the problem and certainly, Hein's later heuristic for tackling the same problem [99] is not very realistic. It assumes only one recombination event may have happened at each position in the alignment. It also leaves open the question of what penalty should be given to a recombination event, and indeed, does not distinguish between a recombination event that makes a small change to the topology of a tree from one which deeply modifies it. There has recently been a paper by Husmeier and Wright [111] which spells the problem in the same terms but proposes a far more interesting solution than Hein's. This new approach uses maximum likelihood criteria to find the optimal partition of an alignment based on the recombination points it may contain. The paper did not present enough results for us to judge the quality of the method proposed, but it seemed promising, and worth exploring further.

Before this paper came to our attention though, we tried to improve Hein's exhaustive method, both in terms of the algorithm and of its biological grounding. This was a very interesting exercise in itself as it led us to a few subproblems all related to tree algorithms, some of which we believe may have an importance that goes far beyond phylogenies. We briefly sketch them below. They are currently being worked upon, first from a quite theoretical point of view, but with applications in mind.

5.2.3.2 Two Important Subproblems

5.2.3.2.1 Recombination Distance between Two Trees – Maximum Agreement Forest

One of the questions posed by Hein, which has been only tentatively addressed by him and others, is the following: given two trees that represent two phylogenies for a same set of species, how to calculate the recombination distance between them ? The recombination distance may be defined as the number of (possibly weighted) moves [101] or swaps [131] of subtrees one has to perform to one of the original trees in order to obtain the other. Hein has shown that, in the case of subtree transfers (not swaps), calculating such a distance is equivalent to finding the maximum agreement forest between two trees, where an agreement forest of two trees $\mathcal{T}_1$ an $\mathcal{T}_2$ is any forest which can be obtained from both $\mathcal{T}_1$ and $\mathcal{T}_2$ by cutting k arcs for some k and forcing arc contractions in each resulting component trees. If the size of a forest is defined as the number of components it contains, then the maximum agreement forest is the agreement forest with the smallest size. This size is always just one more than the recombination distance (formulated in terms of a subtree-transfer distance) we are seeking. This is again a hard problem (*i.e.*, it requires exponential time) and Hein *et al.* [101] propose a 3-approximation algorithm[3] for solving it. The bound is tight as indeed we found a family of examples for which it is attained. Finding these examples has enabled us to determine the exact case where Hein's algorithm behaves "badly"? We are currently using this information to try to obtain a better approximation algorithm.

Our interest in this problem is also in augmenting its range of applicability by considering a related subject (that of enumerating all *distinct* trees at a maximum recombination distance from a given tree $\mathcal{T}$) as well as other operations than just subtree transfers or swaps (which simulate translocations or transfers but not other rearrangements such as deletions or insertions). Thinking about such extensions has brought to our attention another problem which might help in practice to get faster solutions. This concerns identifying maximum homeomorphic or isomorphic subtrees among two or more trees.

[3] A p-approximation algorithm is one whose solution is guaranteed to be no more than p times that given by an optimal algorithm (if the optimum is a minimum as is the case with a distance).

5.2.3.2.2 Maximum Homeomorphic or Isomorphic Subtree

We recall that two trees are said to be isomorphic if there is a one-to-one correspondence between their nodes and arcs such that the incidence relationship is preserved. They are homeomorphic if one can be obtained from the other by the creation of arcs in series (*i.e.*, by the insertion of nodes of degree two) or by the merger of arcs in series. Various algorithms have been elaborated which we have been exploring ([3] [62] [63] seem the most interesting ones) but our attention has focused so far mainly on an older paper (that we believe has a potentially larger scope) due to Karp *et al.* [122], with which the story of the research of this author started (albeit for another problem related to sequences and not trees – see [174] [199]). The tree problem Karp *et al.* tried to solve is that of finding all occurrences of a tree in a (bigger) one, that is, find all subtrees of the big one that are isomorphic to the small. Their trees are oriented and binary. They could be labeled or unlabeled. If they are labeled, nodes and not arcs receive a label. It seems possible to adapt the algorithm to a case that would already interest us more, which is, given a set of trees, find all isomorphic subtrees that are common to the set. We have some very preliminary results on finding all isomorphic subtrees that appear repeated in a given tree and are working on extending the algorithm to handle various trees with labels at both the nodes and arcs, as well as look for homeomorphic common subtrees instead of isomorphic.

In the phylogenetic case that has concerned us in this chapter, locating common subtrees could lead to faster algorithms in practice for calculating the recombination distance between two trees. This will be the case if the trees in question have isomorphic subtrees which can be reduced to a single node. The recombination distance could then be calculated on the shrunken trees only.

The possible application to RNA structure comparison and prediction has been evoked in section 3.4.1.2.

5.3 Genome Evolution and Rearrangement Distance

Similarity between biological sequences is frequently measured by counting point mutations, that is, counting the minimum number of operations among substitution, insertion and deletion (of one nucleotidic base or amino acid) that are necessary to transform one sequence into another. However, as we saw, mutations happen also at a much larger scale. Whole fragments of genomes, involving sometimes thousands of DNA bases, often containing one or more genes, may thus be deleted, moved about, reversed, copied (duplicated) or exchanged (between chromosomes in the case of multi-chromosomal genomes). One may therefore wish to measure intra or inter-species distances based on such larger scale rearrangement events happening at a genomic level instead of on the history of point mutations. The distance we shall be concerned with here is calculated independent of there existing a phylogenetic tree relating the genomes between them. We also assume we know what these rearrangements are and just wish to use them as a way of measuring (and possibly reconstructing) evolution.

Methods for measuring large-scale evolutionary distances may be classified into three main categories depending on what type of rearrangement events are allowed and what kind of data is given as input to the algorithm. Data basically consist of genome representations. The representations come under the form of either one or more sequences of genes (this will be the case when the order of the genes along the chromosome is taken into account), or sets of genes (when that order is ignored). Each sequence or set corresponds to a chromosome. The types of rearrangement events allowed by the various approaches may of course depend on the data considered. Most *rearrangement distances* allow for one or more type of event among deletion, duplication, reversal, inter or intra-chromosome transposition and inter-chromosome translocation, and the gene order is taken into account [10] [37] [42] [95] [96] [97]. In the case of *breakpoint distances* instead [43], the gene order is considered, but the type of rearrangement events that may have changed the order is not fixed. Finally, algorithms which compute a *syntenic distance* [67] are exclusively reserved for multi-chromosomal genomes. The gene order is ignored (a genome consists therefore of a number of unordered sets, each one corresponding to a chromosome and containing elements that are the genes belonging to that chromosome) and only three types of rearrangement events are considered. These are *fusion* (of two sets into one), *fission* (of one set into two), and *translocation* (two sets exchange part of their contents). Hence, the only information that syntenic distance is able to capture is the *synteny* of genes (if we take the narrow view of considering that two genes are *syntenic* if they belong to the same chromosome). Syntenic distance cannot keep track of intra-chromosomal events. Obtaining such distance requires, as usual, determining the minimum number of operations among the three just mentioned that are necessary to transform a first genome into a second. Since genomes are, in this case, seen as mere bags of genes, this is equivalent to calculating the minimum number of moves needed to obtain the same bags.

Syntenic distance is proved to be a hard problem in [51] and a 2-approximation algorithm is suggested (algorithm $\mathcal{H}$). The bound is tight. This bound is the same for Liben-Nowell's algorithm [132] which is an improvement of Ferretti's [67]. Liben-Nowell also proves that, assuming the instance is partitioned into *components* with distinct genes, no approximation ratio better than 2 can be reached as long as only intra-component moves are performed which is what all algorithms proposed so far do. Both in [51] and [132], many interesting properties of syntenic distances are proved and some particular classes of instances, such as *linear synteny* and *exact synteny*, are examined. For one particular class of linear synteny, it is proved in [133] that under certain conditions (requiring inclusion properties among the sets of syntenic genes, resulting in what is called a *nested synteny*), the distance is computable in polynomial time. Finally, in [132], the *syntenic diameter*, that is, the length of the longest optimal move sequence for an instance of size n, is discussed. It is conjectured that its size is $2n-3$.

The work we have done so far on this problem has led us to some new theoretical and algorithmical results concerning both nested synteny and general synteny. Finding an optimal solution for the syntenic distance, in particular for nested synteny, is deeply dependent on being able to calculate the syntenic diameter (see [132]), and thus provide an optimal move sequence for solving an instance of it. We therefore started by solving the conjecture by Liben-Nowell for the syntenic diameter. We showed that this diameter is not $2n-3$ but $2n-4$. We generalized the result to include what we call *bidimensional syntenic diameter*. When the initial instance has more than one component, we then suggested a way for performing inter-component moves, something that was not done in previous algorithms. We followed this by introducing an algorithm which is, we believe, simpler, more general and efficient than in [133] for obtaining a move sequence that is optimal for the exact and linear cases. In addition, we sketched an extension which may optimally solve the connected general nested case. Finally, concerning general synteny, we proved an inclusion theorem equivalent to the one shown in [133] for linear synteny.

Details on all these results may be found in [151]. The wary reader should be warned that this is a quite technical paper. For this reason, and to keep with the spirit of this memoir (that it should be accessible to all), I chose not to go further into this subject here. It is difficult though to resist the temptation to encourage the reader with a taste for combinatorial puzzles to take a look at the paper and try playing with the problem in his mind.

