

META-ALIGNMENT OF BIOLOGICAL SEQUENCES

Enrique Blanco García

PhD Thesis

Barcelona, November 2006

META-ALIGNMENT OF BIOLOGICAL SEQUENCES

Enrique Blanco García

PhD Thesis

Barcelona, November 2006

CopyLeft 2006 by Enrique Blanco García.

First Edition, May 2006.

Printed at:

COPISTERIA MIRACLE

Rector Ubach, 6–10 (Aribau corner)

08021 — Barcelona

Phone: +034 93 200 85 44

Fax: +034 93 209 17 82

Email: miracle at miraclepro.com



META-ALIGNMENT OF BIOLOGICAL SEQUENCES

Enrique Blanco García

Memòria presentada per optar al grau de Doctor en Informàtica
per la Universitat Politècnica de Catalunya (UPC)

Aquesta Tesi Doctoral ha estat realitzada sota la direcció del
Dr. **Xavier Messeguer Peypoch**[†] i el Dr. **Roderic Guigó i Serra**[‡]

[†] Departament de Llenguatges i Sistemes Informàtics,
Universitat Politècnica de Catalunya (UPC)

[‡] Centre de Regulació Genòmica (CRG) /
Universitat Pompeu Fabra (UPF)

PhD dissertation in the area of Computer Science,
Technical University of Catalonia (UPC)

PhD advisors:

Dr. **Xavier Messeguer Peypoch**[†] and Dr. **Roderic Guigó i Serra**[‡]

[†] Software Department, Technical University of Catalonia (UPC)

[‡] Centre for Genomic Regulation (CRG) /
Universitat Pompeu Fabra (UPF)

Barcelona, November 2006

"I have a dream that one day this nation will rise up and live out the true meaning of its creed: We hold these truths to be self-evident, that all men are created equal.

I have a dream that one day on the red hills of Georgia, the sons of former slaves and the sons of former slave owners will be able to sit down together at the table of brotherhood.

I have a dream that one day even the state of Mississippi, a state sweltering with the heat of injustice, sweltering with the heat of oppression, will be transformed into an oasis of freedom and justice.

I have a dream that my four little children will one day live in a nation where they will not be judged by the color of their skin but by the content of their character.

I have a dream today!

I have a dream that one day, down in Alabama, with its vicious racists, with its governor having his lips dripping with the words of interposition and nullification – one day right there in Alabama little black boys and black girls will be able to join hands with little white boys and white girls as sisters and brothers.

I have a dream today!

I have a dream that one day every valley shall be exalted, and every hill and mountain shall be made low, the rough places will be made plain, and the crooked places will be made straight; and the glory of the Lord shall be revealed and all flesh shall see it together."

MARTIN LUTHER KING, JR.

28 AUGUST 1963, AT THE LINCOLN MEMORIAL, WASHINGTON D.C. (USA)

Preface

AS A FAMOUS DIRTY DETECTIVE ONCE SAID, there must be a hundred good reasons why I shouldn't have just initiated a PhD thesis. But right now, I can't think of a single one. On the contrary, I wonder who would have rejected the appealing proposal to investigate the genomic world, which is actually the center of the life, designing programs on a high-performance computational environment.

The construction of the first modern computers was one of the major landmarks achieved by the human being in the past century. Since then, the application of computers on many intriguing problems and the constant evolution of the programs that govern them have permitted the researchers in many areas to discover new concepts that would have been otherwise unreachable for our generation without this technology.

Molecular biology is not an exception. The sequencing of the human genome would be still an impossible challenge if many automatic procedures that are now familiar to us would have not been developed before. In this context, Bioinformatics has been the relevant driving force responsible for stimulating the advance in the study of the biology of our cells. Particularly, many clues to understand the life in our planet can be found in the regulation of gene expression. Nonetheless, to be sincere I have to admit that we are still completely ignorant: a huge amount of new biological information is constantly released so that the global picture that we want to reconstruct becomes today somehow even more complex than the day before.

Understanding life is an enormous challenge. In other scale, a PhD is also an exciting challenge for a student. It is a period in which not only such a person acquires a valuable education in many aspects of his life. At the same time, this individual is supposed to be capable of applying such knowledge in the investigation of a real problem, sometimes in competition with other people that have much more experience. In my case, the task became even more complex as a computer scientist needs a solid biological background to approach this kind of problems.

This thesis not only pretends to communicate the different phases of my work during the PhD period of research. Before starting to write, it was also my commitment to elaborate a manuscript fulfilling the highest requirements of quality and accuracy in the the material that is presented. This manuscript attempts to follow a logical and continuous argument from the introductory parts to the specific chapters devoted to the presentation of the results of the thesis. In addition, a DVD with supplementary materials such as the electronic

thesis, the bibliography, the software or several educational resources, is also released as an excellent complement to the thesis.

The experience and the abilities I have personally acquired during this period do not fit in just two hundred pages. From my point of view, the most relevant result of a thesis is not the compilation of scientific papers published during that time (these should be seen as a relevant consequence of a good work). On the contrary, I am totally convinced that the essential result of a PhD thesis is the improvement of the individual that positively changes his life in many aspects, producing an amazing enrichment of his personality.

In our childhood, many of us have got an intimate and naive desire of changing the world to improve it. Surprisingly after so many years, I still have this feeling although I am quite conscious that some things are not so easy to be changed whereas others simply can not be changed. However, I am happy to see that I have acquired a solid education that will be very useful to face more complicate situations throughout my life. In fact, this PhD thesis has not represented for me a central objective but an excellent opportunity to stop and learn, driving me to more ambitious challenges.

The education of our society has been always among my priorities. To be able to teach is necessary to learn to teach before. This is reflected in the fact that I have voluntarily performed hundreds of teaching activities during my thesis, always with a high degree of motivation in my presentations. Throughout our lives we do not cease to gain new knowledge. But we investigators have the duty of communicating rigorously our achievements with honesty to our people at schools, institutes, universities, meetings and mass media. To reach this ambitious objective is necessary to be engaged and involved in such a project. If we fail now in this attempt, I suspect that the gap between those that have the power to learn and investigate and those that do not, will be dangerously large, probably too much.



Barcelona, November 2006

Contents

Preface	vii
Contents	xi
List of Tables	xiii
List of Figures	xvii
Acknowledgments	xix
Abstract	xxi
Resumen	xxiii
Resum	xxv
I Preliminaries	1
1 Introduction	3
1.1 General objectives	4
1.2 Objectives	4
1.3 Thesis chronology	5
1.4 Outline of this thesis	7
1.5 Particular considerations	8
2 The post-genomic era	9
2.1 The genomic landscape	10
2.2 The genomic era	17
2.3 The post-genomic era	24

II	State of the Art	29
3	The golden age of sequence analysis	31
3.1	Foundations of sequence comparison	32
3.2	Alphabets, sequences and alignments	35
3.3	An anthology of algorithms for global alignments	40
3.4	A short overview on local sequence alignment	61
3.5	A short overview on multiple sequence alignment	69
3.6	Map alignments	72
4	Computational Gene and Promoter Characterization	87
4.1	Genes and promoters	88
4.2	Computational approaches	95
4.3	Detection of signals	96
4.4	Content recognition	101
4.5	Sequence comparison	103
4.6	The state of the art in gene identification	107
4.7	The state of the art in promoter characterization	111
4.8	Looking forward	113
III	Meta-Alignment of Sequences	123
5	Meta-alignment of Biological Sequences	125
5.1	Biological maps: promoters	126
5.2	Transcription Factor maps	128
5.3	TF-map pairwise alignment	128
5.4	TF-map alignment training	136
5.5	TF-map alignments in orthologous genes	144
5.6	TF-map alignments in co-regulated genes	148
5.7	TF-map alignments and matrix specificity	155
5.8	Local TF-map alignments	158
5.9	Discussion	162
6	Multiple Non-Collinear TF-map Alignment	171
6.1	The need for multiple TF-map alignment	172
6.2	Basic definitions	174
6.3	The algorithms	176
6.4	Non-collinear TF-map alignments	181
6.5	Biological results	184
7	Conclusions	197

IV Appendices	199
<i>Curriculum Vitae</i>	201
Software	209
List of Publications	211
Publications	215
Posters	229
Miscellanea	237
WebSite References	241
Index	245

List of Tables

2.1	Comparison of the sizes of several eukaryotic genomes	17
3.1	The IUPAC extended genetic alphabet	38
3.2	The amino acid alphabet	39
4.1	The common accuracy measures in sequence analysis	109
5.1	TF-map alignment accuracy results on the HR SET	140
5.2	BLASTN accuracy results on the HR SET	142
5.3	TF-map alignment results on several genomic samples.	146
5.4	Promoter identification with human-chicken TF-map alignments	149
5.5	Reconstruction of the TTR gene promoter	153
5.6	Q-value and PWM matrix specificity	158
5.7	Evolution of the matrix specificity	159
5.8	JASPAR and TRANSFAC specific subsets	160
6.1	Results when distinguishing promoters with MMAs	186

List of Figures

2.1	Electron micrograph of a chicken chondrocyte	11
2.2	The molecular processes involved in the protein synthesis pathway	13
2.3	The genetic code table	14
2.4	A comparison of chromatin with a mitotic chromosome	16
2.5	The organization of the human genome	18
2.6	Growth of the GENBANK (1982-2004)	19
2.7	An example of GENBANK entry	21
2.8	The human URO-D gene in the UCSC GENOME BROWSER and ENSEMBL	23
2.9	Using SNPs to locate susceptibility genes	25
3.1	Gene evolution events	36
3.2	The maximum-match operation for necessary pathways	41
3.3	The Needleman and Wunsch algorithm	44
3.4	The dynamic programming matrix	45
3.5	The Sellers algorithm	46
3.6	The Hirschberg linear space approach	47
3.7	An algorithm to compute $D(i, j)$ in $O(n)$ space cost	49
3.8	The Hirschberg linear space algorithm	50
3.9	The Needleman and Wunsch algorithm revisited	52
3.10	The generalized dynamic programming matrix	56
3.11	The Sellers algorithm generalized	57
3.12	The Gotoh algorithm	59
3.13	The Smith and Waterman algorithm	63
3.14	Identification of sequence similarities by FASTA	66
3.15	BLAST processing	68
3.16	Generalized MSA dynamic programming matrix	70
3.17	The basic CLUSTALW alignment procedure	71
3.18	DNA nucleotide sequences recognized by restriction nucleases	73
3.19	A restriction map alignment	75

3.20 The [Waterman et al.](#) map alignment algorithm 76

3.21 Mapping the D matrix over a grid 77

3.22 An illustration of a f-curve 78

3.23 An illustration of an i-profile 80

3.24 An illustration of a R-profile and a L-profile 81

3.25 The [Myers and Huang](#) map alignment algorithm 82

4.1 The typical gene structure 89

4.2 Other forms of gene structures 91

4.3 Transcription of two tandem genes 92

4.4 A schematic representation of a promoter 93

4.5 Nucleosomes and chromatin structure can influence gene expression 94

4.6 Sources of information in the ab-initio gene-finding process 95

4.7 Pattern-driven algorithms 97

4.8 Alignment and representation of a set of TFBSs 98

4.9 A Position Weight Matrix 99

4.10 Information content of TRANSFAC 6.3 matrices 100

4.11 An example of coding statistic 102

4.12 Comparative analysis of a gene 104

4.13 Phylogenetic footprinting 105

4.14 A microarray experiment 106

4.15 Sequence-driven algorithms 108

4.16 `geneid` dataflow 110

4.17 Transcriptional regulatory module architectures 112

5.1 The human genome map 127

5.2 TF-maps: construction and alignment 130

5.3 The Naive TF-map alignment algorithm 133

5.4 Sparse matrices 134

5.5 The Enhanced TF-map alignment algorithm 135

5.6 Number of accessions to the matrix S 136

5.7 Examples of the ABS data retrieval system 141

5.8 TF-map alignment of the human and mouse PLA1A gene 143

5.9 TF-map alignment on several genomic samples 145

5.10 TF-map alignment in promoter detection 147

5.11 Alignment experiment with the CISRED genes 150

5.12 Score distribution of the CISRED TF-map alignments 152

5.13 Experimental annotation of the TTR gene 154

5.14 Construction and use of a PWM 156

5.15 The Q-value distribution in JASPAR and TRANSFAC 157

5.16 Using local meta-alignment in pattern identification 161

5.17 Local meta-alignment using the distance metric 162

5.18 Gumbel distribution of local meta-alignments	163
6.1 TF-mapping in a simple example	173
6.2 TF-mapping of the human promoter NM_015900	174
6.3 Progressive multiple map alignment algorithm	177
6.4 MMA algorithm: data structures and matrix	179
6.5 Pairwise alignment of two clusters of TF-maps	180
6.6 Two examples of non-collinear MMAs	181
6.7 Diagonal filling of the alignment matrix	182
6.8 The non-collinearity parameter	184
6.9 Distinguishing promoters from other genomic regions	185
6.10 Multiple promoter characterization	188
6.11 MMA of the MMP13 promoter in 9 species	191
6.12 Using MEME as a mapping function	193

Acknowledgments

THIS SECTION IS USUALLY THE PART OF THE THESIS in which the authors mention those people that have decisively contributed to the presented work. As I am a generous person but my gratitude is not infinite, I want to express the following acknowledgments only to those that really deserve the reward of being cited here.

I am totally convinced about how to begin and to end this section. Honestly, there is only one person that deserves the honor of appearing in the first place of this section: myself. This thesis has not been an easy work at all. In our society, most computer scientists are working on the private sector, so that the orientation of their careers to investigation is a rare fact nowadays. And I have learned to live with this pressure as well. For a computer engineer like me, it has been a rich experience to work in a research environment devoted to the biological discovery. However, it has also been very demanding, because this thesis is not only about the development of new theoretical algorithms. It was also an exercise of application of such methods in real data to obtain novel biological conclusions. To sum up, it was like doing two thesis: one about computer science, and one about molecular biology. And I am very proud to have fulfilled both aspects of my work. Therefore, I want to thank myself for not abandoning, for supporting myself, for carrying on when the main objectives of the thesis seemed to be very far, when things were going too slow, or when the adaptation to the academic world was difficult because of its competitiveness.

I want to warmly thank you my two PhD advisors, Xavier Messeguer and Roderic Guigó for the correct direction of my work. We started in 1999 with the program `geneid` to successfully obtain my degree in computer science the summer of 2000. A few years later, I am very happy to see that the majority of people in my lab have used it in their investigations with a lot of success, and some of them have even been able to modify some of its modules without difficulty. Thanks then to Xavier, for your calm, for your wisdom, for your patience with me, specially when continuous communication was sometimes difficult because I was physically working outside the department, and for believing in my work in many occasions that I will never forget. Also thanks to Roderic, for the computational facilities, for the opportunity to work with so good people in the IMIM lab, for let me learning involuntarily from your experience, for the funding, for the international meetings that increased a lot my knowledge and for being always ambitious in whatever task you are doing.

Also thanks to my colleagues at work from which I learn a lot of useful things. Many of them have also decisively contributed to improve the quality of this thesis. Specially thanks

to Josep Francesc Abril that have assisted me in uncountable occasions with his priceless help. Thanks also to those that were in the lab when I arrive over there: Moisès Burset, Sergi Castellano and Genís Parra. To those that arrived later, many thanks as well: Robert Castelo, Jan-Jaap Wesselink, Mar Albà, Eduardo Eyras, Charles Chapple, Nicolás Bellora and Miguel Pignatelli. Further thanks to our system administrators, Alfons González, Xavier Fustero and Òscar González. I want to specially acknowledge Robert Castelo for the excellent template in \LaTeX from his PhD thesis. This template was later adapted by Sergi Castellano and Genís Parra, and substantially improved by Josep Francesc Abril, for their theses. This manuscript is an adaptation of such templates following my own style.

At this point I want to remember those teachers from many disciplines that have contributed positively to my education throughout my life. First of all, thanks to those in the teaching staff that positively contributed to my education at my school Hermanos Maristas de Les Corts. Second, to those good teachers I have found in my university Facultat d'Informàtica de Barcelona. Finally, many thanks to my teachers at Escola Oficial d'Idiomes de Barcelona, that help me to speak and to write correctly in English and Italian.

During this time, I have been involved in many educational activities related to teach about Bioinformatics in Masters and other programs. Specially thanks among others for your cooperation and your advice to Manuel Gómez (Centro Nacional de Astrobiología, Madrid), Silvia Atriain (Universitat de Barcelona, Barcelona) and Alfonso Valencia (Centro Nacional de Biotecnología, Madrid).

Many thanks also to Dr. Montserrat Corominas and Dr. Jorge Ferrer for two fruitful and interesting collaborations, using the expression data produced during the research performed in their labs.

For formal reasons, I have to thank the Ministerio de Educación y Ciencia of Spain and the Institut Municipal d'Investigacions Mèdiques (IMIM) for the funding for my thesis. Also thanks to Cold Spring Harbor Labs for several travel grants to attend their excellent meetings.

Specially during the latest stages of my thesis I have not much time for my friends so that it is now a good moment to thank you for being there. Specially thanks to David Sánchez for your friendship and for your help, and to David Valldosera for your proximity and wise advice. Also thanks to Josep Vallverdú, Roberto García and Oriol Teixidó for conserving our friendship since we first met at university.

As I said before, it was very clear to me how to begin and end this section. Now that we have arrived at the end, I would like to express my acknowledgments to those that deserve the honor of closing this section: my parents. What I have reached in my life is due to your courage and decision. You can be sure that I will never forget my roots. Thanks to both, for being always my support. Time goes by in my life but you are always here with me. This work is entirely dedicated to you.

Abstract

The sequences are very versatile data structures. In a straightforward manner, a sequence of symbols can store any type of information. Systematic analysis of sequences is a very rich area of algorithmics, with lots of successful applications. The comparison by sequence alignment is a very powerful analysis tool. Dynamic programming is one of the most popular and efficient approaches to align two sequences. However, despite their utility, alignments are not always the best option for characterizing the function of two sequences. Sequences often encode information in different levels of organization (meta-information). In these cases, direct sequence comparison is not able to unveil those higher-order structures that can actually explain the relationship between the sequences.

We have contributed with the work presented here to improve the way in which two sequences can be compared, developing a new family of algorithms that align high level information encoded in biological sequences (meta-alignment). Initially, we have redesigned an existent algorithm, based in dynamic programming, to align two sequences of meta-information, introducing later several improvements for a better performance. Next, we have developed a multiple meta-alignment algorithm, by combining the general algorithm with the progressive schema. In addition, we have studied the properties of the resulting meta-alignments, modifying the algorithm to identify non-collinear or permuted configurations.

Molecular life is a great example of the sequence versatility. Comparative genomics provide the identification of numerous biologically functional elements. The nucleotide sequence of many genes, for example, is relatively well conserved between different species. In contrast, the sequences that regulate the gene expression are shorter and weaker. Thus, the simultaneous activation of a set of genes only can be explained in terms of conservation between configurations of higher-order regulatory elements, that can not be detected at the sequence level. We, therefore, have trained our meta-alignment programs in several datasets of regulatory regions collected from the literature. Then, we have tested the accuracy of our approximation to successfully characterize the promoter regions of human genes and their orthologs in other species.

Resumen

Las secuencias son una de las estructuras de datos más versátiles que existen. De forma relativamente sencilla, en una secuencia de símbolos se puede almacenar información de cualquier tipo. El análisis sistemático de secuencias es un área muy rica de la algorítmica, con numerosas aproximaciones llevadas a cabo con éxito. En concreto, la comparación de secuencias mediante el alineamiento de éstas es una herramienta muy potente. Una de las aproximaciones más populares y eficientes para alinear dos secuencias es el uso de la programación dinámica. Sin embargo, a pesar de su evidente utilidad, un alineamiento de dos secuencias no es siempre la mejor opción para caracterizar su función. Muchas veces, las secuencias codifican la información en diferentes niveles (meta-información). Es entonces cuando la comparación directa entre dos secuencias no es capaz de revelar aquellas estructuras de orden superior que podrían explicar la relación establecida entre éstas.

Con este trabajo hemos contribuido a mejorar el modo en el que dos secuencias pueden ser comparadas, desarrollando una familia de algoritmos de alineamiento de la información de alto nivel codificada en secuencias biológicas (meta-alineamientos). Inicialmente, hemos rediseñado un antiguo algoritmo, basado en programación dinámica, capaz de alinear dos secuencias de meta-información, procediendo después a introducir varias mejoras para acelerar su velocidad. A continuación hemos desarrollado un algoritmo de meta-alineamiento capaz de alinear un número múltiple de secuencias, combinando el algoritmo general con un esquema de clustering jerárquico. Además, hemos estudiado las propiedades de los meta-alineamientos producidos, modificando el algoritmo para identificar alineamientos con una configuración no necesariamente colineal, lo que permite entonces la detección de permutaciones en los resultados.

La vida molecular es un ejemplo paradigmático de la versatilidad de las secuencias. Las comparaciones entre genomas, ahora que su secuencia está disponible, permiten identificar numerosos elementos biológicamente funcionales. La secuencia de nucleótidos de muchos genes, por ejemplo, se encuentra aceptablemente conservada entre diferentes especies. En cambio, las secuencias que regulan la expresión de los propios genes son más cortas y variables. Así que la activación simultánea de un conjunto de genes se puede explicar sólo a partir de la conservación de configuraciones comunes de elementos reguladores de alto nivel, y no a partir de la simple conservación de sus secuencias. Por tanto, hemos entrenado nuestros programas de meta-alineamiento en una serie de conjuntos de regiones reguladoras recopiladas por nosotros mismos de la literatura y después, hemos probado la utilidad biológica de nuestra aproximación, caracterizando automáticamente con éxito las regiones activadoras de genes humanos conservados en otras especies.

Resum

Les seqüències són una de les estructures de dades més versàtils que existeixen. De forma relativament senzilla, en una seqüència de símbols es pot emmagatzemar informació de qualsevol tipus. L'anàlisi sistemàtic de seqüències es un àrea molt rica de l'algorísmica amb nombroses aproximacions desenvolupades amb èxit. Particularment, la comparació de seqüències mitjançant l'alineament d'aquestes és una de les eines més potents. Una de les aproximacions més populars i eficients per alinear dues seqüències es l'ús de la programació dinàmica. Malgrat la seva evident utilitat, un alineament de dues seqüències no és sempre la millor opció per a caracteritzar la seva funció. Moltes vegades, les seqüències codifiquen la informació en diferents nivells (meta-informació). És llavors quan la comparació directa entre dues seqüències no es capaç de revelar aquelles estructures d'ordre superior que podrien explicar la relació establerta entre aquestes seqüències.

Amb aquest treball hem contribuït a millorar la forma en que dues seqüències poden ser comparades, desenvolupant una família d'algorismes d'alineament de la informació d'alt nivell codificada en seqüències biològiques (meta-alineaments). Inicialment, hem redissenyat un antic algorisme, basat en programació dinàmica, que és capaç d'alinear dues seqüències de meta-informació, procedint després a introduir-hi varies millores per accelerar la seva velocitat. A continuació hem desenvolupat un algorisme de meta-aliniament capaç d'alinear un número múltiple de seqüències, combinant l'algorisme general amb un esquema de clustering jeràrquic. A més, hem estudiat les propietats dels meta-alineaments produïts, modificant l'algorisme per tal d'identificar alineaments amb una configuració no necessàriament col·lineal, el que permet llavors la detecció de permutacions en els resultats.

La vida mol·lecular és un exemple paradigmàtic de la versatilitat de les seqüències. Les comparacions de genomes, ara que la seva seqüència està disponible, permeten identificar nombrosos elements biològicament funcionals. La seqüència de nucleòtids de molts gens, per exemple, es troba acceptablement conservada entre diferents espècies. En canvi, les seqüències que regulen l'expressió dels propis gens són més curtes i variables. Així l'activació simultànea d'un conjunt de gens es pot explicar només a partir de la conservació de configuracions comunes d'elements reguladors d'alt nivell, i no pas a partir de la simple conservació de les seves seqüències. Per tant, hem entrenat els nostres programes de meta-alineament en una sèrie de conjunts de regions reguladores recopilades per nosaltres mateixos de la literatura i després, hem provat la utilitat biològica de la nostra aproximació, caracteritzant automàticament de forma exitosa les regions activadores de gens humans conservats en altres espècies.

PART I

Preliminaries

Chapter 1

Introduction

Summary

This chapter details the general questions of the document. It provides a brief explanation of the motivation for this work. Then, the list of objectives of the thesis is introduced. The completion of these tasks and the final calendar of execution of the project (year by year) is included as well. The manuscript is logically divided into three different parts: Preliminaries, State of the Art and Meta-alignments. There is also a brief description of the chapters of each part. Finally, some particular considerations about how to read the book and the layout of the document are presented.

1.1	General objectives	4
1.2	Objectives	4
1.3	Thesis chronology	5
1.4	Outline of this thesis	7
1.5	Particular considerations	8

1.1 General objectives

THE PRINCIPAL OBJECTIVE OF THE THESIS DISSERTATION is to explain in detail the topic on which the work of several years has been focused. In addition, the experience of the author at different areas has been reflected here in the numerous descriptions and solutions to several biological and computational problems. Speculation about future research and criticism have been a valuable ingredient as well.

This is a thesis about computational sequence analysis, particularly applied to characterize genomic sequences. The way in which this synergy between a biological problem and a computational solution is expressed was considered to be crucial for the success of this document. The generality of the proposed solutions, which can be applied to any type of sequence (biological or not), is also underlined in the corresponding sections.

The core of the thesis is the development of a new family of algorithms to align transcription regulatory regions. Among them, a global pairwise algorithm and a global progressive multiple algorithm have been shown to be useful in the characterization of a gene promoter region, specially when the amount of predictions by other systems is excessive. Sketches of other versions are also provided (parallel, local).

The work performed about the meta-alignment strategy has been interestingly complemented and enriched with a serious approach to the algorithms that originated the concept of sequence analysis several decades ago. Such a chapter is an interesting opportunity to review for the first time some of the classic papers in the field that are still very relevant, in spite of the deluge of new proposals and publications continuously released. The introduction of this material in the document improves without any doubt the quality of the final manuscript.

In addition, several references about the relationship between current advances in genomics and society can be found in the text. In my opinion, ethics must be part of any human achievement. Genomics and other 'Omics' disciplines promise to radically change our way of life. Medicine, biotech farming, crime investigation and personal privacy among others will be severely affected.

To sum up, this thesis aims to become an educational book reference. This is an excellent opportunity to explain in detail the topic of the meta-alignment but also to construct an exciting portrait of sequence analysis in computational biology. To satisfy all of these requirements, the use of current technologies to produce an outstanding work was also mandatory. Thus, a DVD with additional materials (electronic thesis, relevant bibliography, source code, educational material, ...) supporting the main text is a good complement to the PhD dissertation.

1.2 Objectives

The characterization of gene regulatory regions is a fundamental step toward understanding the great existing variability between different species. However, it is still an open problem due to the peculiar features of the regulatory elements. The research in this PhD thesis

has been oriented to the development of new computational methods of alignment to deal with such information. However, it is important to mention that the algorithms presented here can deal with other problems that show a similar theoretical framework, lacking of a biological background.

In short, the following objectives were established in 2001 for this thesis:

- ① To study the biological problem of gene regulation in eukaryotes. This includes the control of gene expression, specially through the transcription of the genes: promoters, transcription factors, DNA-protein binding sites, chromatin effect, CpG islands.
- ② To analyze the current computational methods to search regulatory elements in a promoter region. This includes the algorithms based on pattern matching using catalogues of regulatory elements and the pattern discovery algorithms that extract useful information from a set of related sequences.
- ③ To investigate the more recent comparative approaches based on phylogenetic footprinting and microarrays. To understand the biological concepts behind the gene orthology. To study the biological and technological concepts of the high-throughput expression experiments.
- ④ To analyze the existent sequence pairwise sequence alignment algorithms. To study the concept of map, the mapping functions and the map alignment problem.
- ⑤ To design novel algorithms to align two regulatory sequences that produce a minor amount of false positives. To present real biological scenarios in which these approaches show to be more efficient than the conventional sequence alignment algorithms.
- ⑥ To compile and to maintain a public dataset of regulatory annotations suitable for training these and other algorithms that deal with data from comparative genomics and microarray experiments.
- ⑦ To study several alternatives to extend the basic pairwise approach developed before to align multiple sequences. Test this approach on orthologous datasets and microarray expression data.
- ⑧ Public distribution of the software and the databases produced during this thesis to the scientific community. To write web servers that implement most of the methods presented above.

1.3 Thesis chronology

This is a short enumeration of the main tasks implemented during the PhD thesis and their associated results, year by year:

➡ 2001

- ① Planning: decide the main lines and the objectives of the thesis.

- ② Biological problem: bibliographical research in general molecular biology books about the eukaryotic transcription and other forms of gene regulation.
- ③ State of the art: bibliographical research in published papers about the classical algorithms and strategies to analyze gene promoter regions. Including the study of the advanced techniques such phylogenetic footprinting and microarray experiments.
- ④ Attended conferences: Intelligent Systems in Molecular Biology (ISMB) at Copenhagen, Denmark.

➤ 2002

- ① Analysis of co-expressed genes in *Drosophila melanogaster*: gene characterization, G+C content, clustering, gene function, promoter characterization including phylogenetic analysis.
- ② Analysis of co-expressed genes in *Mus musculus*: the results of several microarrays were analyzed with the existing computational tools, including phylogenetic footprinting.

➤ 2003

- ① Developing the global and local meta-alignment first prototypes.
- ② Bibliographical research to find regulatory data for training the meta-alignment approach.
- ③ Attended conferences: Research in Computational Biology (RECOMB) at Berlin, Germany.

➤ 2004

- ① Tuning the meta-alignment. Improving the efficiency of the basic implementation with lists.
- ② Writing the web server of the pairwise meta-alignment program.
- ③ Training the meta-alignment on a small dataset of annotated promoters.
- ④ First prototypes for multiple meta-alignment.
- ⑤ Attended conferences: Systems Biology at Cold Spring Harbor Labs, New York, USA.

➤ 2005

- ① Creation of a public database of annotated promoters (ABS).
- ② Final tests: pairwise meta-alignment approach on the CISRED database.
- ③ Evaluation of the quality of weight matrices using the meta-alignment.
- ④ Tuning the multiple meta-alignment. Improving the computational efficiency.
- ⑤ Variations to allow the existence of non-colinear alignments in the results.
- ⑥ Starting to write the thesis dissertation.
- ⑦ Attended conferences: Systems Biology at Cold Spring Harbor Labs, New York, USA.

⇒ 2006

- ① Final training of the multiple meta-alignment on a set of orthologous of multiple species.
- ② Finishing the thesis dissertation.
- ③ Public defense of the PhD thesis.

1.4 Outline of this thesis

This thesis has been written following the format of a text book. The main text is divided into three parts: introduction, state of the art and results. Every part consists of a set of chapters, each one devoted to a given topic. Chapters can be read separately to facilitate the accession to individual parts of the book, but the thesis has been written following a linear and continuous logical script.

This is a brief description of the content of each chapter:

- ① Introduction: general motivation of the thesis containing the objectives, the calendar and other considerations about the project and the format of the book.
- ② The post-genomic era: biological description of genomic concepts (genes, DNA, mRNA), the genome sequencing projects, bioinformatics, future implications of the genomic research in medicine.
- ③ The golden age of sequence analysis: a comprehensive historical review of the pioneering algorithms in sequence and map alignment in the seventies and eighties, including a detailed analysis of the most relevant ones.
- ④ Computational gene and promoter characterization: a survey of the state of the art in the analysis of genomic sequences (genes and regulatory regions), and a study of the different techniques implemented such as the representation of signals, the detection of biased content regions or the homology search.
- ⑤ Pairwise meta-alignment of regulatory sequences: the mapping functions, the TF-map approach, basic implementations, the accurate construction of collections of examples, the training, the application on a database of co-regulated genes, the detection of promoter regions, the use of meta-alignment to evaluate the specificity of matrices. Other versions: local and parallel meta-alignment.
- ⑥ Multiple meta-alignment of regulatory sequences: the progressive approach, the design of the final solution, the modification to produce non-colinear alignments, the tests on orthologous promoters from multiple species.
- ⑦ Conclusions: the enumeration of the results of this thesis.
- ⑧ Appendix section: curriculum vitae, software and web servers, publications, posters, web glossary.

1.5 Particular considerations

The following are some individual considerations about the thesis:

- The electronic version of this document has hyper links for the table of contents, for the bibliographic references, but most important of all, also for the web addresses on the Internet—from now on, their Uniform Resource Locator (URL). This means that you can visit the corresponding web page by clicking your pointer on them, in case that you have your PDF viewer properly customized. Many of the URLs presented in this book have been collected in a web links reference index available on page 241. URLs within paragraphs have been moved into that web glossary in order to avoid unbalanced line breaks and for a more pleasant reading. A reference to the corresponding page in the web reference index is provided instead.
- An attempt has been made to keep software names as provided by their authors. Those names appear in a `monospaced serif font`. Database names are typeset in a `SMALLCAPS SANS-SERIF FONT`. A *emphasized font* was used for gene names.
- The first time an acronym appears in the document, the full name will be provided and the acronym itself will be shown in parentheses.
- The publications and submissions of papers in which the author of this thesis was involved are included at the end of the thesis as an appendix.
- The use of colour is considered to be essential to accurately highlight some contents of the thesis such as the equations, the algorithms or the figures and the tables.
- The author of this thesis has carefully selected the bibliography of each chapter. Following such references, a detailed reconstruction of such a topic can be performed with great accuracy. Some of these references are also included as electronic supplementary material in the DVD companion to this thesis.

Chapter 2

The post-genomic era

Summary

This chapter is a basic survey of the molecular and cell biological concepts that will be used throughout this thesis, with special emphasis on the topics of genetics and genomics. In addition, the relatively new discipline of bioinformatics is examined, focusing on the genomic databases and the integration of data from different biological domains. The dramatic changes that medicine and drug design are going to experience after the sequencing of the human genome project are explored at the end of the chapter.

2.1 The genomic landscape	10
2.2 The genomic era	17
2.3 The post-genomic era	24

2.1 The genomic landscape

The universe of the cells

THE CELL, a small membrane-bounded compartment filled with a concentrated aqueous solution of chemicals, is the essential constituent of life. Bacteria, plants, birds or humans, all living organisms on Earth are made of at least one cell. Because of their apparent simplicity and flexibility, cells have been able to achieve an incredible success in their perpetuation efforts (Alberts et al., 1994).

All living beings and the cells that form them are believed to have descended from a common ancestor cell through evolution by natural selection. This process involves two simple steps: (1) random variation in the genetic information passed from an individual to its descendants and (2) selection of the genetic information that permits its possessors to survive and propagate in their environment.

Evolution began billions of years ago in our planet. Simple organic molecules (molecules containing carbon) such as amino acids and nucleotides are likely to have been produced under primitive conditions on Earth. Later, these molecules associated to form polymers or more complex structures such as proteins and nucleic acids (DNA and RNA). The competition between such primitive structures for the available precursor materials in that unstable environment produced many of the biological processes present in many cells now. The interplay between DNA and RNA in the protein synthesis pathway is the best example of this. At present, DNA acts as the permanent repository of genetic information in most cells while RNA, originally the molecule from which rudimentary peptides were produced, remains as an intermediary between DNA and proteins (Alberts et al., 1994).

The isolation from the external medium was one of the crucial events leading to the formation of the first cell. The development of an outer membrane by phospholipids around some of these primitive structures provided a brand new capability: the protection of the information that could contribute selectively in the competition against other similar systems (e.g. hereditary material such as a variant RNA that made a superior type of enzyme).

These primitive cells that have survived successfully until our days are the bacteria (also known as prokaryotes). The structure of a bacteria is a simple cell wall beneath which a plasma membrane encloses a single cytoplasmatic compartment containing the genetic material, proteins and small molecules. Basically, survival in bacterial terms means to achieve the fastest speed of replication or cell division to incorporate as many genetic changes as possible on their DNA through each generation. Genetic variability facilitates a rapid adaptation of the species to a changing environment.

The action of millions of these organisms slowly caused revolutionary changes on Earth. The atmosphere was transformed through cyanobacterial photosynthesis or respiration from a mixture with practically no oxygen to one in which oxygen constitutes 21% of the total (Alberts et al., 1994). This dramatic change in the environment produced the extinction of many types of cells but also induced the symbiosis between ancient cells adapted to the prebiotic environment without oxygen (anaerobic) with those possessing the ability to process the oxygen (aerobic).



Figure 2.1 Electron micrograph of a chicken chondrocyte. Chondrocytes are cells from the cartilage (connective tissue). Adapted from UBC BIOMEDIA IMAGE AND MOVIE DATABASE (see Web Glossary, page [244](#)).

This transition to more structured cells named eukaryotes implied numerous additional changes in response to the new situation: bigger size, a rich array of internal membranes to facilitate the transport of the materials for biosynthetic reactions occurring inside the cell and finally, a new inner membrane to protect the increasing genetic material. The stability of the DNA double helix made the storage of higher quantities of genetic information easier. Additional packaging mechanisms were required to manipulate the growing hereditary material inside this second membrane, also known as nuclear membrane (Alberts et al., 1994).

The next step in evolution was the appearance of multicellular organisms. By collaboration and division of tasks, the efficient exploitation of resources that no single cell could utilize before was now possible. Multicellularity enables an individual to separately specialize groups of its cells to perform absolutely different tasks in a collaborative manner. An electron micrograph of an eukaryotic cell from connective tissue is shown in Figure 2.1. All of the cells of every multicellular organism have the same genetic material and are generated by repeated division from a single precursor cell. But, surprisingly, despite having an identical genetic composition when they grow, they become differentiated from others, adopting a different structure and different functions (Alberts et al., 1994).

The mechanisms that governed this amazing ability for specialization are intimately related with the management of the basic units that form the genetic information of a cell: the genes.

Genes and inheritance

The basic component of deoxyribonucleic acid or DNA is the nucleotide, defined by its chemical base: Adenine (A), Cytosine (C), Guanine (G), and Thymine (T). The DNA that constitutes the genetic material of cells is a double-stranded molecule consisting of two chains of nucleotides running in opposite directions. The A-T and G-C base pairs are complementary because these bases form hydrogen bonds that keep them together. Thus, each strand of the molecule is a template to make a copy of the other sequence of bases.

The genes are the basic physical and functional units of heredity. Genes are fragments of DNA with a specific sequence of bases that encodes instructions on how to control a discrete hereditary characteristic. The set of genes belonging to an individual is the genotype. The phenotype is the set of traits expressed in an individual with a certain genotype. A polymorphic gene is a gene in which small variations in its sequence from two different individuals produce different observable physical traits. Each one of the set of alternative forms of a gene is an allele or variant¹.

In sexually reproducing organisms, such as humans, each gene in an individual is represented by two copies or alleles, one from each parent. A dominant allele is an allele that is almost always expressed, even if only one copy is present, overshadowing the other. Known examples of dominant alleles are Huntington's disease and polydactyly (extra fingers and toes). On the contrary, a recessive phenotype will only be expressed if both copies contain the recessive allele. When a recessive allele is overshadowed by a dominant allele and the recessive trait is not expressed, the individual is said to be a carrier for that trait. Recessive disorders in humans include sickle cell anemia and Tay-Sachs disease (NCBI report: genomics, see Web Glossary, page 243).

There are exceptions to these basic laws, usually complex interactions among various allelic conditions:

- Co-dominant alleles both contribute to a phenotype, for example in the case of human blood group.
- Pleiotropy is the phenomenon in which a single gene is responsible for producing multiple and apparently distinct traits.
- A gene that masks the phenotype of another gene is an epistatic gene while the subordinated gene is the hypostatic gene such as in the case of the albinism gene.
- There are traits that are multigenic because they result from the expression of several different genes such as the three genes at least that determine eye colour.

The cell cycle is the process that a cell follows to replicate. To produce a copy of the original cell having an identical genetic composition, the hereditary material is duplicated. Errors are not unusual to happen during the copy. Moreover, dramatical changes in the environment such as exposure to ultraviolet radiation or toxic chemicals can promote changes in the DNA as well. Genetic variations are usually the result of mutations in the sequence of a functional element: substitutions, deletions or insertions of nucleotides. Mutations that

¹See Lander and Weinberg (2000) for a comprehensive historical review of genetics.