5.4 Critique and Perspectives

5.4.1 Critique and Perspectives on Recombinations

Although reasonably interesting results seem to have been obtained with a statistical approach to the problem of detecting the presence and exact location of recombinations, we believe this can be much improved. More in-depth tests are also needed to pinpoint current limitations of the algorithm and ways of improving it. Although also the combinatorial approach appears an unrealistic alternative for now, it is not impossible to imagine a method that would mix the two in a felicitous way. As mentioned in the introduction to this chapter, it remains unclear, at least to this author, how far from biology we are with our current methods and, if far, what needs to be added or changed to our model to get closer to it. An answer to these questions is important as we shall go to considering a number of other problems related to the presence of recombination. There are at least two of them that have called our attention. One is the problem of phylogenetic reconstruction in the presence of recombination and the other the problem of a species tree versus a gene tree when recombinations have happened (these are not considered in current algorithms that compare the two kinds of trees [134] [146]). The case of phylogenetic reconstruction in the presence of recombination is particularly daunting. Among the current approaches to tackle this problem, probably the most frequently used consists in establishing the recombination points "by eye" and then reconstructing as many trees are there are phylogenetically different regions determined by the recombination points [139]. The "by eye" detection is often made on a multiple alignment of a set of sequences. Yet precisely because of the presence of recombination events, obtaining such an alignment may be an error-fraught operation. There is a same sort of circularity here as that observed in algorithms that perform a multiple alignment based on a score according to a tree while deducing the tree from an alignment (in general pairwise) of the sequences.

5.4.2 Critique and Perspectives on Rearrangements

5.4.2.1 Reconstructing Ancestral Operons

A pathogenic bacterium, such as *Escherichia coli* K-12, may cause different types of diseases: diarrhoea if localized in the intestine; cystitis or pyelonephritis when situated in the urinary bladder; meningitis when the habitat is the brain. In some cases, a same chromosomal region, often an operon, may give rise to two different pathology. An example is the operon *afa*-3 in *Escherichia coli* which induces the development of cystitis as well as diarrhoea. On the other hand, different operons may trigger the same pathology (for instance, again in *Escherichia coli*, *afa*, *cs*, *aaf*, *fim*, etc [56] [73] [75] [189]). One may wonder about the reasons for such a diversity of behaviour. The possibility of an escape of the pathogen from the immune system of the host has been suggested. A therapeutical escape might also be possible although the existence in bacteria of a gene causing resistance to antibiotics has never been demonstrated. The pathogenic function may itself vary slightly depending on the operon and its regulation.

If one considers the example of pathogenic operons in *Escherichia coli* which are the cause of intestinal or urinary infection (with the exception of secretory operons of type III), one may see that the genes in each operon may be divided into subgroups with a well-defined function. Four determinants may be described which, respectively:

- code for the pathogenicity (adhesin, invasin);
- allow for the secretion of the pathogenetic factors;
- code for the pili that are sometimes needed for the fixation of the adhesin to the bacterial wall;
- code for the transcriptional regulators specific to the operon.

The two subgroups always represented are the pathogenicity factors and their secretory system. The transcriptional regulators specific to each are either coded by a gene located in the same operon (in *E. coli*, we have the example of the family *papI-papB*), or are genetically not linked (again in *E. coli*, example of the family *cfaR*). Considerable rearrangements have often been observed (deletion of a gene, inversion, translocation). Two types of similarity remain which nevertheless link these genes together. One is the similarity in terms of the sequences of the genes having same function and the other is the similarity of the functional subgroups. One may therefore ask whether such similarities are the signature of an ancestral operon. Such an operon could be extinct or still be present among current ones. It seems important to be able to reconstruct this ancestral operon if we want to understand why there is such an incredible diversity for the expression of a single function. The root and structure of the reconstructed tree could inform us on the development of the family of operons. It would then be possible, by genetic engineering, to materially recreate such an ancestral operon in the laboratory and study its behaviour.

5.4.2.2 Reconstructing the Evolution of Non Coding Regions

It would be interesting also to be able to reconstruct the phylogenetic tree of the rearrangements having possibly taken place in the non coding regions of a genome. The structure of the tree could enable us to see how the regulation system has evolved, in particular, to determine, among other things, whether the regulatory sequences have evolved in parallel with the genes or in a completely independent fashion, and whether they are genetically linked with the genes up and downstream from them. It may appear strange at first sight to want to reconstruct such trees as we attribute in general a function to a gene while we would be dealing instead with non coding sequences. Our work on the identification of regulatory sequences could however enable us to describe a consensus for each type of regulator/promoter. Such consensi could in turn allow us to assign one or more regulatory functions to the non coding sequences. It is from this approach that we wish to develop the concept of a phylogeny of non coding sequences.

5.4.2.3 Exploring Other (More Realistic) Distances

It is clear that, in relation to both cases described in the previous sections, we need to explore more realistic rearrangement distances than we have been doing so far (although we believe the algorithm in [151] may be extended and improved and we intend to do so). One reason is, of course, that we shall be dealing with genomes, or genome fragments, that are located in a single chromosome. Another more important reason is that we want not only to compare such genomes in terms of the rearrangements they have suffered and to measure their general syntenic distance, but also to use this information to reconstruct, *a posteriori* or simultaneously, the evolution of the genomes or fragments of genomes. By general syntenic distance in this case, we mean the number of duplications, deletions, transpositions, inversions and insertions needed to make two chromosomal segments syntenic, *i.e.*, having the same genes, in the same order and orientation. We propose to do that in two steps.

The first one requires being able to locate the recombinations that happened. If we are in a situation where the genome fragments are pre-aligned (perhaps based on some other information), then we have a first algorithm which should be able to do that but needs to be improved. If the genome fragments are not aligned, then we need another approach.

The second step involves exploring the models for calculating a rearrangement distance which were introduced by Sankoff and co-workers. We are particularly interested in what has been called "breakpoint distance" and "breakpoint phylogeny" [26] [28] [179]. A breakpoint indicates the presence of two genes adjacent in a genome but not in the other. Other more complex distances were considered by Sankoff which take into account not just the presence and location of a rearrangement event, but also its type. For the purpose of a phylogenetic reconstruction in particular, this poses the problem of which weight to affect to each type [27]. Some of these methods must also be extended to treat deletions and insertions of genes or genome fragments.

Chapter 6

Some Perspectives and a Last Personal Note in Guise of Conclusion

Co-workers and/or collaborators

Sébastien Provencher

Whoever else feels like, at least talk, perhaps more

How happy is the little Stone
That rambles in the Road alone,
And doesn't care about Careers
And Exigencies never fears –
Whose Coat of elemental Brown
A passing Universe put on,
And independent as the Sun
Associates or glows alone,
Fulfilling absolute Decree
In casual simplicity –

Poem #1510. Emily Dickinson.
The Complete Poems of Emily Dickinson.
Thomas H. Johnson (Ed.).
Faber and Faber, 1975.

It may appear to the reader that there were enough perspectives in the previous chapters to keep one busy for a good many years without wanting to add a few more. This is true.

As briefly alluded to in the introduction, it seems however impossible to wish to understand even the smallest issue without trying to have an idea of the whole. The whole for biology could be, among others: one (or many) complete genome(s); the cell; the environment; time, within which all things are happening. The whole for algorithmics includes a huge repertoire of models (strings but also trees, graphs etc and the characteristics and properties thereof) and methods (discrete, numerical, probabilistic; for searching, sorting, organizing, optimizing (given a certain function), estimating (parameters) etc).

In the previous pages, we also made a reference to a lot of information we are not currently considering, or have barely (and, quite possibly, inadequately) started looking at. The main ones are precisely those that could give us a glimpse of the whole: metabolic and regulatory pathways on the one hand, comparative genomics and evolution on the other.

The quantity of information coming to our databases is indeed huge (a look at a single encyclopedia such as http://www.genome.ad.jp/kegg/ should be enough to convince the reader), and will become awesome in the next few years. This has led even some of the more skeptic to say that, really, we need the computer to treat this mass of data. This is true but is just a very small part of the truth if one means by that the computer's capacity to physically store, handle and quickly sift through big quantities of information. What computer science, in particular algorithmics, and mathematics, bring to biology is also, above all, an experience with and taste for trying to integrate and organize data, discover regularities, find rules, explore the inner structure of problems or objects of investigation, build general models that try to weave threads among such objects and, finally, obtain the means to represent, test, exploit, question and revise the theories it elaborates (of course, even when these involve very big quantities of data, but not only).

What does the researcher in computer science, or mathematics, gain from biology ? The (easy) answer is, he gets a lot of applications for his theoretical work but, to be honest, probably no more than he would get in other areas (he might even happily do without any applications at all). Why then has he chosen biology ? There are two answers to this. One is: why not ? This may seem a little provocative. Please, do not see it as being so. This is said only to better stress the importance of the second answer. This latter will take us back to where we started: because he loves it very deeply. Between his world of abstract mathematical concepts and that of intricate biological organisms, there is the same fascinating complexity of objects and relations, the same alluring mixture of apparent disorder and wished for order, and, for the person contemplating all this, the same urge to grasp, as much as and as large as one possibly can.

As a last (very) personal note, when I said that it seems impossible to wish to understand even the smallest issue without trying to have an idea of the whole, to tell you the truth, what I really wished to say is that, it is impossible *for me*. It has happened to me many times to attend to seminars, or have a talk with someone, and feel just dizzy with happiness to be living at such a time. Among the latest subjects presented or discussed (in disorder and to name just a few) were biochemical pathways, the virtual cell, microsporidies, "shrinking" genomes, natural selection and regulatory gene circuitry. You will have noticed that all these concern biology. They also go beyond the subjects that were covered in this memoir, and I mentioned none on algorithmical subjects although many exist. There will be times when one will have to be reasonable. Like one of the poems quoted at the start of this memoir said, we are all alone, and soon it will be night; but before that, there is the sun shining and perhaps we do not have to do like in the other poem, just sit and wait for it to set down. Perhaps, sometimes, we have to be a little daring and free in our minds.

Is therefore the encounter of biology and algorithmics a meeting of sorts or a marriage made in heaven ? Pick your choice.

Le voleur
M'a tout emporté, sauf
La lune qui était à ma fenêtre.

Ryôkan.
Anthologie de la poésie japonaise classique.
G. Renondeau (Ed. et trad.).
Gallimard, 1971.

All things
Silent:
Peace and quiet

Kyorai.
The Penguin Book of Japanese Verse.
G. Bownas and A. Thwaite (Eds. and transls.).
Penguin Books, 1964.

Appendix A

Recapitulative list of papers published, in press, submitted and in preparation

A.1 Published

A.1.1 International Journals

- M.-F. Sagot, A. Viari, J. Pothier and H. Soldano. Finding flexible patterns in a text – An application to 3D molecular matching. *Comput. Appl. Biosci.*, 11:59-70, 1995, work initially presented at *First International IEEE Workshop on Shape and Pattern Matching in Computational Biology*, Seattle, Washington, USA, 1995.

- M.-F. Sagot, A. Viari and H. Soldano. Multiple comparison: a peptide matching approach. *Theor. Comput. Sci.*, 180(1-2):115-137, 1997, extended version of work presented at *Combinatorial Pattern Matching Conf. 95*, Helsinki, Finland, 1995.

- M.-F. Sagot and G. Myers. Identifying satellites and periodic repetitions in biological sequences. *Journal of Computational Biology*, 5(3):539-554, extended version of work presented at *RECOMB 98*, New York, USA, 1998.

- A. Vanet, L. Marsan and M.-F. Sagot. Promoter sequences and algorithmical methods for identifying them. *Research in Microbiology*, 150:779-799, 1999.

- A. Vanet, L. Marsan, A. Labigne and M.-F. Sagot. Inferring regulatory elements from a complete genome. An application to the analysis of *Helicobacter pylori* σ^{80} family of promoter signals. *Journal of Molecular Biology*, 297:335-353, 2000.

- L. Marsan and M.-F. Sagot. Algorithms for extracting structured motifs using a suffix tree with an application to promoter and regulatory site consensus identification. *Journal of Computational Biology*, in press, extended version of work presented at *RECOMB 00*, Tokyo, Japan, 2000.

A.1.2 International Conferences

- M.-F. Sagot, V. Escalier, A. Viari and H. Soldano. Searching for repeated words in a text allowing for mismatches and gaps. in R. Baeza-Yates and U. Manber, editors, *Proc. of the Second South American Workshop on String Processing*, Viñas del Mar, Chile, pages 87-100. University of Santiago, Chile, 1995.
- M.-F. Sagot, A. Viari and H. Soldano. Multiple comparison: a peptide matching approach. in Z. Galil and E. Ukkonen, editors, *Combinatorial Pattern Matching*, Helsinki, Finland, volume 937 of *Lecture Notes in Computer Science*, pages 366-385. Springer-Verlag, 1995.
- M.-F. Sagot, A. Viari and H. Soldano. A distance-based block searching algorithm. in C. Rawlings, D. Clark, R. Altman, L. Hunter, T. Lengauer and S. Wodak, editors, *Proc. Third International Symposium on Intelligent Systems for Molecular Biology*, Cambridge, England, pages 322-331. AAAI Press, 1995.
- M.-F. Sagot and A. Viari. A double combinatorial approach to discovering patterns in biological sequences. in D. Hirschberg and G. Myers, editors, *Combinatorial Pattern Matching*, Laguna Beach, California, USA, volume 1075 of *Lecture Notes in Computer Science*, pages 186-208. Springer-Verlag, 1996.
- M.-F. Sagot and A. Viari. Flexible identification of structural objects in nucleic acid sequences: palindromes, mirror repeats, pseudoknots and triple helices. in J. Hein and A. Apostolico, editors, *Combinatorial Pattern Matching*, Aarhus, Danemark, volume 1264 of *Lecture Notes in Computer Science*, pages 224-246, Springer Verlag, 1997.
- M.-F. Sagot and E. W. Myers. Identifying satellites in nucleic acid sequences. in S. Istrail, P. Pevzner and M. Waterman, editors, *RECOMB 98*, New York, USA, *Proceedings of the Second Annual International Conference on Computational Molecular Biology*, pages 234-242, ACM Press, 1998.
- M.-F. Sagot. Spelling approximate repeated or common motifs using a suffix tree. in C. L. Lucchesi and A. V. Moura, editors, *LATIN'98, Theoretical Informatics.*, Campinas, Brazil, volume 1380 of *Lecture Notes in Computer Science*, pages 111-127, Springer Verlag, 1998.
- L. Marsan and M.-F. Sagot. Extracting structured motifs using a suffix tree – Algorithms and application to promoter consensus identification. in R. Shamir, S. Miyano, S. Istrail, P. Pevzner and M. Waterman, editors, *RECOMB 00*, Tokyo, Japan, *Proceedings of the Second Annual International Conference on Computational Molecular Biology*, pages 210-219, ACM Press, 2000.

A.2 Submitted

- P. Blayo, P. Rouzé and M.-F. Sagot. Orphan gene finding – An exon assembly approach.
- Nadia Pisanti and M.-F. Sagot. Further thoughts on syntenic distance between genomes.
- Y. Wakabayashi and M.-F. Sagot. Pattern inference under many guises.

A.3 In Preparation

- J.-J. Daudin, S. Robin, M.-F. Sagot and S. Schbath. (On the probability of an inferred motif, no fixed title for now).
- S. Lhullier and M.-F. Sagot. (On inferring RNA structural motifs, no fixed title for now).
- M.-F. Sagot and A. Viari. Combinatorial sampling.
- M.-F. Sagot and A. Viari. Application of a combinatorial sampling approach to identifying sequence motifs that are structurally and/or functionally related.
- C. Mathé, T. Schiex, M.-F. Sagot, P. Rouzé *et al.* (Survey on eukaryotic gene finding methods, no fixed title for now).
- P. Blayo, P. Rouzé and M.-F. Sagot. (On finding all suboptimal gene assemblies, no fixed title for now).
- M. Zelwer, P. Darlu and M.-F. Sagot. (On detecting the presence and location of recombination points, no fixed title for now).

Appendix B

List of algorithmical codes available or soon to be and status thereof

All algorithms are written in C, except for the one on finding a (single) best assembly for orphan genes which is written in C++. Although they may be used in practice, and have indeed been used many times, all current implementations represent just basic cores, not fully developed software.

B.1 Gene Expression Regulation

- Inferring single models on DNA alphabet, one (repeats) or many sequences (common motifs)
  - Name of algorithm: Moivre (see thesis)
  - Paper(s) describing it:

    M.-F. Sagot, V. Escalier, A. Viari and H. Soldano. Searching for repeated words in a text allowing for mismatches and gaps. in R. Baeza-Yates and U. Manber, editors, *Proc. of the Second South American Workshop on String Processing*, Viñas del Mar, Chile, pages 87-100. University of Santiago, Chile, 1995.

    * M.-F. Sagot, A. Viari and H. Soldano.
  - Person(s) who implemented: V. Escalier, M.-F. Sagot and A. Viari
  - Year of last modification: 1998
  - Status of the implementation: functional, primitive input/output, needs revision
  - Availability: now as is

- Inferring single models, combinatorial DNA alphabet, one (repeats) or many sequences (common motifs),
  - Name of algorithm: Combi (no errors) and LeCombi (with errors) (see thesis)
  - Paper(s) describing it:

    M.-F. Sagot and A. Viari. A double combinatorial approach to discovering patterns in biological sequences. in D. Hirschberg and G. Myers, editors, *Combinatorial Pattern Matching*, Laguna Beach, California, USA, volume 1075 of *Lecture Notes in Computer Science*, pages 186-208. Springer-Verlag, 1996.

- Person(s) who implemented: M.-F. Sagot (plus libraries for reading sequences in Fasta format and stack manipulation borrowed from A. Viari)
- Year of last modification: 1997
- Status of the implementation: functional, primitive input/output, needs revision
- Availability: now as is

• Inferring structured models on DNA alphabet, one (repeats) sequence (using a suffix tree)

- Name of algorithm: No name yet
- Paper(s) describing it:

  M.-F. Sagot. Spelling approximate repeated or common motifs using a suffix tree. in C. L. Lucchesi and A. V. Moura, editors, *LATIN'98, Theoretical Informatics.*, Campinas, Brazil, volume 1380 of *Lecture Notes in Computer Science*, pages 111-127, Springer Verlag, 1998.
- Person(s) who implemented: L. Marsan and J. Allali (plus one library from A. Viari)
- Year of last modification: being improved still
- Status of the implementation: not yet functional
- Availability: end of 2000 or beginning of 2001

• Inferring structured models on DNA or degenerate alphabet, many sequences

- Name of algorithm: No name yet
- Paper(s) describing it:

  M.-F. Sagot. Spelling approximate repeated or common motifs using a suffix tree. in C. L. Lucchesi and A. V. Moura, editors, *LATIN'98, Theoretical Informatics.*, Campinas, Brazil, volume 1380 of *Lecture Notes in Computer Science*, pages 111-127, Springer Verlag, 1998.

  L. Marsan and M.-F. Sagot. Extracting structured motifs using a suffix tree – Algorithms and application to promoter consensus identification. in , editors, *RECOMB 00*, Tokyo, Japan, *Proceedings of the Second Annual International Conference on Computational Molecular Biology*, pages -, ACM Press, 2000.