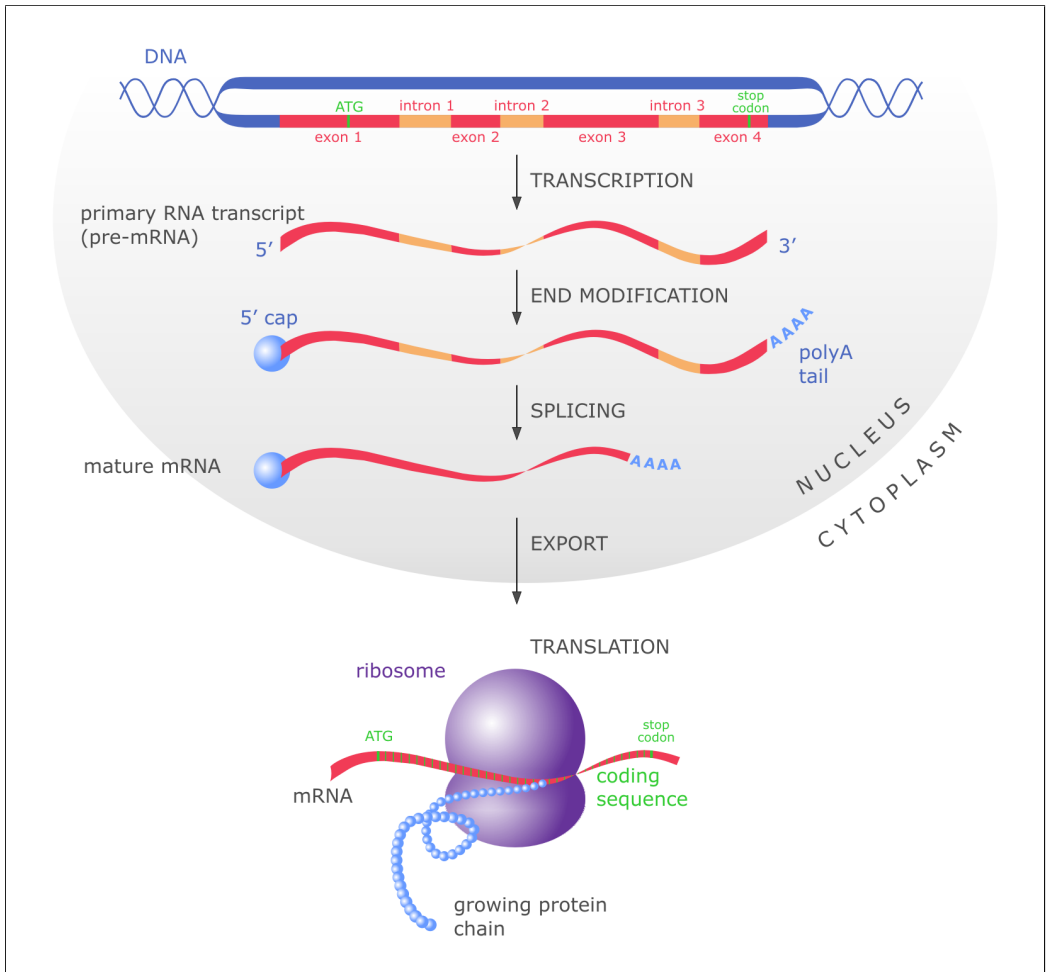


Figure 2.2 The molecular processes involved in the pathway leading from DNA to protein. See main text for further details. Adapted from Blanco and Guigó (2005).

occur in germ cells will be passed on to the next generation while those changes in ordinary cells will only affect the individual.

Although most defective cells die quickly, some can persist and may even become cancerous if the mutation affects cell growth control. However, not all mutations are negative. The main effect of mutations is the opportunity to adapt to a new environment by following the rules of the natural selection: most mutations do not produce any observable result in an organism, others are terribly pernicious causing severe damage, and a minority of them substantially improve the probability of success in the propagation of its genes (Alberts et al., 1994).

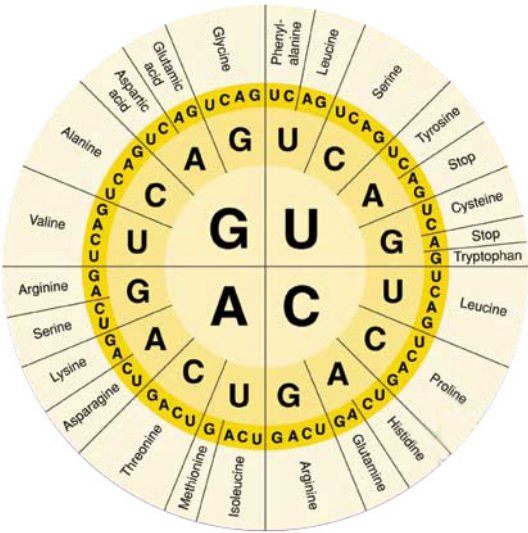


Figure 2.3 The genetic code table. Translation begins from the inner circle to the outer ones. For instance, the codon AUG is translated as *Methionine*.

Genes and proteins

Ribonucleic acid or RNA molecules are single-stranded chains of nucleotides that are constructed using one of the two DNA strands of a given gene as a template, with the substitution of Thymine (T) for Uracil (U). Each gene produces a functional RNA molecule (Alberts et al., 1994). Transcription from DNA to RNA is the first step in the protein synthesis pathway, schematically represented in Figure 2.2. Each RNA molecule can encode a protein or, alternatively, constitute other structures such as ribosomal RNAs, transfer RNAs or small nuclear RNAs.

RNAs that are the result of transcribing protein-coding genes undergo different modifications. First, the ends of these primary transcripts are modified to stabilize the molecule. Second, an editing process called splicing cuts and removes some fragments of the transcript (the introns) and pastes together the remaining ones that contain the information to build the protein (the exons). The processed RNA receives the name of messenger RNA or mRNA because it is then ready to leave the nucleus of the cell. For many genes, more than one splicing form is already known, increasing the volume of information contained in a given gene (Alberts et al., 1994).

The final step is the translation of the mRNA, mediated by the rybosomes. The information contained in the sequence of nucleotides from the mRNA is used to produce a protein. Each group of three nucleotides (a codon) is translated into an amino acid that is added to the growing protein using the genetic code (see Figure 2.3). In eukaryotes, translation initiates at the start codon ATG while it is terminated when one of the stop codons TAA, TAG, or TGA is reached. Because of the length of a codon and the dual nature of the DNA molecule, there are always six different forms to translate a nucleic acid sequence: three reading frames (0,1,2) and two directions (forward and reverse).

DNA is only the carrier of genetic information in a cell. Proteins (often in combination with RNA molecules) are the biomolecules actually responsible for main cellular functions: they catalyze nearly all chemical processes in cells, give them their shape and movement capability, transmit signals through the body, recognize foreign molecules, or transport other elements.

Genes are not continuously being transcribed during each stage of the lifetime of the cell. According to every specific situation inside and outside of the cell, the need for some proteins to perform a given function launches the transcription of a subset of genes encoding those products. Contrarily, the excess of other proteins prevents or stops the transcription of their genes. The activation of a gene is a complex procedure in which many actors play different roles in the genetic material of the cells.

Genome anatomy

In eukaryotes, DNA molecules are long linear polymers that can contain millions of base pairs arranged in an ordered sequence that encodes the genetic information of the cell. A million nucleotides measures a distance of approximately 0.03 cm, only occupying a volume of 10^{-15} cm³. These tightly coiled packets consist of the double helical DNA structure wrapped around specific protein complexes called histones (Alberts et al., 1994).

The genetic material of an organism is part of an apparently chaotic organization called chromatin during the entire lifetime of the cell except replication. However, the chromatin is condensed in individual units that receive the name of chromosomes when the cell is undergoing a nuclear division process. In both configurations, the complete set of DNA of an organism constitutes its genome. In Figure 2.4, a fragment of chromatin, a duplicated chromosome and the complete set of human chromosomes are shown. Only when the process of duplication of genetic material has been finished, the genome of the cell is arranged in two copies of the chromosomes to be distributed into the two new cells. In the meantime, the genome is in a semi-decondensed state in which the regions of chromatin containing genes are accessible for being transcribed.

Genomes widely vary in size because of many causes. The complexity of an organism is not directly related with the size of its genome or the number of genes encoded within. The size of several genomes in millions of base pairs is listed in Table 2.1. Interestingly, a substantial proportion of the genes are relatively conserved between different genomes due to the evolution process. The differences we observe between species are mostly because of minimal changes. For instance, the human genome sequence is 99% identical to the chimpanzee sequence while the difference between two people is estimated to be less than 0.1%. One of the main types of sequence variation between individuals are the single nucleotide polymorphisms (SNPs). SNPs are sites in the genome where individuals differ in the DNA sequence by a single base. It is believed that there are at least 10 million SNPs in the human genome (DOE report, see Web Glossary, page 241).

The genome is not exclusively a container of genes. On the contrary, the genomic landscape is rich and complex. Using the human genome as a reference, the protein coding fraction of the genome is only 2%. What is more, genes and related gene regulatory sequences actually occupy together a third part of the total three billion base pairs. As is represented in Figure 2.5, there is a huge part of the human genome called intergenic DNA which has been

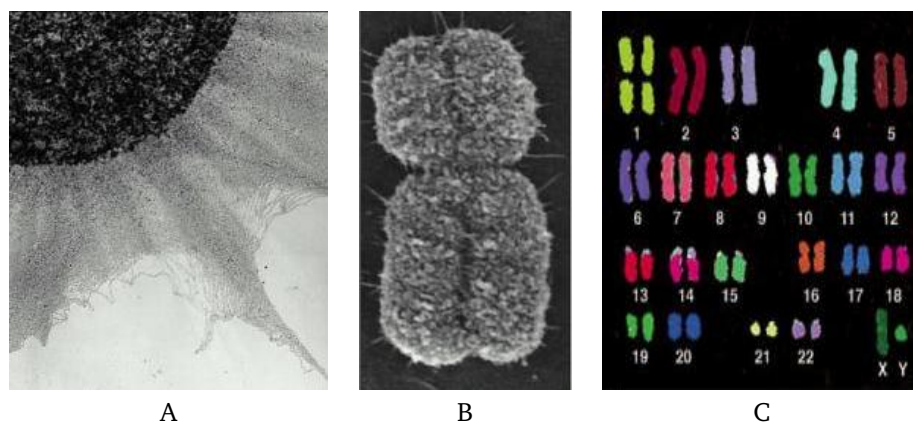


Figure 2.4 A comparison of chromatin with a mitotic chromosome and the karyotype.

(A) An electron micrograph showing a tangle of chromatin spilling out from a nucleus. (B) A scanning electron micrograph of a mitotic chromosome. The two copies are still linked. (C) Human chromosomes (karyotype). Staining is performed by exposing them to a collection of DNA molecules that have been coupled to a combination of fluorescence dyes. Adapted from [Alberts et al. \(1994\)](#).

structurally characterized into different elements for which no known function has been assigned yet. They could play some role in chromosome structure and dynamics or might simply arise through an error in the process of copying the genome during cell division ([Brown, 2002](#)).

The bulk of this intergenic DNA is made up of repeated sequences. Repetitive DNA can be divided into two categories:

- ① Genome-wide or interspersed repeats. Repeat units distributed around the genome in an apparently random fashion. Transposable elements or transposons are mobile segments of DNA that are able to move around the genome from one place to another, leaving a copy of themselves in the original place.
- ② Satellite or tandemly repeated DNAs. Repeat units that are placed next to each other in an array. The commonest type of satellites are dinucleotide repeats and single nucleotide repeats.

Because of the complex nature of genomes, the annotation of the different elements that constitute the whole genomic landscape of a species is a non-trivial task and it requires many years and a lot of effort. Computers have been playing a key role in the major sequencing projects. Furthermore, they are still essential in the unveiling of the thousands of relationships between the genomic components that govern cell behavior.

SPECIES	COMMON NAME	GENOME SIZE	GENES
<i>Saccharomyces cerevisiae</i>	Yeast	12,156,590	6,680
<i>Caenorhabditis elegans</i>	Nematode worm	100,585,160	20,065
<i>Anopheles gambiae</i>	Mosquito	278,253,050	13,277
<i>Apis mellifera</i>	Honey Bee	228,567,597	13,448
<i>Drosophila melanogaster</i>	Fruit fly	144,138,837	13,985
<i>Fugu rubripes</i>	Pufferfish	393,296,343	22,008
<i>Gallus gallus</i>	Chicken	1,054,197,620	18,632
<i>Mus musculus</i>	Mouse	2,676,244,419	24,256
<i>Rattus norvegicus</i>	Rat	2,718,897,321	21,952
<i>Bos taurus</i>	Cow	1,741,208,718	23,231
<i>Pan troglodytes</i>	Chimpanzee	2,733,948,177	22,475
<i>Homo sapiens</i>	Human	3,433,077,231	23,341
<i>Triticum aestivum</i>	Wheat*	17,000,000,000	50,000

Table 2.1 Comparison of the sizes of several eukaryotic genomes. Data extracted from ENSEMBL (May, 2006). Estimated values for wheat.

2.2 The genomic era

Bioinformatics

With major advances in the technologies that supply molecular data and the posterior explosive growth in the amount of available biological information, the application of computers to organize and understand this enormous volume of knowledge became essential. Bioinformatics is the field of science in which biology, computer science, information technology, mathematics and statistics converge to form a single discipline. The ultimate goal of bioinformatics is the combination of many sources of biological information to develop a comprehensive picture of normal cellular activities (NCBI report: bioinformatics, see Web Glossary, page 243).

Broadly, bioinformatics tasks can be divided into three categories:

- ① Implementation of databases to organize existing information from many areas of biological research such as genomics, transcriptomics and proteomics, allowing the public scientific community to efficiently access the data and to avoid redundancy and multiplicity. Doubtlessly, the advent of internet has played a central role in the achievement of this challenge (Goodman, 2002).
- ② Development of new algorithms and statistics that aid the analysis of the data such as sequence alignment methods, motif detection techniques, phylogenetic studies or protein folding simulation. Advanced algorithmic methods and mathematical frameworks are essential to extract biological knowledge from the databases.

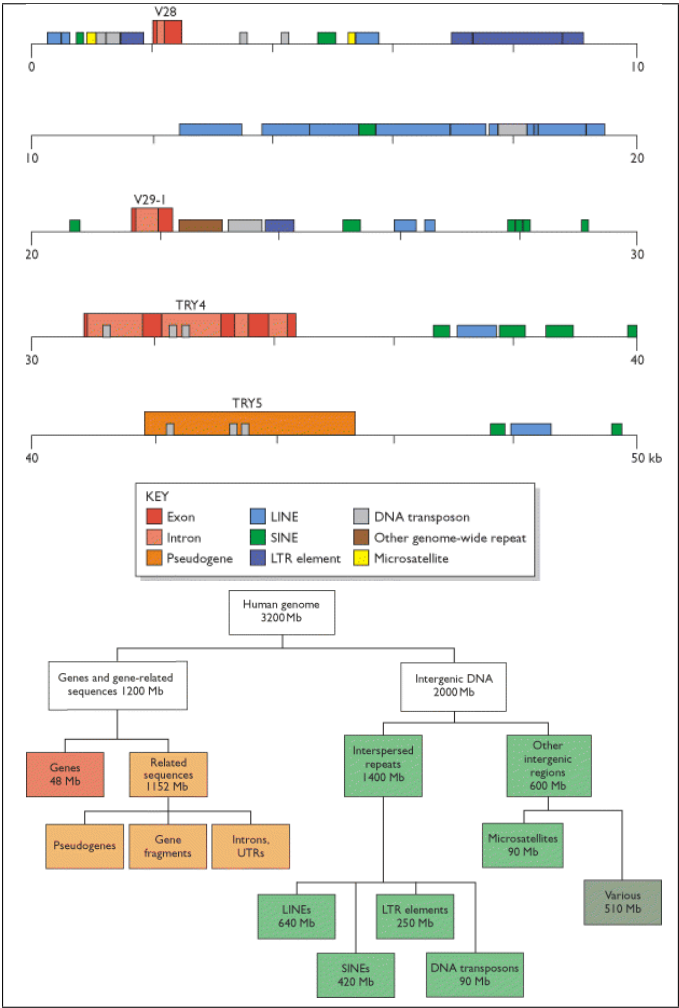


Figure 2.5 The organization of the human genome. (Top) A segment of the human genome. (Bottom) The contribution of different genomic elements to the human genome. Adapted from Brown (2002).

③ The analysis of such data and the interpretation of the results in a biologically meaningful manner to provide a more global perspective (new testable hypotheses) in future experimental designs. So far, it is far often easier to produce sequence data than to understand its function so that this is the most complicate of the three tasks (Bogusky, 1998; Claverie, 2000; Pearson, 2001).

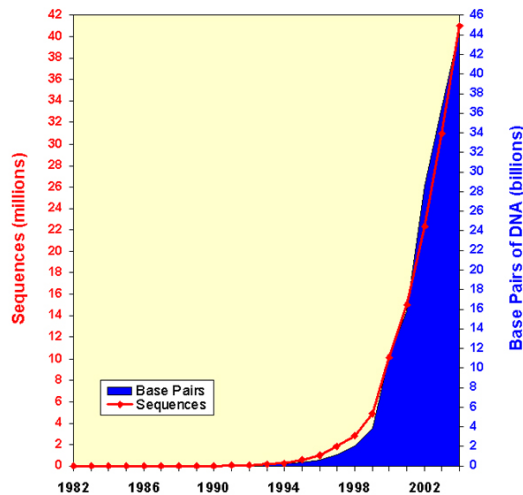


Figure 2.6 Growth of the GENBANK (1982-2004). Adapted from GENBANK (see Web Glossary, page 242).

Sequence databases

A biological database is a large, organized body of persistent data designed to be queried and retrieved in a very efficient manner by the scientific community. Because of the nature of the first data, ancient databases were merely collections of sequences of proteins distributed as a printed work (Dayhoff et al., 1965). Nonetheless, the need for an electronic format became obvious just when the amount of sequences was unmanageable (Baxevanis and Ouellette, 2005; Mount, 2001). With substantial experimental sequencing improvements and the advent of DNA sequence databases initiated by the European Molecular Biology Laboratory (EMBL, Germany) and Los Alamos National Laboratory (LANL, United States), the number of available sequences experienced an exponential growth (see Figure 2.6).

Major public nucleotide and protein sequence databases such as EMBL (Kulikova et al., 2004, see Web Glossary, page 242) or GENBANK² (Benson et al., 2003, see Web Glossary, page 242) are repositories of sequences submitted by researchers in order to make them accessible for the rest of the biological community. An accession number and a set of annotations are provided for each sequence entry. Using flat files as a standard format, the features of each sequence are displayed in a simple format that divides each line of information into two elements: a field descriptor and a value. The popular FASTA format is one of the *de facto* standards that have been adopted to represent a sequence of nucleotides or amino acids (see Figure 2.7 for an example of a GenBank entry and the associated FASTA file).

Because of the relative lack of control over the quality and quantity of the data stored in the sequence databases during the first years, there was soon a necessity to maintain collections of data free of redundancy and errors constructed from the original repositories.

²GenBank is now under the auspices of the National Center for Biotechnology Information (NCBI, United States).

Since then, numerous curated databases, also known as secondary databases, have appeared aiming to avoid any type of multiplicity and low quality data (Baxeavanis and Ouellette, 2005).

A successful example of these refined catalogues is the REFSEQ collection (Pruitt et al., 2005, see Web Glossary, page 243). The major goal of this database is to provide a unique sequence for each molecule in the protein synthesis pathway (DNA, mRNA and protein). To reduce the noise produced by the representation of a single biological entity with many entries in the sequence databases, each biological entity is represented only once in REFSEQ, maintaining a non-redundant repository.

Genomic databases

Once the complete assembly of first eukaryotic genomes such *Saccharomyces cerevisiae* (Goffeau et al., 1996) or *Drosophila melanogaster* (Adams et al., 2000) was achieved, the principal focus of computational biology research shifted from individual sequences to chromosomes and whole genomes. With the release of the human genome (Lander et al., 2001; Venter et al., 2001; International Human Genome Sequencing Consortium, IHGSC, 2004), it became necessary to introduce an important change in the way the assemblies and the genome annotations were presented. Finally, the recent availability of the mouse genome (Waterston et al., 2002), the chicken genome (Hillier et al., 2004) and the sequencing of other model organisms has augmented the need for a new kind of tools to permit the annotation and comparison of many genomes in a more sophisticated form. In addition, support for genomes that have not been finished yet has also been crucial (archives of traces and preview releases).

There are three well established genome browsers that aim to fulfill this need:

- The ENSEMBL project (Birney et al., 2004, see Web Glossary, page 242), a collaboration between the European Bioinformatics Institute and the Sanger Institute. The main browser currently provides a set of gene, transcript and protein predictions for each genome. Data is presented on pages called Views, each View showing a different level of detail.
- The UCSC GENOME BROWSER (Karolchik et al., 2003, see Web Glossary, page 244), produced by the University of California, Santa Cruz Genome Bioinformatics Group. It serves annotations for many eukaryotic genomes, presenting the information in the form of tracks. Each track corresponds to a certain genomic feature.
- The NCBI MAP VIEWER (Wheeler et al., 2005, see Web Glossary, page 243), provides maps for a lot of organisms, many of them without finished assembly. The browser is tightly linked to most services of the NCBI web. The information is displayed using maps. Maps are vertical representations of annotations along a given chromosome. There is a map associated to each genomic feature.

The core of the three browsers is the internal gene annotation pipeline that must be executed on every new sequence assembly of each genome. Genes are annotated according to experimental evidence and computational predictions. Comparisons between different

GENBANK

■ 1: U30787, Reports Human uroporphyrinogen decarboxylase [gi:1322018]	
LOCUS	HSU30787 4514 bp DNA linear PRI 16-MAY-1996
DEFINITION	Human uroporphyrinogen decarboxylase (URO-D) gene, complete cds.
ACCESSION	U30787 X06048
VERSION	U30787.1 GI:1322018
KEYWORDS	.
SOURCE	Homo sapiens (human)
ORGANISM	Homo sapiens Eukaryota; Metazoa; Chordata; Craniata; Vertebrata; Euteleostomi; Mammalia; Eutheria; Euarchontoglires; Primates; Catarrhini; Homnidae; Homo.
REFERENCE	1 (bases 785 to 1143)
AUTHORS	Romana,M., Dubart,A., Beaupain,D., Chabret,C., Goossens,M. and Romeo,F.H.
TITLE	Structure of the gene for human uroporphyrinogen decarboxylase
JOURNAL	Nucleic Acids Res. 15 (18), 7343-7356 (1987)
PMID	3658695
REFERENCE	2 (bases 1 to 4514)
AUTHORS	Moran-Jimenez,M.J., Ged,C., Romana,M., Enriquez De Salamanca,R., Taieb,A., Topi,G., D'Alessandro,L. and de Verneuil,H.
TITLE	Uroporphyrinogen decarboxylase: complete human gene sequence and molecular study of three families with hepatoerythropoietic porphyria
JOURNAL	Am. J. Hum. Genet. 58 (4), 712-721 (1996)
PMID	8644733
REFERENCE	3 (bases 1 to 4514)
AUTHORS	Romana,M., Ged,C. and Verneuil,H.
TITLE	Direct Submission
JOURNAL	Submitted (30-JUN-1995) Cecile Ged, Biochimie Medicale et Moleculaire, University of Bordeaux II, 146 rue Leo Saignat, Bordeaux, France
FEATURES	
source	Location/Qualifiers 1..4514 /organism="Homo sapiens" /mol_type="genomic DNA" /db_xref="taxon:3606"
protein_bind	1028..1037 /bound_moiety="Sp1"
TATA_signal	1066..1073
gene	1089..4514 /gene="URO-D"
mRNA	/product="uroporphyrinogen decarboxylase" join(1107..1126,1748..1860,1976..2055,2132..2194,2434..2631,2749..2910,3279..3416,3576..3676,3780..3846,4179..4340) /gene="URO-D"
CDS	/product="uroporphyrinogen decarboxylase" join(1107..1126,1748..1860,1976..2055,2132..2194,2434..2631,2749..2910,3279..3416,3576..3676,3780..3846,4179..4340) /gene="URO-D" /codon_start=1 /product="uroporphyrinogen decarboxylase" /protein_id="AAC50482.1" /db_xref="GI:1322019" /translation="MEANGLGPGQGFELKNDTFLRAAWGBETDYTFWCMRQAGRYLP EFRETRAADPFSTCFEACCELTLQPLRFLLDAALIFSDLLVVEQALGMEVTHVP GKGFSPFELREQLERLDFEYVASELGVFOAITLFRGLAGRVPLIGFAFWET LMTYVMEGGSGSTMAQKRWLYRFOQASHQLRLITLDALVPLVGVVAGAQALQLPE SHAGHLGQPLNKFALFYIRDVAKQVKRLRAGLAPEVMIIFAKDGHFALEELAQAG YEVVGLDWTVPKFKARECVGKVTVTLQNLDFCALYASEEIEGLQVLKMLDDGPHRYI ANLHGSLYDMDPEHVGAFVDAVHKHSRLLRQ"
ORIGIN	
1 aagcttgcta agcacctctc ggggaacgaa agccagcgct gcttaggggc gcggcgggc 61 gaagctctaa cctctgcaaa gaagcgaac ggcgcggag ctgcggggg gactcaaga 121 ccggcgccgg ccattggacc gcattgagtc agctggcgcc acgcgggaca gagcttccca 181 ccacgccctt ccccgcttt ggcagcctt tgccgtatgt tctggactaa ggcgaaccca 241 cccacccccc tcttcccc tcttcccc tcttcccc tcttcccc tcttcccc	

FASTA

```

>gi|1322018|gb|U30787.1|HSU30787 Human uroporphyrinogen decarboxylase
AAGCTTCGTAAGCACCTCTCGCGGCACGAAAGCCAGCGCTGCCTAGGCGCCGCCGGCGGAGGCTCTCA
CCTCTGCCAAGAGCGCACCGGCCAGCAGCTGCCGGGGGACTCCAGCACCGCGCGGGCCATGGAGCCC
GCCATGAGTCAGCTGGCGCGACCGCGGACAGAGCTTCCACACGCCCTTCCCCGCCTTTGGCCAGCCTT
TGCGCTATGTTCTGGACTAAGCGCACCCAGCTCTCACTGTATTGGACTGTGTACTCCCACTCAACCA
TATTACTTATCTCTGTCACCCCTAACCCAGCCGACCAACCCAAAGATTGGTGATTGTCACCTGATCAAT
CTCCCTCTCTCCATTCTCTGTGACTACCATTTTATCTCTACTGCTACTACCCCTATTCAAGTCACCAATT
CTAGCTAGCCTGGGTCAATGCCAACAGTCATTTTCTGGTTCTTCGGCCTGCTGTTTCTCTCCACTCC
CAGCGAATCTGCTGGACTCCCTATCTATGGGTGGTGTGATTAAAGTGTTTGAGACAATGCCCTTTCCC
CTGCCACTGACAGGATCTTGAGTCATTAGGGTGGTGTCTGTTTGACACTCCTAATCCCAAGGACACTG
GAGATCATTATTCATTTTAAATGTGATTGCTGATTCTGTTTCCCCAGCTCTTGAGCTCTTTAAAGGCTGG
GGTGTCTTGAGCAGAGCTAACCTCTGCACCTACTATAGTCCAGGCTATAGTATGAGCTCTGGCTGGATAA
GACTGTTGGTATCATAGTTGGGACTTGCGCCAAGCTCCGATACCCAGACTGTCAGATGAGACAATAATTC
CTCATGTCAACCGTAAGATACATTTACAGCGGAGTTTCTTTTGGGCCCTTTGTTGTTTCTCGCTACAGCA
AACTTTTACGCTGAAAAAAGTAGGGGCTTACGGCAGCAGCAGGGCAGCCCTGGAGCTGCTCGCTGGAGTCC
GATCATGTGATCTTCAACATGGCAGCCTCTTGGTTCCTACAGAAAGGGCGGAGCTCGGACTGGGGGG
CAGGCTCAGATTCAAGTTAAATTGTGATTGAGCTCGCAATTACAGACAGCTGACCATGGAAGCGAATGG
GTTGGGGTGAATCTCCAGAGCAGCGGCTGGCTAGCGGGGCTCTTAATTGAGTCTTCAACTCAGGA
TCTCTATCCCTTACTCCCTTTCCCCACCTGGAGAACCTCCCAACCTGAACCTCGTTAGCTGGATCCG
AATCCTAAACCATGGATTTTGGATGTTCTATCCAGGGCCTTAATTCAAGGGATGCCTCAGGATTTCC
AAACAGGATCTCTATTCTGGGACCATCAACTCTGATCCCTTTTATCCCCAGCGCTGGTATTCTTCAGC
CCCTGAACCGACCAAGTACATTTCCCGGTTTCTGAGGCTCACTAGTTCGAAGACCCCCAAATATCCTT
AGTGGGCTCTCATTCCTCCCCCAGTCCCTCTGTTGCTTCGAGCTTGAAGAGTAGAGACTAAGTGA

```

Figure 2.7 An example of GENBANK entry and a FASTA sequence.

genomes are also employed to improve the results. Moreover, other genome features such as regulatory regions, repeats, transcripts or sequencing markers are integrated with the sequence and the annotated genes. In Figure 2.8, a screenshot of the same gene displayed in the UCSC GENOME BROWSER and ENSEMBL is shown.

Data integration (integromics)

The biological information that can be now accessed in the databases has not been generated during a continuous process with several steps following an increasing order of complexity. On the contrary, different and discontinuous waves of genome-wide data have overlapped to form the current body of knowledge. The new high-throughput technologies that have arisen in the last decades have been the main catalyst conducting the progress. The first wave was the large-scale production of fragments of transcripts also named expressed sequence tags (ESTs). The second wave was originated by the sequencing of whole microbial organisms and was quickly followed by the achievement of the genomic sequence of many eukaryotic organisms including human. Simultaneously, microarrays and related technology have produced an overwhelming amount of expression data for which new analysis methods are still being designed (Searls, 2000).

In the near future, new waves of information are expected, such as the generation of maps of functional SNPs (see section 2.3), or the complex interaction networks produced by emerging systems biology (Kitano, 2002). Information technologies have adapted to the changing nature of the new data. With every explosion of new knowledge, previous procedures have been reused and others have been created from scratch to integrate the new type of data with the already existing information. The power of data integration arises not from the value of every separate kind of information but from the gain produced by the fusion of all of them. With the advent of more waves of knowledge, integromics will become absolutely essential to manage an amount of 'Omic' information that will exceed exabyte (10^{18} bytes) quantities (Searls, 2005).

Biological databases are essential resources used by biologists around the world. However, each one contains only a subset of biological knowledge. This specificity increases the complexity of finding the answer for the majority of questions. Thus, several databases must be explored in order to obtain the expected results. Cross-database queries require complex mechanisms of data integration that are often not implemented properly (Stein, 2003).

For instance, the name of biological objects such as genes in the genomic browsers of several species (e.g. Rad24, rad24 or RAD24) or the definition of simple entities such as the gene concept (considering only transcript or transcript and regulatory region) can be a source of disagreement. Consequently, the role of ontologies to facilitate data integration must not be neglected. The popular GENE ONTOLOGY (The Gene Ontology Consortium, 2000, see Web Glossary, page 242) establishes a taxonomy of controlled vocabulary that is used by most genome annotation projects to uniformly annotate the function of genes.

There are several ways in which databases developers have tried to integrate databases:

- Link integration. Hypertext links are used to jump from one database to another. Although it is the most popular solution, it has two severe drawbacks: links are vulnerable to name ambiguities and their updating is laborious.

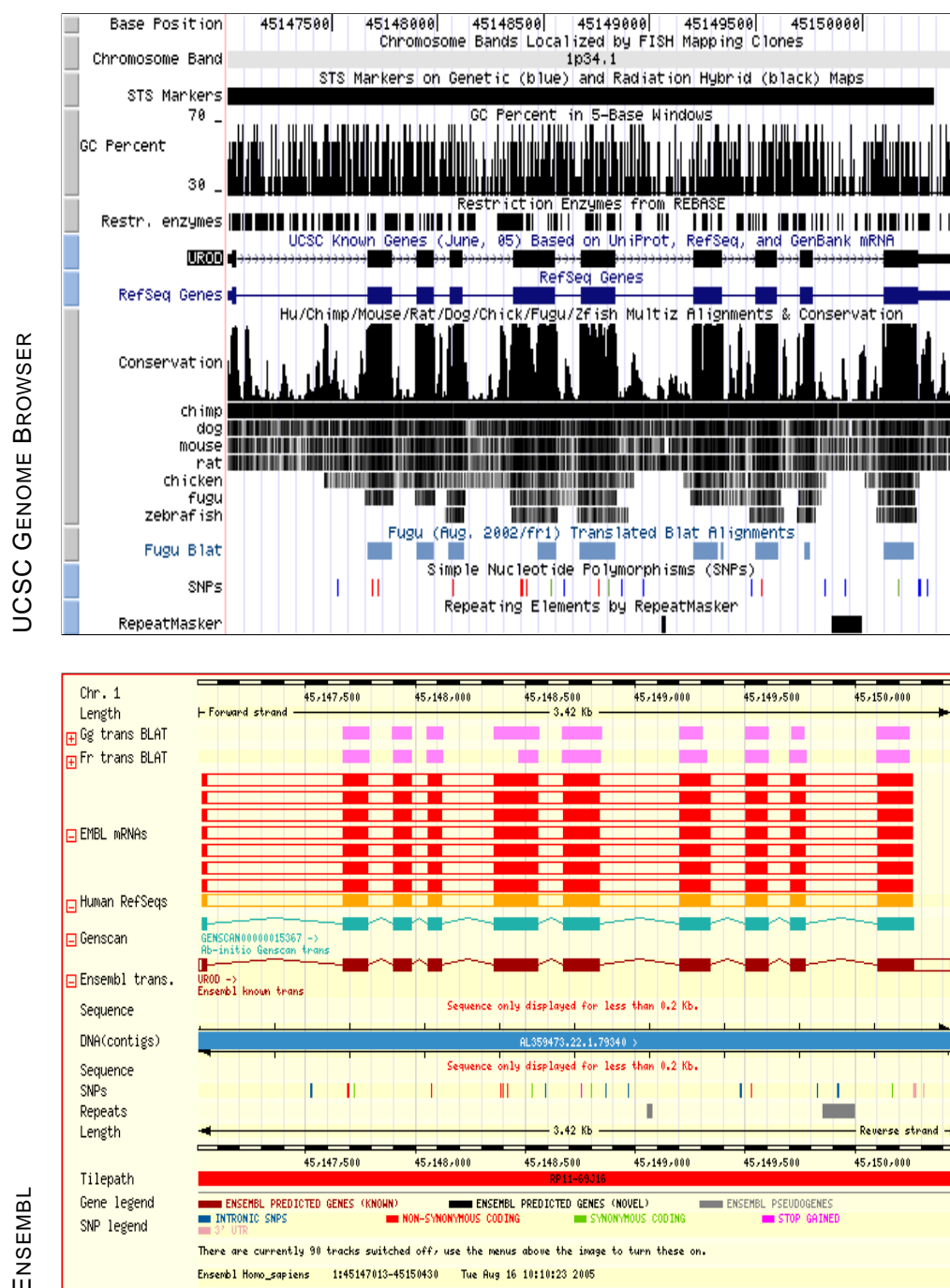


Figure 2.8 The human URO-D gene in the UCSC GENOME BROWSER and ENSEMBL.

- View integration. An environment around the databases is built to create the illusion of a unique resource formed by different sources of data with specific data drivers to retrieve the information. The complexity of such a design is the main disadvantage of this strategy.
- Data warehousing. Merge all of the databases into a single database. Due to the continuous updating of biological databases and the impossibility of reusing the software from one release to the next, this approach is unfeasible in practice.

2.3 The post-genomic era

New forms of investigation

The availability of many genomes and the improvement of very large-scale gene expression experiments have substantially modified the form in which current research is focused (Searls, 2005). The classical hypothesis-driven research paradigm, in which a specific proposition is addressed over a set of targets, is progressively being substituted with data-driven investigations, in which high-throughput explorations are performed typically over the whole collection of genes of an organism to detect previously unknown relationships. Data-diving excursions have several risks derived from their massive exploration. Correct normalization and replication of the results is extremely difficult. In addition, there is usually a high probability of finding pure artefactual relations due to the low signal/noise ratio observed in such experiments (Searls, 2005).

Much effort must be invested to make bioinformatics become part of the wet-dry cycles of research (Searls, 2000). Such discovery processes occur whenever a computational method is linked to a biological one, such that predictions from the former can be tested at the bench, within a feedback strategy. Once the computational candidates have been delivered, they should be monitored during the experimental pipeline, using such results to refine the original computational model (Searls, 2000).

Genomics and health

Virtually every human illness has a hereditary component (Collins and McKusick, 2001). The characterization of the genetic determinants of disease would provide remarkable opportunities for clinical medicine. Current clinical practice is still based on phenotypic criteria to define most diseases rather than studying the underlying mechanisms. Obtaining the sequence of the human genome is only the end of the beginning (Collins and McKusick, 2001). Among the grand challenges to achieve after the sequence of many genomes is available is the development of strategies for identifying the genetic contributions to disease and the gene variants that promote good health and resistance to disease (Collins et al., 2003). Progress is slow but evidence suggests that while public health and antibiotics have played the major roles in the past 50 years, the next 50 are likely to belong to genetics and molecular medicine (Bell, 2003).

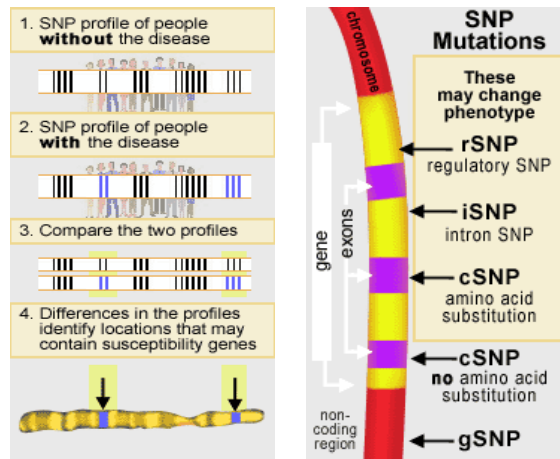


Figure 2.9 Using SNPs to locate susceptibility genes. (Left) SNP profiling of two groups of people. (Right) Categories of SNPs according to their location. Adapted from GSK report: Genes and diseases.

Simple changes in our genes can lead to disease. Single gene mutations, which are already commonly used in diagnostic practice (genetic and disease markers), cause approximately 6,000 inherited diseases also known as monogenic diseases. Disorders like cystic fibrosis, anemia or hemophilia affect millions of people worldwide. For more common diseases such as heart disease, diabetes, or Alzheimer's disease, the interplay of multiple genes and multiple non-genetic factors (environment effects) that contribute to disease susceptibility is still being characterized (GSK report: Genes and diseases, see Web Glossary, page 242, NHGRI/NIH report: Genetics, the Future of Medicine, see Web Glossary, page 243).

For example, loss of control in the growth mechanisms of cells results in cancer. The transformation of a normal cell into a cancerous one is caused by molecular changes that underly growth-signal independence, insensitivity to anti-growth signals, evasion of immunosurveillance, apoptosis evasion, unlimited replicative potential, tissue invasion and metastasis. These molecular changes involving several genes can be produced by certain events that alter the genome such as point mutations, gene amplifications and deletions, and chromosomal translocations. The intimate relationship between cancer and genome sequencing projects has originated the recent launch of several cancer genome projects (Strausberg et al., 2003).

Pharmacogenomics

Before the end of this century, shortly after a person is born, her genotype will be saved at her physician's office to record the presence or absence of specific variations known to be relevant for assessing disease susceptibility and prediction response to drug types. Biomolecular profiling throughout her life will complement this information to provide recommendations about life-style or diet and to detect early stages of a disease. This future scenario in which personalized medicine and therapy are present in our lives to increase the quality of life and

life-span is not unrealistic (Sander, 2000).

In 1998, adverse drug reactions produced over 100,000 deaths in the United States, being one of the leading causes of hospitalization and death. The one-size-fits-all formula typically works for only 60% of the population at best. The way a person responds a drug (positively or negatively) is a complex trait influenced by many different genes. Pharmacogenomics³ is the science that examines the gene variations that dictate drug response and explores how to use them to predict whether a patient will have a good reaction, a bad reaction or no reaction to a given drug (Evans and Relling, 1999, NCBI report: pharmacogenomics, see Web Glossary, page 243).

First studies focused on the broadest categories of inheritance: ethnicity, geography, language and race. Several SNPs mapping projects are working to provide a catalogue of observed one-letter differences between individuals in a population. SNPs are present throughout the human genome with an average frequency of 1 per 1,000 base pairs. Their relatively even distribution make them valuable as genetic markers. To be helpful, the polymorphism must be shared by at least 1% of the population tested, thus becoming a shared SNP. Mutations are less common differences, occurring in a smaller proportion.

With these SNP maps, genetic profile comparison of patients who may suffer from serious side effects and those that may not, might be useful to detect one or more SNPs that differ between both groups. Careful examination of the small area of the genome where the differences are found will classify them into functional and non-functional SNPs (see Figure 2.9). For instance, SNPs found in protein coding regions (cSNPs) would be good candidates to elaborate a hypothetic explanation of the observed drug response as long as they produce a change in the translated amino acid sequence (non synonymous changes).

The haplotype is the set of closely related genes (alleles) that tend to be inherited together as a single unit. The International HapMap Project is currently in charge of developing the haplotype map of the human genome (The International HapMap Consortium, 2003). The official repository of SNPs mined by this project is the NCBI dbSNP database (see Web Glossary, page 241) that contains information for other genomes as well. SNP annotation is also integrated in the genomic browsers explained in Section 2.2. For further information about sequence polymorphisms, see Mullikin and Sherry (2005).

Bibliography

- M.D. Adams, S.E. Celniker, R.A. Holt, C.A. Evans, J.D. Gocayne, P.G. Amanatides, S.E. Scherer, P.W. Li, R.A. Hoskins, R.F. Galle, R.A. George, S.E. Lewis, S. Richards, M. Ashburner, S.N. Henderson, et al. The genome sequence of *Drosophila melanogaster*. *Science*, 287:2185–95, 2000.
- B. Alberts, D. Bray, J. Lewis, M. Raff, K. Roberts, and J. D. Watson. *Molecular biology of the cell*. Garland publishing, third edition, 1994. ISBN 0-8153-1620-8.
- A.D. Baxeavanis and B.F.F. Ouellette. *Bioinformatics: A Practical Guide to the Analysis of Genes and Proteins*. John Wiley & Sons Inc., New York, USA, third edition, 2005. ISBN 0-471-47878-4.
- J.I. Bell. The double helix in clinical practice. *Nature*, 421:414–416, 2003.