  L. Marsan and M.-F. Sagot. Algorithms for etracting structured motifs using a suffix tree with an application to promoter and regulatory site consensus identification. *Journal of Computational Biology*, in press, extended version of work presented at *RECOMB 00*, Tokyo, Japan, 2000.
- Person(s) who implemented: L. Marsan and J. Allali (plus one library borrowed from A. Viari and a random sequence generator from E. Coward [47])
- Year of last modification: being improved still
- Status of the implementation: functional for models composed of one or two boxes (one of the algorithms in Recomb paper only), with statistical evaluation done by simulation, being improved/implemented for more than two boxes and other methods of evaluation
- Availability: now as is for functional part, end of 2000 or beginning of 2001 for part being improved/implemented

B.2 Role and Relation of Geometry and Topology to Sequence and Function

- Inferring small RNA (or DNA) structural motifs (palindromes, pseudo-knots, mirror or identical images, triplexes), one (single motif or repeated) or many sequences
  - Name of algorithm: No name yet
  - Paper(s) describing it:
    * M.-F. Sagot and A. Viari. Flexible identification of structural objects in nucleic acid sequences: palindromes, mirror repeats, pseudoknots and triple helices. in J. Hein and A. Apostolico, editors, *Combinatorial Pattern Matching*, Aarhus, Danemark, volume 1264 of *Lecture Notes in Computer Science*, pages 224-246, Springer Verlag, 1997.
    * S. Lhullier and M.-F. Sagot. (in preparation).
  - Person(s) who implemented: M.-F. Sagot for earlier version (plus two libraries from A. Viari), S. Lhullier for later extensible (to more RNA structure matching) version
  - Year of last modification: 1997 (MFS), being implemented still (SL)
  - Status of the implementation:
    * MFS: three different implementations (palindromes, pseudo-knots and triplexes) all functional, primitive input/output, needs revision
    * SL: close to functional for small motifs
  - Availability: now for MFS, beginning of 2001 for SL if not military service
- Inferring models in proteins, one or many sequences
  - Name of algorithm: Poivre (Protein + mOIVRE) (see thesis)
  - Paper(s) describing it:
    * M.-F. Sagot, A. Viari and H. Soldano. Multiple comparison: a peptide matching approach. *Theor. Comput. Sci.*, 180(1-2):115-137, 1997, extended version of work presented at *Combinatorial Pattern Matching Conf. 95*, Helsinki, Finland, 1995.
    * M.-F. Sagot and A. Viari. A double combinatorial approach to discovering patterns in biological sequences. in D. Hirschberg and G. Myers, editors, *Combinatorial Pattern Matching*, Laguna Beach, California, USA, volume 1075 of *Lecture Notes in Computer Science*, pages 186-208. Springer-Verlag, 1996.
  - Person(s) who implemented: M.-F. Sagot (plus two libraries borrowed from A. Viari)
  - Year of last modification: 1998
  - Status of the implementation: functional, primitive input/output, needs revision
  - Availability: now
- Inferring models in proteins, one or many sequences
  - Name of algorithm: Klast (Kmr + bLAST) (see thesis)
  - Paper(s) describing it:

    * M.-F. Sagot, A. Viari and H. Soldano. A distance-based block searching algorithm. in C. Rawlings, D. Clark, R. Altman, L. Hunter, T. Lengauer and S. Wodak, editors, *Proc. Third International Symposium on Intelligent Systems for Molecular Biology*, Cambridge, England, pages 322-331. AAAI Press, 1995.
  - Person(s) who implemented: M.-F. Sagot (plus two libraries borrowed from A. Viari)
  - Year of last modification: 1997
  - Status of the implementation: functional, primitive input/output, needs revision
  - Availability: now

- Combinatorial sampling, proteins, many sequences
  - Name of algorithm: CoSamp
  - Paper(s) describing it:
    * M.-F. Sagot and A. Viari. Combinatorial sampling.
    * M.-F. Sagot and A. Viari. Application of a combinatorial sampling approach to identifying sequence motifs that are structurally and/or functionally related.
  - Person(s) who implemented: M.-F. Sagot (model combinatorial sampling) and A. Viari (model sieving with some modifications from MFS)
  - Year of last modification: 1998
  - Status of the implementation: functional, needs revision
  - Availability: now

B.3 Genome Structure

- Finding a (single) best assembly for orphan genes
  - Name of algorithm: No name yet
  - Paper(s) describing it:
    * P. Blayo, P. Rouzé and M.-F. Sagot. Orphan gene finding – An exon assembly approach.
  - Person(s) who implemented: P. Blayo
  - Year of last modification: 2000
  - Status of the implementation: functional (but still being improved)
  - Availability: now

- Detecting satellites
  - Name of algorithm: satellites
  - Paper(s) describing it:
    * M.-F. Sagot and E. W. Myers. Identifying satellites in nucleic acid sequences. in S. Istrail, P. Pevzner and M. Waterman, editors, *RECOMB 98*, New York, USA, *Proceedings of the Second Annual International Conference on Computational Molecular Biology*, pages 234-242, ACM Press, 1998.

* M.-F. Sagot and G. Myers. Identifying satellites and periodic repetitions in biological sequences. *Journal of Computational Biology*, 5(3):539-554, extended version of work presented at *RECOMB 98*, New York, USA, 1998.

- Person(s) who implemented: G. Myers (filter) and M.-F. Sagot (satellite model detection and evaluation)
- Year of last modification: 2000
- Status of the implementation: functional, not the most decent input/output
- Observation: exists with a nice web interface (done by E. Poiret) and the possibility of looking for mixed direct/inverted repeats (also done by E. Poiret) but his version contains a bug
- Availability: now for MFS version, by the end of 2000 for revised (by MFS) version with nice interface

B.4 Genome Dynamics and Evolution

- Detecting the presence and exact location of recombination points in aligned (closely related) DNA sequences
  - Name of algorithm: No name yet
  - Paper(s) describing it:
    * M. Zelwer, P. Darlu and M.-F. Sagot. (in preparation).
  - Person(s) who implemented: M. Zelwer (uses J. Farris's ILD test as implemented in Paup and J. Felsenstein's DnaPars as implemented in Phylip)
  - Year of last modification: being implemented still
  - Status of the implementation: part of it is functional (see text)
  - Availability: by the end of 2000

Bibliography

[1] S. F. Altschul, W. Gish, W. Miller, E. W. Myers, and D. J. Lipman. Basic local alignment search tool. *J. Mol. Biol.*, 215:403–410, 1990.

[2] C. Ambrose and M. Bina. Strategy for statistical-mapping of potential regulatory regions in the human genome. *J. Mol. Biol.*, 216:485–490, 1990.

[3] A. Amir and D. Keselman. Maximum agreement subtree in a set of evolutionary trees – metrics and efficient algorithms. *SIAM J. on Computing*, 26:1656–1669, 1997.

[4] A. Apostolico and G. Bejerano. Optimal amnesic probabilistic automata or how to learn and classify proteins in linear time and space. In R. Shamir, S. Miyano, S. Istrail, P. Pevzner, and M. Waterman, editors, *RECOMB'00. Proceedings of Fourth Annual International Conference on Computational Molecular Biology*, pages 25–32. ACM Press, 2000.

[5] L. Arvestad. Aligning coding DNA in the presence of frameshift errors. In Farach-Colton M, editor, *Combinatorial Pattern Matching, 1997*, volume 1264 of *Lecture Notes in Computer Science*, pages 180–190. Springer-Verlag, 1997.

[6] N. E. Assali, R. Mache, and S. L. de Goer. Evidence for a composite phylogenetic origin of the plastid genome of the brown algae *Pylaiella littoralis*. *Plant Mol. Biol.*, 15:307–315, 1990.

[7] S. Bachellier, J.-M. Clément, M. Hofnung, and E. Gilson. Bacterial interspersed mosaic elements (BIMEs) are a major source of sequence polymorphism in *Escherichia coli* intergenic regions including specific associations with a new insertion sequence. *Genetics*, 145:551–562, 1997.

[8] S. Bachellier, J.-M. Clément, M. Hofnung, and E. Gilson. Short palindromic repetitive DNA elements in enterobacteria: a survey. *Research in Microbiology*, 150:627–639, 1999.

[9] S. Bachellier, W. Saurin, D. Perrin, M. Hofnung, and E. Gilson. Structural and functional diversity among bacterial interspersed mosaic elements (BIMEs). *Mol. Microbiol.*, 12:61–70, 1994.

[10] V. Bafna and P. Pevzner. Sorting by transpositions. *SIAM Journal on Discrete Mathematics*, 11(2):224–240, 1998.

[11] T. L. Bailey. *Discovering motifs in DNA and protein sequences: The approximate common substring problem.* Ph.D. thesis, University of California, San Diego, USA, 1995.

[12] T. L. Bailey and C. Elkan. Fitting a mixture model by expectation maximization to discover motifs in biopolymers. In *Second International Symposium on Intelligent Systems for Molecular Biology*, pages 28–36, Stanford University, Stanford, CA, USA, 1994.

[13] T. L. Bailey and C. Elkan. Unsupervised learning of multiple motifs in biopolymers using EM. *Machine Learn.*, 21:51–80, 1995.

[14] T. L. Bailey and C. Elkan. The value of prior knowledge in discovering motifs with MEME. In C. Rawlings, D. Clark, R. Altman, L. Hunter, T. Lengauer, and S. Wodak, editors, *Third International Symposium on Intelligent Systems for Molecular Biology*, pages 21–29, Cambridge, England, 1995.

[15] T. L. Bailey and C. Elkan. The megaprior heuristic for discovering sequence patterns. In D. J. States, P. Agarwal, T. Gaasterland, L. Hunter, and R. F. Smith, editors, *Fourth International Symposium on Intelligent Systems for Molecular Biology*, pages 15–24, Saint Louis, USA, 1996.

[16] T. L. Bailey and C. Elkan. ParaMEME: a parallel implementation and a web interface for a DNA and protein motif discovery tool. *Comput. Appl. Biosci.*, 12:303–310, 1996.

[17] T. L. Bailey and M. Gribskov. Score distributions for simultaneous matching to multiple motifs. *J. Comp. Biol.*, 4:45–59, 1997.

[18] A. Bairoch. Prosite: A dictionary of protein sites and patterns. *Nucleic Acids Res.*, 20:2013–2018, 1992.

[19] A. Bairoch and C. Boeckmann. The Swiss-Prot protein sequence data bank. *Nucleic Acids. Res.*, 19:2247–2249, 1991.

[20] G. Benson. An algorithm for finding tandem repeats of unspecified pattern size. In M. Waterman S. Istrail, P. Pevzner, editor, *Proc. of the Second Annual International Conference on Computational Molecular Biology (RECOMB 98)*, pages 20–29. ACM Press, 1998.

[21] G. Benson. Tandem repeats finder: a program to analyze DNA sequences. *Nucleic Acids Research*, 27:573–580, 1999.

[22] G. Benson and M. Waterman. A method for fast database search for all k-nucleotide repeats. *Nucleic Acids Research*, 22:4828–4836, 1994.

[23] O. G. Berg and P. H. von Hippel. Selection of DNA binding sites by regulatory proteins. II. The binding specificity of cyclic AMP receptor protein to recognition sites. *J. Mol. Biol.*, 200:709–723, 1988.

[24] E. M. Bik, A. E. Bunschoten, R. D. Gouw, and F. R. Mooi. Genesis of the novel epidemic *Vibrio cholerae* o139 strain: evidence for horizontal transfert of genes involved in polysaccharide synthesis. *EMBO J.*, 14:209–216, 1995.

[25] B. Billoud, M. Kontic, and A. Viari. Palingol: a declarative programming language to describe nucleic acids' secondary structures and to scan sequence databases. *Nucleic Acids Res.*, 24:1395–1403, 1996.

[26] M. Blanchette, G. Bourque, and D. Sankoff. Breakpoint phylogenies. In *Genome Informatics 1997*, pages 5–34, 1997.

[27] M. Blanchette, T. Kunisawa, and D. Sankoff. Parametric genome rearrangement. *Gene*, 172:11–17, 1996.

[28] M. Blanchette, T. Kunisawa, and D. Sankoff. Gene order breakpoint evidence in animal mitochondrial phylogeny. *J. Mol. Evol.*, 49:193–203, 1999.

[29] P. Blayo, P. Rouzé, and M.-F. Sagot. Orphan gene finding – an exon assembly approach. 2000. submitted.

[30] D. Bouthinon and H. Soldano. A new method to predict the consensus secondary structure of a set of unaligned RNA sequences. *Bioinformatics*, 15:785–798, 1999.

[31] A. Brazma, I. Jonassen, I. Eidhammer, and D. Gilbert. Approaches to the automatic discovery of patterns in biosequences. *J. Comp. Biol.*, 5:279–305, 1998.

[32] G. S. Brodal, R. Lyngsø, C. N. S. Pedersen, and J. Stoye. Finding maximal pairs with bounded gap. *J. Discrete Algorithms.* to appear in the Special Issue on Matching Patterns.

[33] C. Burge. *Identification of genes in human genomic DNA*. Ph.D. thesis, Stanford University, Stanford, California, USA, 1997.

[34] C. Burge. Modeling dependencies in pre-mRNA splicing signals. In S. Salzberg, D. Searls, and S. Kasif, editors, *Computational Methods in Molecular Biology*, pages 127–163. Elsevier Science, 1998.

[35] C. Burge and S. Karlin. Prediction of complete gene structures in human genomic DNA. *J. Mol. Biol.*, 268:78–94, 1997.

[36] M. Burset and R. Guigó. Evaluation of gene structure prediction programs. *Genomics*, 34:353–367, 1996.

[37] A. Caprara. Sorting by reversals is difficult. In *1st ACM International Conference on Computational Molecular Biology, (RECOMB 97)*, pages 75–83, 1997.

[38] Y. D'Aubenton Carafa, E. Brody, and C. Thermes. Prediction of Rho-independent *Escherichia coli* transciption terminators. A statistical analysis of their RNA stem-loop structures. *J. Mol. Biol.*, 216:835–858, 1990.

[39] L. R. Cardon and G. D. Stormo. Expectation Maximization algorithm for identifying protein-binding sites with variable lengths from unaligned DNA fragments. *J. Mol. Biol.*, 223:139–170, 1992.

[40] K.-M. Chao. Computing all suboptimal alignments in linear space. In M. Crochemore and D. Gusfield, editors, *Combinatorial Pattern Matching*, volume 807 of *Lecture Notes in Computer Science*, pages 31–42. Springer-Verlag, 1994.

[41] B. Charlesworth, P. Sniegowski, and W. Stephan. The evolutionary dynamics of repetitive DNA in eukaryotes. *Nature*, 371:215–220, 1994.

[42] D. A. Christie. Sorting permutations by block-interchanges. *Information Processing Letters*, 60(4):165–169, 1996.

[43] D. A. Christie. *Genome rearrangement problems.* PhD thesis, University of Glasgow, 1998.

[44] J.-M. Claverie. Computational methods for the identification of genes in vertebrate genomic sequence. *Human Molecular Genetics*, 6:1735–1744, 1997.

[45] B. Combrugghe, S. Busby, and H. Buc. Cyclic AMP receptor protein: role in transcription activation. *Science*, 224:831–838, 1984.

[46] F. Corpet and B. Michot. RNAlign program: alignment of RNA sequences using both primary and secondary structures. *Comput. Appl. Biosci.*, 10:389–399, 1994.

[47] E. Coward. Shufflet: Shuffling sequences while conserving the k-let counts. *Bioinformatics*, 15:1058–1059, 1999.

[48] M. Crochemore and W. Rytter. *Text algorithms.* Oxford University Press, 1994.

[49] M. Crochemore and R. Vérin. Direct construction of compact Directed Acyclic Word Graphs. In A. Apostolico and J. Hein, editors, *Combinatorial Pattern Matching*, number 1264 in Lecture Notes Computer Science, pages 116–129. Springer-Verlag, 1997.

[50] A. Danchin. *La barque de Delphes.* Odile Jacob, 1998.

[51] B. DasGupta, T. Jiang, S. Kannan, M. Li, and E. Sweedyk. On the complexity and approximation of syntenic distance. In *1st ACM International Conference on Computational Molecular Biology, (RECOMB 97)*, pages 99–108, 1997.

[52] M.O. Dayhoff, R.M. Schwartz, and B.C. Orcutt. A model of evolutionary change in proteins. In *Atlas of Protein Sequence and Structure*, volume 5 suppl.3, pages 345–352. Natl. Biomed. Res. Found., 1978.

[53] B. Dujon. The yeast genome project: what did we learn? *Trends Genet.*, 12:263–270, 1996.

[54] R. Durbin, S. R. Eddy, A. Krogh, and G. Mitchison. *Biological Sequence Analysis. Probabilistic Models of Proteins and Nucleic Acids.* Cambridge University Press, 1998.

[55] L. Duret. personal communication.

[56] W. P. Elias and J. R. Czeczulin *et al.* Organization of biogenesis genes for aggregative adherence fimbria II defines a virulence gene cluster in enteroaggregative *Escherichia coli. J. Bacteriology*, 181:1779–1785, 1999.

[57] D. Eppstein, Z. Galil, R. Giancarlo, and G. F. Italiano. Sparse dynamic programming. I: linear cost functions; II: convex and concave cost functions. *J. Assoc. Comput. Mach.*, 39:519–567, 1992.

[58] V. Escalier. *Algorithmes pour la comparaison de structures moléculaires tridimensionnelles.* PhD thesis, 1997. Thèse de doctorat - Université de Paris VII.

[59] J. Allali *et al.* personal communication.

[60] P. A. Evans. Finding common subsequences with arcs and pseudoknots. In M. Crochemore and M. Paterson, editors, *Combinatorial Pattern Matching*, volume 1645 of *Lecture Notes in Computer Science*, pages 270–280. 1999.

[61] M. Farach. Optimal suffix tree construction with large alphabets. In *38th Annual IEEE Symposium on Foundation of Computer Science (FOCS 97)*, pages 581–592, 1995.

[62] M. Farach, T. Przytycka, and M. Thorup. On the agreement of many trees. *Information Processing Letters*, 55:297–301, 1995.

[63] M. Farach and M. Thorup. Fast comparison of evolutionary trees. In *Proceedings of the 5th ACM-SIAM Symposium on Discrete Algorithms (SODA 1994)*, 1994.

[64] J. S. Farris, M. Kallersjo, A. G. Kluge, and C. Bult. Testing significance of incongruence. *Cladistics*, 10:315–319, 1995.

[65] P. Fay. Oxygen relations of nitrogen fixation in Cyanobacteria. *Microbiol. Rev.*, 56:340–373, 1992.

[66] J. Felsenstein. Phylip – phylogeny inference package (version 3.2). *Cladistics*, 5:164–166, 1989.

[67] V. Ferretti, J. H. Nadeau, and D. Sankoff. Original synteny. In D. S. Hirschberg and E. W. Myers, editors, *Combinatorial Pattern Matching*, volume 1075 of *Lecture Notes in Computer Science*, pages 159–167. Springer-Verlag, 1996.

[68] J. W. Fickett. Recognition of protein coding regions in DNA sequences. *Nucleic Acids Res.*, 10:5503–5518, 1982.

[69] J. W. Fickett. Finding genes by computer: the state of the art. *Trends in Genetics*, 12:316–320, 1996.

[70] L. Florea, G. Hartzell, Z. Zhang Z, G. M. Rubin, and W. M. Miller. A computer program for aligning a cDNA sequence with a genomic DNA sequence. *Genome Res.*, 8:967–974, 1998.