³The related term pharmacogenetics appeared in the 1950s describing the study of inherited genetic variation in drug metabolism and response.

- D.A. Benson, I. Karsch-Mizrachi, D.J. Lipman, and J. Ostella and D.L. Wheeler. Genbank: update. *Nucleic Acids Research*, 32:D23–D26, 2003.
- E. Birney, D. Andrews, P. Bevan, M. Caccamo, G. Cameron, Y. Chen, L. Clarke, G. Coates, T. Cox, J. Cuff, V. Curwen, T. Cutts, T. Down, R. Durbin, E. Eyraes, et al. ENSEMBL 2004. *Nucleic Acids Res*, 32:D468–70, 2004.
- E. Blanco and R. Guigó. *Bioinformatics: A Practical Guide to the Analysis of Genes and Proteins.*, chapter “Predictive Methods using DNA Sequences”, pages 115–142. John Wiley & Sons Inc., New York, USA, 2005. ISBN 0-471-47878-4.
- M. Bogusky. Bioinformatics - a new era. *Trends in genetics (trends guide to bioinformatics)*, pages 1–3, 1998.
- T.A. Brown. *Genomes*. BIOS Scientific Publishers, Oxford, UK, second edition, 2002. ISBN 1-85996-029-4.
- J.M. Claverie. From bioinformatics to computational biology. *Genome Research*, 10:1277–1279, 2000.
- F.S. Collins, E.D. Green, A.E. Guttmacher, and M.S. Guyer. A vision for the future of genomics research. *Nature*, 422:1–13, 2003.
- F.S. Collins and V.A. McKusick. Implications of the human genome project for medical science. *Journal of the American Medical Association*, 285:540–544, 2001.
- M.O. Dayhoff, R.V. Eck, M.A. Chang, and M.R. Sochard. *Atlas of protein sequence and structure*, volume 1. National Biomedical Research Foundation, Silver Spring, Maryland, 1965.
- W.E. Evans and M.V. Relling. Pharmacogenomics: translating functional genomics into rational therapeutics. *Science*, 286:487–, 1999.
- A. Goffeau, B.G. Barrell, H. Bussey, R.W. Davis, BB. Dujon, H. Feldmann, F. Galibert, J.D. Hoheisel, C. Jacq, and M. Johnston. Life with 6000 genes. *Science*, 274:546, 563–567, 1996.
- N. Goodman. Biological data becomes computer literate: new advances in bioinformatics. *Current Opinion in Biotechnology*, 13:68–71, 2002.
- L.W. Hillier, W. Miller, E. Birney, W. Warren, R.C. Hardison, C.P. Ponting, P. Bork, D.W. Burt, M.A. Groenen, M.E. Delany, J.B. Dodgson, G. Fingerprint Map Sequence, Assembly, A.T. Chinwalla, PF Clifton, S.W. Clifton, and others (International Chicken Genome Sequencing Consortium, ICGSC). Sequence and comparative analysis of the chicken genome provide unique perspectives on vertebrate evolution. *Nature*, 432:695–716, 2004.
- International Human Genome Sequencing Consortium, IHGSC. Finishing the euchromatic sequence of the human genome. *Nature*, 431:931–45, 2004.
- D. Karolchik, R. Baertsch, M. Diekhans, T.S. Furey, A. Hinrichs, Y.T. Lu, K.M. Roskin, M. Schwartz, C.W. Sugnet, D.J. Thomas, R.J. Weber, D. Haussler, and W.J. Kent. The UCSC GENOME BROWSER database. *Nucleic Acids Res*, 31:51–54, 2003.
- H. Kitano. Systems biology: a brief overview. *Science*, 295:1662–1664, 2002.
- T. Kulikova, P. Aldebert, N. Althorpe, W. Baker, K. Bates, P. Browne, A. van den Broek, G. Cochrane, K. Duggan, R. Eberhardt, et al. The EMBL nucleotide sequence database. *Nucleic Acids Research*, 32: D27–D30, 2004.

- E.S. Lander, L.M. Linton, B. Birren, C. Nusbaum, M.C. Zody, J. Baldwin, K. Devon, K. Dewar, M. Doyle, W. FitzHugh, R. Funke, D. Gage, K. Harris, A. Heaford, J. Howland, and others (International Human Genome Sequencing Consortium, IHGSC). Initial sequencing and analysis of the human genome. *Nature*, 409:860–921, 2001.
- E.S. Lander and R.A. Weinberg. Genomics: journey to the center of biology. *Science*, 287:1777–1782, 2000.
- D.W. Mount. *Bioinformatics: sequence and genome analysis*. Cold Spring Harbor Laboratory Press, first edition, 2001. ISBN 0-87969-608-7.
- J.C. Mullikin and S.T. Sherry. *Bioinformatics: A Practical Guide to the Analysis of Genes and Proteins*, chapter “Sequence polymorphisms”, pages 171–193. John Wiley & Sons Inc., New York, USA, 2005. ISBN 0-471-47878-4.
- W.R. Pearson. Training for bioinformatics and computational biology. *Bioinformatics*, 17:761–762, 2001.
- K.D. Pruitt, T. Tatusova, and D.R. Maglott. NCBI Reference Sequence (REFSEQ): a curated non-redundant sequence database of genomes, transcripts and proteins. *Nucleic Acids Research*, 33 Database Issue:D501–D504, 2005.
- C. Sander. Genomic medicine and the future of health care. *Science*, 287:1977–1978, 2000.
- D.B. Searls. Using bioinformatics in gene and drug discovery. *Drug Discovery Today*, 5:135–143, 2000.
- D.B. Searls. Data integration: challenges for drug discovery. *Nature Reviews Drug Discovery*, 4:45–58, 2005.
- L.D Stein. Integrating biological databases. *Nature Reviews Genetics*, 4:337–345, 2003.
- R.L. Strausberg, A.J.G. Simpson, and R. Wooster. Sequence-based cancer genomics: progress, lessons and opportunities. *Nature Reviews Genetics*, 4:409–418, 2003.
- The Gene Ontology Consortium. Gene ontology: tool for the unification of biology. *Nature Genetics*, 25:25–29, 2000.
- The International HapMap Consortium. The international hapmap project. *Nature*, 426:789–796, 2003.
- J.C. Venter, M.D. Adams, E.W. Myers, P.W. Li, R.J. Mural, G.G. Sutton, H.O. Smith, M. Yandell, C.A. Evans, R.A. Holt, J.D. Gocayne, P. Amanatides, R.M. Ballew, D.H. Huson, J.R. Wortman, et al. The sequence of the human genome. *Science*, 291:1304–1351, 2001.
- R.H. Waterston, K. Lindblad-Toh, E. Birney, J. Rogers, J.F. Abril, P. Agarwal, R. Agarwala, R. Ainscough, M. Alexandersson, P. An, S.E. Antonarakis, J. Attwood, R. Baertsch, J. Bailey, K. Barlow, and others (International Mouse Genome Sequencing Consortium, IMGSC). Initial sequencing and comparative analysis of the mouse genome. *Nature*, 420:520–562, 2002.
- D.L. Wheeler, T. Barrett, D.A. Benson, S.H. Bryant, K. Canese, D.M. Church, M. DiCuccio, R. Edgar, S. Federhen, W. Helmberg, D.L. Kenton, O. Khovayko, D.J. Lipman, T.L. Madden, D.R. Maglott, et al. Database resources of the national center for biotechnology information. *Nucleic Acids Research*, 33 Database Issue:D39–45, 2005.

PART II

State of the Art

Chapter 3

The golden age of sequence analysis

Summary

This chapter aims to be a historical survey of the sequence comparisons algorithms analyzing the most relevant solutions. The algorithms that represented innovative changes in the field are described in detail, covering the concepts of global, local and multiple alignment of sequences. In addition, the theoretical framework of the map alignment problems necessary to understand the rest of work presented in this thesis is also formalized here.

3.1	Foundations of sequence comparison	32
3.2	Alphabets, sequences and alignments	35
3.3	An anthology of algorithms for global alignments	40
3.4	A short overview on local sequence alignment	61
3.5	A short overview on multiple sequence alignment	69
3.6	Map alignments	72

3.1 Foundations of sequence comparison

THE TOPIC OF BIOSEQUENCE COMPARISON has a rich history dating back over 40 years. It is certainly very difficult to trace a line in some moments to establish the order in which every new development was presented because of the enormous body of publications that have contributed substantially to improve this field. Several general reviews have been used to reconstruct the history of biological sequence comparisons [Mount \(2001\)](#); [Myers \(1991\)](#); [Ouzounis and Valencia \(2003\)](#); [Sankoff and Kruskal \(1983\)](#); [Meidanis and Setubal \(1997\)](#); [Waterman \(1984b\)](#).

Molecular evolution began to be studied in the 1960s when a few protein sequences were available, being published into the protein sequence atlas ([Dayhoff et al., 1965](#)). Soon, pioneering analysis appeared to infer the evolutionary relationships from these sequences, depicted as distances in phylogenetic trees ([Fitch and Margoliash, 1967](#)).

Outside the molecular biology, other significant advances in mathematics and in the emerging discipline of computer science contributed decisively to the current state of the art. For instance, it is impossible to understand the history of modern sequence alignment without mentioning the birth of a new technique in the 1950s to solve multistage decision process problems called dynamic programming ([Bellman, 1957](#); [Dreyfus, 2002](#)). A problem is solved by dynamic programming if the answer can be efficiently determined by computing a table of optimal answers to progressively larger subproblems. The principle of optimality requires that the optimal answer to a given subproblem is expressible in terms of optimal answers to smaller subproblems. During all this time, despite innumerable optimal and heuristic approaches have been proposed to obtain the best alignments between two sequences with the minimum cost, dynamic programming is still the most stable technique to solve the original problem and many of its variations.

Another key concept is the definition of several metrics of distance between sequences in the coding theory field. Since noise in a transmission channel introduces errors into the signal reception, several mechanisms were developed for detection and correction of such errors. The Hamming distance, defined as the number of positions in which two sequences differ, was oriented to detect only substitutions ([Hamming, 1950](#)). Next, [Levenshtein \(1966\)](#) presented the edit distance, which was the earliest known use of a distance function that is appropriate to detect insertions and deletions of symbols in the original message.

It is not clear when the basic dynamic programming algorithm for molecular sequence comparison first appeared. It was probably rediscovered many times in different contexts. The well-known paper by [Needleman and Wunsch \(1970\)](#) who presented an algorithm for maximizing the number of matches minus the number of insertions and deletions is generally considered to be the first important contribution. Although no complexity analysis was provided, the original [Needleman and Wunsch](#) algorithm measured the homology between two sequences in a $O(n^3)$ time.

A more rigorous approach with solid mathematical foundations arised from the problem of computing the distance between two sequences ([Ulam, 1972](#); [Beyer et al., 1985](#)). [Sellers \(1974\)](#) presented a dynamic algorithm based on the Levenshtein metric distance. Though less flexible for future variations of the problem, this new approach fitted better with the perspective of evolutionary distance analysis developed earlier. Under the realistic

assumption that both sequences have n nucleotides, the Sellers algorithm have computation time proportional to $O(n^2)$. A comprehensive study of equivalence between similarity and distance was presented in Smith et al. (1981).

Within the field of computer science, sequence comparison appeared in simpler incarnations of the molecular biology problems, for comparing the contents of files or correcting the spelling of words. For example, the longest common subsequence problem (LCS) consists on finding an alignment that maximizes the number of identical aligned pairs between two sequences (see Apostolico and Guerra (1987) for a review). Interestingly for long sequences, Hirschberg (1975) applied the divide and conquer strategy to solve the LCS problem in $O(2n^2)$ time with a linear space cost instead of the established quadratic cost. Myers and Miller (1988) generalized this technique to align two sequences using $O(n)$ space.

Nonetheless, the treatment of gaps was still biologically unrealistic as a deletion of n symbols and n deletions of one symbol were punished indistinctly. Waterman et al. (1976) accommodated the same algorithm to deal with multiple deletions and insertions, introducing the concept of general gap penalty functions. Gotoh (1982) reduced the asymptotic cost from $O(n^3)$ to $O(n^2)$, under the application of the affine gap penalty functions in which there was an initial penalty for opening a gap and an additional minor penalty for extending an existent one. Apart from general and affine gap functions, Waterman (1984b) introduced the concept of concave gap function in which the cost of extending an existent gap grows with the logarithm of the length of the gap as a continuous curve. Later, Eppstein et al. (1988) and Miller and Myers (1988) independently arrived at $O(n^2 \log n)$ solutions of the problem.

DNA and protein sequences are the result of an evolutionary process that tend to preserve those parts that are key to perform a function, permitting variation in the rest. Thus a global comparison can easily produce a very poor alignment of two sequences that have some parts in common while others are completely free of conservation. Smith and Waterman (1981b) introduced the concept of local alignment with a simple variation in the basic global similarity algorithm without increasing its cost. Under the premise of a negative gap penalty, reported alignments are regions of high similarity with a positive score within. Sellers (1984) tried to export the same concept to the distance metric. Only, those paths in the matrix whose density of mismatches was below a certain threshold were reported.

Thousands of genomic and proteomic sequences, that is millions of nucleotides and amino acids, are rapidly being accumulated in the biological databases. However, searching a database with a query sequence for similarities to other sequences using the optimal algorithms enumerated above is clearly unfeasible when this simple operation involves thousands of comparisons between two sequences. To overcome this problem, a new family of heuristic procedures that produce nearly correct answers in a simple and cheaper fashion was designed. The most popular representatives of these are the program FASTA (Pearson and Lipman, 1988) and the program BLAST (Altschul et al., 1990). The FASTA heuristic is based on identifying the identities between two sequences (diagonals in the matrix) and then applying some more expensive procedures only on those subalignments. BLAST processing relies on first, detecting ungapped segment pairs of high score and then, extending them from both ends until a threshold value is reached.

A collateral effect of producing hundreds of alignments was the concern about the quality of a given alignment between two sequences. The significance of a local alignment score can be tested by comparing with the distribution of scores expected by aligning two random

sequences with the same length and composition (Karlin and Altschul, 1990). These random sequence alignment scores follow a distribution called the extreme value distribution (also known as the Gumbel distribution), which is similar to a normal distribution but with a positively skewed tail in the higher score (Gumbel, 1962). Less interest has traditionally been focused on global comparisons because of a global alignment is always produced by definition even between random or unrelated sequences, growing the score proportionally to the length of them.

In attempt to distinguish more distant relationships, the implementation of comparisons for more than two sequences is the logical evolution to locate elements with function that are conserved for instance in several homologous sequences. Waterman et al. (1976) naturally extended the basic dynamic programming recurrence for k sequences, with an exponential cost $O(n^k)$. As this approach is generally impractical, some heuristics appeared to solve the problem with a minor cost. The most popular of them is the hierarchical or clustering method called progressive alignment that first takes $O(k^2 n^2)$ to perform all pairwise alignments and second, produce a multiple alignment following a guide tree to merge these alignments (Feng and Doolittle, 1987). The program CLUSTALW (Thompson et al., 1994) combines this strategy with different weighting schemes according to the progression in the distances tree. Previously, Carrillo and Lipmann (1988) developed another method based on identifying the projections of the pairwise alignments that can form the multiple alignment. Moreover, hidden Markov models have been used to produce multiple alignments of a family of sequences to which more members can be dynamically be added (profile HMMs, see Durbin et al., 1998).

Pattern discovery and local multiple sequence alignment have been very closely related problems (Brazma et al., 1998). For instance, a conserved pattern or a block of ungapped common motifs in a set of sequences defines a local multiple alignment. In any case, the problem is even more difficult than pure global alignment and optimal approaches were discarded beforehand. Some heuristic approaches have been proposed to circumvent the complexity. Iterative methods do not necessarily find the best pattern, but may converge to a local maximum. Gibbs sampling (Lawrence et al., 1993) and expectation maximization (Bailey and Elkan, 1994) are successful examples of these stochastic techniques.

Some pattern recognition problems are too complex or too ambiguous to be expressed as a simple pattern matching operations over a sequence. In these cases, a richer environment over the basic sequences is needed to describe the comparison of such elements (Knight and Myers, 1995). For example, for most sequence comparison problems there is a corresponding map comparison algorithm. Map comparisons were introduced to model the alignment of restriction enzyme maps. These were used in the construction of physical maps prior to genome sequencing projects. The basic definition of the problem by Waterman et al. (1984) contained an $O(n^4)$ time cost algorithm although it was noticed the dynamic programming matrix was very sparse. Later, Myers and Huang (1992) improved the time efficiency by using an analytical approach that reduced the cost to $O(n^2 \log n)$. Additional refinements of the problem produced new algorithms to deal with map data errors (Huang and Waterman, 1992) or to align specifically short maps to longer ones (Miller et al., 1990).

Not only analytical approaches have been employed for comparing sequences. Dot matrix comparisons, also known as dotplots, are visual comparisons that can be useful to conduct afterwards a deeper research with dynamic programming algorithms only on those conserved regions (Gibbs and McIntyre, 1970). Sequence logos are graphs that illustrate the amount of information in each column of an alignment or motif (Schneider and Stephens,

1990).

Sequence comparison algorithms that were developed to solve biological problems have been recreated and applied in other scientific fields (Sankoff and Kruskal, 1983). For instance, applications can be found in geology (stratigraphic sequences), in dendrochronology (time dating based on tree rings), or in bird song recognition (animal communication).

3.2 Alphabets, sequences and alignments

Biological significance of sequence comparison

Gene evolution is thought to occur by gene duplication, creating two tandem copies of the gene in a given ancestor species. In rare cases, new mutations in one of the copies can provide an advantageous change in function. The two copies then evolve along separate pathways. At a certain evolutionary point, a speciation event gives rise to two separate branches (two new species) of the tandem gene preserving a similar sequence due to the single gene ancestor (see Figure 3.1). The four copies of the original gene are said to be homologous: the two corresponding units of the tandem gene in each species are orthologous while the two units of each tandem gene in the same species are paralogous. Molecular evolution events include substitutions of one nucleotide or amino acid for another as well as insertions and deletions (indels) of others. More complex genetic rearrangements such as inversions, transpositions, translocations or duplications can shuffle larger parts of the genes or of the proteins, producing chimeric products in which some regions are homologous and others are not (Mount, 2001).

Sequence comparison consists of finding which parts of the sequences are alike and which parts differ. This operation is extremely useful for discovering functional, structural and evolutionary information in biological sequences. If two sequences from different organisms are similar, there may have been a common ancestor sequence that would make these sequences to be homologous. Phylogenetic analyses are usually conducted starting from multiple sequence comparisons, and then producing hierarchical trees that would explain the evolution of the species.

Alphabets and sequences

A finite alphabet is a set of symbols or characters. For instance, the four-letter DNA and RNA alphabets are defined as:

$$\Sigma_{\text{DNA}} = \{A, C, G, T\} \text{ and } \Sigma_{\text{RNA}} = \{A, C, G, U\}.$$

To support some degree of variation or ambiguity in a symbol, the IUPAC extended genetic alphabet of 15 elements allows for special symbols possessing multiple letters (see Table 3.1). The single-letter amino acid alphabet contains 20 elements¹ from which all proteins are built (see Table 3.2).

¹Nowadays, new amino acids are still being unveiled such as Selenocysteine.

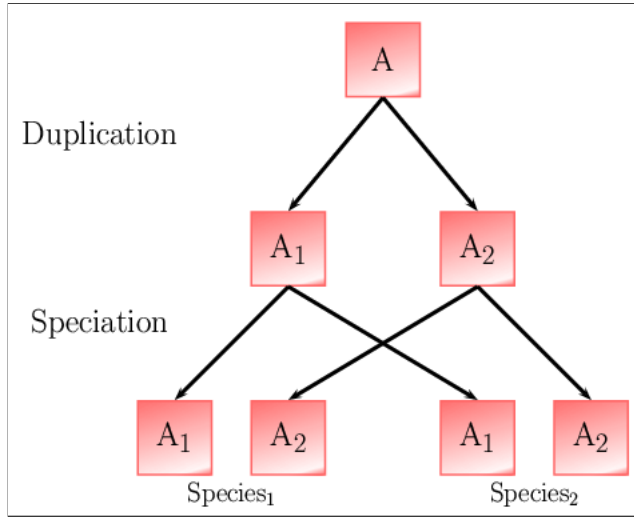


Figure 3.1 Gene evolution events.

Σ^* denotes the set of all finite sequences of characters from Σ including the empty sequence λ . A generic sequence S of length $|S| = n$ symbols over a finite alphabet Σ is defined as:

$$S = s_1 s_2 \dots s_n \text{ where } \forall i : 1 \leq i \leq n : s_i \in \Sigma.$$

A subsequence of S between positions i and j of S is the contiguous series of elements between both positions². If $i = 1$, the subsequence is called a prefix of S . If $j = n$, the subsequence is a suffix:

$$S_{i,j} = s_i \dots s_j \text{ where } 1 \leq i \leq j \leq n \text{ and } \forall k : i \leq k \leq j : s_k \in S.$$

Sequence alignments

Given two sequences $A = a_1 a_2 \dots a_m$ and $B = b_1 b_2 \dots b_n$ in a finite alphabet Σ , a sequence alignment of A and B is a correspondence C between the symbols from the two sequences

$$C(A, B) = \{(a_{i_1}, b_{j_1}), (a_{i_2}, b_{j_2}), \dots, (a_{i_T}, b_{j_T})\} \text{ where } 1 \leq i_1 \leq i_2 \leq \dots \leq i_T \leq m, 1 \leq j_1 \leq j_2 \leq \dots \leq j_T \leq n$$

such that:

²As defined in computer science, subsequences are subsets of characters of S possibly not contiguous but arranged in their original relative order.

- ① Each a_k (or b_l) not appearing in the subsequence $a_{i_1} \dots a_{i_T}$ (or $b_{j_1} \dots b_{j_T}$) is considered to be an insertion in the other sequence (or a deletion in this one).
- ② If the pair $(a_i, b_j) \in C \Rightarrow \forall k : b_k \in B \wedge k \neq j : (a_i, b_k) \notin C$ (one symbol only matches another symbol at most).
- ③ If the pairs $(a_i, b_j), (a_k, b_l) \in C$ and $i < k \Rightarrow j < l$ (no inversions are allowed).

For example, a possible alignment of the sequence $A = \text{AAGTTC}$ and the sequence $B = \text{AGCCC}$ is

A =	A	A	G	T	T	C
B =	A	-	G	C	C	C.

This alignment represents a certain hypothesis about the evolution of the two sequences (Waterman et al., 1990): three of the nucleotides have not changed since the common ancestor of A and B (matches), there have been at least two substitutions (mismatches), and one nucleotide has been either inserted or deleted (a gap), which is denoted with the symbol “-”.

If we adopt a scoring function that assigns a given value to a match, a mismatch and a gap, every column of the alignment will receive a score and the total score of the alignment will be the sum of the values assigned to its columns. The best alignment will be the one that optimizes the total score. In the literature, two different types of measures have been devised to construct such a scoring function : similarity and distance (see Smith and Waterman (1981a) for a review).

Sequence similarity

Similarity is a measure of how alike two sequences are. An alignment is scored by rewarding the identities and in less degree, the substitutions, and punishing the gaps.

Let (a_i, b_j) be a match (or a mismatch) of type k with a weight α_k and let w_l be the weight associated to a gap of length l . Then, the similarity of an alignment C of A and B with λ_x matches of type x and Δ_y gaps of length y is

$$S(C) = \sum_x \lambda_x \alpha_x - \sum_y \Delta_y w_y. \quad (3.1)$$

The best alignment is the one that maximizes the similarity between A and B. The similarity can increase and decrease during the computation of an alignment score from $-\infty$ to ∞ (from dissimilarity to similarity, where 0 means absence of any type of similarity).

SYMBOL	LETTERS	ORIGIN OF DESIGNATION
A	A	Adenine
C	C	Cytosine
G	G	Guanine
T	T	Thymine
R	A or G	puRine
Y	C or T	pYrimidine
M	A or C	aMino
K	G or T	Keto
S	C or G	Strong interaction (3 H-bonds)
W	A or T	Weak interaction (2 H-bonds)
B	C or G or T	not A, B follows A
D	A or G or T	not C, D follows C
H	A or C or T	not G, H follows G
V	A or C or G	not T (not U), V follows U
N	A or C or G or T	aNy

Table 3.1 The IUPAC extended genetic alphabet.

Sequence distance

Distance (also called edit distance) is the minimal number of changes (indels and substitutions) needed to transform one sequence into another. An alignment is scored by charging a cost to each difference in the aligned sequences (0 for exact matches).

Let (a_i, b_j) be a match (or a mismatch) of type k with a weight β_k and let w_l be the weight associated to a gap of length l . Then, the distance of an alignment C of A and B with λ_x matches of type x and Δ_y gaps of length y is

$$D(C) = \sum_x \lambda_x \beta_x + \sum_y \Delta_y w_y.$$

(3.2)

The best alignment is the one that minimizes the distance between A and B . Distance metric provides a more biologically natural way to compare sequences, estimating the evolutionary time that has elapsed since the sequences diverged from a common ancestor. The distance value can only increase during the computation of an alignment score, starting with a value of 0.

The number of alignments

The number of possible alignments between two sequences of n symbols can be computed with the following function (Waterman, 1984b, 1995):

$$g(n) \sim \frac{2^{2n}}{4\sqrt{n\pi}}.$$

(3.3)

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

Table 3.2 The amino acid alphabet.

For two sequences of 1,000 nucleotides, $g(n) > 10^{600}$. As direct examination of all these alignments is in practice impossible, computational approaches are therefore essential to calculate the optimal alignment without exploring all of the combinations.

Classes of sequence alignments

According to the type of comparison that must be performed between sequences, sequence alignments can be classified as (Mount, 2001):

- ➔ Global alignments: the entire sequence length must be aligned to include the maximum number of matches. Sequences that are quite similar and approximately have the same length are good candidates for global alignment.

L	G	P	S	S	K	Q	T	G	K	G	S	–	S	R	I	W	D	N
L	N	–	I	T	K	S	A	G	K	G	A	I	M	R	L	G	D	A

- ➔ Local alignments: only the stretches of the sequences with the highest density of matches are aligned. Sequences that differ in length or that only share certain regions are suitable candidates for local alignment.

-	-	-	-	-	-	-	-	T	G	K	G	-	-	-	-	-	-	-	-
-	-	-	-	-	-	-	-	A	G	K	G	-	-	-	-	-	-	-	-

When the number of sequences is two, such alignments receive the name of pairwise alignments as the examples above. If the number of input sequences is higher, they are called multiple sequence alignments:

- Global multiple alignments: the whole set of sequences is aligned at their entire length. Simply known as multiple alignments, they are the starting point for evolutionary modeling. Each column of the alignment is examined and significant changes observed in this position collaborate in the construction of a phylogenetic tree.

L	G	P	S	S	K	Q	T	G	K	G	S	-	S	R	I	W	D	N
L	N	-	I	T	K	S	A	G	K	G	A	I	M	R	L	G	D	A
L	N	-	K	Q	Q	S	A	G	K	C	A	I	M	-	L	G	D	A

- Local multiple alignments: they are equivalent to searching a pattern conserved in a set of sequences. Rather than be defined as a form of alignment, it is conceptually considered a pattern discovery problem.

-	-	-	-	-	-	-	-	T	G	K	G	-	-	-	-	-	-	-
-	-	-	-	-	-	-	-	A	G	K	G	-	-	-	-	-	-	-
-	-	-	-	-	-	-	-	A	G	K	C	-	-	-	-	-	-	-



3.3 An anthology of algorithms for global alignments

This section aims to be a catalogue of different approaches to solve the global pairwise alignment which was the first problem introduced in the field of sequence comparisons. Naturally, the extension to the multiple alignment of sequences has been also treated although optimal solutions were discarded because of their expensive time and space costs. Different heuristics to cope with multiple alignment are explained in detail in Section 3.5.

The Needleman and Wunsch algorithm (1970)

For the authors, the similarity or maximum match value between two proteins depends on the largest number of amino acids from the first protein that can be matched with those of the second one allowing possible interruptions in either sequence.

	A	B	C	N	J	R	O	C	L	C	R	P	M
A	1												
J					1								
C			1					1		1			
J					1								
N				1									
R						1	4	3	3	2	2	0	0
C	3	3	4	3	3	3	3	4	3	3	1	0	0
K	3	3	3	3	3	3	3	3	3	2	1	0	0
C	2	2	3	2	2	2	2	3	2	3	1	0	0
R	2	1	1	1	1	2	1	1	1	1	2	0	0
B	1	2	1	1	1	1	1	1	1	1	1	0	0
P	0	0	0	0	0	0	0	0	0	0	0	1	0

Figure 3.2 The maximum-match operation for necessary pathways. The cell (R, R, 1) corresponds to the current $M(i, j)$. Adapted from Needleman and Wunsch (1970).

Each pair of amino acids from each sequence is the smallest unit of significance. All possible pair combinations are represented in a two-dimensional matrix M . The pathways through the cells of the matrix are representations of every possible comparison of the two sequences. If a given value is assigned to each identity and mismatch, the maximum match between two sequences A and B is then the largest number that would result from the sum of the cell values of every pathway.

The original Needleman and Wunsch algorithm is actually a description of a method to systematically count the number of identities (denoted as 1's in the simplest formulation) between both sequences. No complexity analysis was provided although a careful analysis determines the cost of the process is cubic (see next section). In addition, the authors implicitly suggested the extension of the method to allow multiple comparison of several proteins or the inclusion of a gap penalty factor as a function depending on the length of the gap.

The assessment of the significance of a given match value was also proposed: first, two sets of random sequences with the same composition of the original proteins are constructed; second, the maximum-match between pairs of these sequences is determined several times and is compared to the value obtained between real proteins; third, the match between one of the real proteins and several of the random sequences is also computed and evaluated. In all of the cases, the difference between the real match and the artificial ones should be statistically significant. Otherwise, the match between both proteins would be explained in part only by a similar composition.

Formulation and cost

The objective of the algorithm is to compute the pathway in the matrix M that according to a certain scoring schema is assigned the maximum value. The procedure to efficiently compute this value consists of two stages (see Figure 3.2):

- ① Each cell of the matrix $M(i, j)$ is assigned the corresponding value whether there is a match or a mismatch in this position (e.g. 1 for identities, void or 0 for mismatches).
- ② Beginning at the terminals of the sequences and proceeding toward the origins in the matrix, the value of the maximum-match starting at each cell $M(i, j)$ can be obtained by adding to its value, the maximum value from among all the cells which lie on a pathway to it. The pathways are negatively weighted with the value g according to the number of gaps they contain.

$$M(i, j) = M(i, j) + \max \begin{cases} M(i+1, j+1) \\ M(i', j+1) + g \times (i' - i + 1), & i+2 \leq i' \leq |A| \\ M(i+1, j') + g \times (j' - j + 1), & j+2 \leq j' \leq |B|. \end{cases} \quad (3.4)$$

If $|A| = |B| = n$, then the cost of visiting each cell of the matrix is $O(n^2)$. Additionally, for each cell the best pathway among all of the possible ones in the previous row, in the previous column and in the diagonal is searched. The cost of accessing the values of the pathways in a given column or row is $O(n)$, while accessing the diagonal is constant $O(1)$. Therefore, the final cost of the [Needleman and Wunsch](#) algorithm is $O(n^3)$.

Implementation

The implementation of the algorithm is shown in Figure 3.3. The matrix is processed following a systematic order. Both processing steps described above are integrated in a single one. For each pair of amino acids from both sequences represented by a cell $M(i, j)$ in the matrix, the optimal pathway starting there is constructed selecting the best pathway in the diagonal, and in the $i+1$ row and the $j+1$ column (here weighting according to the number of gaps) that have been previously computed.

The matrix P is used to record the cell from which the maximum pathway was selected. The retrieval of the solution, not shown here, consists on (1) searching the maximum value (cell x, y) both in the first row and in the first column and (2) using recursively the coordinates in $P(x, y)$, to construct the arrangement of both sequences until a cell at the last column or row is reached.

The Sellers algorithm (1974)

In the 1970s, most techniques used in taxonomic tree construction depended on the introduction of a measure of distance between sequences ([Fitch and Margoliash, 1967](#)). The work on distances or metrics on protein sequences was essentially based on discovering what genetic mutations were required to change one sequence into another.

A metric space is a function $\rho : S \times S \rightarrow \mathcal{Z}^+$ on a generic set S , with the following properties:

<i>Non-negative</i>	$\forall a, b \in S : \rho(a, b) \geq 0$
<i>Identity</i>	$\forall a, b \in S : \rho(a, b) = 0 \Leftrightarrow a = b$
<i>Reflexivity</i>	$\forall a, b \in S : \rho(a, b) = \rho(b, a)$
<i>Transitivity</i>	$\forall a, b, c \in S : \rho(a, b) \leq \rho(a, c) + \rho(c, b).$

[Sellers \(1974\)](#) described the construction of an evolutionary tree, which assumes that evolutionary distance is a metric. The minimum distance $D(A, B)$ between two sequences A and B is defined as the smallest possible weighted sum of insertions, deletions, and substitutions which transforms one sequence into the other.

[Sellers](#) showed that if a scoring function $d(a, b)$ ³ forms a metric space over the underlying alphabet of symbols then the minimum distance function $D(A, B)$ forms a metric space over the set of finite sequences constructed with such an alphabet. In addition, he proportioned the dynamic programming recurrence to efficiently compute the minimum distance D between two sequences using several scoring functions. In fact, many comparison algorithms that use distance functions with a given weighting scheme provide an optimal alignment only if such a scheme is a metric ([Tyler et al., 1991](#)).

Formulation and cost

[Sellers](#) generalized the algorithm to allow for various weighting schemes. Let a and b be two symbols. The simplest scheme d to score this match is defined as:

$$d(a, b) = \begin{cases} 0 & \text{if } a = b \\ 1 & \text{if } a \neq b. \end{cases} \quad (3.5)$$

Using this scoring function d , the following recurrence calculates the optimal distance between two sequences $A = (a_1, a_2, \dots, a_m)$ and $B = (b_1, b_2, \dots, b_n)$, and provides the initial values as well:

$$D(i, j) = \min \begin{cases} D(i-1, j-1) + d(a_i, b_j) & \text{Match} \\ D(i-1, j) + d(a_i, -) & \text{Gap in B} \\ D(i, j-1) + d(-, b_j) & \text{Gap in A} \end{cases}, \quad (3.6)$$

$$D(i, 0) = \sum_{k=0}^i d(a_k, -),$$

$$D(0, j) = \sum_{k=0}^j d(-, b_k).$$

To avoid the exponential number of combinations to construct an alignment between two sequences, this dynamic programming recurrence decompose the problem in smaller alignments of prefixes of the original sequences. Thus, starting from the one-letter prefixes, the minimum distance of the alignment ending at the prefixes $A_{1,i}$ and $B_{1,j}$ can be calculated from the three different forms of finishing such an alignment:

³Also known as a weighting scheme.

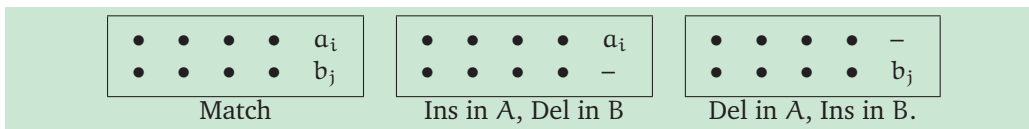
```

Pre  $\equiv$  A, B: sequences; id, mis, gap  $\in \mathcal{Z}$ 

(* Begin the series of sums from last row and column *)
for i = |A| to 1 do
  for j = |B| to 1 do
5:   (* Setting the identity or mismatch value for the cell *)
    if  $a_i = b_j$  then
       $M(i, j) \leftarrow \text{id}$ ;
    else
       $M(i, j) \leftarrow \text{mis}$ ;
10:  if  $i \neq |A|$  and  $j \neq |B|$  then
    (* Search the maximum-match pathway beginning here *)
    (* A. The maximum from diagonal *)
     $\text{max} \leftarrow M(i + 1, j + 1)$ ;
     $P(i, j) \leftarrow (i + 1, j + 1)$ ;
15:  (* B. The maximum value from previous column *)
     $\text{ngaps} \leftarrow 1$ ;
    for  $i' = i + 2$  to |A| do
       $\text{value} \leftarrow M(i', j + 1) + \text{gap} * \text{ngaps}$ ;
      if  $\text{value} > \text{max}$  then
20:         $\text{max} \leftarrow \text{value}$ ;
         $P(i, j) \leftarrow (i', j + 1)$ ;
         $\text{ngaps} \leftarrow \text{ngaps} + 1$ ;
    (* C. The maximum value from previous row *)
     $\text{ngaps} \leftarrow 1$ ;
25:  for  $j' = j + 2$  to |B| do
     $\text{value} \leftarrow M(i + 1, j') + \text{gap} * \text{ngaps}$ ;
    if  $\text{value} > \text{max}$  then
       $\text{max} \leftarrow \text{value}$ ;
       $P(i, j) \leftarrow (i + 1, j')$ ;
30:   $\text{ngaps} \leftarrow \text{ngaps} + 1$ ;
    (* The maximum-match pathway is formed *)
     $M(i, j) \leftarrow M(i, j) + \text{max}$ ;

```

Figure 3.3 The Needleman and Wunsch algorithm.



If both sequences have the same length n , the cost of the [Sellers](#) algorithm is $O(n^2)$ which is the time to visit all of the cells of the dynamic programming matrix (see [Figure 3.4](#)). For each cell, only three neighbours are consulted: in the diagonal, in the horizontal and in the vertical.

The procedure to trace-back the distance matrix, reconstructing the alignment was

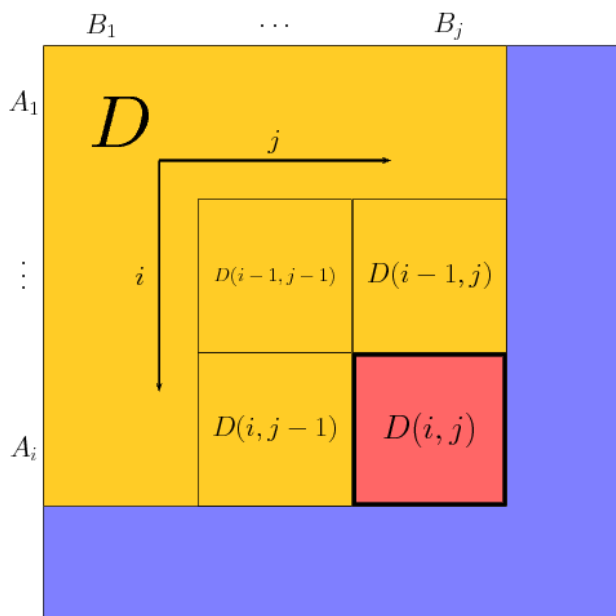


Figure 3.4 The dynamic programming matrix. In yellow, the part of the alignment matrix that has been computed. In blue, the part that must be still calculated. The cell $D(i, j)$ is the match currently in process.

adapted from Needleman and Wunsch by Sellers. A second matrix of pointers is needed for recording from which direction was taken the value to update a given cell matrix.

Implementation

The Sellers algorithm requires to fit the Needleman and Wunsch $m \times n$ matrix in an artificial 0-column and 0-row to increase the initial distance when starting the alignment with gaps⁴.

Then, the algorithm starts at $D(1, 1)$ and the matrix is filled by rows (from top to bottom) and within a row by columns (from left to right). Thus, when a cell $D(i, j)$ is reached, its neighbours $D(i - 1, j - 1)$, $D(i - 1, j)$ and $D(i, j - 1)$ have been already calculated.

Contrarily to the Needleman and Wunsch algorithm (in which the maximum match was searched in the last column and the last row), the minimum distance between both sequences will be saved at the end into the cell $D(m, n)$ because of the different initialization.

As in the case of the Needleman and Wunsch, there is an auxiliary matrix P that saves the source of each calculation in a given cell to recursively reconstruct the alignment with such a distance.

⁴There is an easy modification of the algorithm to permit not to punish this kind of gaps.

```

Pre  $\equiv$  A, B: sequences; d: metric on  $\Sigma$ 

(* Initialize the 0-column and the 0-row *)
for i = 0 to |A| do
    D(i, 0)  $\leftarrow$  i  $\times$  d(ai, -);
5: for j = 1 to |B| do
    D(0, j)  $\leftarrow$  j  $\times$  d(bj, -);
    (* Filling the matrix *)
    for i = 1 to |A| do
        for j = 1 to |B| do
10:     (* A. Match *)
        min  $\leftarrow$  D(i - 1, j - 1) + d(ai, bj);
        P(i, j)  $\leftarrow$  (i - 1, j - 1);
        (* B. Gap in sequence B *)
        value  $\leftarrow$  D(i - 1, j) + d(ai, -);
15:     if value < min then
        min  $\leftarrow$  value;
        P(i, j)  $\leftarrow$  (i - 1, j);
        (* C. Gap in sequence A *)
        value  $\leftarrow$  D(i, j - 1) + d(-, bj);
20:     if value < min then
        min  $\leftarrow$  value;
        P(i, j)  $\leftarrow$  (i, j - 1);
        D(i, j)  $\leftarrow$  min;

```

Figure 3.5 The Sellers algorithm.

A linear space algorithm: Hirschberg (1975)

In some occasions when aligning two sequences, the limiting factor is not the time but the space (memory). Any algorithm that solves the alignment of two sequences can not decrease the quadratic time cost unless any assumption is made over the length of the inputs. However, the quadratic cost in terms of space can be reduced to a linear cost.

Hirschberg (1975) designed a divide and conquer algorithm to solve the LCS problem in linear space without increasing the asymptotic time cost. Later, Myers and Miller (1988) demonstrated how this technique could optimally deal with general sequence alignment problems.

The key point of the algorithm is based on the fact that in the alignment between the sequences A and B, any element of A will be aligned either to a gap or another element in B. Thus, the problem of aligning both sequences can be expressed in terms of making this decision for a current element a_i , assuming the optimal alignments between the subsequences from A and B around this element are already computed.

Another important fact is the ability to compute the distance between two sequences in linear space. If the dynamic programming matrix is filled in from top to bottom (row by row), and fixing a row, from left to right (column by column), then the values in a row i

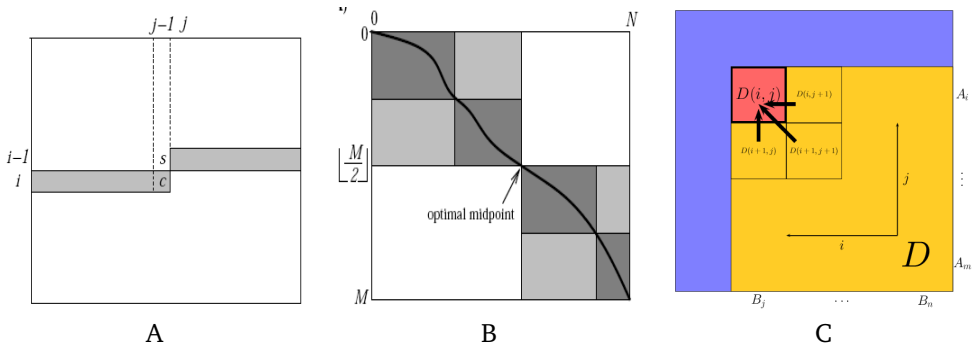


Figure 3.6 The Hirschberg linear space approach. (A) Using a single array to compute $D(i, j)$. (B) The divide and conquer strategy applied over the dynamic programming approach. (C) The backward propagation of values.

depend only on the values stored at the previous row $i - 1$ and on the values in the same row i . The other previous rows are therefore not necessary to obtain the final value $D(m, n)$ (Myers, 1991; Meidanis and Setubal, 1997).

Furthermore, instead of using two arrays to represent the rows i and $i + 1$, the computation can be performed in a single array D (see Figure 3.6 (A)), overwriting the old values on the left of the current column j . The equivalence between each cell $D(i, j)$ in the original dynamic programming matrix and the content of this unidimensional array D when the row i is being processed is:

$$\begin{aligned} D(k) &\approx D(i, k) \text{ when } k < j \text{ (current row, } i) \\ D(k) &\approx D(i - 1, k) \text{ when } k \geq j \text{ (previous row, } i - 1). \end{aligned} \quad (3.7)$$

Formulation and cost

In the optimal alignment between two sequences A and B , a given element a_i from A will be either matched to another element b_j from B or aligned to a gap between a certain b_j and b_{j+1} . Then, this optimal alignment can be decomposed in three parts:

- ① The optimal alignment between the elements from both sequences on the left (prefixes).
- ② The match between a_i with a certain b_j or a gap.
- ③ The optimal alignment between the elements from both sequences on the right (suffixes).

For a given i , the optimal point j can be unveiled with the application of the algorithm to compute only the distance between two sequences in linear space time. Such a solution

provides the point in which the optimal alignment path will cross the i -row in the the dynamic programming matrix. As it is shown in Figure 3.6 (B), once the points i and j are established, the general problem is divided into two subproblems and recursively the same procedure is applied until reaching the base case (empty sequences).

The algorithm that computes only the distance between two sequences in linear space is in fact a method to provide the minimum distance between the first sequence and any of the prefixes of the second sequence. As the right parts are also aligned in the main procedure, a modification of such an algorithm is necessary to obtain the minimum distance between the first sequence and any of the suffixes of the second one.

In fact, the dynamic programming scheme is not restricted to construct the final alignment from alignments between prefixes of the input sequences. The same recurrence is appropriate for building it from alignments between suffixes of them. The procedure now begins in the position $D(|A|, |B|)$, and propagates the values from bottom to top, and from right to left (see Figure 3.6 (C)). The Equation 3.6 must be slightly modified to accommodate this backward propagation:

$$D(i, j) = \min \begin{cases} D(i+1, j+1) + d(a_i, b_j) & \text{Match} \\ D(i+1, j) + d(a_i, -) & \text{Gap in B} \\ D(i, j+1) + d(-, b_j) & \text{Gap in A} \end{cases}, \quad (3.8)$$

$$D(i, |B| + 1) = \sum_{k=0}^i d(a_k, -),$$

$$D(|A| + 1, j) = \sum_{k=0}^j d(-, b_k).$$

The cost of obtaining just the value $D(i, j)$ following the forward or the backward manner is again quadratic in terms of time. However, the cost in terms of space of this function is linear as only a single array is used in both cases.

To compute the cost of a recursive divide and conquer function, a different cost scheme must be applied. Let us consider n the length of both input sequences that are aligned. At the beginning, the routine performs some computations and then, there is an approximate $\frac{n}{2}$ reduction in the size of the input data for the subsequent two recursive calls. The cost $T(n)$ of computing such a recursive function can be expressed in terms of its children as:

$$T(n) = \begin{cases} g(n) & 0 \leq n < c \quad \text{Base case} \\ aT(\frac{n}{c}) + bn^k & n \geq c \quad \text{Recursive case} \end{cases}, \quad (3.9)$$

where a is the number of recursive calls, c is the size of the fragmentation and bn^k is the cost of the non-recurrent operations performed on each call. From the relationship between a and c^k , the corresponding cost function is inferred following the Master Theorem of recurrent equations (see [Cormen et al. \(2001\)](#), Sections 4.3 and 4.4). In the [Hirschberg](#) recurrence is easy to notice $g(n) \in O(n)$, $a = 2$, $b = 2$, $c = 2$ and $k = 2$. Therefore as $a < c^k$ or $2 < 2^2$, according to the Master Theorem $T(n) \in \Theta(n^k)$, that is $T(n) \in \Theta(n^2)$.

Nevertheless, the spatial cost of the [Hirschberg](#) algorithm is linear. All of the computations on the theoretical dynamic programming matrix are performed over single rows implemented with unidimensional arrays.

Procedure ComputeOnlyDistanceForward	Procedure ComputeOnlyDistanceBackward
Pre \equiv A, B: sequences; d: metric on Σ Post \equiv D: array ($ B + 1$);	Pre \equiv A, B: sequences; d: metric on Σ Post \equiv D: array ($ B + 1$);
(* Simulating the initialization of the 0-row *) for j = 0 to $ B $ do D(j) $\leftarrow j \times d(-, b_j)$; 5: for i = 1 to $ A $ do diag \leftarrow D(0); (* Simulating the initialization of the i-row *) D(0) $\leftarrow i \times d(a_i, -)$; for j = 1 to $ B $ do 10: (* This cell will be the next diagonal *) temp \leftarrow D(j); (* A. Match (\nearrow) *) min \leftarrow diag + d(a_i, b_j); (* B. Gap in sequence A (\uparrow) *) value \leftarrow D(j) + d($a_i, -$); 15: if value < min then min \leftarrow value; (* C. Gap in sequence B (\leftarrow) *) value \leftarrow D(j - 1) + d($-, b_j$); 20: if value < min then min \leftarrow value; D(j) \leftarrow min; (* Update diagonal *) diag \leftarrow temp	(* Simulating the initialization of the last row *) for j = $ B $ to 1 do D(j) $\leftarrow j \times d(-, b_j)$; 5: for i = $ A - 1$ to 1 do diag \leftarrow D($ B + 1$); (* Simulating the initialization of the i-row *) D($ B + 1$) $\leftarrow i \times d(a_i, -)$; for j = $ B $ to 1 do 10: (* This cell will be the next diagonal *) temp \leftarrow D(j); (* A. Match (\nwarrow) *) min \leftarrow diag + d(a_i, b_j); (* B. Gap in sequence A (\downarrow) *) value \leftarrow D(j) + d($a_i, -$); 15: if value < min then min \leftarrow value; (* C. Gap in sequence B (\rightarrow) *) value \leftarrow D(j + 1) + d($-, b_j$); 20: if value < min then min \leftarrow value; D(j) \leftarrow min; (* Update diagonal *) diag \leftarrow temp

Figure 3.7 An algorithm to compute $D(i, j)$ in $O(n)$ space cost. (Left) The computation is done from $D(0, 0)$ to $D(|A|, |B|)$. (Right) The computation is done from $D(|A| + 1, |B| + 1)$ to $D(1, 1)$.

Implementation

Computing the value $D(i, j)$ in linear space

To implement the fusion of the previous row and the current one in a single array in a forward manner, the temporary variables *diag* and *temp* are necessary to save the values $D(i - 1, j - 1)$ –diagonal– and $D(i - 1, j)$ –gap in A–, respectively. At the end of the forward computation, the array D will contain the same values as the last row of the bidimensional classic dynamic programming matrix, that is, the distance between the sequence A and any of the prefixes of the sequence B. In particular, the array position $D(|B|)$ will contain the distance between the sequences $|A|$ and $|B|$.

The backward computation is symmetrical to the forward processing. The propagation of values starts now in the position $D(|B| + 1)$, moving the values from right to left, and from bottom to top. At the end of the backward computation, the array D will contain the same values as the last row of the bidimensional reverse dynamic programming matrix, that is the distance between the sequence A and any of the suffixes of the sequence B. In particular, the array position $D(1)$ will contain the distance between the sequences $|A|$ and $|B|$.

Procedure Alignment

Pre \equiv A, B: sequences; d: metric on Σ ; $i_1, i_2, j_1, j_2, al_1, al_2$ in \mathcal{Z}

Post \equiv alA : array ($al_1..al_2$), alB : array ($al_1..al_2$);

```

if  $A_{i_1, i_2} = \emptyset$  then
  (* Base case 1 *)
  for  $k = j_1$  to  $j_2$  do
5:    $alA(al_1 + k) \leftarrow -$ ;
     $alB(al_1 + k) \leftarrow B(j_1 + k)$ ;
     $al_2 \leftarrow al_1 + k$ ;
  else if  $B_{j_1, j_2} = \emptyset$  then
    (* Base case 2 *)
10:  for  $k = i_1$  to  $i_2$  do
     $alA(al_1 + k) \leftarrow A(i_1 + k)$ ;
     $alB(al_1 + k) \leftarrow -$ ;
     $al_2 \leftarrow al_1 + k$ ;
  else
15:  (* General case *)
    (* Select the point i *)
     $i \leftarrow \lfloor \frac{i_1 + i_2}{2} \rfloor$ 
    (* Compute the distance to the prefixes/suffixes of B *)
     $prefDist \leftarrow \text{ComputeOnlyDistanceForward}(A_{i_1, i}, B_{j_1, j_2}, d)$ ;
20:   $suffDist \leftarrow \text{ComputeOnlyDistanceBackward}(A_{i+1, i_2}, B_{j_1, j_2}, d)$ ;
    (* The column 0 *)
     $posmin \leftarrow j_1 - 1$ ;
     $typemin \leftarrow \text{SPACE}$ ;
     $vmin \leftarrow prefDist(j_1 - 1) + d(a_i, -) + suffDist(j_1 - 1)$ ;
    (* A sweep along the row i *)
25:  for  $j = j_1$  to  $j_2$  do
    (* Match *)
     $value \leftarrow prefDist(j - 1) + d(a_i, b_j) + suffDist(j + 1)$ ;
    if  $value < vmin$  then
30:     $vmin \leftarrow value$ ;
     $posmin \leftarrow j$ ;
     $typemin \leftarrow \text{SYMBOL}$ ;
    (* Gap *)
     $value \leftarrow prefDist(j) + d(a_i, -) + suffDist(j + 1)$ ;
35:  if  $value < vmin$  then
     $vmin \leftarrow value$ ;
     $posmin \leftarrow j$ ;
     $typemin \leftarrow \text{SPACE}$ ;
    (* Divide and conquer with these values of i and j *)
40:  if  $typemin \leftarrow \text{SPACE}$  then
     $\text{Align}(A, B, d, i_1, i - 1, j_1, posmin, al_1, al_{tmp})$ ;
     $alA(al_{tmp}) \leftarrow A(i)$ ;
     $alB(al_{tmp}) \leftarrow -$ ;
     $\text{Align}(A, B, d, i + 1, i_2, posmin + 1, j_2, al_{tmp}, al_2)$ ;
45:  else
     $\text{Align}(A, B, d, i_1, i - 1, j_1, posmin - 1, al_1, al_{tmp})$ ;
     $alA(al_{tmp}) \leftarrow -$ ;
     $alB(al_{tmp}) \leftarrow B(posmin)$ ;
     $\text{Align}(A, B, d, i + 1, i_2, posmin + 1, j_2, al_{tmp} + 1, al_2)$ ;

```

Figure 3.8 The Hirschberg linear space algorithm.

The divide and conquer algorithm

The [Hirschberg](#) linear space algorithm is a function *Alignment* that computes the position of a given symbol a_i from A in the optimal alignment (aligned to a gap or to a certain b_j from B) and then splits the general problem into two smaller subproblems (left and right halves of the corresponding sequences).

The initial call is *Alignment*($A, B, d, 1, |A|, 1, |B|, 0, 0$) where $(0, 0)$ are the boundaries (a_1, a_2) of the optimal alignment that is in construction. Additionally, a pair of arrays alA, alB will save the correspondence between the symbols from both sequences.

As we divide the problem into two minor parts, the base cases are the empty sequences (alignment of the rest of the symbols in one sequence with gaps in the other one). The general case selects a middle point or symbol a_i from A . Then, the distance between the prefix $A_{1,i-1}$ to all of the prefixes of the sequence B is computed by the routine *ComputeOnlyDistanceForward*. The routine *ComputeOnlyDistanceBackward* likewise calculates the same value between the suffix $A_{i+1,|A|}$ and all of the suffixes of the sequence B .

Now, a sweep shifting j along the whole row is performed to detect the point j in which the alignment constituted by the prefix of A and a given prefix of B , the symbol a_i and a gap or a symbol b_j , and the suffix of A and a given suffix of B is optimal. This operation is easily implemented by accessing with the proper indexes the arrays *prefDist* and *suffDist* that were filled in by the corresponding *ComputeOnlyDistance* functions.

Once the optimal j for the current a_i has been found, a recursive call to discover the part of the optimal alignment on the left of this symbol is launched. Then, the conquer step assigns the correct position to a_i aligned to a gap or a certain b_j in the alignment (arrays alA, alB). Finally, a second recursive call is performed to place correctly the right part of the optimal alignment.

The variables *posmin*, *vmin*, *typemin* save at each moment the minimum distance value in the loop along j and the position and the symbol to be aligned to a_i . The type of symbol is important to correctly split the sequence B at the divide step.

The Needleman and Wunsch algorithm revisited by Smith *et al.* (1981)

Although the denomination of [Needleman and Wunsch](#) algorithm and [Sellers](#) algorithm have survived throughout the years, the standard formulation in terms of distance and similarity methods that is widely known today was provided by [Smith et al. \(1981\)](#). In their work, they adapted the [Needleman and Wunsch](#) method to a dynamic programming recurrence complementary to that introduced by [Sellers](#) and presented an analysis of equivalence between both measures (see next section).

The similarity measure did not conserve the mathematical properties of the distance metrics. Nonetheless, this revisited version of the algorithm became very popular because it was easily extended to cope with the local alignment problem (see Section 3.4).

Pre \equiv A, B: sequences; s: substitution matrix

```

(* Initialize the 0-column and the 0-row *)
for i = 0 to |A| do
    S(i, 0)  $\leftarrow$  i  $\times$  s(ai, -);
5: for j = 1 to |B| do
    S(0, j)  $\leftarrow$  j  $\times$  s(bj, -);
(* Filling the matrix *)
for i = 1 to |A| do
    for j = 1 to |B| do
10:    (* A. Match *)
    max  $\leftarrow$  S(i - 1, j - 1) + s(ai, bj);
    P(i, j)  $\leftarrow$  (i - 1, j - 1);
    (* B. Gap in sequence B *)
    value  $\leftarrow$  S(i - 1, j) + s(ai, -);
15:    if value > max then
        max  $\leftarrow$  value;
        P(i, j)  $\leftarrow$  (i - 1, j);
    (* C. Gap in sequence A *)
    value  $\leftarrow$  S(i, j - 1) + s(-, bj);
20:    if value > max then
        max  $\leftarrow$  value;
        P(i, j)  $\leftarrow$  (i, j - 1);
    S(i, j)  $\leftarrow$  max;

```

Figure 3.9 The Needleman and Wunsch algorithm revisited.

Formulation and cost

If matches are positively rewarded and gaps are punished negatively, the recurrence for computing the maximum similarity between two sequences A and B is:

$$\begin{aligned}
 S(i, j) &= \max \begin{cases} S(i-1, j-1) + s(a_i, b_j) & \text{Match} \\ S(i-1, j) + s(a_i, -) & \text{Gap in B} \\ S(i, j-1) + s(-, b_j) & \text{Gap in A} \end{cases}, \\
 S(i, 0) &= \sum_{k=0}^i s(a_k, -), \\
 S(0, j) &= \sum_{k=0}^j s(-, b_k).
 \end{aligned} \tag{3.10}$$

where the function $s(a_i, b_j)$ provides a positive or a negative value for a given match (mismatch) according to the aligned elements. If this is an alignment of proteins, the function s can be a popular amino acid substitution matrix with an additional penalty for aligning a symbol to a gap.

The cost of this revisited Needleman and Wunsch algorithm is $O(n^2)$, being correct the same analysis explained in the Sellers approach.

Implementation

This implementation is a symmetric translation from the implementation of the [Needleman and Wunsch](#) algorithm. The same procedures to fill the matrix in and to retrieve the optimal alignment are performed.

Equivalence between distance and similarity: Smith *et al.* (1981)

From the formulation of the [Needleman and Wunsch](#) similarity algorithm and the [Sellers](#) distance algorithm a couple of relevant questions quickly arised: (1) When are both algorithms equivalent? (2) When do they provide the same set of optimal alignments? [Smith and Waterman \(1981a\)](#) stated that the two algorithms are defined to be equivalent if given the scoring scheme for one algorithm, there is a choice of a scoring scheme for the second algorithm such that the set of alignments achieving the maximum similarity is equal to the set of alignments achieving the minimum distance.

Given two sequences A and B , the optimal alignment \mathcal{A} that maximizes $S(A, B)$ or minimizes $D(A, B)$ can be decomposed into two sections: the matched elements (λ_i) and the elements in one sequence that are aligned with gaps in the other one (Δ_k):

$$\begin{array}{cc}
 s(a_i, b_j) = \alpha_k & \quad \quad \quad d(a_i, b_j) = \beta_k \\
 g(k) \geq 0 & \quad \quad \quad g'(k) \geq 0 \\
 \lambda_i \rightarrow \# \text{ of aligned symbols of type } i & & \\
 \Delta_k \rightarrow \# \text{ of gaps of length } k & & \\
 S(A, B) = \max_{\mathcal{A}} \{ \sum_i \alpha_i \lambda_i - \sum_k g(k) \Delta_k \} & \mid & D(A, B) = \min_{\mathcal{A}} \{ \sum_i \beta_i \lambda_i + \sum_k g'(k) \Delta_k \}.
 \end{array} \tag{3.11}$$

The following consideration that relates the length of the input sequences to the number of aligned symbols and gaps is essential for the next equations:

$$|A| + |B| = 2 \sum_i \lambda_i + \sum_k \Delta_k. \tag{3.12}$$

For instance, this equation applied on the alignment

$$\begin{array}{cccccc}
 A: & A & A & T & T & C & A \\
 & | & & & | & | & \\
 B: & T & - & - & T & C & -
 \end{array}$$

with $|A| = 6$ and $|B| = 3$ with three matches, one gap of two positions and one gap of one position, produces:

$$6 + 3 = 2 \times 3 + 1 \times 2 + 1 \times 1.$$

Smith et al. (1981) showed that with a certain scoring model for both algorithms, the optimal alignments are equivalent. Let α_M be $\alpha_M = \max_i \alpha_i$ (the maximum value of similarity). Then, the other scoring model must be defined as $\beta_i = \alpha_M - \alpha_i$. Intuitively, the higher the similarity, the lower the distance. Thus, maximum similarity (α_M) equals to minimum distance (0). The development of the Equation 3.11 produces:

$$\begin{aligned}
 S(A, B) &= \max_{\mathcal{A}} \{ \sum_i \alpha_i \lambda_i - \sum_k g(k) \Delta_k \} \\
 \star \beta_i &= \alpha_M - \alpha_i \star \\
 &= \max_{\mathcal{A}} \{ \sum_i (\alpha_M - \beta_i) \lambda_i - \sum_k g(k) \Delta_k \} \\
 &= \max_{\mathcal{A}} \{ \alpha_M \sum_i \lambda_i - \sum_i \beta_i \lambda_i - \sum_k g(k) \Delta_k \} \\
 \star |A| + |B| &= 2 \sum_i \lambda_i + \sum_k \Delta_k \star \\
 &= \max_{\mathcal{A}} \{ \alpha_M (\frac{|A|+|B|}{2}) - \sum_k \frac{k}{2} \Delta_k - \sum_i \beta_i \lambda_i - \sum_k g(k) \Delta_k \} \\
 &= \max_{\mathcal{A}} \{ \alpha_M (\frac{|A|+|B|}{2}) - \sum_k \frac{\alpha_M k}{2} \Delta_k - \sum_i \beta_i \lambda_i - \sum_k g(k) \Delta_k \} \\
 &= \max_{\mathcal{A}} \{ \alpha_M (\frac{|A|+|B|}{2}) - \sum_i \beta_i \lambda_i - \sum_k (\frac{\alpha_M k}{2} + g(k)) \Delta_k \} \\
 &= \alpha_M (\frac{|A|+|B|}{2}) + \max_{\mathcal{A}} \{ - \sum_i \beta_i \lambda_i - \sum_k (\frac{\alpha_M k}{2} + g(k)) \Delta_k \} \\
 &= \alpha_M (\frac{|A|+|B|}{2}) - \min_{\mathcal{A}} \{ \sum_i \beta_i \lambda_i + \sum_k (\frac{\alpha_M k}{2} + g(k)) \Delta_k \} \\
 &= \alpha_M (\frac{|A|+|B|}{2}) - D(A, B).
 \end{aligned} \tag{3.13}$$

To sum up, the minimum distance $D(A, B)$ is equivalent to the maximum similarity $S(A, B)$ when the following scoring model for the distance scheme is employed:

$$\begin{aligned}
 \beta_i &= \alpha_M - \alpha_i \\
 g'(k) &= \frac{\alpha_M k}{2} + g(k).
 \end{aligned} \tag{3.14}$$

Given the similarity scoring model $(s, g(k))$, the following distance scheme is therefore compatible:

$$s(a, b) = \begin{cases} 0 & \text{if } a \neq b \\ 1 & \text{if } a = b \end{cases} \tag{3.15} \quad \left| \quad d(a, b) = \begin{cases} 0 & \text{if } a = b \\ 1 & \text{if } a \neq b \end{cases} \tag{3.16}$$

where $\alpha_M = 1$, $\beta_i = 1 - \alpha_i$, $d(a, b) = 1 - s(a, b)$ and $g'(k) = \frac{k}{2} + g(k)$.

Obviously, not all the possible s functions will have a compatible d counterpart (see local alignment, Section 3.4).

The Sellers algorithm generalized by Waterman *et al.* (1976)

From an evolutionary point of view, a single mutation event involving a gap with k positions is more probable than the same number of distinct mutations of k isolated spaces. In the previous algorithms, gaps have been treated as another symbol producing simple mismatches. However, longer indels should not be weighted as the sum of single indels.

Let $g(k)$ an arbitrary function that determines the penalty for a gap of length k , in which the existence of any relationship between the penalty of a gap having k characters and a gap of $k + 1$ is not assumed (general gap scoring model):

In Needleman and Wunsch / Sellers	A more realistic weighting scheme
$g(k) = kg(1)$	$g(k) \leq kg(1)$

Waterman, Smith, and Beyer introduced a new metric. Let $\tau = \{T|T : S \rightarrow S\}$ be a set of transformations (including identity) applied over an input sequence. Every transformation has an associated weight w . Given two sequences A and B , a sum of weights $\sum_{i=1}^k w(T_i)$ can be computed for each sequence of transformations T_1, T_2, \dots, T_k from τ such that $T_1 \circ T_2 \circ \dots \circ T_k(A) = (B)$. The minimum sum of weights of such sequences of transformations can be viewed as the distance from A to B and a metric space is obtained⁵.

τ can be employed with different sets of transformations and weights (Waterman *et al.*, 1976). Specifically, the authors defined a τ -metric which included longer deletions and insertions, and generalize the Sellers algorithm for computing the new distance.

Formulation and cost

In the Sellers algorithm, the optimal alignment between the prefixes $A_{1,i}$ and $B_{1,j}$ could contain a match between a_i and b_j or an alignment of one of them to a gap in the other sequence. In this new generalized gap model, an alignment of one of them to a gap of length k in the other sequence is also possible.

To deal with gaps that have different scores according to their lengths, given a cell $D(i, j)$ in the dynamic programming matrix, all of the possible gaps of $1..(i - 1)$ symbols (scanning a column, fixing j) and all of the possible gaps of $1..(j - 1)$ symbols (scanning a row, fixing i) must be evaluated (see Figure 3.10).

This modification also receives the name of block indel variation because there are now three classes of implicit blocks to establish the optimal alignment between two symbols: either a match between both or an alignment between a substring of symbols in one of the sequences to a block of gaps in the other.

⁵The weights associated to every class of transformation must be non-negative.

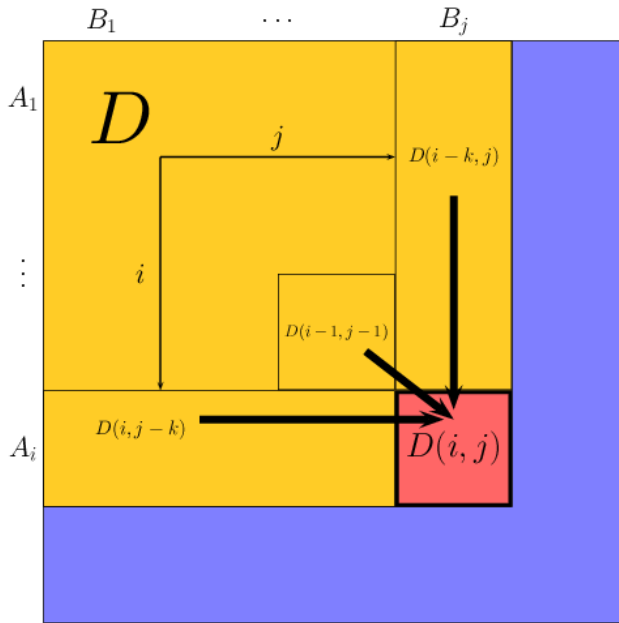


Figure 3.10 The generalized dynamic programming matrix. In yellow, the part of the alignment matrix that has been computed. In blue, the part that must be still calculated. The cell $D(i, j)$ is the match currently in process.

The following recurrence represents the generalization of the Sellers algorithm by Waterman, Smith, and Beyer:

$$D(i, j) = \min \begin{cases} D(i-1, j-1) + d(a_i, b_j) & \text{Match} \\ \min_{1 \leq k \leq i} \{D(i-k, j) + g(k)\} & \text{Gap of length } k \text{ in } B \\ \min_{1 \leq k \leq j} \{D(i, j-k) + g(k)\} & \text{Gap of length } k \text{ in } A \end{cases}, \quad (3.17)$$

$$D(k, 0) = g(k),$$

$$D(0, k) = g(k).$$

The algorithm must evaluate for each cell $D(i, j)$, all of the previously computed cells in that row and column. If the length of the sequences is m and n respectively, the cost of performing such an alignment with a general gap scoring model is therefore $O(mn(m+n))$, that is, $O(m^2n + mn^2)$ or $O(n^3)$ if both sequences have the same length.

Implementation

This algorithm requires the existence of an artificial 0-column and 0-row to compute the distance when starting the alignment with gaps. Then, the algorithm starts at $D(1, 1)$ and

Pre \equiv A, B: sequences; d: metric on Σ ; g(k): gap scoring function

```

(* Initialize the 0-column and the 0-row *)
for i = 0 to |A| do
  D(i, 0)  $\leftarrow$  g(i);
5: for j = 1 to |B| do
  D(0, j)  $\leftarrow$  g(j);
(* Filling the matrix *)
for i = 1 to |A| do
  for j = 1 to |B| do
10:   (* A. Match *)
   min  $\leftarrow$  D(i - 1, j - 1) + d(ai, bj);
   P(i, j)  $\leftarrow$  (i - 1, j - 1);
   (* B. Gap of length k in sequence B *)
   for k = 1 to i - 1 do
15:     value  $\leftarrow$  D(i - k, j) + g(k);
     if value < min then
       min  $\leftarrow$  value;
       P(i, j)  $\leftarrow$  (i - k, j);
   (* C. Gap of length k in sequence A *)
20:   for k = 1 to j - 1 do
     value  $\leftarrow$  D(i, j - k) + g(k);
     if value < min then
       min  $\leftarrow$  value;
       P(i, j)  $\leftarrow$  (i, j - k);
25:   D(i, j)  $\leftarrow$  min;

```

Figure 3.11 The Sellers algorithm generalized.

the matrix is filled by rows (from top to bottom) and within a row by columns (from left to right). For a given a cell $D(i, j)$ in the matrix, its neighbour in the diagonal $D(i - 1, j - 1)$ is evaluated (match) and additionally, all of the previous cells at that row and column must be separately visited to measure the contribution of $g(k)$ to their final value.

The minimum distance between both sequences will be saved at the end into the cell $D(m, n)$. The optimal alignment with such a distance value can be recursively retrieved from the auxiliary matrix P that saved the direction of the alignment for each cell.

The Waterman *et al.* algorithm revisited by Gotoh (1982)

Despite its cubic cost, the Waterman, Smith, and Beyer algorithm provided a more realistic gap treatment model from the biological standpoint. Several posterior proposals were presented to reduce that cost by simplifying the gap model. Gotoh (1982) proposed a less general model called the affine gap scoring model in which the gap scoring function presents a linear schema based on a different penalty for opening a gap and for extending an existing

one.

Let $g(k)$, the affine gap model to score a gap of k positions, then

$$\begin{aligned} g(k) &= \begin{cases} a, & \text{if } k = 1 \\ a + bk, & \text{if } k > 1 \end{cases} \quad \text{where } a, b \geq 0 \\ g(k+1) &= a + b(k+1) = a + bk + b = g(k) + b. \end{aligned} \quad (3.18)$$

With such a function, if $a > b$ then the first space in a gap of length k is more expensive than the rest of $k - 1$ spaces that extend the gap. As the value $g(k+1)$ can be computed only using the previous value $g(k)$, there is no need to perform an exhaustive scanning of a given row and column for each pair i, j in the dynamic programming matrix.

Formulation and cost

[Gotoh](#) rewrote the general recurrence by [Waterman et al. \(1976\)](#), introducing two additional functions E and F that substituted the two loops along the column and the row of a given cell $D(i, j)$ to evaluate the gaps of length k :

$$\begin{aligned} D(i, j) &= \min\{D(i-1, j-1) + d(a_i, b_j), E(i, j), F(i, j)\} \\ E(i, j) &= \min_{1 \leq k \leq i} \{D(i-k, j) + g(k)\} \\ F(i, j) &= \min_{1 \leq k \leq j} \{D(i, j-k) + g(k)\} \\ D(k, 0) &= g(k), \\ D(0, k) &= g(k). \end{aligned} \quad (3.19)$$

Unfolding the value of $E(i, j)$ in the k and $k+1$ iterations, the combination between Equation 3.18 and Equation 3.19 produced the following result ([Gotoh, 1982](#)):

$$\begin{aligned} E(i, j) &= \min_{1 \leq k \leq i} \{D(i-k, j) + g(k)\} \\ &= \min\{D(i-1, j) + g(1), \min_{2 \leq k \leq i} \{D(i-k, j) + g(k)\}\} \\ &= \min\{D(i-1, j) + a, \min_{1 \leq k \leq i-1} \{D(i-(k+1), j) + g(k+1)\}\} \\ &= \min\{D(i-1, j) + a, \min_{1 \leq k \leq i-1} \{D(i-1-k, j) + g(k)\} + b\} \\ &= \min\{D(i-1, j) + a, E(i-1, j) + b\}. \end{aligned} \quad (3.20)$$

The same recursion is applied to the function F producing

$$F(i, j) = \min\{D(i, j-1) + a, F(i, j-1) + b\}. \quad (3.21)$$

Pre \equiv A, B: sequences; d: metric on Σ ; $g(k) = a + bk$;

```

(* Initialize the 0-column and the 0-row *)
for i = 0 to |A| do
  D(i, 0)  $\leftarrow$  g(i);
5:   E(i, 0)  $\leftarrow$  g(i);
   F(i, 0)  $\leftarrow$  g(i);
for j = 1 to |B| do
  D(0, j)  $\leftarrow$  g(j);
  E(0, j)  $\leftarrow$  g(j);
10:  F(0, j)  $\leftarrow$  g(j);
(* Filling the matrix *)
for i = 1 to |A| do
  for j = 1 to |B| do
    (* A. Update the E matrix *)
15:    min  $\leftarrow$  D(i - 1, j) + a;
    value  $\leftarrow$  E(i - 1, j) + b;
    if value < min then
      min  $\leftarrow$  value;
    E(i, j)  $\leftarrow$  min;
20:    (* B. Update the F matrix *)
    min  $\leftarrow$  D(i, j - 1) + a;
    value  $\leftarrow$  F(i, j - 1) + b;
    if value < min then
      min  $\leftarrow$  value;
25:    F(i, j)  $\leftarrow$  min;
    (* C. Minimum between Match, E and F *)
    min  $\leftarrow$  D(i - 1, j - 1) + d(ai, bj);
    P(i, j)  $\leftarrow$  D(i - 1, j - 1);
    if E(i, j) < min then
30:      min  $\leftarrow$  E(i, j);
      P(i, j)  $\leftarrow$  E(i, j);
    if F(i, j) < min then
      min  $\leftarrow$  F(i, j);
      P(i, j)  $\leftarrow$  F(i, j);
35:    D(i, j)  $\leftarrow$  min;

```

Figure 3.12 The Gotoh algorithm.

Now, there are only three operations that must be performed to compute each $D(i, j)$: the match between both symbols, and the alignment of one of them to a gap (of any length) in the other sequence. The cost of using the affine gap scoring model is therefore $O(n^2)$, notably smaller than the cubic cost of the original general solution.

Implementation

To implement the functions E and F, two additional matrices are necessary. Then, the value $D(i, j)$ is selected (minimum) among the value of $D(i - 1, j - 1)$ (a match) and the values of $E(i, j)$ (a gap in the second sequence) and $F(i, j)$ (a gap in the first sequence).

The matrix P is used again to maintain the pathway associated to the optimal distance between any prefix of the input sequences.

Concave gap penalty functions: Waterman (1984)

In the affine gap penalty model, the same penalty is associated to the second space and to all of the next spaces in a gap. [Fitch and Smith \(1983\)](#) studied the behaviour of the multiple indels scoring models in the coding region of the chicken α and β hemoglobin genes. They determined that a specific range of gap penalties was necessary to obtain correct alignments. Later, [Waterman \(1984a\)](#) formally introduced the concave gap functions.

In this scoring scheme, the gaps after the first one are not punished proportionally as in the case of the affine model. Once there is a gap, it must be biologically easier to incorporate more gaps. Let $g(k)$ the function that provides the penalty for a gap of length k , then:

$$g(k + 1) - g(k) \leq g(k) - g(k - 1). \quad (3.22)$$

The affine model arises directly when the equality is required. Strict inequality corresponds to those increasing functions with decreasing differences between consecutive gaps, also referred to as concave downward or simply concave. For instance, the function

$$g(k) = a + b \log(k) \text{ where } a, b \geq 0. \quad (3.23)$$

Let $f(k) = a + bk$ the affine gap penalty function, for a given length k the difference with the behavior of g is clear. For instance, if $k = 16$ then $f(16) = a + 16b$ whereas $g(16) = a + b \log_2(16) = a + 4b$, being less penalized this large gap in comparison to smaller gaps.

Formulation and cost

Lying between the general gap model, with a $O(n^3)$ cost, and the affine gap model, with a $O(n^2)$ cost, the concave gap problem has been proved to have an algorithm with a cost $O(n^2 \log n)$. [Waterman \(1984a\)](#) introduced the concept and conjectured such a cost. Posteriorly, two independent groups arrived at different solutions with such a cost ([Eppstein et al., 1988](#); [Miller and Myers, 1988](#)).

3.4 A short overview on local sequence alignment

Local alignments are usually more meaningful than global alignments because they only detect the patterns that are conserved in the sequences. The statistical significance of these patterns is usually evaluated. Uncommon degree of conservation of these segments in long sequences could be explained in terms of conservation of biological function.

Two alternative lines using dynamic programming approaches were proposed to rigorously detect such fragments: algorithms based on similarity and algorithms based on distance metrics. Traditionally, similarity schemes have shown to be easier to be implemented whereas distance measures are more complex to be adapted to this problem. Additional works about pattern discovery and multiple local alignments are provided in Chapter 4.

The Smith and Waterman algorithm (1981)

In a short communication, [Smith and Waterman \(1981b\)](#) published a slight modification of the [Needleman and Wunsch](#) algorithm revisited by [Smith et al. \(1981\)](#) to deal with local alignments. The main objective was to find the pair of segments, one from each of two long sequences, such that there is no other pair of segments with greater similarity (homology).

The key point is to stop the traceback that starts from the cell having the maximum similarity whenever a negative similarity zone is detected. The score function s must therefore include negative values for mismatches to provide optimal alignments with this strategy.

Posterior refinements by [Waterman and Eggert \(1987\)](#) allow to report the second best path disjoint from the first one, the third best and so on. Essentially, the positions of the visited previous maximum paths are marked up and a new recomputation of some parts of the matrix is done to repeat the traceback.

Formulation and cost

In this formulation, a cell $S(i, j)$ of the dynamic programming matrix whose value after evaluating its neighbours is negative must be automatically set to 0 (the value for representing the lack of similarity of any local alignment ending at this cell). In fact, all of the positions in the matrix with a 0 are candidates to become the left boundary of the optimal local alignment between two sequences A and B.

The Equation 3.10 is just slightly modified to accommodate this concept:

$$S(i, j) = \max \begin{cases} S(i-1, j-1) + s(a_i, b_j) & \text{Match} \\ S(i-1, j) + s(a_i, -) & \text{Gap in B} \\ S(i, j-1) + s(-, b_j) & \text{Gap in A} \\ 0 & \text{Segment termination} \end{cases}, \quad (3.24)$$

$$S(i, 0) = 0,$$

$$S(0, j) = 0.$$

As long as the scoring function $s(a, b)$ with $a \neq b$ (mismatch) returns negative values, the similarity of every path in the matrix will increase and decrease according to the associated alignment. Once the matrix has been completed, the cell having the highest value will be the right boundary of the optimal local alignment. From this point, the rest of the maximum similarity segment must be retrieved going back until a 0 is reached.

The natural generalization to support multiple insertions/deletions ($g(k)$) is naturally derived:

$$S(i, j) = \max \begin{cases} S(i-1, j-1) + s(a_i, b_j) & \text{Match} \\ \max_{1 \leq k \leq i} \{S(i-k, j) + g(k)\} & \text{Gap of length } k \text{ in } B \\ \max_{1 \leq k \leq j} \{S(i, j-k) + g(k)\} & \text{Gap of length } k \text{ in } A \\ 0 & \text{Segment termination} \end{cases}, \quad (3.25)$$

$$S(i, 0) = 0,$$

$$S(0, j) = 0.$$

Reduction to $O(n^2)$ can be achieved applying the [Gotoh \(1982\)](#) results as in the global alignment case. The time cost function of the versions above is the same as in their global counterparts as no additional operations are needed.

Implementation

In contrast to the global alignment algorithm, the initialization procedure reset to 0 the 0-row and the 0-column. In this implementation 0 means termination of current segment at the traceback process. The matrix P is used again to save the optimal pathway of the segment maximizing similarity ending at each position of the matrix S .

To retrieve such a segment, the position $S(i, j)$ which contains the maximum value in the matrix is found. Then, an ordinary traceback in P must be performed, reconstructing this local alignment until a cell whose value is 0 is reached, terminating then.