[71] K. Frech, G. Herrmann, and T. Werner. Computer-assisted prediction, classification, and delimitation of protein binding sites in nucleic acids. *Nucleic Acids Res.*, 21:1655–1664, 1993.

[72] K. Frech, K. Quandt, and T. Werner. Software for the analysis of dna sequence elements of transcription. *Comput. Appl. Biosci.*, 13:89–97, 1993.

[73] B. J. Froehlich and A. Karakashian *et al.* Genes for CS2 pili of enterotoxigenic *Escherichia coli* and their interchangeability with those for CS1 pili. *Infect. Immun.*, 63:4849–4856, 1995.

[74] A. E. Gabrielian, D. Landsman, and A. Bolshoy. Curved DNA in promoter sequences. *In Silico Biology*, 1:http://www.bioinfo.de/isb/1999/01/0017, 1999.

[75] M. I. Garcia and A. Labigne *et al.* Nucleotide sequence of the afimbrial-adhesin-encoding *afa*-3 gene cluster and its translocation via flanking IS1 insertion sequences. *J. Bacteriology*, 176:7601–7613, 1994.

[76] D. Gautheret, F. Major, and R. J. Cedergren. Pattern searching/alignment with RNA primary and secondary structures: an effective descriptor for tRNA. *Comput. Appl. Biosci.*, 6:325–331, 1990.

[77] M. S. Gelfand. Computer prediction of the exon-intron structure of mammalian pre-mRNAs. *Nucleic Acids Res.*, 18:5865–5869, 1990.

[78] M. S. Gelfand, A. Mironov, and P. A. Pevzner. Spliced alignment problem: A new approach to gene recognition. In D. S. Hirschberg and E. W. Myers, editors, *Combinatorial Pattern Matching*, volume 1075 of *Lecture Notes in Computer Science*, pages 141–158. Springer-Verlag, 1996.

[79] M. S. Gelfand, A. A. Mironov, and P. A. Pevzner. Gene recognition via spliced sequence alignment. *Proc. Natl. Acad. Sci. (USA)*, 93:9061–9066, 1996.

[80] A. J. Gibbs and P. K. Keese. In search of origins of viral genes. In C. H. Callisher A. J. Gibbs and F. Garcia-Arenal, editors, *Molecular Basis of Virus Evolution*. Cambridge University Press, 1995.

[81] R. Giegerich, S. Kurtz, and J. Stoye. Efficient implementation of lazy suffix trees. In *Proceedings of the Third Workshop on Algorithmic Engineering (WAE99)*, Lecture Notes in Computer Science, pages 30–42. Springer and Verlag, 1999.

[82] E. Gilson, S. Bachellier, S. Perrin, D. Perrin, P. A. Grimont, F. Grimont, and M. Hofnung. The BIME family of bacterial highly repetitive sequences. *Res. Microbiol.*, 1142:217–222, 1991.

[83] E. Gilson, W. Saurin, D. Perrin, S. Bachellier, and M. Hofnung. Palindromic units are part of a new bacterial interspersed mosaic element (BIME). *Nucleic Acids Res.*, 19:1375–1383, 1991.

[84] L. Goldstein and M. S. Waterman. Approximations to profile score distributions. *J. Comp. Biol.*, 1:93–104, 1994.

[85] J. Gorodkin, L. J. Heyer, and G. D. Stormo. Finding the most significant common sequence and structure motifs in a set of RNA sequences. *Nucleics Acids Research*, 25:3724–3732, 1997.

[86] O. Gotoh. An improved algorithm for matching biological sequences. *J. Mol. Biol.*, 162:705–708, 1982.

[87] N. C. Grassly and E. C. Holmes. Detecting and characterizing gene conversions between multigene family members. *Mol. Biol. Evol.*, 16:1369–1390, 1999.

[88] M. Gribskov, R. Luthy, and D. Eisenberg. Profile analysis. *Meth. Enzymol.*, 183:146–159, 1990.

[89] M. Gribskov, M. McLachlan, and D. Eisenberg. Profile analysis: detection of distantly related proteins. *Proc. Natl. Acad. Sci. USA*, 84:4355–4358, 1987.

[90] C. A. Gross, M. Lonetto, and R. Losick. Bacterial sigma factors. In S. L. Knight and K. R. Yamamoto, editors, *Transcriptional Regulation*, volume 1, pages 129–176. Cold Spring Harbor Laboratory Press, 1992.

[91] X. Guan and E. C. Uberbacher. Alignments of DNA and protein sequences containing frameshift errors. *Comput. Appl. Biosci.*, 12(1):31–40, 1996.

[92] R. Guigó. Assembling genes from predicted exons in linear time with dynamic programming. *J. Comput. Biol.*, 5(4):681–702, 1998.

[93] D. Gusfield. *Algorithms on Strings, Trees, and Sequences: Computer Science and Computational Biology.* Cambridge University Press, 1997.

[94] D. Gusfield and J. Stoye. Linear time algorithms for finding and representing all the tandem repeats in a string. Report CSE-98-4, Department of Computer Science, University of California, Davis, 1998.

[95] S. Hannenhalli. Polynomial-time algorithm for computing translocation distance between genomes. *Discrete Applied Mathematics*, 71:137–151, 1996.

[96] S. Hannenhalli and P.Pevzner. Transforming cabbage into turnip (polynomial algorithm for sorting signed permutations by reversals). In *27th ACM Symposium on Theory of Computing (STOC 95)*, pages 178–189, 1995.

[97] S. Hannenhalli and P.Pevzner. Transforming man into mice (polynomial algorithm for genomic distance problem). In *36th Annual IEEE Symposium on Foundation of Computer Science (FOCS 95)*, pages 581–592, 1995.

[98] J. Hein. Reconstructing the evolution of sequences subject to recombination using parsimony. *Mathematical Biosciences*, 98:185–200, 1990.

[99] J. Hein. A heuristic method to reconstruct the history of sequences subject to recombination. *J. of Mol. Evol.*, 36:396–405, 1993.

[100] J. Hein. An algorithm combining DNA and protein alignment. *J. Theor. Biol.*, 167:169–174, 1994.

[101] J. Hein, T. Jiang, L. Wang, and K. Zhang. On the complexity of comparing evolutionary trees. *Discrete Applied Mathematics*, 71:153–169, 1996.

[102] J. D. Helmann. Compilation and analysis of *Bacillus subtilis* α-dependent promoter sequences: evidence for extended contact between RNA polymerase and upstream promoter DNA. *Nucleic Acids Res.*, 23:2351–2360, 1995.

[103] S. Henikoff and J. G. Henikoff. Amino acid substitution matrices for protein blocks. *Proc. Natl. Acad. Sci. USA*, 89:10915–10919, 1992.

[104] G. Z. Hertz and G. D. Stormo. *Escherichia coli* promoter sequences: Analysis and prediction. *Meth. Enzymol.*, 273:30–42, 1996.

[105] D. S. Hirschberg. A linear space algorithm for computing maximal common subsequences. *ACM*, 18:341–343, 1975.

[106] E. C. Holmes, M. Worobey, and A. Rambaut. Phylogenetic evidence for recombination in dengue virus. *Mol. Biol. Evol.*, 16:405–409, 1999.

[107] J. B. Howard and D. C. Rees. Nitrogenase: A nucleotide-dependent molecular switch. *Ann. Rev. Biochem.*, 94:235–264, 1994.

[108] Y. Hua, T. Jiang, and B. Wu. Aligning DNA sequences to minimize change in protein. In M. Farach-Colton, editor, *Combinatorial Pattern Matching*, volume 1448 of *Lecture Notes in Computer Science*, pages 221–234. Springer-Verlag, 1998.

[109] X. Huang and W. Miller. A time efficient, linear-space local similarity algorithm. *Advances in Applied Mathematics*, 12:337–357, 1991.

[110] X. Q. Huang, , R. C. Hardison, and W. Miller. A space-efficient algorithm for local similarities. *Comput. Appl. Biosci.*, 6:373–381, 1990.

[111] D. Husmeier and F. Wright. Detection of recombination in DNA multiple alignments with bayesian hidden Markov models and variational free energy einimization. http://www.bioss.sari.ac.uk/ dirk/My_publications.html.

[112] G. B. Hutchinson and M. R. Hayden. The prediction of exons through an analysis of spliceable open reading frames. *Nucleic Acids Res.*, 20:3453–3462, 1992.

[113] L. J. Maher III. personal communication.

[114] I. Jakobsen and S. Easteal. A program for calculating and displaying compatibility matrices as an aid in determining recticulate evolution in molecular sequences. *Compt. Appli. Biosci.*, 12:291–295, 1996.

[115] P. Jean, J. Pothier, P. Dansette, D. Mansuy, and A. Viari. Automated multiple analysis of protein structures: application to homology modeling of cytochromes P450. *Proteins: struct., funct., and genetics*, 28:388–404, 1997.

[116] T. Jiang, G. H. Lin, B. Ma, and K. Zhang. The longest common subsequence problem for arc-annotated sequences. In R. Giancarlo and D. Sankoff, editors, *Combinatorial Pattern Matching*, volume 1848 of *Lecture Notes in Computer Science*. Springer-Verlag, 2000.

[117] I. Jonassen. Efficient discovery of conserved patterns using a pattern graph. *Comput. Appl. Biosci.*, 13:509–522, 1997.

[118] I. Jonassen, J. F. Collins, and D. G. Higgins. Finding flexible patterns in unaligned protein sequences. *Protein Science*, 4:1587–1595, 1995.

[119] I. Jonassen, I. Eidhammer, and W. R. Taylor. Discovery of local packing motifs in protein structures. *Proteins: struct., funct., and genetics*, 34:206–219, 1999.

[120] D. Kandel, Y. Matias, R. Unger, and P. Winkler. Shuffling biological sequences. *Discrete Applied Mathematics*, 71:171–185, 1996.

[121] S. Karlin, F. Ost, and B. E. Blaisdell. Patterns in DNA and amino acid sequences and their statistical significance. In M. S. Waterman, editor, *Mathematical Methods for DNA Sequences*, pages 133–158. CRC Press, 1989.

[122] R. M. Karp, R. E. Miller, and A. L. Rosenberg. Rapid identification of repeated patterns in strings, trees and arrays. In *Proc. 4th Annu. ACM Symp. Theory of Computing*, pages 125–136, 1972.

[123] K. Karplus. Evaluating regularizers for estimating distributions of amino acids. In C. Rawlings, D. Clark, R. Altman, L. Hunter, T. Lengauer, and S. Wodak, editors, *Third International Symposium on Intelligent Systems for Molecular Biology*, pages 188–196, Cambridge, England, 1995.

[124] K. Kean. personal communication.

[125] S. Kurtz. Reducing the space requirement of suffix trees. *Software – Practice and Experience*, 29:1149–1171, 1999.

[126] G. Landau and J. Schmidt. An algorithm for approximate tandem repeats. In Z. Galil A. Apostolico, M. Crochemore and U. Manber, editors, *Combinatorial Pattern Matching*, volume 684 of *Lecture Notes in Computer Science*, pages 120–133. Springer-Verlag, 1993.

[127] C. E. Lawrence, S. F. Altschul, M. S. Boguski, J. S. Liu, A. F. Neuwald, and J. C. Wooton. Detecting subtle sequence signals: a Gibbs sampling strategy for multiple alignment. *Science*, 262:208–214, 1993.

[128] C. E. Lawrence and A. A. Reilly. An expectation maximization (EM) algorithm for the identification and characterization of common sites in unaligned biopolymer sequences. *Proteins: struct., funct., and genetics*, 7:41–51, 1990.

[129] G. Lecointre, L. Rachdi, P. Darlu, and E. Denamur. *Escherichia coli* molecular phylogeny using the incongruence length difference test. *Mol. Biol. Evol.*, 15:1685–1695, 1998.

[130] V. I. Levenshtein. Binary codes capable of correcting deletions, insertions, and reversals. *Cyber. Contr. Theory*, 10:707–710, 1966.

[131] M. Li, J. Tromp, and L. Zhang. On the nearest neighbour interchange distance between evolutionary trees. *Journal on Theoretical Biology*, 182(4):463–467, 1996.

[132] D. Liben-Nowell. On the structure of syntenic distance. In M. Crochemore and M. Paterson, editors, *Combinatorial Pattern Matching*, volume 1645 of *Lecture Notes in Computer Science*, pages 50–65. 1999.

[133] D. Liben-Nowell and J. Kleinberg. Structural properties and tractability results for linear synteny. In R. Giancarlo and D. Sankoff, editors, *Combinatorial Pattern Matching*, volume 1848 of *Lecture Notes in Computer Science*, pages 248–263. 2000.

[134] B. Ma, M. Li, and L. Zhang. From gene trees to species trees. In S. Istrail, P. Pevzner, and M. Waterman, editors, *RECOMB'98. Proceedings of Second Annual International Conference on Computational Molecular Biology.* ACM Press, 1998.

[135] F. Major. personal communication.

[136] L. Marsan and M.-F. Sagot. Algorithms for extracting structured motifs using a suffix tree with an application to promoter and regulatory site consensus identification. *J. Comp. Biol.*, 2000. in press.

[137] L. Marsan and M.-F. Sagot. Extracting structured motifs using a suffix tree – algorithms and application to promoter consensus identification. In R. Shamir, S. Miyano, S. Istrail, P. Pevzner, and M. Waterman, editors, *RECOMB'00. Proceedings of Fourth Annual International Conference on Computational Molecular Biology*, pages 210–219. ACM Press, 2000.

[138] J. Maynard-Smith. The detection and measurement of recombination from sequence data. *Genetics*, 153:1021–1027, 1999.

[139] M. Muller. personal communication.

[140] J. I. Munro, V. Raman, and S. S. Rao. Space efficient suffix trees. In V. Arvind and R. Ramanujam, editors, *Proceedings of the Foundations of Software Technology and Theoretical Computer Science Symposium*, volume 1530 of *Lecture Notes in Computer Science*, pages 1–12. Springer-Verlag, 1998.

[141] E. W. Myers and W. Miller. Optimal alignments in linear space. *Comput. Appl. Biosci.*, 4:11–17, 1988.

[142] E. W. Myers and W. Miller. Approximate matching of regular expressions. *Bull. Math. Biol.*, 51:5–37, 1989.

[143] D. Naor and D. Brutlag. On suboptimal alignments of biological sequences. In A. Apostolico, M. Crochemore, Z. Galil, and U. Manber, editors, *Combinatorial Pattern Matching*, volume 684 of *Lecture Notes in Computer Science*, pages 179–196. Springer-Verlag, 1993.

[144] K. E. Nelson, R. A. Clayton, S. R. Gill, M. L. Gwinn, R. J. Dodson, D. H. Haft, E. K. Hickey, J. D. Peterson, W. C. Nelson, K. A. Ketchum, L. McDonald, T. R. Utterback, J. A. Malek, K. D. Linher, M. M. Garrett, A. M. Stewart, M. D. Cotton, M. S. Pratt, C. A. Phillips, D. Richardson, J. Heidelberg, G. G. Sutton, R. D. Fleischmann, J. A. Eisen, , and C. M. *et al.* Evidence for lateral gene transfer between archaea and bacteria from genome sequence of *Thermotoga maritima*. *Nature*, 399:323–329, 1999.

[145] O. N. Ozoline, A. A. Deev, and M. V. Arkhipova. Non-canonical sequence elements in the promoter structure. cluster analysis of promoters recognized by *Escherichia coli* RNA polymerase. *Nucleic Acids Res.*, 25:4703–4709, 1998.

[146] R. D. M. Page and M. A. Charleston. From gene to organismal phylogeny: Reconciled trees and the gene tree/species tree problem. *Mol. Phyl. Evol.*, 7:231–240, 1997.

[147] L. Parida, I. Rigoutsos, A. Floratos, D. Platt, and Y. Gao. Pattern discovery on character sets and real-valued data: linear bound on irredundant motifs and polynomial time algorithms. In *Proceedings of the eleventh ACM-SIAM Symposium on Discrete Algorithms (SODA 2000)*, pages 297–308. ACM Press, 2000.

[148] N. Pavy, S. Rombauts, P. Déhais, C. Mathé, D. V. V. Ramana, P. Leroy, and P. Rouzé. Evaluation of gene prediction software using a genomic data set: Application to *Arabidopsis thaliana* sequences. *BioInformatics*, 1999. in press.

[149] C. N. S. Pedersen. personal communication.

[150] C. N. S. Pedersen, R. Lyngso, and J. Hein. Comparison of coding DNA. In M. Farach-Colton, editor, *Combinatorial Pattern Matching*, volume 1448 of *Lecture Notes in Computer Science*, pages 153–173. Springer-Verlag, 1998.

[151] N. Pisanti and M.-F. Sagot. Further thoughts on the synteny distance between genomes. 2000. submitted.

[152] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery. *Numerical Recipes in C : The Art of Scientific Computing.* Cambridge Univ. Press, 1993.

[153] A. Rambaut and N. C. Grassly. Seq-gen: An application for the Monte Carlo simulation of DNA sequence evolution along phylogenetic trees. *Comput. Appl. Biosci.*, 13:235–238, 1996.

[154] M. T. Record, W. S. Reznikoff, M. L. Craig, K. L. McQuade, and P. J. Schlax. *Escherichia coli* RNA polymerase σ^{70} promoters, and the kinetics of the steps of transcription initiation. In F. C. Neidhardt, editor, *Escherichia coli and Salmonella*, volume 1, pages 792–820. ASM Press, 1996.

[155] M. Régnier. A unified approach to word statistics. In S. Istrail, P. Pevzner, and M. Waterman, editors, *RECOMB'98. Proceedings of Second Annual International Conference on Computational Molecular Biology*, pages 207–213. ACM Press, 1998.

[156] M. Régnier, Alexandre Lifanov, and V. Makeev. Three variations on word counting. manuscript.

[157] M. Régnier and W. Szpankowski. On the approximate pattern occurrences in a text. 1997. manuscript.

[158] G. Reinert and S. Schbath. Compound poisson and poisson process approximations for occurrences of multiple words in markov chains. *J. Comp. Biol.*, 5:223–253, 1998.

[159] J.-L. Risler, M.-O. Delorme, H. Delacroix, and A. Henaut. Amino acid substitutions in structurally related proteins. A pattern recognition approach. *J. Mol. Biol.*, 204:1019–1029, 1988.

[160] R. L. Robson and J. R. Postgate. Oxygen and hydrogen in biological nitrogen fixation. *Ann. Rev. Microbiol.*, 34:183–207, 1980.

[161] E. Rocke. Using suffix trees for gapped motif discovery. In R. Giancarlo and D. Sankoff, editors, *Combinatorial Pattern Matching*, volume 1848 of *Lecture Notes in Computer Science*. Springer-Verlag, 2000.

[162] E. Rocke and M. Tompa. An algorithm for finding novel gapped motifs in DNA sequences. In S. Istrail, P. Pevzner, and M. Waterman, editors, *RECOMB'98. Proceedings of Second Annual International Conference on Computational Molecular Biology*, pages 228–233. ACM Press, 1998.

[163] F. P. Roth, J. D. Hughes, P. W. Estep, and G. M. Church. Finding DNA regulatory motifs within unaligned noncoding sequences clustered by whole-genome mRNA quantitation. *Nat. Biotechnol.*, 16:939–945, 1998.