Distance-based scoring schemes

As it has been shown in Section 3.3, [Smith et al. \(1981\)](#) determined the following relationship between a metric distance $D(A, B)$ and a homology function $S(A, B)$:

$$S(A, B) + D(A, B) = \alpha_M \frac{(m+n)}{2}, \quad (3.26)$$

where α_M is the maximum score for a match, and m and n are the lengths of the respective sequences A and B . From this, it might seem that the problem of finding segments of maximum similarity can be simply reformulated into a problem of finding segments of minimum distance. However, several differences between both measures prevent the establishment of such an equivalence:

Pre \equiv A, B: sequences; s: substitution matrix

```

(* Initialize the 0-column and the 0-row *)
for i = 0 to |A| do
    S(i, 0)  $\leftarrow$  0;
5: for j = 1 to |B| do
    S(0, j)  $\leftarrow$  0;
(* Filling the matrix *)
for i = 1 to |A| do
    for j = 1 to |B| do
10:     (* A. Segment termination *)
        max  $\leftarrow$  0;
        P(i, j)  $\leftarrow$  (0, 0);
        (* B. Match *)
        value  $\leftarrow$  S(i - 1, j - 1) + s(ai, bj);
15:     if value > max then
        max  $\leftarrow$  value;
        P(i, j)  $\leftarrow$  (i - 1, j - 1);
        (* C. Gap in sequence B *)
        value  $\leftarrow$  S(i - 1, j) + s(ai, -);
20:     if value > max then
        max  $\leftarrow$  value;
        P(i, j)  $\leftarrow$  (i - 1, j);
        (* D. Gap in sequence A *)
        value  $\leftarrow$  S(i, j - 1) + s(-, bj);
25:     if value > max then
        max  $\leftarrow$  value;
        P(i, j)  $\leftarrow$  (i, j - 1);
        S(i, j)  $\leftarrow$  max;

```

Figure 3.13 The Smith and Waterman algorithm.

- ➔ The maximum similarity is a positive number that depends on the aligned segments. On the contrary, the minimum distance is always 0.
- ➔ The similarity scoring scheme typically has a negative reward for mismatches and gaps and a positive reward for matches. However, the distance metric has no positive reward for matches: the extension of an alignment with a minimum distance d can only receive a score equal or worse than the original one (continuously growing function).
- ➔ In the similarity model, during the traceback a local alignment starting at the cell having the maximum value $S(i, j)$ is extended. Then, 0 is employed as a limit of such an extension. In the distance model, there is not a simple minimum value $D(i, j)$ in the matrix to start the traceback because smaller segments would be better by definition. Furthermore, there is not here an equivalent of the 0 in the similarity model during the traceback procedure.

To overcome some of these limitations, [Goad and Kanehisa \(1982\)](#) considered to include the length of the segments in the scoring scheme as a way to favor longer alignments against shorter alignments with distance 0. The mismatch density of an alignment \mathcal{A} between two segments is defined as the ratio of the minimum distance D between both sequences and the length L of the alignment. In addition, only those alignments with a mismatch density below a certain positive threshold R must be reported:

$$\frac{D(\mathcal{A})}{L(\mathcal{A})} \leq R. \quad (3.27)$$

Essentially, the segment maximizing the similarity should be equivalent to a segment starting at $D(i_0, j_0)$ and ending at $D(i, j)$ with $i_0 < i$ and $j_0 < j$ such that the difference $\Delta D = D(i, j) - D(i_0, j_0)$ is the minimum taking into account the length of such an alignment.

[Goad and Kanehisa](#) also transformed this distance scheme into a similarity scheme that must be maximized, with the following manipulations:

$$\frac{D(\mathcal{A})}{L(\mathcal{A})} \leq R \equiv D(\mathcal{A}) \leq RL(\mathcal{A}) \equiv RL(\mathcal{A}) - D(\mathcal{A}) \geq 0. \quad (3.28)$$

Formulation and cost

First approaches

Previously to [Goad and Kanehisa \(1982\)](#), [Sellers \(1980\)](#) approached the problem with an algorithm to determine the segments S and T such that for any aligned pair (S', T') in a small neighbourhood, $D(S, T) \leq D(S', T')$. Obviously, $D(S, T)$ was guaranteed to be only a relative minimum in such a set of alignments. Therefore, the procedure provided many alignments like this that needed further screening.

Later, [Goad and Kanehisa](#) used the mismatch density concept to propose an algorithm in two steps. The solution is better understood if alignments are represented by paths in a lattice of points:

- ① Use the [Sellers](#) global alignment algorithm to fill in the matrix D_F (minimize distance). This formulation computes the values from left to right and from top to bottom. This form corresponds to obtain the optimal alignment using the increasing prefixes of the sequences (forward graph).
- ② The same algorithm can be formulated in terms of suffixes of the input sequences (see the explanation about the [Hirschberg](#) algorithm). Then, use such an algorithm over the same sequences to fill in the matrix D_B (backward graph).
- ③ Report those paths that were common in D_F and D_B .

This solution limited the number of paths but there is not a clear procedure to show that these are optimal. The cost of the algorithm is clearly $O(n^2)$.

Multi-sweep algorithm by Sellers (1984)

Sellers (1984) described a more rigorous extension of the Goad and Kanehisa algorithm in which several iterations over a single matrix are necessary to remove the edges of the paths that are not supported in the forward or backward computations.

Given a positive constant R , the algorithm produces all paths P such that:

- ① All prefixes of P have mismatch density less than R .
- ② All suffixes of P have mismatch density less than R .
- ③ The path P is locally maximal. The paths meeting the two previous conditions that intersect with P have a lowest score.

The algorithm starts with a matrix G_0 in which every possible alignment of the two given sequences is represented. First, the forward procedure removes all edges of the paths not being part of any alignment⁶, creating the matrix G_1 . Second, in a backward computation, all edges from G_1 not meeting the alignments of the suffixes are also erased to form the matrix G_2 . Then, alternating forward and backward computations are performed over G_i , removing edges of the paths at each stage until no variation is observed. At the end, all of the disjoint paths present in the matrix are reported as local alignments or segments minimizing the mismatch density criterion.

No more than $O(n)$ sweeps are ever required to converge (Myers, 1991). As every forward or backward operation takes $O(n^2)$ time, the final cost of the approach by Sellers is $O(n^3)$, notably higher than the $O(n^2)$ cost of the simple Smith and Waterman design.

Databases searches

The information available at the sequence databases is useful to infer the function of similar sequences. Anonymous sequences can be aligned to other sequences whose function, structure or biochemical activity is known. As explained in Chapter 2, the size of such databases grows exponentially since the very first days of computational sequence analysis.

It is important to mention that from now on the term database simply refers to a large collection of sequences. It does not imply any extra capabilities of fast access, data sharing, and so on, commonly found in standard database management systems.

Ordinary alignment algorithms based on dynamic programming are very inefficient to search large collections of sequence because of their quadratic time cost. Novel methods based on heuristics have been employed to reduce in several orders of magnitude the time to align two sequences, providing near optimal results. The search on a database for sequences that are similar to a query sequence usually performs hundreds of thousands of such alignments.

This search typically provides a list of sequences with which the query sequence can be aligned better, using certain quality score function. These results can be expanded using each sequence found before to find more distant relatives of the initial sequences.

⁶In the dynamic programming recurrence, each edge corresponds to a decision in the optimization step.

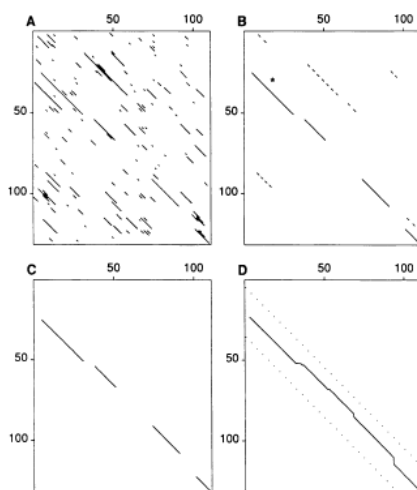


Figure 3.14 Identification of sequence similarities by FASTA. Adapted from [Pearson and Lipman \(1988\)](#).

To speed the search, the sequences of the database are usually preprocessed to store computations about their content (usually word distribution) that will be used during the future searching operations.

FASTA

The FAST family of algorithms is a group of heuristic methods for string comparison, specially to compare a query sequence with each sequence on a database. ([Lipman and Pearson, 1985](#); [Pearson and Lipman, 1988](#)). The FASTA program that is included on such a package is entirely based on the following assumption: good local alignments are likely to contain exact matching subsequences. The FASTA strategy is therefore to locate firstly the segments of both sequences richer in exact matches and secondly, try to reconstruct the final alignment using these specific regions.

The FASTA processing is divided into four main steps (see [Figure 3.14](#)) that are repeated to compare the query sequence to each sequence in the database:

- ① Detection of regions of identity. Determine the words of length k (k -tuples) that are common to both sequences. The offset of an exact word match between a substring s starting at position x and a substring t starting at position y is defined as the difference $x - y$. Matches that are located in the same diagonal of the dotplot comparison have the same offset value (see (A) in [Figure 3.14](#)). An array addressed by the offsets is used to locate those diagonals with more exact matches.

During the preprocessing of each sequence in the database, a hash table is used to store where each word of length k is appearing along such a sequence ([Dumas and Ninio, 1982](#)). Then, the query sequence is scanned and each k -tuple in it is looked up

in the hash table. For all common occurrences, the entry of the corresponding offset is incremented.

Next, each offset is analyzed to merge those exact matches in the same diagonal that are in close proximity (without introducing gaps, including intervening sequence). These merged regions do not contain any insertions or deletions because they are derived from a single diagonal. The score of these diagonal regions is the sum of the exact matches scores combined with a penalty that increases with the distance among them. According to this scoring scheme, the 10 best diagonal regions are selected to constitute the future local alignment (see (B) in Figure 3.14).

- ② Re-scoring. The 10 best diagonals are evaluated again using an amino acid (or nucleotide) substitution matrix to allow conservative replacements and exact matches shorter than k to contribute to the similarity score. The diagonal region with maximal score is identified (highest scoring initial region). Those regions whose score is below a given threshold are discarded (see (C) in Figure 3.14).
- ③ Optimal alignment of diagonal regions. The regions from compatible diagonals are combined following certain rules. The segments that are close to each other (not in the same diagonal) can be part of an alignment whose score is a function of a joining penalty (moving from one diagonal to other involves gap introduction), their scores and their location. The optimal alignment initial region is a combination of compatible regions with maximal score. This score is a reference to rank the library of sequences according to their similarity to the query (see (D) in Figure 3.14).
- ④ The highest scoring library sequences are finally aligned with a modification of the [Needleman and Wunsch](#) and [Smith and Waterman](#) algorithms. Using dynamic programming, all possible alignments of the query and each sequence in the database that fall within a band centered around the highest scoring initial region are considered.

BLAST

As FASTA, the BLAST family programs ([Altschul et al., 1990, 1997](#)) are able to achieve a substantial gain in terms of speed by searching first for common words or k -tuples in the query and in each database sequence. However, FASTA searches for all possible words of the same length whereas BLAST limits the search to those that are the most significant by integrating a substitution matrix in this step.

The central concept of the BLAST strategy is the neighbourhood of a sequence. The T -neighbourhood of a word w is the set of all sequences of the same length that align to w with score better than T . Such an alignment is gapless and the similarity score is the sum of the similarity values for each pair of aligned residues. Thus, searching a match between a given word in the query and other word in a sequence of the database is equivalent to searching a match between a neighbour of the original word in the query with a score greater than T and the same word in the other sequence. BLAST will only seek in the database for those significant words that would form with w a pair with a score of at least T , if any.

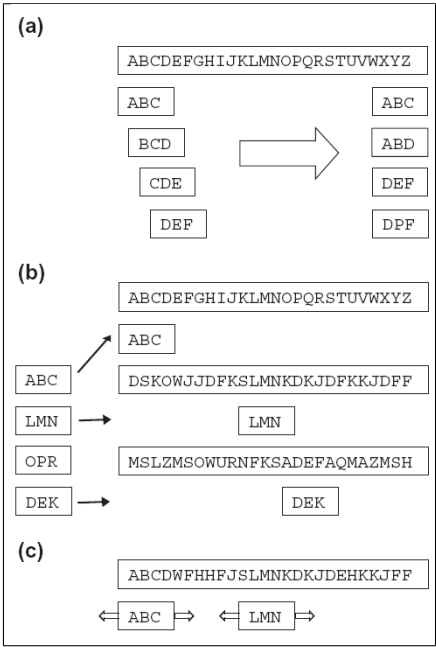


Figure 3.15 BLAST processing. Adapted from [Pertsemlidis and Fondon \(2001\)](#).

As a substitution matrix is used to score this alignment, the words where conservative substitutions have been introduced can also obtain a high score because the matches with them also may be biologically informative. In addition, different amino acid identities are not scored in the same manner: for instance, the alignment between a query word composed by very common amino acids and itself might not achieve a score better than T , and therefore it would not be included in the search process ⁷.

Whenever one of these significant words is found in one entry of the database, the respective word w in the query and the detected neighbour are aligned and form the seed of a segment pair that will be enlarged later. If the extended segment pair is assigned a score better than S , such a sequence is reported to be similar to the query.

The BLAST pipeline is constituted by these steps (see Figure 3.15) that are repeated to compare the query sequence to each sequence in the database:

- ① The query sequence is filtered to remove low-complexity regions (repeats) that can distort the word search (optional) to produce significant alignments.
- ② Generate the T -neighbourhood of every word of length k in the query sequence. Given a word w , the matches between any other combination of k amino acids and w are evaluated with a substitution matrix. For instance, if $k = 3$, there are 8000 possible words to align with w . The neighbours are ranked according to the score of this alignment. A deterministic finite automaton is constructed to recognize the language of the high-scoring neighbours (the most significant ones).

⁷BLAST allows the user to force the inclusion of the original words in the following steps.

- ③ See if any sequence in the database contains one of these strings with the automaton constructed before (a match).
- ④ Every match is used as a seed to find a locally maximal segment pair containing that hit, also called a maximal segment pair (MSP). The alignment between both words in their respective sequences is extended in each direction along the respective sequences, continuing the extension as long as the score does not fall more than a drop-off threshold. Such a score is a cumulative value resulting from evaluating with a substitution matrix the matches, mismatches and gaps of the alignment.
- ⑤ BLAST reports the database sequences with MSPs above a certain threshold S (the high-scoring segment pairs or HSPs). Such significant value is computed for each database according to the size of the query and the database, being unlikely to find a random sequence that achieves a score better than S when compared with the query (Karlin and Altschul, 1990).

The procedure is heuristic: only word pairs with a score above the threshold T can be the core of local similarity regions. Therefore, a segment pair of score better than S that does not contain any subsequence of length k with a score greater than T will not be detected. In addition, the selection of the parameters is not trivial: this method is feasible in practice only when the values of k , T and S are carefully chosen (Altschul et al., 1990; Myers, 1991).

3.5 A short overview on multiple sequence alignment

From a simplistic point of view, a multiple sequence alignment (MSA) is a rectangular array of sequences optimally arranged to obtain the greatest number of similar characters on each column of the alignment. From an evolutionary perspective, however, the alignment of multiple sequences is intimately related to the study of molecular evolution. For example, the number and the class of changes in the residues of a MSA may be used to develop a preliminary phylogenetic analysis. Each column in the alignment of a set of sequences may predict the mutations that occurred at one site during the evolution of such a sequence family, revealing which positions in the sequences were conserved and which diverged from a common ancestor sequence.

The natural extension of the pairwise dynamic programming recurrence produces a multidimensional representation of the similarity matrix, being not possible to be implemented in practice (see the example for three sequences in Figure 3.16 (A)). Because of its $O(n^k)$ cost, where k is the number of sequences and n is the length of them (Waterman et al., 1976)), several approaches have tried to circumvent such a problem by introducing some heuristic functions.

Carrillo and Lipmann (1988) developed a method assuming that the optimal MSA can be constructed from the best pairwise alignments between each pair of sequences (the projections). Thus, each optimal pairwise alignment defines a set of spatial positions within which the optimal MSA is supposed to be when projected on such a plane (see Figure 3.16 (B) and (C)).

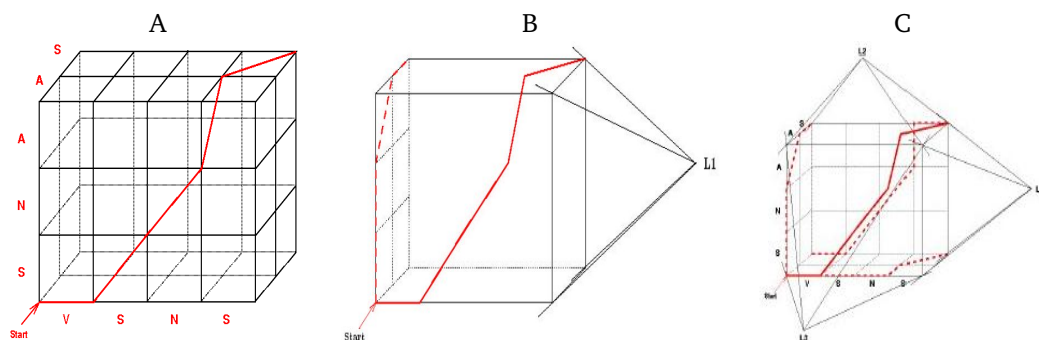


Figure 3.16 Generalized MSA dynamic programming matrix. (A) The $O(n^2)$ matrix is generalized into a $O(n^k)$ matrix in a multiple sequence alignment. (B) The projection of the optimal MSA into one of the pairwise alignments (Carrillo and Lipmann, 1988). (C) The optimal MSA alignment projected into all of the pairwise alignments (Carrillo and Lipmann, 1988).

The generalization to multiple alignment also induced a problem in the dimension of the substitution matrices and in the form of scoring an alignment in general. Let k be the number of aligned characters in a column of a MSA. In principle, $2^k - 1$ combinations with such elements are possible, but a substitution matrix of such dimensions would be absolutely unfeasible. The ordinary approach when scoring a MSA is usually the sum of pairs (the SP-score) that weights the n^2 combinations between two elements in the same column with a normal substitution matrix to provide a final score.

The hierarchical or clustering method called progressive alignment rapidly became popular because of its simplicity and biological feasibility (Feng and Doolittle, 1987). This strategy initially selects the best pairwise alignment and progressively incorporates the rest of sequences to this alignment. However, this dependence on the first alignment produces somehow a loss of flexibility in the rest of the subsequent alignments as most of the conserved positions in such an alignment are preserved throughout the process. The order of sequence selection relies on the creation of a phylogenetic tree that guides the process to create the MSA. There are several well-known techniques to infer the best tree for a set of sequences. Distance based methods are based on minimizing the number of global changes between each pair of input sequences. The neighbour-joining algorithm (Saitou and Nei, 1987) is a distance based method that first joins the clusters of sequences that are close to each other and apart from the rest, minimizing the sum of the branch lengths in the final tree.

The program CLUSTALW (Thompson et al., 1994) incorporates a number of improvements to the progressive alignment implementation. In an initial round, all of the pairwise alignments are performed to calculate a distance matrix in $O(k^2n^2)$. A guide tree is constructed from this matrix using the neighbour-joining method. An initial alignment starting with the two most related sequences is then constructed. Finally, the sequences are gradually aligned according to the branching order in the guide tree (see the complete process in Figure 3.17).

During the construction of the tree, CLUSTALW assigns weights to the sequences to correct unfair sampling across all evolutionary distances in the data set. Highly divergent

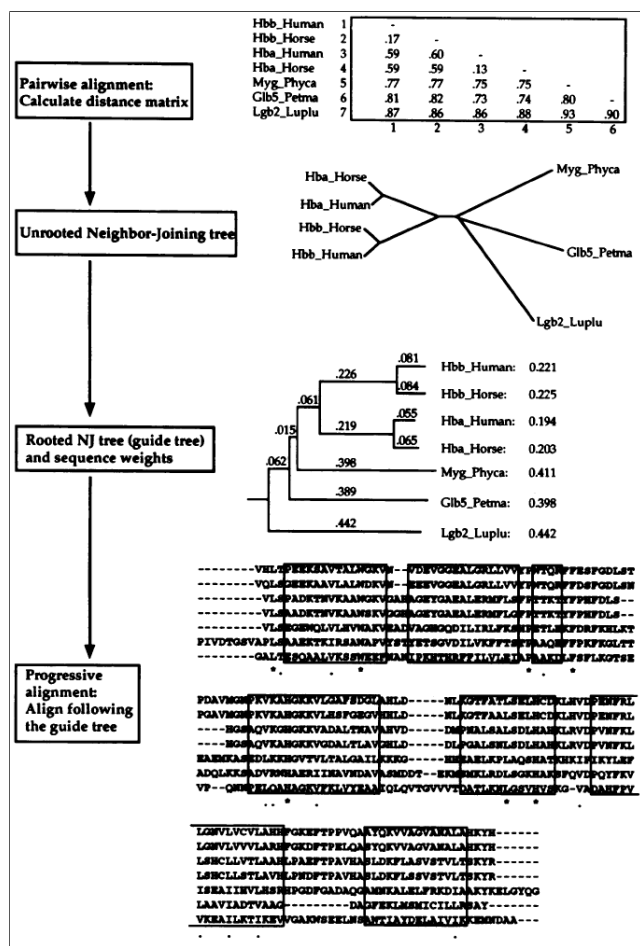


Figure 3.17 The basic CLUSTALW progressive alignment procedure. Adapted from Thompson et al. (1994).

sequences without close relatives receive high weights. For instance, the weight 0.221 for the Hbb_Human gene in Figure 3.17 is calculated in this form:

$$0.221 = 0.081 + \frac{0.226}{2} + \frac{0.061}{4} + \frac{0.015}{5} + \frac{0.062}{6}$$

In addition, different substitution matrices are used on every stage of the alignment. Position-specific gap penalties that depend on several factors such as the existence of other gaps, the type of residues or the length of the sequences are also used during the alignment.

The dynamic programming recurrence must be now adapted to allow the alignment between two profiles or clusters of sequences that have been previously aligned. The score of the alignment between a column in a first alignment and a column in a second alignment is the average of all of the pairwise substitution matrix scores from the residues in the two

sets of sequences multiplied by the weight of the sequences. Let C_i and C_j be two multiple alignments:

$$C_i = \begin{cases} x_1^1 \dots x_1^{l_i} \\ \vdots \\ x_{|i|}^1 \dots x_{|i|}^{l_i} \end{cases} \quad C_j = \begin{cases} y_1^1 \dots y_1^{l_j} \\ \vdots \\ y_{|j|}^1 \dots y_{|j|}^{l_j} \end{cases} \quad (3.29)$$

Let $p \in C_i, q \in C_j$, two columns of the previous alignments. The score $S(C_i^p, C_j^q)$ of the alignment between both columns is computed as:

$$S(C_i^p, C_j^q) = \frac{\sum_{r=1}^{|i|} \sum_{s=1}^{|j|} w_r \cdot w_s \cdot M(x_r^p, y_s^q)}{|i||j|} \quad (3.30)$$

All of the methods above produce a global multiple sequence alignment. Local alignment of several sequences is intimately related to motif finding techniques, all of them heuristics. In Chapter 4, there is a brief overview about several pattern discovery methods.

3.6 Map alignments

Restriction enzymes and genomic maps

The DNA molecules in a cell can be randomly broken into small pieces by mechanical forces. However, the probability of randomly breaking a molecule to produce a fragment that contains a gene is null. Restriction nucleases, which can be purified from bacteria, are enzymes that cut the DNA double helix at specific sites defined by the local nucleotide sequence, producing DNA fragments of defined sizes (Alberts et al., 1994). In fact, every nuclease recognizes a specific sequence of four to eight nucleotides (see Figure 3.18 for examples).

Different species of bacteria make restriction nucleases with different sequence specificities. More than 100 nucleases are now available commercially. It is relatively simple to find a restriction nuclease that create a DNA fragment including a particular gene (Alberts et al., 1994).

After treatment with a combination of several restriction nucleases, a restriction map of a particular genetic region can be constructed showing the location of each restriction site in relation to the neighbour sites (see Figure 3.19 for an example of map comparison). The sites thus act as genetic markers and the map reflects their arrangement in the region. This arrangement allow the comparison of the same region of DNA in different species without having to determine the nucleotide sequence in detail (Alberts et al., 1994). Indeed, mutations at a single letter of a sequence of DNA can cause the appearance or disappearance of a restriction site.

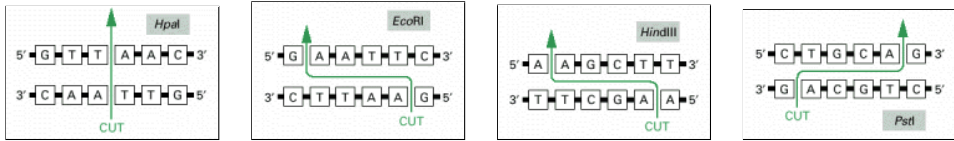


Figure 3.18 The DNA nucleotide sequences recognized by four widely used restriction nucleases. Adapted from [Alberts et al. \(1994\)](#).

Common problems involving restriction maps are:

- ➔ Prior to genomic sequencing projects, to organize genomic DNA, one approach was to make restriction maps of relatively small pieces to utilize these maps later to determine overlap of pieces and thus construct a map that includes larger parts of the genome.
- ➔ The fragment lengths from a digestion of a DNA sequence can be measured after using two enzymes separately, or by both applied together. The problem of determining the positions of the cuts from fragment length data is known as the Double Digest Problem ([Schmitt and Waterman, 1991](#)).

A more general definition can be considered. Genomic mapping is the process of determining where an object of biological interest (e.g. a marker, a gene, a genomic variation, or a disease predisposition locus) lies within a defined genomic sequence. Such a map therefore describes biological attributes of each genomic position (see [White and Matisse \(2005\)](#) for a comprehensive introduction about mapping concepts).

Map alignments

Given a sequence S of m symbols, a site $a_i = (r_i, p_i)$ is an element of a certain type r_i mapped on a certain position p_i relative to the origin of S . A map A is then defined as an ordered set of n sites such that:

$$A = a_1 a_2 \dots a_n \text{ where } \forall i : 1 \leq i \leq n : a_i = (r_i, p_i), r_i \in \Sigma_{\text{sites}}, 1 \leq p_i \leq p_{i+1} \leq m$$

[Waterman et al. \(1984\)](#) first defined the notion of map comparison using alignments and developed an algorithm that handles the distances between the sites as well as the linear sequence of sites. If intersite distances were ignored, then the [Sellers](#) algorithm could be immediately applied to align two maps ([Huang and Waterman, 1992](#)).

Let $A = a_1 a_2 \dots a_m$ and $B = b_1 b_2 \dots b_n$ be two maps of m and n sites respectively with $a_i = (r_i, p_i)$ and $b_j = (s_j, q_j)$. An alignment of A and B is a sequence of ordered matching pairs of sites $(a_{i_1}, b_{j_1})(a_{i_2}, b_{j_2}) \dots (a_{i_T}, b_{j_T})$ such that:

- ① $(a_i, b_j) \in C$ if and only if $r_i = s_j$ (that is, two elements are aligned if and only if they correspond to the same site).

- ② if $(a_i, b_j) \in C$ then there are no other elements $b_l (l \neq j)$ in B such that $(a_i, b_l) \in T$, nor elements $a_k (k \neq i)$ in A such that $(a_k, b_j) \in T$ (that is, each element in A is aligned at most to one element in B , and vice versa)
- ③ if $(a_i, b_j) \in C$ and $(a_k, b_l) \in C$ and $i < k$ then $j < l$ (that is, the alignment maintains the colinearity between the sequence A and B).

For instance, this is an example of a map alignment between the maps $A = \{(B, 1)(D, 15)(A, 20)(E, 32)(D, 50)(F, 95)\}$ and $B = \{(B, 5)(D, 17)(D, 47)(C, 78)(A, 87)(F, 92)\}$:

A =	(B,1)	(D,15)	(A,20)	(E,32)	(D,50)	-	-	(F,95)
B =	(B,5)	(D,17)	-	-	(D,47)	(C,78)	(A,87)	(F,92).

Let α be the reward given to each matching pair (optional), let λ be the penalty associated to each unaligned site from both maps and let μ be the penalty associated to the discrepancy in distance between adjacent aligned pairs $(a_{i_{t-1}}, b_{i_{t-1}})$ and (a_{i_t}, b_{i_t}) . Then, the score of the map alignment C between maps A and B that contains T matched pairs is defined to be:

$$\begin{aligned}
 S(C) = & \alpha T \\
 & -\lambda(m+n-2T) \\
 & -\mu(|q_{i_1} - p_{i_1}|) \\
 & -\mu \sum_{t=2}^T (|(p_{i_t} - p_{i_{t-1}}) - (q_{i_t} - q_{i_{t-1}})|) \\
 & -\mu(|(p_{i_m} - p_{i_T}) - (q_{i_n} - q_{i_T})|).
 \end{aligned} \tag{3.31}$$

That is, the score of the alignment increases with the score of the matches of the aligned elements (α , optional), and decreases with the number of elements not in the alignment (λ), and with the difference in the distance between matches of consecutive aligned elements (μ).

The Waterman *et al.* map alignment algorithm (1984)

Waterman *et al.* (1984) firstly formalized the problem of map alignment and introduced an algorithm distinct from usual sequence comparison algorithms, to investigate the relationships among restriction maps of homologous regions.

This algorithm yields a measure of distance between two maps and provides an alignment of them. Such a distance is the minimum weighted sum of genetic events required to convert one map into the other, where the genetic events are the appearance/disappearance of restriction sites and changes in the number of bases between them. Mutations from one site to other are ignored because this event was considered to be unlikely (Waterman *et al.*, 1984).

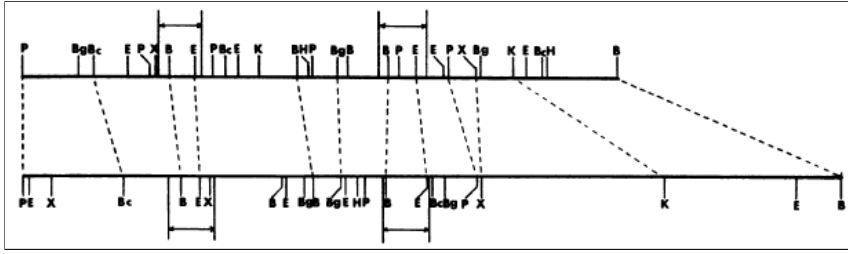


Figure 3.19 A restriction map alignment including the β and δ globin genes from the lowland Gorilla and the Owl Monkey. Adapted from [Waterman et al. \(1984\)](#).

Formulation and cost

Let $A = a_1 a_2 \dots a_m$ be a map of sites where each pair $a_i = (r_i, p_i)$ represents the restriction site r_i occurring at position p_i of a sequence of nucleotides, let $B = b_1 b_2 \dots b_n$ be a second map of sites denoted as $b_j = (s_j, q_j)$: a map alignment between A and B is a correspondence $(a_{i_1}, b_{j_1})(a_{i_2}, b_{j_2}) \dots (a_{i_T}, b_{j_T})$ in which two sites a_{i_t} and b_{j_t} constitute a match if they correspond to the same type of restriction site (see [Figure 3.19](#) for an example).

To measure the distance between two maps, two events must be taken into account:

- ① Each site from A and from B that is not aligned receives a weight λ .
- ② The number of bases between every pair of aligned sites in A that changes by x bases in B receives the weight $\mu(x)$.

Let $D(i, j)$ the minimum sum of weights of events required to convert the map A into the map B where the site a_i is equal to the site b_j (otherwise $D(i, j) = \infty$). Then, $D(i, j)$ is calculated as

$$D(i, j) = \min_{\substack{0 < i' < i \\ 0 < j' < j}} \{D(i', j') + \lambda(i - i' - 1 + j - j' - 1) + \mu(p_i - p_{i'} - q_j + q_{j'})\}.$$

(3.32)

Thus, the optimal alignment ending at a given pair (a_i, b_j) , where r_i is equal to s_j , is optimally computed by:

- ① Searching among the alignments ending at previous matches $(a_{i'}, b_{j'})$.
- ② Evaluating the value $D(i, j)$ if the pair $(a_{i'}, b_{j'})$ was placed immediately before the current pair (a_i, b_j) in the optimal alignment in construction.

Note that to compute the optimal score at $D(i, j)$ with this algorithm, all the cells $D(k, l)$ with $k < i$ and $l < j$ need to be explored. Therefore, if the length of the two maps A and

```

Pre  $\equiv A, B$ : maps;  $\lambda, \mu \in \mathcal{Z}^+$ 
(* Calculating the element  $i, j$  in  $D$  *)
for  $i = 0$  to  $|A| - 1$  do
  for  $j = 0$  to  $|B| - 1$  do
    if  $\text{site}(a_i) = \text{site}(b_j)$  then
      5:  $D(i, j) \leftarrow \text{ComputeInitialDistance}();$ 
      (* Searching the best previous element in  $D$  *)
      for  $i' = 0$  to  $i - 1$  do
        for  $j' = 0$  to  $j - 1$  do
           $y \leftarrow \lambda((i - i' - 1) + (j - j' - 1));$ 
          10:  $z \leftarrow \mu(|(\text{pos}(a_i) - \text{pos}(a_{i'})) - (\text{pos}(b_j) - \text{pos}(b_{j'}))|);$ 
           $\text{currentDist} \leftarrow D(i', j') + y + z;$ 
          if  $\text{currentDist} < D(i, j)$  then
             $D(i, j) \leftarrow \text{currentDist};$ 

```

Figure 3.20 The Waterman et al. map alignment algorithm.

B is m and n respectively, the cost of computing $D(A, B) = D(a_m, b_n)$ is $O(mn \cdot mn) = O(m^2n^2)$. Under the assumption that m and n are similar, the final cost function is $O(n^4)$. However, as there are hundreds of distinct types of sites, the dynamic programming matrix is actually very sparse (there is a smaller number of matches), being less prohibitive such a cost.

Implementation

A direct implementation of the recursion above involves the recursive filling of the cells $D(i, j)$ in the matrix D (Waterman, 1984b). In the pseudocode below, the elements of the maps A and B are represented as structures a_i and b_j , with the functions *site* and *pos* returning the values of the corresponding fields. The variable *currentDist* stores the minimum distance so far computed.

The resulting map alignment can be easily retrieved using a supplementary structure *path(i, j)* which points to the previous cell in the optimal path leading to cell $D(i, j)$. In addition, for each cell $D(i, j)$, the function *ComputeInitialDistance* calculates the initial score of a hypothetical alignment that includes only a_i and b_j .

The Myers and Huang map alignment algorithm (1992)

The formulation of the problem by Waterman (1984b) for aligning two maps A and B of m and n sites respectively, leads directly to a $O(m^2n^2)$ algorithm. Myers and Huang (1992) presented an algorithm for comparing restriction maps based on some works related to sequence comparison algorithms in the cases where gap costs are concave (see Section 3.3). Because of the distance between two maps relies not only on the number of gaps in the lists of sites but also on the physical distances between sites, multiple indels or gaps can be

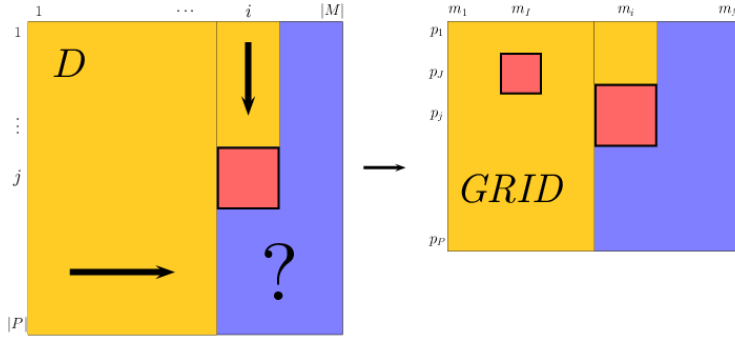


Figure 3.21 Mapping the D matrix over the rectangle $[0, m_M] \times [0, p_P]$.

treated as a unit.

Basically, the $O(n^4)$ cost of the original algorithm can be decomposed into two $O(n^2)$ components:

- ① The worst-case number of possible matches between A and B .
- ② The cost of retrieving the best previous match that minimizes the distance of the alignment ending at the current match.

While the cost of the first component is unavoidable, the second contribution can be reduced in many ways, specially in the cases in which the dynamic programming matrix is very sparse. In [Myers and Huang \(1992\)](#), such a cost is dramatically reduced to a logarithmic function through the application of several analytical methods. First of all, the formulation of the score (distance) of a map alignment is rewritten again: the elements that do not depend of the current match (a_i, b_j) are now isolated to be computed only once. Second, the dynamic programming matrix that is addressed with the sites in A and B is substituted by a grid of points that correspond to the physical positions of the elements from both maps. Finally, a list of candidates (previous matches) that induces a partition in the set of sites from the second map is updated when the matrix is filled in, at the same time the sites in the first map are being processed.

Formulation and cost

Let $M = \{M_1, M_2, \dots, M_M\} = \{(a_1, m_1), (a_2, m_2) \dots (a_M, m_M)\}$ be a map of sites where each pair (a_i, m_i) represents the restriction site a_i occurring at position m_i of a sequence of nucleotides, and let $P = \{P_1, P_2, \dots, P_P\} = \{(b_1, p_1), (b_2, p_2) \dots (b_P, p_P)\}$ be a shorter map of sites (a probe) where each pair (b_j, p_j) represents the restriction site b_j occurring at position p_j of a sequence of nucleotides. Then, the score of an alignment $C = (M_{i_1}, P_{j_1})(M_{i_2}, P_{j_2}) \dots (M_{i_L}, P_{j_L})$ between the map M and the probe P is defined to be:

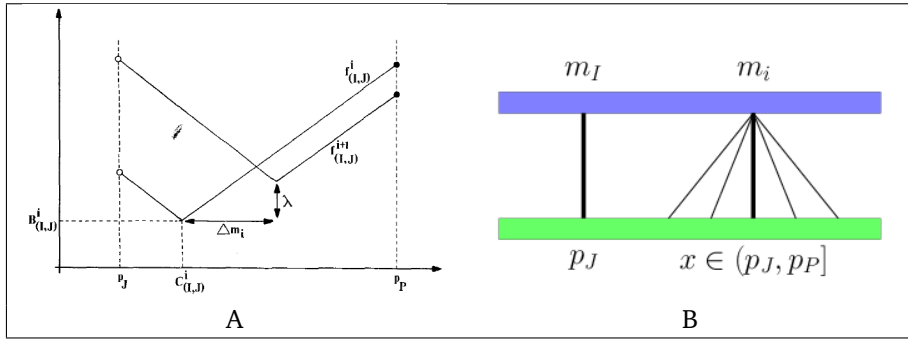


Figure 3.22 Analytical methods in Myers and Huang (1992). (A) An illustration of $f_{I,J}^i(x)$ and $f_{I,J}^{i+1}(x)$. (B) The contribution of a (I, J) to each match point (m_i, x) in the interval $(p_J, p_P]$ of P .

$$\text{Score}(C) = \lambda(P - L) + \mu \sum_{k=2}^L (|(m_{i_k} - m_{i_{k-1}}) - (p_{j_k} - p_{j_{k-1}})|). \quad (3.33)$$

That is, the distance between the map and the probe according to such an alignment increases with the number of elements of P not in the alignment (λ), and with the difference in the distance between matches of consecutive aligned elements (μ).

Let Matchpoints be the matches between the map and the probe $\{(i, j) | a_i = b_j\}$. Then, to compute the minimum distance between a map and a probe, the Equation 3.32 is rewritten by Myers and Huang in terms of the contribution of a previous match to the current one:

$$D(i, j) = \min(\lambda(P - 1), \min_{\substack{(I, J) \in \text{Matchpoints} \\ I < i, J < j}} \text{contrib}_{I,J}(i, j)). \quad (3.34)$$

Such a contribution of a previous match (I, J) to the current one (i, j) is defined as:

$$\text{contrib}_{I,J}(i, j) = D(I, J) + \lambda(i - I - 2) + \mu(|(m_i - m_I) - (p_j - p_J)|). \quad (3.35)$$

Instead of dealing with the classical dynamic programming matrix that is usually accessed using the sites in the maps, Myers and Huang (1992) proposed to map the original problem into a matrix representing the domain of physical positions. Thus, the procedure that completes the original matrix column by column is exported to this new grid whose dimensions are the position of the last site in both maps respectively (see Figure 3.21).

This algorithm computes each column of the matrix D in increasing order of i (M), simultaneously updating a list of match points called candidates. Each one of these previous matches (I, J) are actually associated to a given partition of the probe P , constituting the best previous match for the current point $D(m_i, p_j)$ in this column m_i and row p_j . The step of scanning back the matrix to retrieve the best previous match is then substituted with a list that returns the best element in a logarithmic time. Several additional definitions must be

provided to manage the candidate list. These concepts are all of them based on an analytical description of the computation of the matrix D .

The contribution f of a match point (m_I, p_J) to future points in a given column m_i ($I < i$) on the interval $x \in (p_J, p_P]$ can be divided into the components associated to λ and μ . At the same time, each one can be split into the values that depend on the current m_i and those that were already computed when the match point (m_I, p_J) was reached:

$$\begin{aligned} f_{I,J}^i(x) &= \mu |C_{I,J}^i(x)| + B_{I,J}^i & \text{where} \\ C_{I,J}^i &= m_i + \Delta_{I,J}, & \Delta_{I,J} = p_J - m_I \\ B_{I,J}^i &= \lambda i + E_{I,J}, & E_{I,J} = D(I, J) - \lambda(I + 2). \end{aligned} \quad (3.36)$$

It is direct to see that $\text{contrib}_{I,J}(i, j) = f_{I,J}^i(p_j)$, as the terms in Equation 3.33 have been simply rearranged. For the μ factor, $(m_i - m_I) - (p_j - p_J) = m_i + (p_J - m_I) - x = m_i + \Delta_{I,J} - x$. For the λ factor, $D(I, J) + \lambda(i - I - 2) = D(I, J) - \lambda(I + 2) + \lambda i = \lambda i + E_{I,J}$. In this case, the values $\Delta_{I,J}$ and $E_{I,J}$ do not depend on i , being already precomputed.

The new contribution of a match point (m_I, p_J) to the next position (column m_{i+1}) is easily computed from its contribution to the previous one:

$$f_{I,J}^{i+1}(x) = f_{I,J}^i(x - \Delta m_i) + \lambda \quad (3.37)$$

The updating consists of two changes: (1) a unit of λ is increased because a new site has not been included in the alignment (m_i); (2) the physical position of the match point (m_{i+1}, x) must be updated in the computation of the μ factor by decreasing x with Δm_i to recover the new value of the μ penalty.

As shown in Figure 3.22 (A), a given function $f_{I,J}^i$ can be represented graphically. The minimum value that can be reached is $B_{I,J}^i$ corresponding to the point $x = C_{I,J}^i$. For the rest of x values, the λ penalty and $D(I, J)$ are the same so that changes depend directly from the μ penalty. This value will decrease as long as x approaches the $m_I - m_i$ vertical until $C_{I,J}^i$. From that point, it will increase again due to the progressive movement away such point (see Figure 3.22 (B)). The contribution in the next column can also be represented as a similar function with the corresponding new values of x and $f(x)$.

Let m_i be the current column: each previous match point (m_I, p_J) has a different contribution to each one of the new match points (m_i, x) found in this column. For a given x , the best contribution of the previous match points in the alignment ending at such point is the minimum value among the $f_{I,J}^i(x)$ functions. In addition, the optimal value $D(i, j)$ is either the distance of an alignment containing only this match point or the alignment with the previous match point that showed the highest contribution for (m_i, x) :

$$\begin{aligned} P^i(x) &= \min_{(I,J) \in \text{Matchpoints}} f_{I,J}^i(x) \\ &\quad I < i, J < j \\ D(i, j) &= \min(\lambda(P - 1), P^i(p_j)). \end{aligned} \quad (3.38)$$

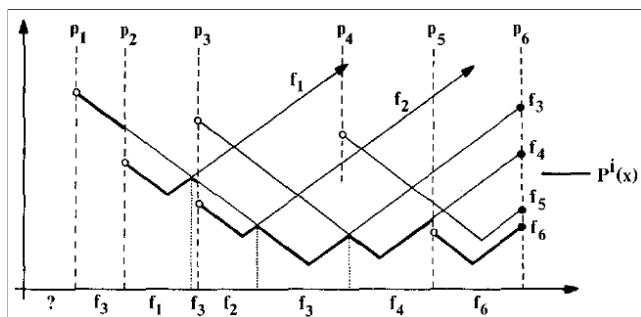


Figure 3.23 An illustration of an i-profile. Adapted from Myers and Huang (1992).

An i-profile is then defined to be the intersection between the contributions of all of the available match points computed before. For each interval between two consecutive points in P , the f function with the lowest value over there is claimed to be the owner of such interval. The calculation of the value $P^i(x)$ consists on locating the representative of an interval to know its contribution. In Figure 3.23, the i-profile represented as the minimum envelope of the f -curves of all match points left to the column m_i is graphically shown.

For simplicity, the list of candidates that form an i-profile is decomposed into two different lists L and R which correspond to the parts of the f -curves that are before and after the point $C_{I,J}^i$. For each one, insertions and updates to the list of candidates must be performed in a slightly different form (see Figure 3.24).

In the case of the R -list, their members are always increasing straight lines. For that reason, whenever one of the candidates has another candidate below, this first element is said to be dominated by the second one. When this candidate is dominated in all of the intervals over P , it becomes dead and it is removed from the list. In the case of the L -list, the processing must take into account the stationarity of the left ends of the curves when shifting horizontally.

For both lists the management is similar: each match point that has just been computed must be inserted into the L -list and the R -list. This insertion can cause the removal of the match points that become dominated by this new element. Similarly, once a column m_i has been processed, the elements of the lists must be processed to be ready for the next column m_{i+1} , involving the recomputation of their values using Δm_i . Again, this operation can force some match points to be removed from the list because they can not contribute positively to any of the future ones.

These type of sorted lists that must provide a direct access to a given element (e.g. the owner of an interval) can be implemented using balanced trees. These trees support logarithmic insertion, deletion and search primitives. Let M and P the length of the map and the probe respectively, there are $R = MP$ potential match points. For each one, the owner of its interval can be retrieved in a logarithmic time $O(\log(P))$. The insertion in the list and the shifting operation are also performed with a logarithmic cost taking into account some particular considerations. Thus, the final cost of this algorithm is $O(R \log(P))$ (for further details see Myers and Huang (1992)).

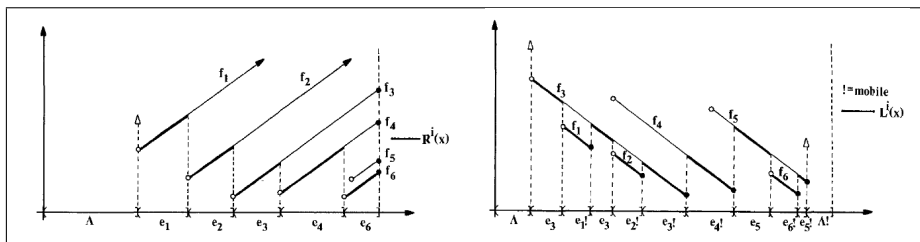


Figure 3.24 An illustration of a R-profile and a L-profile. Adapted from Myers and Huang (1992).

Such an algorithm was designed primarily for comparisons between a map and a probe. However, Myers and Huang also presented some changes to convert the problem in a comparison between two maps of length M and N respectively in $O(MN(\log M + \log N))$ time.

Implementation

The main algorithm of the Myers and Huang (1992) strategy consists of a loop that visits column by column the $m_M \times p_P$ matrix. For each site in m , there is a function $\text{Match}(\text{site}(a_i))$, precomputed only once at the beginning, which returns the sites x in P that share the same restriction enzyme.

Then, the optimal alignment ending at every new match point (m_i, x) is constructed between either an alignment only constituted by this match or the contribution of the owner of its interval, which is directly identified with the function Find_min by accessing the list of candidates implemented as two balanced trees (the L-list and R-list).

The new match points that have been processed in the current column are then inserted in the corresponding lists with the function Insert , removing from the lists those candidates that are categorized as dead.

Once the current column has been completely processed, both lists of candidates must be updated with the function Update to be prepared for the next element m_{i+1} , taking into account the value of Δm_i .

Bibliography

- B. Alberts, D. Bray, J. Lewis, M. Raff, K. Roberts, and J. D. Watson. *Molecular biology of the cell*. Garland publishing, third edition, 1994. ISBN 0-8153-1620-8.
- S.F. Altschul, W. Gish, W. Miller, E.W. Myers, and D.J. Lipman. Basic local alignment search tool. *Journal of Molecular Biology*, 215:403–10, 1990.
- S.F. Altschul, T.L. Madden, A.A. Schaffer, J. Zhang, Z. Zhang, W. Miller, and D.J. Lipman. Gapped BLAST and PSI-BLAST: a new generation of protein database search programs. *Nucleic Acids Research*, 25:3389–3402, 1997.

```

Pre  $\equiv$  M, P: maps;  $\lambda, \mu \in \mathbb{Z}^+$ 
Initialize_candidate_list();
(* Current column  $m_i$  *)
for  $i = 0$  to  $|M_m| - 1$  do
    for  $j = 0 \in \text{Match}(\text{site}(m_i))$  do
5:      $D(i, j) \leftarrow \min(\lambda(P - 1), \text{Find\_min}(i, j));$ 
    if  $i < M$  then
        for  $j \in \text{Match}(\text{site}(a_i))$  and  $j < P$  do
            Insert( $i, j$ );
        Update( $i$ );

```

Figure 3.25 The Myers and Huang map alignment algorithm.

- A. Apostolico and C. Guerra. The longest common subsequence problem revisited. *Algorithmica*, 2: 315–336, 1987.
- T. L. Bailey and C. Elkan. Fitting a mixture model by expectation maximization to discover motifs in biopolymers. *Proceedings of the 2nd International Conference on Intelligent Systems for Molecular Biology (ISMB)*, pages 28–36, 1994.
- R. Bellman. *Dynamic programming*. Princeton University Press, Boston, USA, 1957.
- W.A. Beyer, P.H. Sellers, and M.S. Waterman. Stanislaw m. ulam’s contributions to theoretical theory. *Letters in Mathematical Physics*, 10:231–242, 1985.
- A. Brazma, I. Jonassen, I. Eidhammer, and D. Gilbert. Approaches to the automatic discovery of patterns in biosequences. *Journal of Computational Biology*, 5:279–305, 1998.
- H. Carrillo and D. Lipmann. The multiple sequence alignment problem in biology. *SIAM Journal of Applied Mathematics*, 48:1073–1082, 1988.
- T.H. Cormen, C.E. Leiserson, R.L. Rivest, and C. Stein. *Introduction to Algorithms*. MIT Press and McGraw-Hill, second edition, 2001. ISBN 0-2620-3293-7.
- M.O. Dayhoff, R.V. Eck, M.A. Chang, and M.R. Sochard. *Atlas of protein sequence and structure*, volume 1. National Biomedical Research Foundation, Silver Spring, Maryland, 1965.
- S. Dreyfus. Richard bellman on the birth of dynamic programming. *Operations Research*, 50:48–51, 2002.
- J. Dumas and J. Ninio. Efficient algorithms for folding and comparing nucleic acid sequences. *Nucleic Acids Research*, 10:197–206, 1982.
- R. Durbin, S. Eddy, A. Crogh, and G. Mitchison. *Biological Sequence Analysis: Probabilistic Models of Protein and Nucleic Acids*. Cambridge University Press, first edition, 1998. ISBN 0-521-62971-3.
- D. Eppstein, Z. Galil, and R. Giancarlo. Speeding up dynamic programming. *IEEE Symposium on Foundations of Computer Science*, pages 488–496, 1988.
- D. Feng and R.F. Doolittle. Progressive sequence alignment as a prerequisite to correct phylogenetic trees. *Journal of Molecular Evolution*, 25:351–360, 1987.
- W.M. Fitch and E. Margoliash. Construction of phylogenetic trees. *Science*, 155:279–284, 1967.

- W.M. Fitch and T.F. Smith. Optimal sequence alignments. *Proceedings of the National Academy of Sciences*, 80:1382–1386, 1983.
- A.J. Gibbs and G.A. McIntyre. The diagram, a method for comparing sequences. its use with amino acid and nucleotide sequences. *European Journal of Biochemistry*, 16:1–11, 1970.
- W.B. Goad and M.I. Kanehisa. Pattern recognition in nucleic acid sequences i. a general method for finding local homologies and symmetries. *Nucleic Acids Research*, 10:247–278, 1982.
- O. Gotoh. An improved algorithm for matching biological sequences. *Journal of Molecular Biology*, 162:705–708, 1982.
- E.J. Gumbel. *Contributions to order statistics.*, chapter Statistical theory of extreme values, page 71. Wiley, New York, USA, 1962.
- R.W. Hamming. *Journal of Bell Systems Technology*, 26:147, 1950.
- D.S. Hirschberg. A linear space algorithm for computing longest common sequences. *Communications of the ACM*, 18:341–343, 1975.
- X. Huang and M. S. Waterman. Dynamic programming algorithms for restriction map comparison. *Bioinformatics*, 8:511–520, 1992.
- S. Karlin and S.F. Altschul. Methods for assessing the statistical significance of molecular sequence features by using general scoring schemes. *Proceedings of the National Academy of Sciences*, 87: 2264–2268, 1990.
- J. R. Knight and E. W. Myers. Super-pattern matching. *Algorithmica*, 13:211–243, 1995.
- C. E. Lawrence, S. F. Altschul, M. S. Boguski, J. S. Liu, A. F. Neuwald, and J. C. Wootton. Detecting subtle sequence signals: a gibbs sampling strategy for multiple alignment. *Science*, 262:208–214, 1993.
- VI. Levhenshtein. Binary codes capable of correcting deletions, insertions, and reversals. *Cybernetics and Control Theory*, 10:707–710, 1966.
- D.J. Lipman and W.R. Pearson. Rapid and sensitive protein similarity searches. *Science*, 227:1435–1441, 1985.
- J. Meidanis and J.C. Setubal. *Introduction to computational molecular biology*. PWS Publishing Company, Boston, first edition, 1997. ISBN 0-534-95262-3.
- W. Miller and E.W. Myers. Sequence comparison with concave weighting functions. *Bulletin of Mathematical Biology*, 50:97–120, 1988.
- W. Miller, J. Ostell, and K.E. Rudd. An algorithm for searching restriction maps. *CABIOS*, 3:247–252, 1990.
- D.W. Mount. *Bioinformatics: sequence and genome analysis*. Cold Spring Harbor Laboratory Press, first edition, 2001. ISBN 0-87969-608-7.
- E.W. Myers. An overview of sequence comparison algorithms in molecular biology. *Technical report TR 91-29, University of Arizona, Tucson, Department of Computer Science*, pages 1–25, 1991.
- E.W. Myers and X. Huang. An $o(n^2 \log n)$ restriction map comparison and search algorithm. *Bull. Math. Biol.*, 54:599–618, 1992.
- E.W. Myers and W. Miller. Optimal alignments in linear space. *CABIOS*, 4:11–17, 1988.

- S. B. Needleman and C. D. Wunsch. A general method to search for similarities in the amino acid sequence of two proteins. *Journal of molecular biology*, 48:443–453, 1970.
- C.A. Ouzounis and A. Valencia. Early bioinformatics: the birth of a discipline – a personal view. *Bioinformatics*, 19:2176–2190, 2003.
- W.R. Pearson and D.J. Lipman. Improved tools for biological sequence comparison. *Proceedings of the National Academy of Sciences*, 85:2444–2448, 1988.
- A. Pertsemlidis and J.W. Fondon. Having a blast with bioinformatics (and avoiding blastphemy). *Genome Biology*, 2:2002, 2001.
- N. Saitou and M. Nei. The neighbor-joining method: a new method for reconstructing phylogenetic trees. *Molecular Biology and Evolution*, 4:406–425, 1987.
- D. Sankoff and J.R. Kruskal. *Time warps, string edits, and macromolecules: The theory and practice of sequence comparison*. Addison-Wesley, Don Mills, Ontario, 1983. ISBN 1-57586-217-4.
- W. Schmitt and M.S. Waterman. Multiple solutions of dna restriction mapping problems. *Advances in Applied Mathematics*, 12:412–427, 1991.
- T.D. Schneider and R.M. Stephens. Sequence logos: a new way to display consensus sequences. *Nucleic Acids Research*, 18:6097–6100, 1990.
- P. Sellers. On the theory and computation of evolutionary distances. *SIAM Journal of applied Mathematics*, 26:787–793, 1974.
- P. Sellers. The theory and computation of evolutionary distances: pattern recognition. *Journal of Algorithms*, 1:359–373, 1980.
- P. Sellers. Pattern recognition in genetic sequences by mismatch density. *Bulletin of Mathematical Biology*, 46:501–514, 1984.
- T.F. Smith and M.S. Waterman. Comparison of biosequences. *Advances in Applied Mathematics*, 2: 482–489, 1981a.
- T.F. Smith and M.S. Waterman. Identification of common molecular subsequences. *Journal of Molecular Biology*, 147:195–197, 1981b.
- T.F. Smith, M.S. Waterman, and W.M. Fitch. Comparative biosequence metrics. *Journal of Molecular Evolution*, 18:38–46, 1981.
- J. D. Thompson, D. G. Higgins, and T. J. Gibson. Clustalw: improving the sensitivity of progressive multiple sequence alignment through sequence weighting, position-specific gap penalties and weight matrix choice. *Nucleic Acids Research*, 22:4673–4680, 1994.
- E.C. Tyler, M.R. Horton, and P.R. Krause. A review of algorithms for molecular sequence comparison. *Computers and Biomedical Research*, 24:72–96, 1991.
- S.M. Ulam. *Applications of number theory to numerical analysis.*, pages 1–3. Academic Press, New York, USA, 1972.
- M. S. Waterman and M. Eggert. A new algorithm for best subsequence alignments with application to trna-rRNA comparisons. *Journal of Molecular Biology*, 197:723–728, 1987.
- M. S. Waterman, T. F. Smith, and H. L. Katcher. Algorithms for restriction map comparisons. *Nucleic acids research*, 12:237–242, 1984.

- M.S. Waterman. Efficient sequence alignment algorithms. *Journal of Theoretical Biology*, 108:333–337, 1984a.
- M.S. Waterman. General methods of sequence comparison. *Bulletin of mathematical biology*, 46: 473–500, 1984b.
- M.S. Waterman. *Introduction to computational biology*. Chapman and Hall, UK, 1995. ISBN 0-412-99391-0.
- M.S. Waterman, J. Joyce, and M. Eggert. *Phylogenetic Analysis of DNA Sequences*, chapter “Computer alignment of sequences”, pages 59–72. Oxford University Press, 1990.
- M.S. Waterman, T.F. Smith, and W.A. Beyer. Some biological sequence metrics. *Advances in Mathematics*, 20:367–387, 1976.
- P.S. White and T.C. Matise. *Bioinformatics: A Practical Guide to the Analysis of Genes and Proteins.*, chapter “Mapping Databases.”, pages 25–54. John Wiley & Sons Inc., New York, USA, 2005. ISBN 0-471-47878-4.

Chapter 4

Computational Gene and Promoter Characterization

Summary

The computational identification of genes in an eukaryotic genome and the description of their promoter regions are reviewed here. An important fraction of the information used by the cell to activate the genes and to recognize their protein-coding regions is contained in the genomic sequences. The methods to represent such cellular signals and to detect functional regions presenting unusual statistical content are similar in both cases. This chapter introduces the different alternatives proposed throughout the past years, providing a glimpse of the future.

4.1 Genes and promoters	88
4.2 Computational approaches	95
4.3 Detection of signals	96
4.4 Content recognition	101
4.5 Sequence comparison	103
4.6 The state of the art in gene identification	107
4.7 The state of the art in promoter characterization	111
4.8 Looking forward	113

4.1 Genes and promoters

Towards a catalogue of the genome

ONE OF THE MAJOR PROBLEMS THAT BIOLOGISTS HAVE EVER FACED is how to extract relevant information from millions of nucleotides produced by large-scale genome sequencing projects. The first task is to locate all protein-coding genes encoded in the genomic sequence to able then to characterize the regulatory content of the genome (Blanco and Guigó, 2005).

Genes are switches regulated by cellular mechanisms which turn them on or off according to different situations and circumstances. The identification of the promoter elements required for the correct expression of genes is crucial to understand why many genetic diseases are caused and perhaps, how to prevent or stop them.

Computational gene-finding and promoter characterization have been traditionally strongly related. Both methods process the genomic sequence using similar techniques in order to extract the information that is used by the cells to control the production of genes. However, the elaboration of catalogues of genes in eukaryotes have shown to be more feasible in practice than the construction of regulatory maps because of the specific nature of each problem. Nonetheless, promoters are still very interesting for gene-finding because their detection will help to improve the accuracy of current gene predictions. Therefore, the complete annotation of a gene should include both the protein-coding regions and the promoter elements that govern its expression (Pedersen et al., 1999).

Eukaryotic gene structure

The identification of genes is difficult, specially because of their fragmented nature and the large spacers found between them. Only 2% of the 3,000 million nucleotides in the human genome are estimated to code for proteins (Venter et al., 2001).

As explained in Chapter 2, the splicing machinery removes from the transcript those regions that are not coding for proteins (introns), joining the coding fragments (exons). The mRNA is constituted of the coding sequence (CDS) and the untranslated region (UTR). For further details about the general structure of an eukaryotic gene see Figure 4.1.

Most gene computational tools can only predict the location of the coding exons of a gene. Essentially, the splicing and translation signals are first located in order to construct then the possible reading frames that form the exons. Typically, there are four types of exon-defining signals:

- ① Start codons: the first amino acid of a protein is usually the Methionine, coded with the codon ATG. It represents the beginning of a translation.
- ② Stop codons: there are three codons (TAA, TAG and TGA) that end the translation of a mRNA.

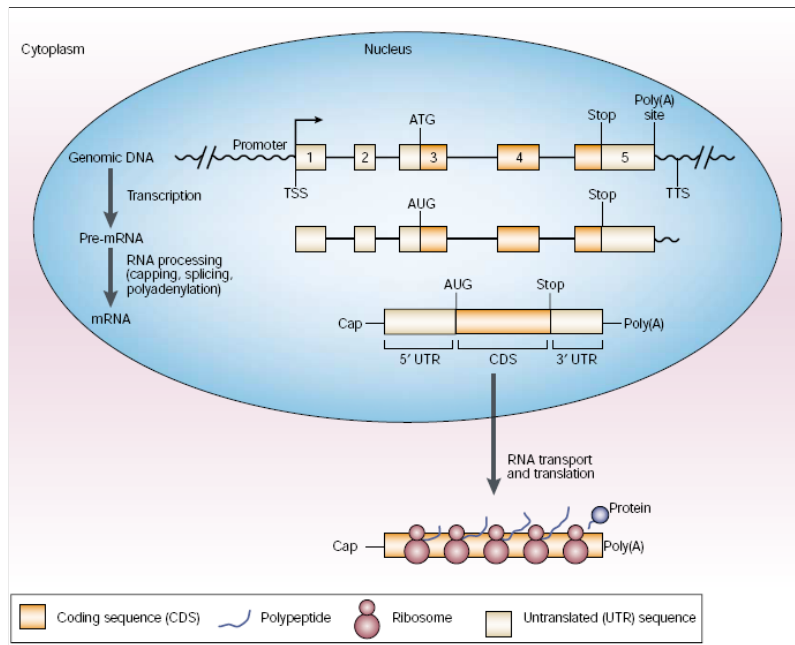


Figure 4.1 The typical gene structure. TSS is the transcription start site. TTS is the transcription termination site. ATG/AUG is the translation start codon. Adapted from Zhang (2002).

- ③ Acceptor splice site: the right part (3') of a removed intron contains this signal. It represents the nucleotides immediately before the beginning of an exon.
- ④ Donor splice site: the left part (5') of a removed intron contains this signal. It represents the nucleotides immediately after the end of an exon.

With such signals, the following types of exons can be defined:

- ① Initial exons (Start codon - Donor site): the first coding exon of a gene
- ② Internal exons (Acceptor site - Donor site): the set of coding exons between the initial and the terminal ones
- ③ Terminal exons (Acceptor site - Stop codon): the last coding exon of a gene

It is important to mention that, due to the existence of exons completely or partially constituting the UTR region at both ends of a gene, the initial and terminal coding exons predicted by a computational approach do not usually correspond to the authentic ends of the transcript.

Other forms of gene structures

Gene identification is not an easy problem. Nowadays, there are still serious discussions to establish the exact number of genes in an organism. One of the reasons for this controversy is the definition of what a gene is. Exceeding the classical definition “one gene for one protein”, biological reality has shown how things are more complex. A better biological understanding of these facts will help to obtain in the future more accurate gene predictions (Pennisi, 2003).

These are other forms of gene structures that exceed the classical definition of a gene:

- Alternative spliced genes: 60 % of human genes can be spliced following different patterns of exons and introns, omitting some exons or altering the length of others to produce different proteins (Ladd and Cooper, 2002). See Figure 4.2 (A) for an example of alternative splicing.
- Pseudogenes: due to the continually changing nature of the genomes, some genes have been inactivated by excess of mutations (conventional pseudogene). Processed pseudogenes are the result of the insertion in the genome of a reversed-transcribed mRNA copy of a gene. See Figure 4.2 (B) for an example.
- Intronless genes: genes without introns (prokaryotic origin).
- Non-coding genes: some genes correspond to specific RNA molecules playing crucial roles in the cell that are not translated into a protein.
- Non-canonical spliced genes: splicing signals in most genes present certain dinucleotides as characteristic signatures. However, other types of splicing signals occurring in a minority of genes are recognized by a different splicing machinery (Burset et al., 2000).
- Genes-within-genes: some human genes have been found to be within long introns of others. These internal genes can be affected by the normal splicing process as well (Brown, 2002).
- Selenoproteins: some codons can be translated into different amino acids according to each situation (context-dependent codon reassignment). For instance, in presence of a secondary structure in the mRNA called SECIS, the codon TGA is translated into the novel amino acid Selenocysteine instead of stopping the process (Low and Berry, 1996).

Eukaryotic promoter structure

The expression of a gene is the appearance of an observable feature or action caused by the effect of the protein encoded by this gene. Gene regulation is the mechanism which determines the amount of protein product that must be synthesized by switching the genes responsible for that protein on or off. Only a subset of genes in an eukaryotic cell are

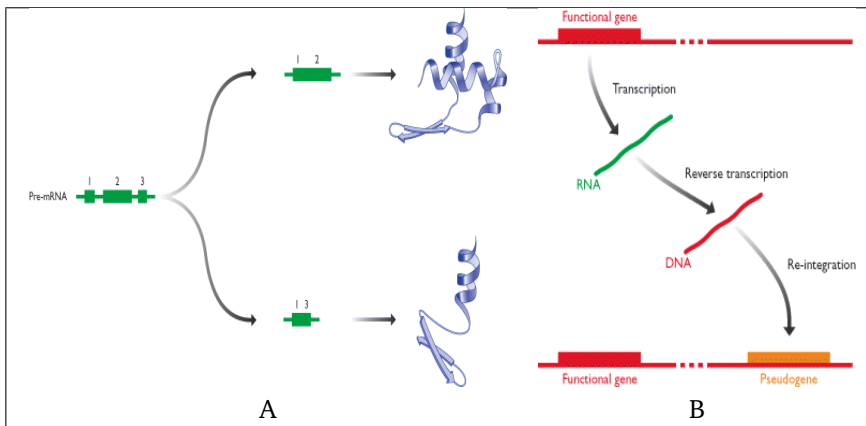


Figure 4.2 Other forms of gene structures. (A) Alternative splicing results in different combinations of exons from the same pre-mRNA. (B) The origin of a processed pseudogene. Adapted from Brown (2002).

expressed at each instant, considerably changing this regulational composition during the life cycle.

But research about gene expression is not trivial: a human cell can be seen in terms of a black box with approximately 20,000 inputs, one per gene. Such box must work with $2^{20,000}$ states, since every gene would be either on or off. This number can be approached to $10^{6,000}$ while the number of particles in the universe is believed to be about 10^{80} . Moreover, the degrees of intensity and the large network of relationships among related genes are neglected in this estimation.

In fact, little is known about the relationship, for instance, between transcription and splicing. More and more evidences are being gathered to postulate that both processes are in fact performed simultaneously or at least in a very intimate manner (Kornblihtt, 2005).

Checkpoints in the pathway from DNA to protein

There are actually two levels of gene expression control along the pathway from DNA to RNA to protein (Brown, 2002). The primary level selects which genes have to be expressed and which not and belongs to the process of transcription (see Figure 4.3). The second level is necessary to modulate the expression of a gene by changing the rate of production or by modifying the nature of the product (RNA, protein) using post-transcriptional methods.

Specifically, this control is implemented through different stages:

- ① Accessibility: What regions of a chromosome are visible for being transcribed
- ② Transcriptional control: When and how often a given gene is transcribed.
- ③ RNA processing control: How the primary transcript is spliced.
- ④ RNA transport control: Which mRNAs are exported to the cytoplasm.

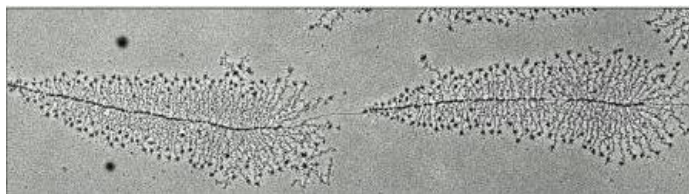


Figure 4.3 Transcription of two tandem genes as observed under the electron microscope. Each gene is being transcribed simultaneously by hundreds of RNA-polymerase II. Adapted from [Alberts et al. \(1994\)](#).

- ⑤ RNA translational control: Which mRNAs are translated by ribosomes.
- ⑥ RNA degradation control: Which and when mRNAs have to be destroyed.
- ⑦ Protein activity control: (In)activating synthesized protein molecules.

Transcriptional regulation: promoters

Transcriptional regulation is a highly dynamic process. Most of genes are governed by variable temporal and spatial heterogeneous profiles. The promoter sequences are functional regions located immediately upstream the transcription start site of the gene (TSS). Many genes usually possess several alternative TSSs, having therefore different promoter regions. The main function of a promoter is the integration of information about the status of the cell, to alter the rate of transcription of a single gene accordingly ([Wray et al., 2003](#)).

In Figure 4.4, a promoter prototype is represented as a gene specific container for the assembly of some special proteins called transcription factors (TFs). The TFs are responsible for recruiting the RNA-polymerase II that performs the transcription from DNA into RNA molecules. Every gene is regulated by a core of general TFs and a combination of gene-specific TFs located upstream the TSS. About 1,800 different TFs are estimated to be encoded in the human genome ([Venter et al., 2001](#)).

The TFs are attracted to the promoter region by very specific motifs imprinted in the DNA called TF binding sites (TFBSs). From the study of a well-characterized set of eukaryotic promoters, the occupation of a promoter has been estimated to be about 10 to 50 TFBSs for 5 to 15 different TFs ([Wray et al., 2003](#)). TFs are usually arranged along the promoter region following very restrictive rules such as minimum/maximum distance or neighbourhood constraints ([Pedersen et al., 1999](#); [Werner, 2000](#)).

The problem of finding regulatory elements is extremely difficult due to many reasons ([Fickett and Hatzigeorgiou, 1997](#)):

- There are thousands of different TFs.
- TFBSs are short: typically 5-15 nucleotides long.
- Each TF can connect to more than one different binding site.

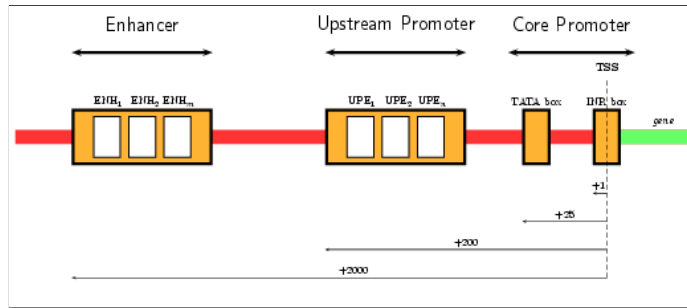


Figure 4.4 A schematic representation of a promoter.

- Each TFBS can recruit different TFs.
- The core promoter is not universal, presenting high diversity as well.
- TFBSs can form clusters of regulatory modules or composites.
- The poor knowledge about the biological interactions between different TFs.

Eventually, some regulatory regions called enhancers are located within intergenic segments, being able to affect several loci in other parts of the genome. First exons and introns are also known to contain some regulatory signals as well. In addition, other promoter regions control the coordinate expression of two bidirectional genes, that is, gene pairs that are arranged head-to-head on opposite strands with less than 1,000 nucleotides separating the TSSs (Trinklein et al., 2004).

Chromatine structure and gene expression

In Eukaryotes the chromatin is packaged into a compact structure with the aid of a class of proteins called histones. The nucleosomes, the fundamental packaging units, are histones with DNA wrapping around (Alberts et al., 1994). Chromatin packaging plays an important function of regulation before the beginning of the transcription. To be transcribed, a promoter must be physically accessible to the RNA polymerase for starting the copy (see Figure 4.5).

If a region containing a gene is not momentarily accessible, that gene is said to be silenced. RNA polymerases can transcribe a region containing attached nucleosomes when they are moved slightly by thermal effects. This process allows the polymerase to copy short regions of DNA while the nucleosome shifts to a position near the end of the transcription. Thus, nucleosome positioning and distribution of genes into visible and not visible regions of chromatin are some types of pre-transcriptional control (Brown, 2002).

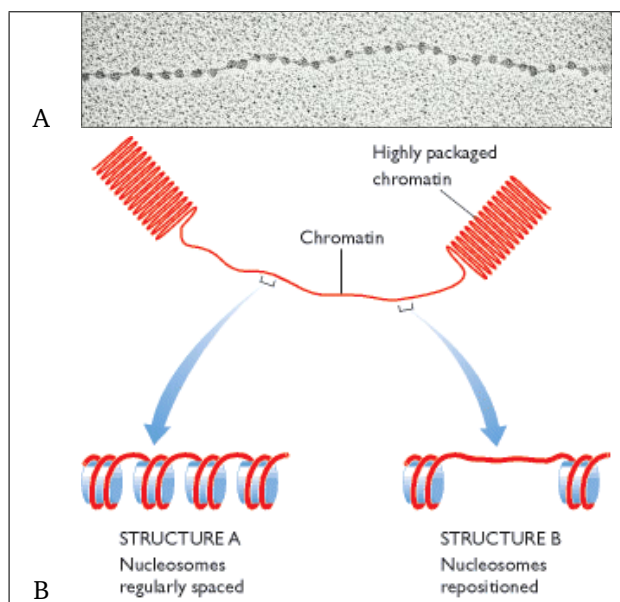


Figure 4.5 Nucleosomes and chromatin structure can influence gene expression. (A) Nucleosomes as seen in the electron microscope. Adapted from (Alberts et al., 1994). (B) A region of unpackaged chromatin in which the genes are accessible is flanked by two more compact segments. On the left, the nucleosomes have regular spacing structure. On the right, the nucleosome positioning has changed and a short stretch of DNA is exposed for transcription. Adapted from Brown (2002).

Methylation and CpG islands

In eukaryotes, Cytosine bases in CpG dinucleotides from chromosomal DNA molecules are sometimes modified with the addition of methyl groups by special enzymes which maintain this feature through the offspring of a cell. Such process is named methylation. The inheritance of methylation patterns is a feasible explanation to the cell memory event and is also associated with repression of gene activity.

Some correlation between the degree of methylation and the level of transcription of genes has been observed. Methylation is thought to be related with the way histones move and stand along the DNA molecules of chromatin and therefore with the silencing of genes as well (Brown, 2002).

CpG islands are regions of several hundreds of nucleotides in which the frequency of the dinucleotide CpG and the G+C content are higher than the average for the rest of genome (Antequera and Bird, 1993). Most of the CpG islands in the human genome are methylated. However, the CpG islands that are adjacent to housekeeping genes¹ are unmethylated, being the genes potentially active.

¹Genes that are expressed generally in every phase of the cell cycle.

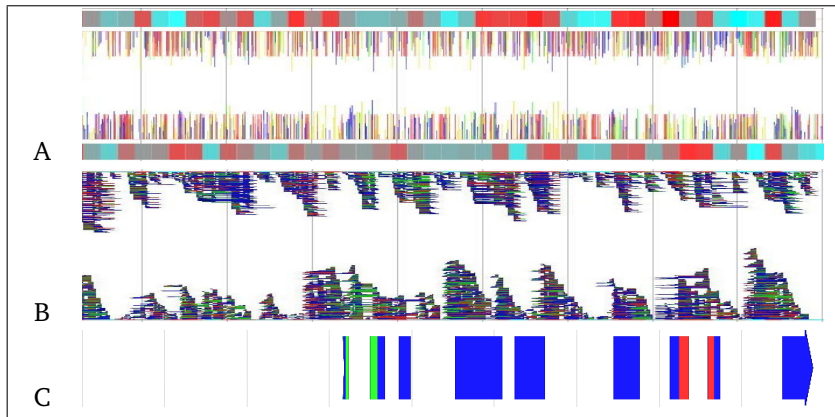


Figure 4.6 Sources of information in the ab-initio gene-finding process (in both strands). (A) Signal and content information: vertical bars are predicted splicing signals; the red-blue code measures the coding potential of the sequence. (B) Predicted set of coding exons. (C) Optimal gene structure assembled from the set of predicted exons with a dynamic programming algorithm.

4.2 Computational approaches

Gene identification and promoter characterization methods essentially process similar input sequences with many common algorithmic approaches. However, the underlying biological problem is slightly different. The genes are regular structures formed by exon-defining signals with several exon features usually well conserved. The promoter regions instead are more flexible arrangements of TFBSs which, in addition, present a higher variability in their motifs.

Gene-finding methods normally use three different types of information to build a prediction: splice sites and translational signals, protein-coding potential measures, and similarity searches. Ab initio methods only rely on the investigation of the statistical properties of annotated coding sequences: signals and coding statistics. As shown in Figure 4.6, the combination of signals and content measures with an assembly algorithm of exons, typically based on dynamic programming, produces a predicted gene (Haussler, 1998; Stormo, 2000b). Homology methods compare directly the sequence of interest to known coding sequences or even orthologous regions of other genomes using alignment programs.

Promoter characterization methods are often based on the detection of the motifs specifying a family of TFBSs. A combinatorial set of rules can be designed to propose arrangements of sites in groups of few elements (composites or modules). There is a severe lack of biological knowledge about the promoter structures (Fickett and Hatzigeorgiou, 1997; Fickett and Wasserman, 2000). Despite this, promising advances have been obtained using homology methods based on the phylogenetic conservation of regulatory elements and the introduction of high-throughput expression data (Blanco and Guigó, 2005).

4.3 Detection of signals

Sequence signals or sites are defined as short, functional DNA elements involved in gene specification or transcriptional regulation. There is not a typical unique sequence of nucleotides that can be associated to each class of signal. Nonetheless, certain trends in the conservation of some base pairs in these motifs are usually detected, being statistically measured.

Because of the importance of these signals to characterize genes and promoter regions, an important family of techniques based on the use of an external catalogue of known examples have been designed for their detection: the pattern-driven algorithms (Brazma et al., 1998), also called the search by signal approaches (Blanco and Guigó, 2005).

A naive procedure for scanning a genomic sequence suspicious to contain a functional element will always produce an enormous list of false positives due to the short length of most genomic signals and the high probability to find the same subsequence by chance in other region. To circumvent this problem, the pattern-driven algorithms usually rely on three steps:

- ① The construction of a catalogue of experimentally annotated sites of a given class
- ② The representation of this set of examples to mask their variability without losing information
- ③ The detection of new sites in other sequences using those representations of real examples, as in the algorithm shown in Figure 4.7.

Construction of a catalogue

Pattern-driven methods need an input set of real (annotated) elements to build a profile that represents such a family of signals. These samples are usually extracted from public databases of annotated gene and promoter regions.

A high-quality collection of exons extracted from the genome browsers annotations must be used to compile a set of real splicing and translation signals. Typically, the real signals are extracted from the boundaries of the exons, while a set of false signals is built from any similar sequence detected in the introns (see Burset and Guigó (1996); Rogic et al. (2001) for an example of construction of evaluation sets).

Due to the lack of experimental high-throughput methods to verificate and annotate regulatory functions, the amount of real regulatory signals is very small in comparison to the exon-defining ones. Despite this, several regulatory catalogues are available such as the databases TRANSFAC (Matys et al., 2003, see Web Glossary, page 244), JASPAR (Sandelin et al., 2004, see Web Glossary, page 242) or PROMO (Farre et al., 2003, see Web Glossary, page 243). New regulatory databases specifically oriented to the training of computational tools are emerging now, such as the Cold Spring Harbor Laboratory Mammalian promoter database (Xuan et al., 2005, see Web Glossary, page 241) or the ABS database of orthologous TFBSs (Blanco et al., 2006, see Web Glossary, page 241).


```

Pre  $\equiv$  S: sequence; M: signal model; L, STEP, T: integer;

i  $\leftarrow$  1;
j  $\leftarrow$  i + L;
(* Apply the model on each window of length L *)
5: while i  $\leq$  |S| - L + 1 do
    (* Evaluate the current candidate with this model *)
    score  $\leftarrow$  M(Si,j);
    (* Report the candidates above a quality threshold *)
    if score  $\geq$  T then
10:     ReportCandidate(Si,j,score);
    i  $\leftarrow$  i + STEP;

```

Figure 4.7 Pattern-driven algorithms.

A correct annotation of the TSS is also crucial for the correct extraction of the promoters. However, such a signal has been poorly characterized so far, being in practice useless to predict its location by computational means. The EPD (Perier et al., 2000, see Web Glossary, page 242) and the DBTSS (Suzuki et al., 2004) databases maintain collections of experimentally determined TSSs.

Representation of functional sites

Representing a biological signal site as a unique string is very unrealistic. A large number of sequences containing the same signal (exon-defining or regulatory) represents a good statistical sample of the sequences that are likely to exist in the genome with the same function. However, the alignment of them will probably show differences in the context or even in the apparently best conserved positions of the core (see the example in Figure 4.8).

This limitation leads to a simple question: given a collection of biological signals, how to develop a representation or model to characterize them. Several data structures have been designed to retrieve enough information from the input sequences to be able to recognize putative sites in other sequences (see Osada et al. (2004); Stormo (2000a) for a review).

➤ Deterministic patterns:

- Consensus sequences: sequences constructed by selecting the nucleotide appearing more often at each position of the motif in the examples.

➤ Probabilistic patterns:

- Position weight matrices: a numerical representation that registers the frequency of each nucleotide at each position of the motif in the examples.
- Hidden Markov models: a stochastic procedure that registers the dependencies between each nucleotide and the previous group of k nucleotides at each position of the motif in the examples.