[164] M.-F. Sagot. *Analyse des structures primaire et tertiaire des protéines : localisation des mots répétés et recherche rapide sur banque.* 1993. Mémoire de DEA d'Informatique Fondamentale - Université de Paris VII, France.

[165] M.-F. Sagot. *Ressemblance lexicale et structurale entre macromolécules - Formalisation et approches combinatoires.* 1996. Thèse de doctorat - Université de Marne-la-Vallée, Noisy le Grand, France.

[166] M.-F. Sagot. Spelling approximate repeated or common motifs using a suffix tree. In C. L. Lucchesi and A. V. Moura, editors, *LATIN'98: Theoretical Informatics*, Lecture Notes in Computer Science, pages 111–127. Springer-Verlag, 1998.

[167] M.-F. Sagot, V. Escalier, A. Viari, and H. Soldano. Searching for repeated words in a text allowing for mismatches and gaps. In R. Baeza-Yates and U. Manber, editors, *Second South American Workshop on String Processing*, pages 87–100, Viñas del Mar, Chile, 1995. University of Chile.

[168] M.-F. Sagot and E. W. Myers. Identifying satellites and periodic repetitions in biological sequences. *J. Comp. Biol.*, 10:10–20, 1998. extended version of paper presented at *RECOMB'98*.

[169] M.-F. Sagot and E. W. Myers. Identifying satellites in nucleic acid sequences. In S. Istrail, P. Pevzner, and M. Waterman, editors, *RECOMB'98. Proceedings of Second Annual International Conference on Computational Molecular Biology*, pages 234–242. ACM Press, 1998.

[170] M.-F. Sagot and A. Viari. Application of a combinatorial sampling approach to identifying sequence motifs that are structurally and/or functionally related. in preparation.

[171] M.-F. Sagot and A. Viari. Combinatorial sampling. in preparation.

[172] M.-F. Sagot and A. Viari. A double combinatorial approach to discovering patterns in biological sequences. In D. Hirschberg and G. Myers, editors, *Combinatorial Pattern Matching*, volume 1075 of *Lecture Notes in Computer Science*, pages 186–208. Springer-Verlag, 1996.

[173] M.-F. Sagot and A. Viari. Flexible identification of structural objects in nucleic acid sequences: palindromes, mirror repeats, pseudoknots and triple helices. In J. Hein and A. Apostolico, editors, *Combinatorial Pattern Matching*, volume 1264 of *Lecture Notes in Computer Science*, pages 224–246. Springer-Verlag, 1997.

[174] M.-F. Sagot, A. Viari, J. Pothier, and H. Soldano. Finding flexible patterns in a text - an application to 3D molecular matching. *Comput. Appl. Biosci.*, 11:59–70, 1995. presented at First International IEEE Workshop on Shape and Pattern Matching in Computational Biology, Seattle, Washington, USA.

[175] M.-F. Sagot, A. Viari, and H. Soldano. A distance-based block searching algorithm. In C. Rawlings, D. Clark, R. Altman, L. Hunter, T. Lengauer, and S. Wodak, editors, *Third International Symposium on Intelligent Systems for Molecular Biology*, pages 322–331, Cambridge, England, 1995. AAAI Press.

[176] M.-F. Sagot, A. Viari, and H. Soldano. Multiple comparison: a peptide matching approach. *Theoret. Comput. Sci.*, 180:115–137, 1997. presented at *Combinatorial Pattern Matching 1995*.

[177] N. Saitou and M. Nei. The neighbor-joining method: a new method for reconstructing phylogenetic trees. *Mol. Biol. Evol.*, 4:406–425, 1987.

[178] D. Sankoff. Simultaneous solution of the RNA folding, alignment and protosequence problems. *SIAM J. Appl. Math.*, 45:810–825, 1985.

[179] D. Sankoff and M. Blanchette. Multiple genome rearrangement and breakpoint phylogeny. *J. Comp. Biol.*, 5:555–570, 1998.

[180] D. Sankoff and J. B. Kruskal. *Time Warps, String Edits, and Macromolecules: the Theory and Practice of Sequence Comparison.* Addison-Wesley, 1983.

[181] T. Schiex. personal communication.

[182] T. D. Schneider, G. D. Stormo, L. Gold, and A. Ehrenfeucht. Information content of binding sites on nucleotide sequences. *J. Mol. Biol.*, 188:415–431, 1986.

[183] B. Shapiro. An algorithm for comparing multiple RNA secondary structures. *Comput. Appl. Biosci.*, 4:387–393, 1988.

[184] B. Shapiro and K. Zhang. Comparing multiple RNA secondary structures using tree comparisons. *Comput. Appl. Biosci.*, 6:309–318, 1990.

[185] J. C. W. Shepherd. Method to determine the reading frame of a protein from the purine/pyrimidine genome sequence and its possible evolutionary justification. *Proc. Natl. Acad. Sci. USA*, 78:1576–1600, 1981.

[186] J. L. Slightom, A. E. Blechl, and O. Smithies. Human fetal $^{G}\gamma$- and $^{A}\gamma$- globin genes complete nucleotide sequences suggest that dna can be exchanged between these duplicated genes. *Cell*, 21:627–638, 1980.

[187] H. Soldano, A. Viari, and M. Champesme. Searching for flexible repeated patterns using a non transitive similarity relation. *Pattern Recognition Letters*, 16:233–246, 1995.

[188] R. Staden and A. D. McLachlan. Codon preference and its use in identifying protein coding regions in long DNA sequences. *Nucleic Acids Res.*, 10:5503–5518, 1982.

[189] B. Stentebjerg-Olesen and T. Chakraborty *et al.* Type 1 fimbriation and phase switching in a natural *Escherichia coli fimB* null strain, nissle 1917. *J. Bacteriology*, 181:7470–7478, 1999.

[190] J. Stoye and D. Gusfield. Simple and flexible detection of contiguous repeats using a suffix tree. In M. Farach-Colton, editor, *Combinatorial Pattern Matching*, volume 1448 of *Lecture Notes in Computer Science*, pages 140–152. Springer-Verlag, 1998.

[191] W. R. Taylor. The classification of amino acid conservation. *J. Theor. Biol.*, 119:205–218, 1986.

[192] N. Terryn, L. Heijnen, A. De Keyser, M. Van Asseldonck, R. De Clercq, H. Verbakel, J. Gielen, M. Zabeau, R. Villarroel, T. Jesse, P. Neyt, R. Hogers, H. Van Den Daele, W. Ardiles, C. Schueller, K. Mayer, P. Déhais, S. Rombauts, M. Van Montagu, P. Rouzé, and P. Vos. Evidence for an ancient chromosomal duplication in *Arabidopsis thaliana* by sequencing and analyzing a 400-kb contig at the APETALA2 locus on chromosome 4. *FEBS Letters*, 445:237–245, 1999.

[193] A. Thomas and M. H. Skolnick. A probabilistic model for detecting coding regions in DNA sequences. *IMA J. Math. Appl. Med. Biol.*, 11:149–160, 1994.

[194] G. Thyjs. personal communication.

[195] M. Tompa. An exact method for finding short motifs in sequences, with application to the ribosome binding site problem. In *Seventh International Symposium on Intelligent Systems for Molecular Biology*, pages 262–271, Heidelberg, Germany, 1999. AAAI Press.

[196] A. Vanet, L. Marsan, A. Labigne, and M.-F. Sagot. Inferring regulatory elements from a whole genome. An analysis of the σ^{80} family of promoter signals. *J. Mol. Biol.*, 297:335–353, 2000.

[197] A. Vanet, L. Marsan, and M.-F. Sagot. Promoter sequences and algorithmical methods for identifying them. *Research in Microbiology*, 150:779–799, 1999.

[198] A. Viari. personal communication.

[199] Y. Wakabayashi and M.-F. Sagot. Pattern inference under many guises. 2001. submitted to Special Volume of Lecture Notes in Computer Science (Springer and Verlag) for Summer School on Algorithms and Combinatorics, Brazilian Symposium on Graphs, Algorithms and Combinatorics, 12th-24th March 2001.

[200] M. S. Waterman. *Introduction to Computational Biology. Maps, Sequences and Genomes.* Chapman and Hall, 1995.

[201] T. Werner. Models for prediction and recognition of eukaryotic promoters. *Mamm. Genome*, 10:168–175, 1999.

[202] C. Wiuf, T. Christensen, and J. Hein. Detection of recombinations in the history of a sample of sequences. *Mol. Biol. Evol.* submitted.

[203] Y. Xu, J. R. Einstein, R. J. Mural, M. Shah, and E. C. Uberbacher. An improved system for exon recognition and gene modeling in human DNA sequences. In *Second International Conference on Intelligent Systems for Molecular Biology*, pages 376–384. AAAAI Press, 1994.

[204] M. Zelwer. *Détection de recombinaisons et de transferts horizontaux.* 2000. Mémoire de DEA de Biomathématiques - Universités de Paris VI et Paris VII, France.

[205] K. Zhang, L. Wang, and B. Ma. Computing similarity between RNA structures. In M. Crochemore and M. Paterson, editors, *Combinatorial Pattern Matching*, volume 1645 of *Lecture Notes in Computer Science*, pages 258–269. 1999.

[206] M. Zuker. Calculating nucleic acid secondary structure. *Curr. Opin. Struct. Biol.*, 10:303–310, 2000.

[207] M. Zuker and D. Sankoff. RNA secondary structures and their prediction. *Bull. Math. Biol.*, 46:591–621, 1984.

[208] M. Zuker and P. Stiegler. Optimal computer folding of large RNA sequences using thermodynamics and auxiliary information. *Nucleic Acids Res.*, 9:133–148, 1981.

Index