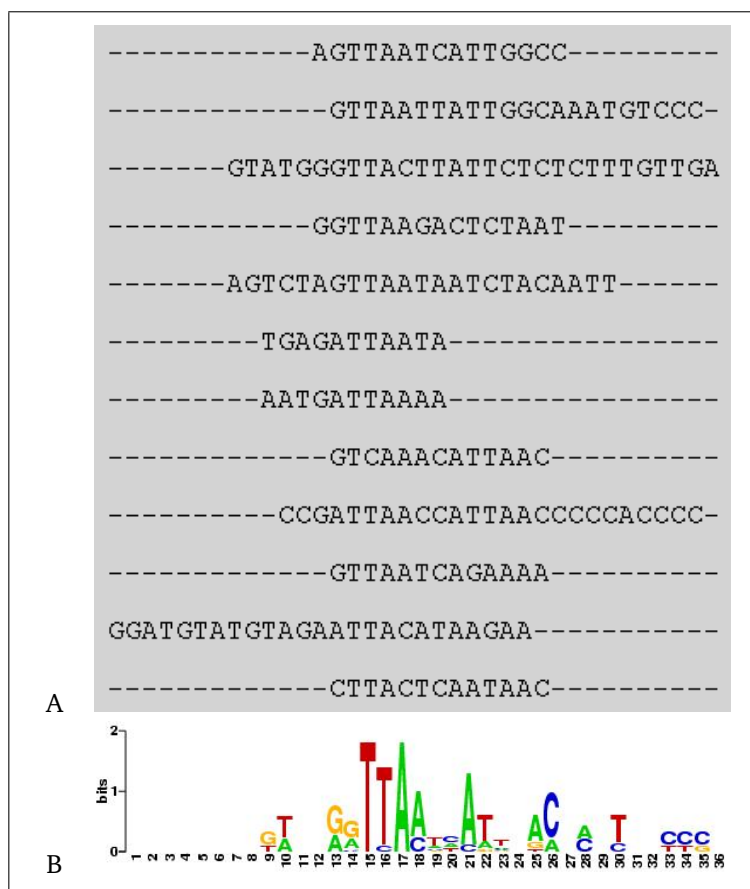


Figure 4.8 Alignment and representation of a set of TFBSs. (A) Global alignment of 12 human sites of HNF-1 α . (B) Sequence logo constructed from the multiple alignment.

⇒ Non-symbolic representations:

⇒ Neural networks: machine-learning methods that represent the stronger dependencies found in the examples with stronger connectivities in an artificial network.

Example: position weight matrices (PWMs)

Once a collection of real binding sites is aligned, a more sophisticated treatment of the information than a simple consensus sequence can be performed. PWMs² are two dimensional arrays of values that represent the score for finding each of the possible sequence characters at each position in the signal that is being analyzed (Staden, 1984).

Such a score is derived from the frequency of each nucleotide observed in a set of real

²PWMs are sometimes called Position-Specific Scoring Matrices (PSSMs).

<i>M</i>	1	2	3	4	5	6
A		6		3	4	
C			1		1	
G	1			3		
T	5		5		1	6
	T	A	T	A	A	T

MaxScore = 29, MinScore = 0

$$\text{Score}(s) = \frac{\sum_i M(s_i, i) - \text{MinScore}}{\text{MaxScore} - \text{MinScore}}$$

$s_1 = \text{TATATT} \rightarrow \text{score} = 26 / 29 = 0.89$
$s_2 = \text{TATGCA} \rightarrow \text{score} = 21 / 29 = 0.72$
$s_3 = \text{CGCTAT} \rightarrow \text{score} = 11 / 29 = 0.37$

Figure 4.9 A Position Weight Matrix. A naive scoring system is also presented. Three candidates are scored. Only the first one would be over a reasonable threshold of 85% of similarity to the original matrix.

functional sites (see Figure 4.9 for an example of PWM). Because some positions are more conserved than others, this is a flexible method to represent sites, under the hypothesis that different positions within the site make independent contributions to the total score. As the most conserved positions are supposed to be relevant for the biological activity of the site, any sequence that differs from the consensus will have a lower score proportional to the significance of the mismatching positions in the motif (Stormo, 2000a).

PWMs are used to score new sequences that could contain a signal of the same family (e.g. splice sites in Guigó et al. (1992) or promoter elements in Bucher (1990)). Each position of the matrix is a weight. Weights are employed to score every position of a candidate signal. The sum of these weights according to the content of such a sequence is the score of the candidate (see Figure 4.9).

There are several types of PWMs (Wasserman and Sandelin, 2004):

- Frequency matrices contain the absolute frequency of a nucleotide at each motif position
- Weight matrices contain the relative frequency of a nucleotide at a motif position as an estimation of the probability of this fact
- Log-likelihood ratio or log-odds matrices contain at each position the log of the quotient between the probability of finding a particular nucleotide at such a position position in sequences containing the real motif and the background frequency of the letter at the same position (usually computed from DNA random sequences). To eliminate null values, pseudocounts are usually added to every weight in the matrix.

PWM main drawbacks are two: first, the need for a threshold to filter candidates once the matrix has been used to search for putative sites in new sequences; second, the difficulty

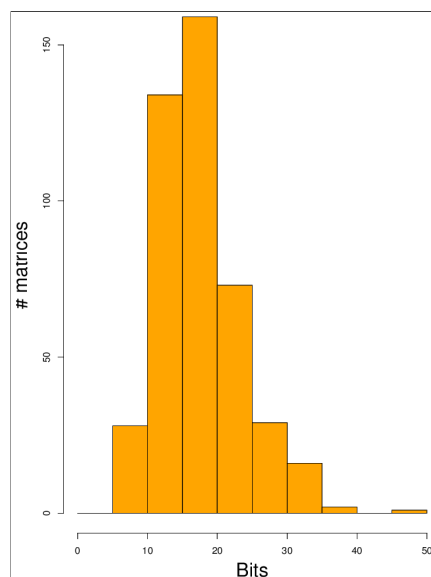


Figure 4.10 Information content of TRANSFAC 6.3 matrices.

to estimate the length of the matrix depending on the interesting positions that show a stronger bias or conservation in comparison with the context (Stormo, 2000a).

In the case of the promoter regulation, an additional serious inconvenient has been detected. Because of the high degree of ambiguity for a TF to select a binding site, the majority of the PWMs representing classes of TFBSs are very unspecific. Recently, Schones et al. (2005) measured the similarity between the matrices of several popular collections, reporting the existence of classes of equivalences between PWMs of different TFs. This unexpected result is probably produced by the small number of cases employed to construct such models (Rahmann et al., 2003).

PWMs and information content

The quality and quantity of information provided by the PWMs is different for each column in the motif and can be explained in terms of entropy or amount of uncertainty, expressed in bits per symbol for each position in a PWM (see Kim et al. (2003) for a review of the topic).

Given i , a position in a PWM, and (p_A, p_C, p_G, p_T) , the relative frequencies of the four possible nucleotides in that column, the information content of this position is defined as (Schneider and Stephens, 1990):

$$H(X) = - \sum_{x=A,C,G,T} p_x \log(p_x). \quad (4.1)$$

According to H , the maximum uncertainty is reached when $p_A = p_C = p_G = p_T = 0.25$. In this situation, no additional information can be assumed to guess what nucleotide will

be found over there. Obviously this is not the preferred situation because no particular trend or bias is observed. The opposite situation happens when one of the nucleotides dominates the rest of them: $p_A = 1, p_C = p_G = p_T = 0$. The absence of uncertainty in that position reflects a high degree of conservation that might be explained in biological terms. In general, some nucleotides tend to dominate the distribution in a subset of consecutive positions in the signal (the footprint or core). Instead, the context around usually shows a weaker conservation although discontinuities may happen along the matrix.

The amount of uncertainty of a PWM can be depicted in a sequence logo as in Figure 4.8 with the most conserved positions clearly highlighted (Schneider and Stephens, 1990). Motif positions are represented along the horizontal axis while the height of every column corresponds to the lack of uncertainty, that is, maximum entropy (2 bits in DNA) minus entropy computed for that position. The higher the column, the more conserved that position is.

The distribution of TRANSFAC matrices (Matys et al., 2003) according to their information content, calculated as shown in Equation 4.1, is presented in Figure 4.10.

4.4 Content recognition

The analysis of word counts has been very relevant in the detection of interesting regions in sequences of DNA. Historically, this analysis has been applied to locate functional sequences whose statistical content was significantly different from the values expected in non-functional regions.

Once a method to count oligo-nucleotides has been implemented, two approaches are possible. On the one hand, the search can be devoted to detect those regions richer in words that are statistically similar to the type of words observed in functional regions. On the other hand, the search can be directed to locate over-representations that are a priori unknown, reporting then such words in a set of related sequences.

Protein-coding regions

The distribution of amino acids in the known families of proteins is not uniform: for each species some amino acids are more common than others. Additionally, not all the synonymous codons of the genetic code that represent the same amino acid are used in the same proportion. Both facts produce a bias in the codon usage that can be statistically measured in the known genes of each species. Obviously, such a biased distribution is not observed in intronic and intergenic regions, improving the discrimination power. At the core of most gene-finding methods are one or more coding measures that evaluate the codingness of a sequence based on the codon bias (see Fickett and Tung (1992) for a review).

A coding statistic is a function that given a DNA sequence computes a real number measuring the likelihood that the sequence is coding for a protein (see Figure 4.11). The most popular coding statistic is the count of the frequency of each hexamer (two codons) in a sequence, to compare it afterwards to the frequencies observed in real protein-coding regions and non-coding regions (introns or intergenic sequences). If the content of such a

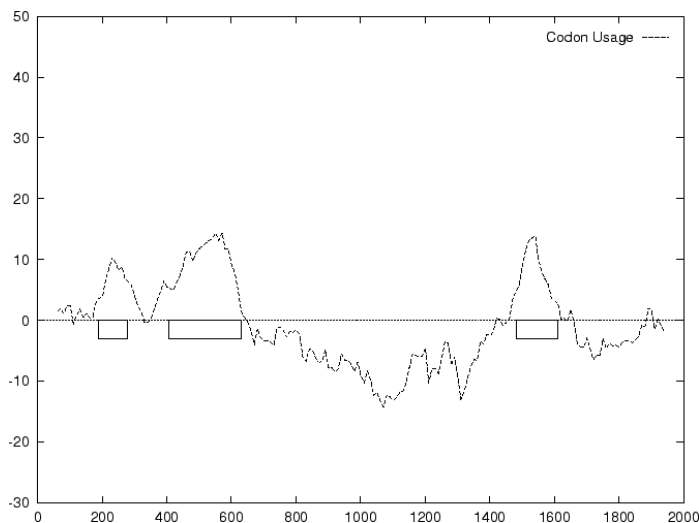


Figure 4.11 An example of coding statistic. The coding Vs non-coding model based on the codon usage along 2,000 bp of the human β -globin gene sequence (3 exons), computed on a sliding window of length 120 with step 10. Adapted from Guigó (1999).

region is similar to the oligomers that are more present in exons than in introns then it is reported as a predicted coding exon (Stormo, 2000b). Markov models are a natural form of counting these oligonucleotides to detect the dependencies between a group of consecutive nucleotides and the current one (Haussler, 1998).

Other type of statistical regularities are independent of a coding model. These statistics only capture the universal features of coding DNA, not requiring a sample of real protein-coding regions. For instance, periodicities or asymmetries are typical deviations from randomness (see Guigó (1999) for a review on DNA composition and codon usage).

Promoter regions

Gene promoter regions consist of clusters of binding sites, with some TFBSs oftenly occurring more than once to favour a higher rate of success in the transcription. Promoters can be therefore detected by taking advantage of this biased composition. However, there is not a general composition present in the majority of promoters, and the bias is not as strong as in the case of the coding regions.

The exact location annotation of the beginning of a transcript (the TSS) is usually very difficult. Basically, oligonucleotide counts are used in combination with other techniques to locate the TSS, as well as the upstream promoter region and the first exon (Davuluri et al., 2001). Such a region is supposed to contain a significant concentration of words representing binding site motifs. The enumerative methods to characterize promoter regions count all possible DNA words of a certain length in promoter sequences, and then evaluate statistically the results to report a list of over-represented words that could reflect the regulatory content of the sequences (Marino-Ramirez et al., 2004).

Simulating the coding and non-coding models constructed for gene prediction, similar methods have been attempted in the case of the promoter prediction. For instance, a model for promoter sequences and a model for coding exons can be used to discriminate promoters from other genic regions (Ohler, 2000).

4.5 Sequence comparison

A region of DNA that is significantly similar to a known sequence is suspicious to possess a similar function. This information may be used to guide or validate the prediction process. When a genomic sequence encodes a protein with a known homolog, methods that are based on the comparison with annotated sequences are preferable (positive evidence). Conversely, a region that matches well to repetitive sequence is unlikely to contain coding regions (negative evidence). Obviously, the main drawback of such methods is the impossibility to find genes and regulatory elements that are completely different from the products in the databases.

Different sources of information can be used to establish the comparison:

- ➔ Comparison to databases of expressed sequence tags (ESTs) or complete transcripts (cDNAs), to identify regions of a contig that could correspond to a processed mRNA.
- ➔ Translation of the input genomic sequence in the six reading frames and alignment to protein databases.
- ➔ Comparison of the predicted peptide in a genomic sequence to protein databases.
- ➔ Comparative analysis with homologous genomic sequences from other organisms to identify conservations of functional elements (binding sites, exons, ...).

Comparative genomics

The complete genomic sequence of a number of eukaryotes is already available. Therefore, it is natural to expect to extract practical results from this data. The rationale behind comparative genomic methods is that functional sequences (e.g. protein-coding regions, regulatory elements) tend to be more conserved than non-functional sequences in other species.

There is a lot of controversy in the scientific community about the use of the terms synteny, orthology/paralogy, homology or similarity. A syntenic region is defined to be a set of gene loci that stay together on the same chromosomal location in two or more species (Passarge et al., 1999). As explained in Chapter 3, two sequences are homologous if both share a common ancestor (Jensen, 2001). In addition, two sequences are similar when an alignment procedure reports a high degree of identity/similarity, not necessarily reflecting an evolutionary relationship (Pertsemlidis and Fondon, 2001).



Figure 4.12 Comparative analysis of the mouse, chicken and fugu orthologs for the human *FOS* gene. The boxes in red are the coding exons in both species. The diagonal lines are conserved segments in the pairwise alignment of the genomic sequences. Notice the better discrimination of the exons in more distant species.

Comparative gene prediction

When two genomes have only recently diverged, the order of many genes, gene numbers, gene positions and even gene structures (exon-intron organization, splice site usage) remain highly conserved (see Figure 4.12). Thus, gene prediction accuracy can be improved by using comparisons between two closely related genomes (Zhang, 2002).

Typically, comparative gene-finding combines sequence alignment and gene prediction. In a first step, the syntenic sequences of both genomes are located by the alignment of both genomes. Due to the importance of a good detection of such sequences, the choice of the genomes to align, the programs, and their parameters is crucial (Korf, 2003; Pertsemlidis and Fondon, 2001; Ureta-Vidal et al., 2003).

In a second step, the gene-finding engines predict genes on these hypothetically homologous regions, enhancing the score of the predicted exons overlapping the conserved parts of both genomes (Batzoglu et al., 2000; Parra et al., 2003).

Phylogenetic footprinting

Transcription regulation and animal diversity are intimately associated. For example, despite the number of genes in common between two different species as human and mouse is extremely high, both animals present different organismal complexity. Emerging evidence suggests that a more sophisticated elaboration of the regulatory mechanisms can be the responsible of this great variability (Levine and Tijan, 2003).

Comparative promoter prediction is based on the hypothesis that patterns of gene regulation are often conserved across species. Interspecies comparisons would help to identify common regulatory sequences (see Figure 4.13).

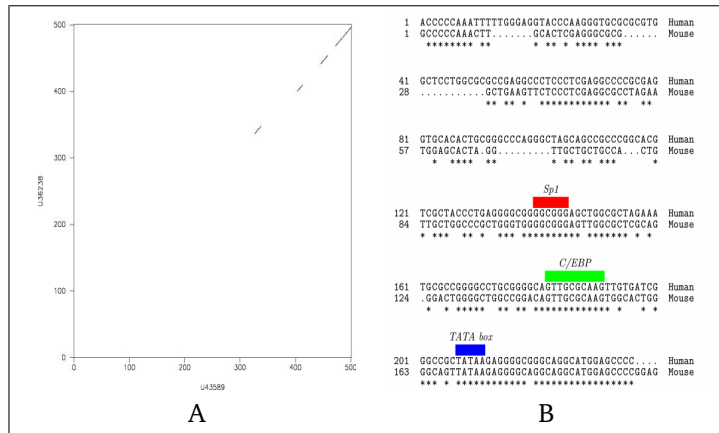


Figure 4.13 Phylogenetic footprinting (A) Dotplot of the promoter regions of the human and mouse *Leptin* gene. (B) Comparative analysis of both promoters.

Tagle et al. (1988) proposed the term 'phylogenetic footprinting' to describe the phylogenetic comparisons that reveal evolutionary conserved functional elements in homologous genes. However, this promising technique also presents some caveats, such as the difficulty to select the proper pair of species to perform the comparisons as every region of the genome evolves at a different speed (Duret and Bucher, 1997), the detection of specific elements of a given genome that are not present in the other one (Dermitzakis and Clark, 2002) or the existence of ultraconserved elements in the genomes of several species whose function must be determined (Bejerano et al., 2004).

Despite their limitations, phylogenetic footprinting has become very popular, being widely extended as an interesting method to locate regulatory elements (see Zhang and Gerstein (2003); Wasserman and Sandelin (2004) for a review).

Microarray data

The advent of the genome projects have favored the development of revolutionary techniques to process such a huge volume of information. High-throughput transcriptional profiling is definitely among these substantial improvements. DNA microarrays are the best representative of this new class of data-driven research paradigm. Microarray data measure the expression of a set of genes in two different cellular samples (knock-out vs. wild type) or after inoculation of some substance during a period of time divided into several stages.

The main principle of the method is the hybridization between unique oligonucleotides that represent a gene: one of which is immobilized on a matrix and the other is the actual RNA that is being transcribed in the sample. By fluorescently tagging each sample with different colours, the amount of transcript present in each sample can be quantified with a posterior image scanning of the hybridized microarray (see an example in Figure 4.14).

Many different implementations of the general microarray concept have been developed. Despite the ambiguity inherent to the high volume of output information, the procedure to

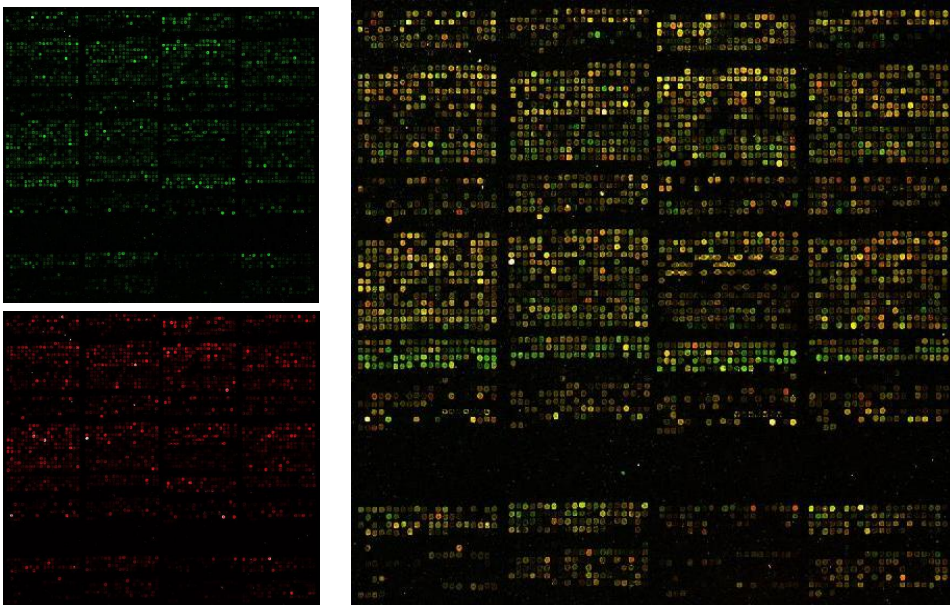


Figure 4.14 A microarray experiment. (Left) Expressed genes in a cell after a specific treatment in green and expressed genes in a normal cell in red. (Right) The ratio between both sets to detect the coexpressed genes.

elaborate and perform a microarray experiment usually consists of these steps (for further details see [Quackenbush \(2005\)](#)):

- ① Selection of the platform to construct the array
- ② Experiment design: choose a set of genes adequate to answer a biological question
- ③ Perform the experiment in the microarray (replications)
- ④ Image processing and estimation of the expression
- ⑤ Data collection and management of the gene expression data
- ⑥ Normalization of the expression data
- ⑦ Data analysis to find significant genes
- ⑧ Clustering the genes according to the pattern of expression
- ⑨ Analysis of the interesting groups (function, promoter elements, ...)

The final result of a microarray experiment is usually a list of genes that are over-expressed or under-expressed according to the state of the cells or the tissues from which the samples were extracted. Each group of genes presenting a similar temporal pattern of expression is said to be co-regulated or co-expressed.

The guilty by association strategy states that genes exhibiting a similar pattern of expression probably possess in common a similar transcriptional regulatory mechanism or play a similar function in such a cell. Thus, co-expressed genes are mainly the target of promoter detection analysis, being also functionally characterized using some catalogue of known biological functions such as the Gene Ontology ([The Gene Ontology Consortium, 2000](#)).

Since their creation, microarray technology has shown to be extremely useful to produce an enormous amount of large scale expression information. Microarrays have been applied at a genome-wide scale to build a regulatory map of *Saccharomyces cerevisiae* ([Harbison et al., 2004](#)), to classify and discover different types of acute leukemia ([Golub et al., 1999](#)), to annotate the human genome ([Shoemaker et al., 2001](#)), to reconstruct the transcriptional network controlled by a TF in *Drosophila melanogaster* ([Beltran et al., 2003](#)), to study alternative splicing ([Religio et al., 2005](#)) or to experimentally annotate the genes controlled by a family of TFs in human ([Odom et al., 2004](#)). Several outstanding reviews on the topic of microarrays have been published ([Various, a,b](#)).

Pattern discovery

Opposite to pattern matching or pattern-driven methods reviewed in Section 4.3, a new family of algorithms called sequence-driven methods appeared for searching novel motifs in a set of sequences that are hypothetically regulated in a similar manner ([Brazma et al., 1998](#)).

Sequence-driven methods, also called pattern discovery, do not rely on the use of any external dictionary or catalogue of elements that must be searched in the sequences. Instead, this approach attempts to detect novel patterns that are conserved in the input sequences. These motifs are not expected to be exact matches so that some mismatches are allowed and positional conservation is somehow neglected during the process.

The procedure described in Figure 4.15 is based on the definition of a fitness function and the implementation of an iterative procedure to distinguish the occurrences of the novel motifs that stops when no improvement is observed. Sequence-driven algorithms have been mainly used to analyze the promoters of co-regulated genes according to microarray expression experiments. Examples are the programs AlignAce ([Roth et al., 1998](#)), MEME ([Bailey and Elkan, 1994](#)) and Gibbs sampling ([Lawrence et al., 1993](#)).



4.6 The state of the art in gene identification

In the early nineties, the first computational gene-finding programs were designed to integrate both signal and content sensors, modeled during the eighties using either linguistic methods, machine learning procedures or purely statistical approaches. These programs used to be applied on single sequences. The seminal works in this field were presented by [Gelfand \(1990\)](#) and [Fields and Soderlund \(1990\)](#). Other members of this first generation of gene finders were: *fgenesh* ([Solovyev and Salamov, 1994](#)), *geneid* ([Guigó et al., 1992](#)), *genelang* ([Dong and Searls, 1994](#)), *genemark* ([Borodovsky and McIninch, 1993](#)) and *grail* ([Uberbacher and Mural, 1991](#)).

```

Pre  $\equiv S_1, S_2, \dots, S_n$ : sequence; M: motif model; F: scoring function;

(* Select a random pool of motifs in the sequences to create M *)
M  $\leftarrow$  CreateInitialModel();
(* Evaluate the fitness of the current model M *)
5: score0  $\leftarrow$  EvaluateModel(M, F);
   score  $\leftarrow$  score0;
   (* Repeat until convergence in the model M *)
   while score  $\geq$  score0 do
       score0  $\leftarrow$  score;
10:  (* Alter the model, trying to locate the motifs in each sequence *)
      UpdateModel(M);
      score  $\leftarrow$  EvaluateModel(M, F);
      (* Use the new model M to search the best motifs on each sequence *)
      for i  $\leftarrow$  1 to n do
15:   PatternDriven(Si, M);

```

Figure 4.15 Sequence-driven algorithms.

The first exhaustive evaluation of the accuracy of those methods on a large set of vertebrate sequences with simple gene structure was published by [Burset and Guigó \(1996\)](#). The results indicated that the predictive accuracy of the programs analyzed was lower than originally expected (the average percentage of exons exactly identified was less than 50%). This low accuracy level was in part explained because of the limited number of sequences used in the training process. Some of the basic accuracy measures used in the field are described in Table 4.1.

At the end of the last decade, a second generation of programs appeared simultaneously with the completion of the first genome sequencing projects. Some of them were even used in the earlier stages of the annotation pipelines. As new data and more powerful computers became accessible, the gene finders were able to deal with sequences containing more than one gene. Examples of programs in this second generation of gene prediction tools include: *geneid* ([Parra et al., 2000](#)), *genie* ([Kulp et al., 1996](#)), *genscan* ([Burge and Karlin, 1997](#)), *hmmgene* ([Krogh, 1997](#)) and *mzef* ([Zhang, 1997](#)).

Moreover, it was evident that sequence similarity to external databases containing known examples (search by homology) should be incorporated into the scoring schema of the programs in order to reinforce the predictions. This paradigm was developed in programs such as *genewise* ([Birney and Durbin, 1997](#)), *grail-exp* ([Xu and Uberbacher, 1997](#)) or *procrustes* ([Gelfand et al., 1996](#)). Some of these approaches were evaluated by [Guigó et al. \(2000\)](#) and [Rogic et al. \(2001\)](#). Although the gain in accuracy was significant in short sequences containing one gene, the performance was still insufficient in long semi-artificial sequences constructed from annotated examples.

Nowadays, after the completion of the first draft of the human genome we are completely immersed in a context of genomic research. The current generation of gene finders is devoted to the automatic reannotation of genomes by using the increasing amount of new information. Comparisons between genomes have proven to be very helpful in the discov-

SHORT	NAME	DESCRIPTION
TP	True positives	Number of real positive examples correctly predicted
TN	True negatives	Number of real negative examples correctly predicted
FN	False negatives	Number of real positive examples not correctly predicted
FP	False positives	Number of real negative examples not correctly predicted
SN	Sensitivity	Proportion of real examples corresponding to any prediction: $\frac{TP}{TP+FN}$
SP	Specificity	Proportion of predictions supported by any real example: $\frac{TN}{TN+FP}$
CC	Correlation coefficient	Correlation between SN and SP: $\frac{(TP \times TN) - (FN \times FP)}{\sqrt{(TP+FN) \times (TN+FP) \times (TP+FP) \times (TN+FN)}}$

Table 4.1 The common accuracy measures in sequence analysis.

ery of novel genes (Guigó et al., 2003). Some representatives of the current generation of gene prediction programs are fgenesh+ (Salamov and Solovyev, 2000), geneid (Blanco et al., 2003) and genomescan (R. Yeh and Burge, 2001), or the comparative analysis systems doublescan (Meyer and Durbin, 2002), rosetta (Batzoglou et al., 2000), slam (Alexandersson et al., 2003), sgp1 (Wiehe et al., 2001), sgp-2 (Parra et al., 2003) and twinscan (Korf and Flicek, 2001).

The latest achievements in the sequencing of other higher eukaryotes have allowed the advent of comparative predictors that consider the alignment of multiple genomes in the prediction model, such as N-scan that simultaneously combines the genomes of human, mouse, rat and chicken (Gross and Brent, 2005). Moreover, new tools such as jigsaw (Allen and Salzberg, 2005) and gaze (Howe et al., 2002) for the assembly of data obtained from external sources of prediction and experimental evidence have been recently developed.

geneid

The current version of geneid (Blanco et al., 2003) is a program that predicts genes in anonymous genomic sequences designed following a simple hierarchical structure (see Figure 4.16 (A)). First, splice sites and start and stop codons are predicted and scored along the sequence. Next, potential exons are constructed from these sites and scored as the sum of the defining sites plus the score of a Markov model for coding DNA. Finally, from the set of predicted exons, the gene structure maximizing the sum of the score of its exons is assembled using a dynamic programming algorithm (Guigó, 1998).

geneid offers two features to integrate external information into the ab initio predictions: (1) sequence homology information can be used to reinforce the predictions that are supported by the alignment and (2) partial or complete genes obtained from other sources can be incorporated before the exon assembly.

As a consequence of its simple design, geneid has been also parallelized. Parallelism of data (distribution of data among processors with shared memory) was finally implemented

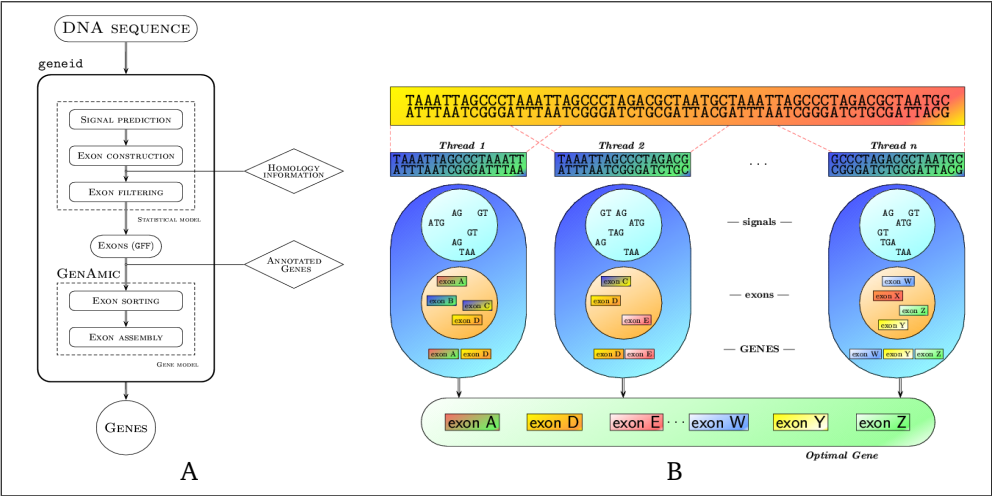


Figure 4.16 *geneid* dataflow. (A) The serial dataflow. (B) The parallel dataflow.

because it was the best solution for distributing the overload in the system. Following the divide and conquer strategy, the best gene structures computed in different processors are assembled introducing some overlap between sequence fragments (see Figure 4.16 (B)).

The simplicity of the architecture of *geneid* is appropriate to deal with problems different from the canonical ones. Taking advantage of the implemented facilities to reannotate sequences, *geneid* has been the main component of two recent genome annotation pipelines:

- ① Identification of novel selenoproteins in eukaryotes. The presence of a secondary structure (SECIS element) in the 3' UTR of the mRNA induces the UGA codon, usually a termination signal, to be translated as Selenocysteine. *geneid* was modified to permit the dual meaning of the UGA triplet, being successfully applied to describe the *Drosophila melanogaster*, human and *Takifugu rubripes* selenoproteomes (Castellano et al., 2001; Kryukov et al., 2003; Castellano et al., 2004). In addition, *geneid* was used to reannotate selenoproteins in the *Tetraodon nigroviridis* genome (Jaillon et al., 2004), being the first eukaryotic genome project to integrate the identification of this particular family into the gene annotation pipeline.
- ② Comparative gene prediction. *snp2* is a method to predict genes in a target genome sequence using the sequence of a second informant or reference genome (Parra et al., 2003). Essentially, *snp2* is a framework to integrate the search program *tblastx* results with *geneid* predictions. The result of the *tblastx* alignment of two sequences is used by *geneid* to rescore the exons supported by the alignment, penalizing the score of the others. *snp2* was successfully used in cooperation with another similar program called TWINSKAN (Korf and Flicek, 2001) to discover a set of novel human and mouse genes. A subset of them was then experimentally validated in a subsequent stage of the genome comparison protocol (Guigó et al., 2003). The same protocol was used to annotate the genomes of human and chicken (Hillier et al., 2004).

4.7 The state of the art in promoter characterization

The first algorithms of sequence alignment were entirely written to analyze proteins (Needleman and Wunsch, 1970). However, it was soon noticed that the same procedures could be applied over any type of biological sequence, including transcription regulatory regions. For instance, Sadler et al. (1983) used consensus and similarity searches to locate some general promoter elements in a set of vertebrate sequences. In (Waterman et al., 1984), two algorithms to detect a common motif that can be known or unknown a priori in a set of sequences were presented. Later, these algorithms were used to characterize the core promoter of several *Escherichia coli* genes (Galas et al., 1985).

Consensus are a rudimentary form for representing regulatory sites so that new proposals to overcome their limitations were published. Staden (1984) suggested the use of weight matrices. These PWMs were constructed from previous alignments of different types of biological sites. Bucher (1990) systematically refined and tested the PWMs for detecting different regulatory signals such as the TATA box, the CAAT-box or the GC-box. At the same time, theoretical studies to relate the information content and the quality of anchored alignments were already published (Schneider and Stephens, 1990). Posterior studies have shown the low specificity of the PWMs when the set of initial examples is small (Schones et al., 2005).

Soon, several databases to store the experimental examples and the constructed matrices were published, such as TRANSFAC (Wingender, 1988). At the same time, efficient programs to scan promoter sequences based on the pattern matching technique (pattern-driven approaches) were designed to use these matrices, being MatInspector the most popular one (Frech et al., 1993; Quandt et al., 1995). However, methods to identify TFBSs in a single sequence demonstrated a very poor performance with an excess of false positives. Certain improvements were observed when using additional information. New heuristic methods to discover unknown patterns in a set of regulatory sequences appeared (sequence-driven approaches): the application of the Gibbs sampling (Lawrence et al., 1993) and the expectation-maximization method (Bailey and Elkan, 1994) are good examples.

In general, however, the experimental investigation of a single promoter in all cell types where it can be active, under all conceivable conditions, at all possible developmental and cell-cycle stages, is in practice impossible. With this limitation in mind, the predictions obtained by any method must be always very carefully evaluated to avoid the rejection of predicted functional sites that have not been experimentally annotated yet.

The identification of the core promoter regions and the annotation of the TSSs have also been two problems associated to the problem of the TFBSs prediction. The presence of significantly over-expressed words or an unusual high percentage of CpG dinucleotides have traditionally been two measures of promoterness. For instance, Davuluri et al. (2001) combined these two sensors with splicing detection to locate the first exon of a gene, predicting therefore the TSS position. Neural networks and genetic algorithms were used in (Knudsen, 1999) to discriminate between promoter and non-promoter sequences. Fickett and Hatzigeorgiou (1997) reviewed the topic, showing the poor accuracy of most methods in the detection of the TSS. Word over-representations have been also used to study the as-

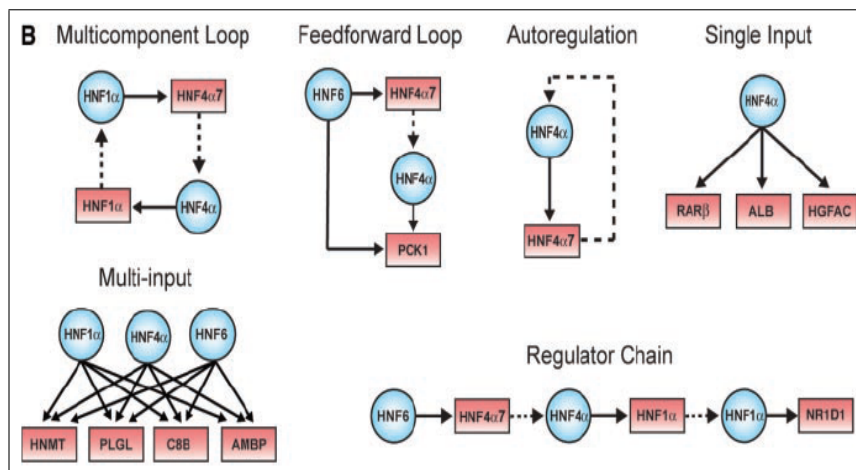


Figure 4.17 Transcriptional regulatory module architectures. Regulatory proteins and their gene targets are represented as blue circles and red boxes, respectively. Solid arrows indicate protein-DNA interactions, and genes encoding regulators are linked to their protein products by dashed lines. Adapted from (Harbison et al., 2004).

sociation of adjacent TFBSs to form regulational modules or clusters with interesting results although the deciphering of a regulatory code seems still too complex (Beer and Tavazoie, 2004; Sharan et al., 2003; Terai and Takagi, 2004; Thompson et al., 2004). An example of such architectures is shown in Figure 4.17.

A new revolution in the study of gene regulation began with the availability of genomic information and the possibility to work with abundant expression data. Phylogenetic footprinting, for instance, is a new form of leaving a great fraction of false positives out (Duret and Bucher, 1997; Fickett and Wasserman, 2000). Promising results have been obtained in several investigations (Blanchette and Tompa, 2002; Krivan and Wasserman, 2001; Lenhard et al., 2003). A review on phylogenetic footprinting can be found in (Wasserman and Sandelin, 2004). Gene expression data from microarrays is the other great hope in the field to elaborate a regulatory map of human. Despite at the beginning, there was a boom of analysis of such data in different biological problems (Beltran et al., 2003; Golub et al., 1999; Shoemaker et al., 2001), the difficulty to analyze and understand such an amount of data has been underscored in many occasions, though. The new generation of arrays based on chromatin immunoprecipitation promise to be an interesting method of prediction validation (Odom et al., 2004). The combination of comparative genomics and expression data will become in a few years the standard way to study a group of genes as in (Xie et al., 2005).

Due to the poor results obtained when analyzing sequences to find pure binding motifs, intensive research has been performed in other areas to understand better the gene regulation problem. For instance, the association between CpG islands and promoters (Cuadrado et al., 2001), DNA structure (Pedersen et al., 1998), nucleosome positioning (Ioshikhes et al., 1999) or protein-DNA physical interactions (Halford and Marko, 2004).

Similarly to the gene-finding accuracy tests, several assessments have been performed

about the quality of promoter characterization tools, always with discouraging results. The lack of stable data sets of regulation sites, and the surprising difficulty to deal sometimes with orthologous sequences are two causes that suggests the need for further improvement (Prakash and Tompa, 2005; Tompa et al., 2005).

4.8 Looking forward

Despite the numerous advances in the basic algorithms of gene and promoter prediction and the unceasing flow of new data, the way to determine the exact number of genes in the human genome remains unclear (Pennisi, 2003) and the elaboration of a regulatory map of the human genome seems today an objective too ambitious (Wasserman and Sandelin, 2004).

In the discipline of gene prediction, the same concepts have been applied since more than 20 years ago. While the basic gene models have been improved to support comparative research, the definition of a gene predicted by a gene-finder is still the same. It is true that some non-canonical gene structures are being slowly incorporated into the programs such as prediction of UTRs, alternative splicing forms or selenoproteins (Brent and Guigó, 2004). Right now, the gene identification problem is still open and many efforts are engaged in the creation of a solid catalogue of human genes (ENCODE Project Consortium, 2004), in which large-scale experimental methods of validation will be crucial (Brent, 2005).

Moreover, gene prediction and promoter recognition should be performed simultaneously. Unfortunately, we are far from reaching such an achievement due to the poor performance in the detection of regulatory elements despite the new and promising research that is currently being done in that direction (Pennacchio and Rubin, 2001). The enormous volume of high-throughput expression data has provided new opportunities in the investigation of the biology of the systems (Davidson et al., 2002). Phylogenetic footprinting is also demonstrating their capability to unveil regulatory blocks conserved in several species (Wasserman et al., 2000). In addition, more accurate catalogues of annotated regulatory elements are appearing, making the training of new pattern discovery methods easier. All together will be part of a future pipeline to automatically identify and annotate the eukaryotic promoter regions. However, much effort must be still invested in understanding better other aspects of the same biological problem such as chromatin effect, methylation, or nucleosome movement (Pedersen et al., 1999).

Perhaps a new line of thought should be established in both fields (Claverie, 2000). So far, we have been only focusing on the sequence and many successful advances have been possible following such an approach. However, it is assumed that the cell machinery works in many levels with uncountable number of interactions that we have not incorporated in our systems yet. Once we have reached the limit with the current methods, and that moment is not too far, it will be essential to move from the current analytical systems to more constructive and dynamic applications, emulating the mechanisms of the cell.

Bibliography

- B. Alberts, D. Bray, J. Lewis, M. Raff, K. Roberts, and J. D. Watson. *Molecular biology of the cell*. Garland publishing, third edition, 1994. ISBN 0-8153-1620-8.
- M. Alexandersson, S. Cawley, and L. Patcher. Slam: cross-species gene finding and alignment with a generalized pair hidden markov model. *Genome Research*, 13:496–502, 2003.
- J.E. Allen and S.L. Salzberg. Jigsaw: integration of multiple sources of evidence for gene prediction. *Bioinformatics*, 21:3596–3603, 2005.
- F. Antequera and Adrian Bird. Number of CpG islands and genes in human and mouse. *Proceedings of National Academy of Sciences*, 90:11995–11999, 1993.
- T. L. Bailey and C. Elkan. Fitting a mixture model by expectation maximization to discover motifs in biopolymers. *Proceedings of the 2nd International Conference on Intelligent Systems for Molecular Biology (ISMB)*, pages 28–36, 1994.
- S. Batzoglou, L. Pachter, J.P. Mesirov, B. Berger, and E.S. Lander. Human and mouse gene structure: comparative analysis and application to exon prediction. *Genome Research*, 10:950–958, 2000.
- M. A. Beer and S. Tavazoie. Predicting gene expression from sequence. *Cell*, 117:185–198, 2004.
- G. Bejerano, M. Pheasant, I. Makunin, S. Stephen, W.J. Kent, J.S. Mattick, and D. Haussler. Ultraconserved elements in the human genome. *Science*, 304:1321–1325, 2004.
- S. Beltran, E. Blanco, F. Serras, B. Perez-Villamil, R. Guigó, S. Artavanis-Tsakonas, and M. Corominas. Transcriptional network controlled by the trithorax-group gene *ash2* in *drosophila melanogaster*. *Proceedings of the National Academy of Sciences*, 100:3293–3298, 2003.
- E. Birney and R. Durbin. Dynamite: a flexible code generating language for dynamic programming methods used in sequence comparison. *Proceedings Intell. Syst. Mol. Bio.*, 5:56–64, 1997.
- M. Blanchette and M. Tompa. Discovery of regulatory elements by a computational method for phylogenetic footprinting. *Genome Research*, 12:739–748, 2002.
- E. Blanco, D. Farre, M. Alba, X. Messeguer, and R. Guigó. ABS: a database of annotated regulatory binding sites from orthologous promoters. *Nucleic Acids Research*, 34:D63–D67, 2006.
- E. Blanco and R. Guigó. *Bioinformatics: A Practical Guide to the Analysis of Genes and Proteins.*, chapter “Predictive Methods using DNA Sequences”, pages 115–142. John Wiley & Sons Inc., New York, USA, 2005. ISBN 0-471-47878-4.
- E. Blanco, G. Parra, and R. Guigó. *Current Protocols in Bioinformatics.*, volume 1, chapter “Using geneid to Identify Genes.”. John Wiley & Sons Inc., New York, USA, 2003. ISBN 0-471-25093-7.
- M. Borodovsky and J. McIninch. GenMark: Parallel gene recognition for both DNA strands. *Computer and Chemistry*, 17:123–134, 1993.
- A. Brazma, I. Jonassen, I. Eidhammer, and D. Gilbert. Approaches to the automatic discovery of patterns in biosequences. *Journal of Computational Biology*, 5:279–305, 1998.
- M.R. Brent. Genome annotation past, present, and future: how to define an orf at each locus. *Genome Research*, 15:1777–1786, 2005.
- M.R. Brent and R. Guigó. Recent advances in gene structure prediction. *Current Opinion in Structural Biology*, 14:264–272, 2004.

- T.A. Brown. *Genomes*. BIOS Scientific Publishers, Oxford, UK, second edition, 2002. ISBN 1-85996-029-4.
- P. Bucher. Weight matrix descriptions of four eukaryotic RNA polymerase II promoter elements derived from 502 unrelated promoter sequences. *Journal of Molecular Biology*, 212:563–578, 1990.
- C. B. Burge and S. Karlin. Prediction of complete gene structures in human genomic DNA. *Journal of Molecular Biology*, 268:78–94, 1997.
- M. Burset and R. Guigó. Evaluation of gene structure prediction programs. *Genomics*, 34:353–67, 1996.
- M. Burset, I.A. Seledtsov, and V.V. Solovyev. Analysis of canonical and non-canonical splice sites in mammalian genomes. *Nucleic Acids Research*, 28:4364–4375, 2000.
- S. Castellano, N. Morozova, M. Morey, M.J. Berry, F. Serras, M. Corominas, and R. Guigó. In silico identification of novel selenoproteins in the drosophila melanogaster genome. *EMBO Reports*, 2: 697–702, 2001.
- S. Castellano, S.V. Novoselov, G.V. Kryukov, A. Lescure, E. Blanco, A. Krol, V.N. Gladyshev, and R. Guigó. Reconsidering the evolution of eukaryotic selenoproteins: a novel nonmammalian family with scattered phylogenetic distribution. *EMBO Reports*, 5:71–77, 2004.
- J.M. Claverie. From bioinformatics to computational biology. *Genome Research*, 10:1277–1279, 2000.
- M. Cuadrado, M. Sacristan, and F. Antequera. Species-specific organization of cpg island promoters at mammalian homologous genes. *EMBO reports*, 21:586–592, 2001.
- E.H. Davidson, J.P. Rast, P. Oliveri, A. Ransick, C. Caletani, C. Yuh, T. Minokawa, G. Amore, V. Hinman, C. Arenas-Mena, O. Otim, C.T. Brown, C.B. Livi, P.Y. Lee, R. Revilla, A.G. Rust, Z. Pan, M.J. Schilstra, P.J.C. Clarke, M.I. Arnone, L. Rowen, R.A. Cameron, D.R. McClay, L. Hood, and H. Bolouri. A genomic regulatory network for development. *Science*, 295:1669–1678, 2002.
- R. Davuluri, I. Grosse, and M.Q. Zhang. Computational identification of promoters and first exons in the human genome. *Nature Genetics*, 29:412–417, 2001.
- E. T. Dermitzakis and A. G. Clark. Evolution of transcription factor binding sites in mammalian gene regulatory regions: conservation and turnover. *Molecular Biology and Evolution*, 7:1114–1121, 2002.
- S. Dong and D.B. Searls. Gene structure prediction by linguistic methods. *Genomics*, 23:540–551, 1994.
- L. Duret and P. Bucher. Searching for regulatory elements in human noncoding sequences. *Current Opinion in Structural Biology*, 7:399–406, 1997.
- ENCODE Project Consortium. The ENCODE (ENCyclopedia Of DNA Elements) Project. *Science*, 306: 636–40, 2004.
- D. Farre, R. Roset, M. Huerta, J. E. Adsuara, LL. Rosello, M. Alba, and X. Messeguer. Identification of patterns in biological sequences at the ALGGEN server: PROMO and MALGEN. *Nucleic Acids Research*, 31:3651–3653, 2003.
- J. W. Fickett and A. Hatzigeorgiou. Eukaryotic promoter recognition. *Genome Research*, 7:861–878, 1997.
- J. W. Fickett and C.S. Tung. Assessment of protein coding measures. *Nucleic Acids Research*, 20:6441–6450, 1992.

- J. W. Fickett and W.W. Wasserman. Discovery and modeling of transcriptional regulatory regions. *Current Opinion in Biotechnology*, 11:19–24, 2000.
- C.A. Fields and C.A. Soderlund. gm: a practical tool for automating dna sequence analysis. *CABIOS*, 6:263–272, 1990.
- K. Frech, G. Herrmann, and T. Werner. Computer-assisted prediction, classification, and delimitation of protein binding sites in nucleic acids. *Nucleic Acids Research*, 21:1655–1664, 1993.
- D.J. Galas, M. Eggert, and M.S. Waterman. Rigorous pattern-recognition methods for dna sequences. *Journal of Molecular Biology*, 186:117–128, 1985.
- M.S. Gelfand. Computer prediction of exon-intron structure of mammalian pre-mrnas. *Nucleic Acids Research*, 18:5865–5869, 1990.
- M.S. Gelfand, A.A. Mironov, and P.A. Pevner. Gene recognition via spliced sequence alignment. *Proceedings of the National Academy of Sciences*, 93:9061–9066, 1996.
- T.R. Golub et al. Molecular classification of cancer: class discovery and class prediction by gene expression monitoring. *Science*, 286:531–536, 1999.
- S.S. Gross and M.R. Brent. Using multiple alignments to improve gene prediction. *Proceedings of the 9th Annual International Conference, RECOMB 2005*, pages 374–388, 2005.
- R. Guigó. Assembling genes from predicted exons in linear time with dynamic programming. *Journal of Computational Biology*, 5:681–702, 1998.
- R. Guigó. *Genetic Databases.*, chapter DNA Composition, Codon Usage and Exon Prediction., pages 53–80. Academic Press, San Diego, California, USA, 1999. ISBN 0-12-101625-0.
- R. Guigó, P. Agarwal, J.F. Abril, M. Burset, and J.W. Fickett. An assessment of gene prediction accuracy in large dna sequences. *Genome Research*, 10:1631–1642, 2000.
- R. Guigó, E.T. Dermitzakis, P. Agarwal, C.P. Ponting, G. Parra, A. Raymond, J.F. Abril, E. Keibler, R. Lyle, C. Ucla, S.E. Antonarakis, and M.R. Brent. Comparison of mouse and human genomes followed by experimental verification yields an estimated 1,019 additional genes. *Proceedings of the National Academy of Sciences*, 100:1140–1145, 2003.
- R. Guigó, S. Knudsen, N. Drake, and T. Smith. Prediction of gene structure. *Journal of Molecular Biology*, 226:141–157, 1992.
- S.E. Halford and J.F. Marko. How do site-specific dna-binding proteins find their targets? *Nucleic Acids Research*, 32:3040–3052, 2004.
- C.T. Harbison, D.B. Gordon, T.I. Lee, N.J. Rinaldi, K.D. MacIsaac, T.W. Danford, N.M. Hannet, J. Tagne, D.B. Reynolds, J. YOO, E.G. Jennings, J. Zeitlinger, D.K. Pokholok, M. Kellis, P.A. Rolfe, K.T. Takusagawa, E.S. Lander, D.K. Gifford, E. Fraenkel, and R. A. Young. Transcriptional regulatory code of a eukaryotic genome. *Nature*, 431:99–104, 2004.
- D. Haussler. Computational genefinding. *Trends in Genetics (Trends guide to bioinformatics)*, pages 12–15, 1998.
- L.W. Hillier, W. Miller, E. Birney, W. Warren, R.C. Hardison, C.P. Ponting, P. Bork, D.W. Burt, M.A. Groenen, M.E. Delany, J.B. Dodgson, G. Fingerprint Map Sequence, Assembly, A.T. Chinwalla, P.F. Clifton, S.W. Clifton, and others (International Chicken Genome Sequencing Consortium, ICGSC). Sequence and comparative analysis of the chicken genome provide unique perspectives on vertebrate evolution. *Nature*, 432:695–716, 2004.

- K.L. Howe, T. Chothia, and R. Durbin. Gaze: a generic framework for the integration of gene-prediction data by dynamic programming. *Genome Research*, 12:1418–1427, 2002.
- I. Ioshikhes, E. Trifonov, and M.Q. Zhang. Periodical distribution of transcription factor sites in promoter regions and connection with chromatine structure. *Proceedings of National Academy of Sciences*, 96:2891–2895, 1999.
- O. Jaillon et al. Genome duplication in the teleost fish *Tetraodon nigroviridis* reveals the early vertebrate proto-karyotype analysis of the draft sequence of the compact. *Nature*, 431:946–957, 2004.
- R.A. Jensen. Orthologs and paralogs - we need to get it right. *Genome Biology*, 2:1002, 2001.
- J.T. Kim, T. Martinetz, and D. Polanti. Bioinformatic principles underlying the information content of transcription factor binding sites. *Journal of Theoretical Biology*, 220:529–544, 2003.
- S. Knudsen. Promoter 2.0: for the recognition of pol ii promoter sequences. *Bioinformatics*, 15: 356–361, 1999.
- I. Korf. Serial blast searching. *Bioinformatics*, 19:1492–1496, 2003.
- I. Korf and P. Flicek. Integrating genomic homology into gene structure prediction. *Bioinformatics*, 17: S140–S148, 2001.
- A.R. Kornbliht. Promoter usage and alternative splicing. *Current Opinion in Cell Biology*, 17:262–268, 2005.
- W. Krivan and W. W. Wasserman. A predictive model for regulatory sequences detecting liver-specific transcription. *Genome Research*, 11:1559–1566, 2001.
- A. Krogh. Two methods for improving performance of an hmm and their application for gene-finding. *Proceedings Intell. Syst. Mol. Bio.*, pages 179–186, 1997.
- G.V. Kryukov, S. Castellano, S.V. Novoselov, A.V. Lobanov, O. Zehtab, R. Guigó, and V.N. Gladyshev. Characterization of mammalian selenoproteomes. *Science*, 300:1439–1443, 2003.
- D. Kulp, D. Haussler, M.G. Reese, and F.H. Eeckman. A generalized hidden markov model for the recognition of human genes in dna. *Proceedings Intell. Syst. Mol. Bio.*, 4:134–142, 1996.
- A.N. Ladd and T.A. Cooper. Finding signals that regulate alternative splicing in the post-genomic era. *Genome Biology*, 3:reviews0008, 2002.
- C. E. Lawrence, S. F. Altschul, M. S. Boguski, J. S. Liu, A. F. Neuwald, and J. C. Wootton. Detecting subtle sequence signals: a gibbs sampling strategy for multiple alignment. *Science*, 262:208–214, 1993.
- B. Lenhard, A. Sandelin, L. Mendoza, P. Engstrom, N. Jareborg, and W. W. Wasserman. Identification of conserved regulatory elements by comparative genome analysis. *Journal of Biology*, 2:13, 2003.
- M. Levine and R. Tijan. Transcriptional regulation and animal diversity. *Nature*, 424:147–151, 2003.
- S.C. Low and M.J. Berry. Knowing when not to stop: selenocysteine incorporation in eukaryotes. *Trends in Biochemical Sciences*, 21:203–208, 1996.
- L. Marino-Ramirez, J.L. Spouge, G.C. Kanga, and D. Landsman. Statistical analysis of over-represented words in human promoter sequences. *Nucleic Acids Research*, 32:949–958, 2004.
- V. Matys et al. TRANSFAC: transcriptional regulation, from patterns to profiles. *Nucleic Acids Research*, 31:374–378, 2003.

- I.M. Meyer and R. Durbin. Comparative ab initio prediction of gene structures using pair hmms. *Bioinformatics*, 18:1309–1318, 2002.
- S. B. Needleman and C. D. Wunsch. A general method to search for similarities in the amino acid sequence of two proteins. *Journal of molecular biology*, 48:443–453, 1970.
- D.T. Odom, N. Zizlsperger, D.B. Gordon, G.W. Bell, N.J. Rinaldi, H.L. Murray, T.L. Volkert, J. Schreiber, P.A. Rolfe and D.K. Gifford, E. Fraenkel, G.I. Bell, and R.A. Young. Control of pancreas and liver gene expression by hnf transcription factors. *Science*, 303:1378–1381, 2004.
- U. Ohler. Promoter prediction on a genomic scale - the Adh experience. *Genome research*, 10:539–542, 2000.
- R. Osada, E. Zaslavsky, and M. Singh. Comparative analysis of methods for representing and searching for transcription factor binding sites. *Bioinformatics*, 18:3516–3525, 2004.
- G. Parra, P. Agarwal, J.F. Abril, T. Wiehe, J.W. Fickett, and R. Guigó. Comparative gene prediction in human and mouse. *Genome Research*, 13:108–117, 2003.
- G. Parra, E. Blanco, and R. Guigó. Geneid in drosophila. *Genome Research*, 10:511–515, 2000.
- E. Passarge, B. Horsthemke, and R.A. Farber. Incorrect use of the term synteny. *Nature Genetics*, 23:387, 1999.
- A. G. Pedersen, P. Baldi, Y. Chauvin, and S. Brunak. Dna structure in human rna polymerase ii promoters. *Journal of Molecular Biology*, 281:663–673, 1998.
- A. G. Pedersen, P. Baldi, Y. Chauvin, and S. Brunak. The biology of eukaryotic promoter prediction - a review. *Computers and Chemistry*, 23:191–207, 1999.
- L.A. Pennacchio and E.M. Rubin. Genomic strategies to identify mammalian regulatory sequences. *Nature Reviews Genetics*, 2:100–109, 2001.
- E. Pennisi. Bioinformatics. Gene counters struggle to get the right answer. *Science*, 301:1040–1041, 2003.
- R. C. Perier et al. The eukaryotic promoter database (EPD). *Nucleic Acids Research*, 28:302–303, 2000.
- A. Pertselmidis and J.W. Fondon. Having a blast with bioinformatics (and avoiding blastphemy). *Genome Biology*, 2:2002, 2001.
- A. Prakash and M. Tompa. Discovery of regulatory elements in vertebrates through comparative genomics. *Nature Biotechnology*, 23:1249–1256, 2005.
- J. Quackenbush. *Bioinformatics: A Practical Guide to the Analysis of Genes and Proteins.*, chapter Using DNA microarrays to assay gene expression, pages 409–444. John Wiley & Sons Inc., New York, USA, 2005. ISBN 0-471-47878-4.
- K. Quandt, K. Frech, H. Karas, E. Wingender, and T. Werner. Matind and matinspector: new fast and versatile tools for the detection of consensus matches in nucleotide sequence data. *Nucleic Acids Research*, 23:4878–4884, 1995.
- L.P. Lim R. Yeh and C.B. Burge. Computational inference of homologous gene structures in the human genome. *Genome Research*, 11:803–816, 2001.
- S. Rahmann, T. Muller, and M. Vingron. On the power of profiles for transcription factor binding site detection. *Statistical Applications in Genetics and Molecular Biology*, 2:7, 2003.

- A. Religio, C. Ben-Dov, M. Baum, M. Ruggiu, C. Gemund, V. Benes, R.B. Darnell, and J. Valcarcel. Alternative splicing microarrays reveal functional expression of neuron-specific regulators in hodgkin lymphoma cells. *Journal of Biology and Chemistry*, 280:4779–4784, 2005.
- S. Rogic, A.K. Mackworth, and F.B. Ouellette. Evaluation of gene-finding programs on mammalian sequences. *Genome Research*, 11:817–832, 2001.
- F.R. Roth, J.D. Hughes, P.E. Estep, and G.M. Church. Finding dna regulatory motifs within unaligned non-coding sequences clustered by whole-genome mrna quantitation. *Nature Biotechnology*, 16: 939–945, 1998.
- J.R. Sadler, M.S. Waterman, and T.F. Smith. Regulatory pattern identification in nucleic acid sequences. *Nucleic Acids Research*, 11:2221–2231, 1983.
- A.A. Salamov and V.V. Solovyev. Ab initio gene finding in *Drosophila melanogaster*. *Genome Research*, 10:516–522, 2000.
- A. Sandelin, W. Alkema, P. Engstrom, W. W. Wasserman, and B. Lenhard. JASPAR: an open-access database for eukaryotic transcription factor binding profiles. *Nucleic Acids Research*, 32:D91–D94, 2004.
- T.D. Schneider and R.M. Stephens. Sequence logos: a new way to display consensus sequences. *Nucleic Acids Research*, 18:6097–6100, 1990.
- D. E. Schones, P. Sumazin, and M. Q. Zhang. Similarity of position frequency matrices for transcription factor binding sites. *Bioinformatics*, 21:307–313, 2005.
- R. Sharan, I. Ovcharenko, A. Ben-Hur, and R. M. Karp. Creme: a framework for identifying cis-regulatory modules in human-mouse conserved segments. *Bioinformatics*, 19:(Suppl. 1) i283–i291, 2003.
- D.D. Shoemaker et al. Experimental annotation of the human genome using microarray technology. *Nature*, 409:922–927, 2001.
- V.V. Solovyev and A.A. Salamov. Predicting internal exons by oligonucleotide composition and discriminant analysis of spliceable open reading frames. *Nucleic Acids Research*, 22:5156–5163, 1994.
- R. Staden. Computer methods to locate signals in nucleic acid sequences. *Nucleic Acids Research*, 12: 505–519, 1984.
- G.D. Stormo. Dna binding sites: representation and discovery. *Bioinformatics*, 16:16–23, 2000a.
- G.D. Stormo. Gene-finding approaches for eukaryotes. *Genome Research*, 10:394–397, 2000b.
- Y. Suzuki, R. Yamashita, S. Sugano, and K. Nakai. Dbtss: Database of transcriptional start sites: progress report 2004. *Nucleic Acids Research*, 32:D78 – D81, 2004.
- D.A. Tagle, B.F. Koop, M. Goodman, J.L. Slightom, and D.L. Hess. Embryonic ϵ and γ globin genes of a prosimian primate, nucleotide and amino acid sequences, developmental regulation and phylogenetic footprints. *Journal of Molecular Biology*, 203:439–455, 1988.
- G. Terai and T. Takagi. Predicting rules on organization of cis-regulatory elements, taking the order of elements into account. *Bioinformatics*, 20:1119–1128, 2004.
- The Gene Ontology Consortium. Gene ontology: tool for the unification of biology. *Nature Genetics*, 25:25–29, 2000.

- W. Thompson, M.J. Palumbo and W.W. Wasserman, J.S. Liu, and C.E. Lawrence. Decoding human regulatory circuits. *Genome Research*, 14:1967–1974, 2004.
- M. Tompa et al. Assessing computational tools for the discovery of transcription factor binding sites. *Nature Biotechnology*, 23:137–144, 2005.
- N.D. Trinklein, S.F. Aldred, S.J. Hartman, D.I. Schroeder, R.P. Otilar, and R.M. Myers. An abundance of bidirectional promoters in the human genome. *Genome Research*, 14:62–66, 2004.
- E.C. Uberbacher and R.J. Mural. Locating protein-coding regions in human dna sequences by a multiple sensor-neural network approach. *Proceedings of the National Academy of Sciences*, 88:11261–11265, 1991.
- A. Ureta-Vidal, L. Ettwiller, and E. Birney. Comparative genomics: genome-wide analysis in metazoan eukaryotes. *Nature Reviews Genetics*, 4:251–262, 2003.
- Various. The chipping forecast (supplement). a.
- Various. Functional genomics (supplement). b.
- J.C. Venter, M.D. Adams, E.W. Myers, P.W. Li, R.J. Mural, G.G. Sutton, H.O. Smith, M. Yandell, C.A. Evans, R.A. Holt, J.D. Gocayne, P. Amanatides, R.M. Ballew, D.H. Huson, J.R. Wortman, et al. The sequence of the human genome. *Science*, 291:1304–1351, 2001.
- W.W. Wasserman, M. Palumbo, W. Thompson, J.W. Fickett, and C.E. Lawrence. Human-mouse genome comparisons to locate regulatory sites. *Nature Genetics*, 26:225–228, 2000.
- W.W. Wasserman and A. Sandelin. Applied bioinformatics for the identification of regulatory elements. *Nature Reviews Genetics*, 5:276–287, 2004.
- M.S. Waterman, R. Arratia, and D.J. Galas. Pattern recognition in several sequences: consensus and alignment. *Bulletin of Mathematical Biology*, 46:515–527, 1984.
- T. Werner. Identification and functional modelling of DNA sequence elements of transcription. *Briefings in bioinformatics*, 1:372–380, 2000.
- T. Wiehe, S. Gebauer-Jung, T. Mitchell-Olds, and R. Guigó. Sgp-1: Prediction and validation of homologous genes based on sequence alignments. *Genome Research*, 11:1574–1583, 2001.
- E. Wingender. Compilation of transcription regulating proteins. *Nucleic Acids Research*, 16:1879–1902, 1988.
- G.A. Wray, M.W. Hahn, E. Abouheif, J.P. Balhoff, M. Pizer, M.V. Rockman, and L.A. Romano. The evolution of transcriptional regulation in eukaryotes. *Molecular Biology and Evolution*, 20:1377–1419, 2003.
- X. Xie, J. Lu, E.J. Kulbokas, T.R. Golub, V. Mootha, K. Lindblad-Toh, E. Lander, and M. Kellis. Systematic discovery of regulatory motifs in human promoters and 3' utrs by comparison of several mammals. *Nature*, 434:338–345, 2005.
- Y. Xu and E.C. Uberbacher. Automated gene identification in large-scale genomic sequences. *Journal of Computational Biology*, 4:325–338, 1997.
- Z. Xuan, F. Zhao, J. Wang, G. Chen, and M.Q. Zhang. Genome-wide promoter extraction and analysis in human, mouse, and rat. *Genome Biology*, 6:R72, 2005.
- M.Q. Zhang. Identification of protein coding regions in the human genome by quadratic discriminant analysis. *Proceedings of the National Academy of Sciences*, 94:565–568, 1997.

- M.Q. Zhang. Computational prediction of eukaryotic protein-coding genes. *Nature Review Genetics*, 3: 698–709, 2002.
- Z. Zhang and M. Gerstein. Of mice and men: phylogenetic footprinting aids the discovery of regulatory elements. *Journal of Biology*, 2:11, 2003.

PART III

Meta-Alignment of Sequences

Chapter 5

Meta-alignment of Biological Sequences

Summary

This chapter contains the description of an efficient algorithm to align higher order elements mapped over biological sequences. The relationship between sequence alignments and meta-alignment is also reviewed. Such an approach is trained on a set of well annotated promoters. The ability of the meta-alignment to identify functional elements conserved at high level, such as regulatory elements in co-regulated genes, in absence of sequence conservation is shown in several situations. In addition, the meta-alignment is used to evaluate the specificity of the weight matrices in a genome wide approach.

5.1	Biological maps: promoters	126
5.2	Transcription Factor maps	128
5.3	TF-map pairwise alignment	128
5.4	TF-map alignment training	136
5.5	TF-map alignments in orthologous genes	144
5.6	TF-map alignments in co-regulated genes	148
5.7	TF-map alignments and matrix specificity	155
5.8	Local TF-map alignments	158
5.9	Discussion	162

5.1 Biological maps: promoters

SEQUENCE COMPARISONS ARE AMONG THE MOST USEFUL COMPUTATIONAL TECHNIQUES in molecular biology. Sequences of characters in the four-letter nucleotide alphabet and in the twenty-letter amino acid alphabet are extremely good symbolic representations of the underlying DNA and protein molecules, and encode substantial information on their structure, function and history.

Primary sequence comparisons, however, have limitations. Although similar sequences do tend to play similar functions, the opposite is not necessarily true. Often similar functions are encoded in higher order sequence elements –such, for instance, structural motifs in amino acid sequences– and the relation between these and the underlying primary sequence may not be univocal. As a result, similar functions are frequently encoded by diverse sequences.

As reviewed in Chapter 3, a biological map is a description of functional objects (e.g. genes or regulatory sites) that are identified in a sequence at a given position. The annotation of the human genome in Figure 5.1 is a clear example of genomic mapping (Venter et al., 2001). Comparison operations between maps are then necessary to elucidate functional relationships that are undetectable at the sequence level.

Promoter regions controlling eukaryotic gene expression are a case in point. As reviewed in Chapter 4, the information for the control of the initiation of the gene transcription is mostly contained in the gene promoter, a region upstream of the gene transcription start site (TSS). Transcription factors (TFs) interact in these regions with sequence specific elements or motifs (the TF binding sites, TFBSs). TFBSs are typically 5-15 nucleotides long and one promoter region usually contains many of them to harbor different TFs (Wray et al., 2003). The interplay between these factors is not well understood, but the motifs appear to be arranged in specific configurations that confer on each gene an individualized spatial and temporal transcription program (Wray et al., 2003). It is assumed, in consequence, that genes exhibiting similar expression patterns would also share similar configurations of TFs in their promoter.

However, TFBSs associated to the same TF are known to tolerate sequence substitutions without losing functionality, and are often not conserved. Consequently, promoter regions of genes with similar expression patterns may not show sequence similarity, even though they may be regulated by similar configurations of TFs. For instance, only about 30 to 40% of the promoter regions are conserved between human and chicken orthologous genes (Hillier et al., 2004), and the conservation of human-mouse orthologous promoter regions is only slightly higher than that observed in intergenic regions (Waterston et al., 2002). Indeed, despite the recent progress due to the development of techniques based in the so-called phylogenetic footprinting, lack of nucleotide sequence conservation between functionally related promoter regions may partially explain the still limited success of current available computational methods for promoter characterization (see Chapter 4 for a review of these methods).

In the approach described in this chapter (Blanco et al., 2006b), we attempt to overcome this limitation by abstracting the nucleotide sequence, and representing a promoter region by a sequence in a new alphabet in which the different symbols denote different TFs. Using an external mapping function (for instance, a look-up table or a collection of position weight



matrices, PWMs) that associates each TF to the nucleotide sequence motifs the factor is known to bind, we can translate the nucleotide sequence of the promoter into a sequence in this new alphabet. These sequences can be aligned. If the scoring of the alignment takes into account not only the presence/absence of a given symbol, but its relative position on the primary nucleotide sequence, the optimal alignment between the promoter regions of two genes with similar expression patterns may reflect the underlying common configuration of TFBSs. We refer to these alignments either as meta-alignments, as they are performed between sequences in a meta-alphabet, or map alignments, since they are obtained after mapping the nucleotide sequence in a higher order alphabet.

5.2 Transcription Factor maps

Analogously to the restriction enzyme maps initially formalized by [Waterman et al. \(1984\)](#) that are described in Chapter 3, we translate in our approach ([Blanco et al., 2006b](#)) the nucleotide sequence of a promoter region $S = s_1 s_2 \dots s_k$ into a sequence of 4-tuples $A = a_1 \dots a_n$ where each $a_i = \langle a_i^f, a_i^{p_1}, a_i^{p_2}, a_i^s \rangle$ denotes the match with score a_i^s of a binding site for the TF a_i^f occurring between the position $a_i^{p_1}$ and the position $a_i^{p_2}$ over the sequence S .

We obtain the translation from S to A by running on S a collection of PWMs representing binding motifs for TFs (such as, for instance, the collection in TRANSFAC ([Matys et al., 2003](#))). For each match over a given threshold, we register in A the positions $(a_i^{p_1}, a_i^{p_2})$, the score (a_i^s) , and the label (a_i^f) of the TF associated to the PWM. The translation preserves the order of S in A , that is if $i < j$ in A then $a_i^{p_1} \leq a_j^{p_1}$ (the \leq is because matches to different TFs may occur at the same position). We will refer to the resulting sequence A as a Transcription Factor Map (TF-map) or simply a map (see Figure 5.2). Note that other mapping functions, instead of collections of PWMs, can also be used to translate S into A .

In the implementation here, matches to PWMs are considered strandless, that is, they are annotated at a given location, irrespective of the orientation in which they occur. While biological evidence suggests that some TFBSs are functional only when present in a given strand, in other cases TF activity appears to be independent of the orientation of the binding site ([Strachan and Read, 1999](#)). Since in general, we do not have information of the strand in which a binding site may be functional, we have not considered strand in our analysis.

5.3 TF-map pairwise alignment

The same types of sequence alignments that were reviewed in Chapter 3 are also possible with maps: pairwise or multiple, global or local alignments. In this chapter, we described the algorithms of global and local pairwise TF-map alignment. The approach for multiple map alignment is detailed in the next chapter.

Formally, the pairwise alignment of the TF-maps $A = a_1 \dots a_m$ and $B = b_1 \dots b_n$ is a correspondence T , maybe empty, between A and B such that (Blanco et al., 2006b):

1. $(a_i, b_j) \in T$ if and only if $a_i^f = b_j^f$ (that is, two elements are aligned if and only if they correspond to the same TF).
2. if $(a_i, b_j) \in T$ then there are no other elements b_l ($l \neq j$) in B such that $(a_i, b_l) \in T$, nor elements a_k ($k \neq i$) in A such that $(a_k, b_j) \in T$ (that is, each element in A is aligned at most to one element in B , and vice versa).
3. if $(a_i, b_j) \in T$ and $(a_k, b_l) \in T$ and $i < k$ then $j < l$ (that is, the alignment maintains the colinearity between the sequences A and B).
4. if $(a_i, b_j) \in T$ and $(a_k, b_l) \in T$ with $i < k$ and $j < l$ then $a_i^{p_2} < a_k^{p_1}$ and $b_j^{p_2} < b_l^{p_1}$ (that is, no overlap in the primary sequences is permitted between the sites corresponding to the aligned elements).

Usually there are many possible alignments between two given A and B maps (see Figure 5.2 for an example). Given an alignment T

$$T = \{(a_{I_1}, b_{J_1}), (a_{I_2}, b_{J_2}), \dots, (a_{I_t}, b_{J_t})\} \quad (5.1)$$

where $T_k = (a_{I_k}, b_{J_k})$ is the match between the 4-tuple in position I_k from A and the 4-tuple in position J_k from B , we compute the score of the alignment $s(T)$ in the following way:

$$s(T) = \alpha \sum_{k=1}^t a_{I_k}^s + b_{J_k}^s - \lambda(m + n - 2t) - \mu \sum_{k=2}^t |(a_{I_k}^{p_1} - a_{I_{k-1}}^{p_1}) - (b_{J_k}^{p_1} - b_{J_{k-1}}^{p_1})| \quad (5.2)$$

where $\alpha, \lambda, \mu > 0$. That is, the score of the alignment increases with the score of the aligned elements (α), and decreases with the number of unaligned elements (λ), and with the difference in the distance between adjacent aligned elements (μ).

Finding the optimal alignment

The optimal alignment between two given maps A and B is the one scoring the maximum among all possible alignments. To obtain such an alignment efficiently, we have implemented an algorithm reminiscent of that proposed by Waterman et al. (1984) to align and compare restriction enzyme maps. This algorithm was developed to find the distance between two homologous restriction maps in terms of minimum weighted sum of genetic events necessary to convert one restriction map into another, where the genetic events are the appearance/disappearance of restriction sites and changes in the number of bases between restriction sites (see Chapter 3 for further details).

Here to align TF-maps A and B , we adapted the recursion in Waterman et al. (1984) to optimize similarity instead (Blanco et al., 2006b). In addition, we included a term (α) into

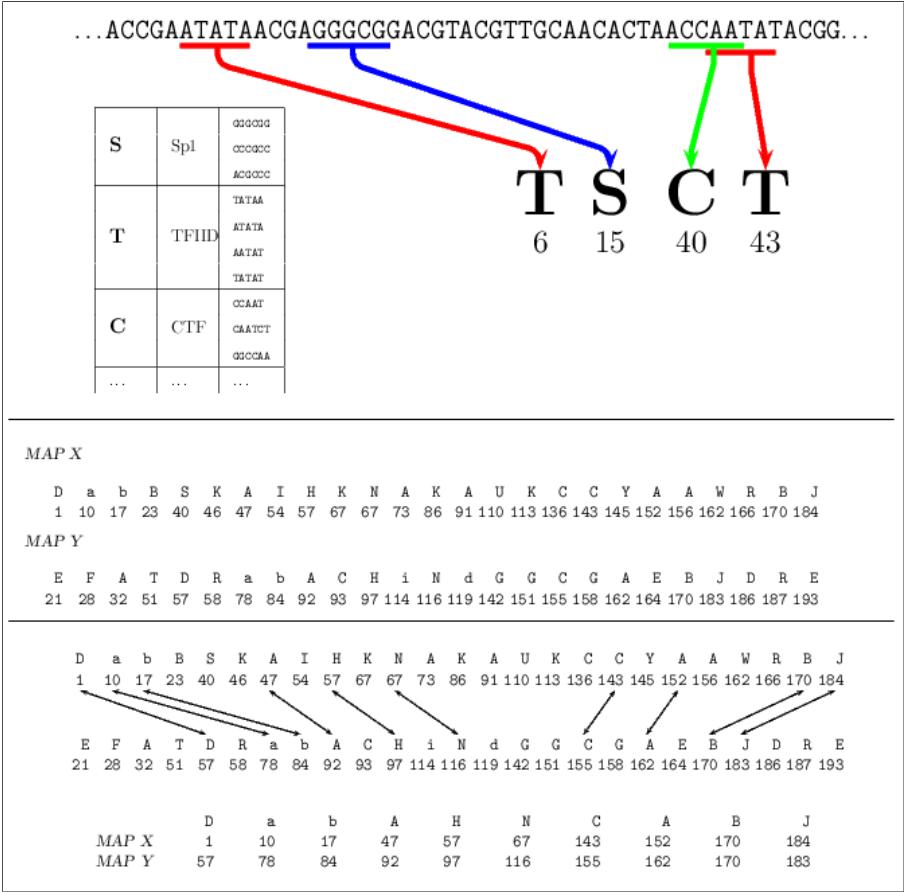


Figure 5.2 TF-maps: construction and alignment. (A) The sequence of a promoter is searched for occurrences of known binding motifs for transcription factors (TFs). Matches are annotated with the position of the match in the primary sequence, and the label of the TF. Because TFs can bind to motifs showing no sequence conservation, labels of the same TF at different positions may correspond to different underlying nucleotide sequences. We refer here to these sequences of pairs (“label”, “position”), transcription factor maps (or TF-maps). TF-maps are actually more complicated. First, we do not only register the position of each match, but also its length. Second, while in the example here, sequence motifs are associated to TFs by means of a (binary) look-up table, in our work we have instead used collections of position weight matrices. Matches to transcription factor binding sites (TFBSs) are thus scored, and this score is also registered. (B) TF-map of the promoter region of two hypothetically co-regulated genes X and Y. Each letter corresponds to a different TF. We assume that 200 nucleotides upstream of the annotated transcription start site (TSS) have been considered, with position 1 corresponding to position -200 from the TSS. (C) Global pairwise alignment of the two co-regulated genes X and Y. Only positions with identical labels can be aligned. Essentially, the alignment finds the longest common substrings constrained to maximizing the sum of the scores (not shown here) of the aligned positions, and minimizing the differences in the distances on the primary sequence between adjacent positions.

the scoring function to weight the scores of the TFBSs. We also explicitly prohibited overlap between the sites.

Thus, the maximum similarity S_{ij} between TF-maps $A = a_1 \dots a_i$ and $B = b_1 \dots b_j$ where the site a_i^f is equal to the site b_j^f , can be computed as:

$$S_{ij} \equiv S(a_i, b_j) = \alpha(a_i^s + b_j^s) + \max_{\substack{i' < i \\ j' < j \\ a_i^{p^2} < a_i^{p^1} \\ b_j^{p^2} < b_j^{p^1}}} \{S_{i'j'} - \lambda(i - i' - 1 + j - j' - 1) - \mu(|(a_i^{p^1} - a_{i'}^{p^1}) - (b_j^{p^1} - b_{j'}^{p^1})|)\}. \quad (5.3)$$

Sequence alignments and meta-alignments

There is an intimate relationship between the Equation 5.3 and the Needleman and Wunsch recurrence as revisited by Smith et al. (1981) in which the conventional pairwise sequence alignment is based (see Chapter 3, Section 3.3).

In fact, the sequence alignment class of algorithms are a particular case of the more general class of map alignment algorithms. Let us analyze the form in which the conventional sequence alignment calculates any value in the similarity matrix S , trying to detect for each element in such a recurrence its counterpart in Equation 5.3:

- ① The matches and the substitutions between two symbols x and y are assigned the value of the corresponding scoring function $s(x, y)$ in a sequence alignment. The matches between two elements in a meta-alignment are also scored using a similar function (the α parameter in Equation 5.3). Let us consider $\alpha = (\alpha_1, \alpha_2 \dots \alpha_k)$ the family of scoring functions for evaluate any type of identity and substitution between two symbols x and y . If the mapping quality score of each element is omitted, the scoring functions s and α are equivalent.
- ② The number of gaps in a sequence alignment is punished by the scoring function $s(x, -) = s(-, x)$. There is not an explicit penalty for introducing a single gap into a meta-alignment. However, the λ parameter punishes the number of elements in two maps that are not included in the optimal met-alignment. Because such unaligned elements are implicitly aligned to gaps in the other map, the λ parameter is the equivalent of the scoring function $s(x, -)$.
- ③ The μ parameter must be silenced due to the lack of mapping information in conventional sequences.

A trivial mapping function to translate a sequence of nucleotides into a map that can be meta-aligned consists on using the position of the elements in the sequence also as the position in the map. The length of every feature is in this case one position. The score of each feature is neglected as nucleotides do not have this value. With these considerations in mind, the sequence of nucleotides $S = \text{ATTACTG}$ can be transformed into the map M :

S:	A	T	T	A	C	T	G
M:	(A, 1, 1, ·)	(T, 2, 2, ·)	(T, 3, 3, ·)	(A, 4, 4, ·)	(C, 5, 5, ·)	(T, 6, 6, ·)	(G, 7, 7, ·).

The meta-alignment class of algorithms can deal, therefore, with any sequence alignment problem. However, the opposite is not true, as meta-alignments involve management of higher-order level features that are not supported in the classical sequence comparisons.

Naive implementation

A naive implementation of the recursion above (Equation 5.3) involves the recursive filling of the cells S_{ij} in the matrix S (Waterman et al., 1984). In the pseudocode shown in Figure 5.3, the elements of the maps A and B are represented as structures a_i and b_j , with the functions *factor*, *score*, *pos1* and *pos2* returning the values of the corresponding fields. The variable *currentSim* stores the optimal score so far computed. The resulting meta-alignment can be easily retrieved using a supplementary structure *path*(i,j) which points to the previous cell in the optimal path leading to cell S_{ij} . In addition, for each cell S_{ij} , the function *ComputeInitialSimilarity* calculates the initial score of a hypothetical alignment that includes only a_i and b_j .

Note that to compute the optimal score at S_{ij} with this algorithm, all the cells S_{kl} ($k < i$, $l < j$) need to be explored (see Figure 5.3). Therefore, if the lengths of the TF-maps A and B are m and n respectively, the cost of computing $S(A, B) = S(a_m, b_n)$ is $O(mn \cdot mn) = O(m^2n^2)$. Under the assumption that m and n are similar lengths, the final cost function is $O(n^4)$.

Enhanced implementation

Myers and Huang (1992) described an improved algorithm for computing in $O(mn(\log m + \log n))$ time the minimum distance between two restriction maps of length m and n respectively under the original framework proposed by Waterman (1984). The algorithm, reviewed in Chapter 3, is basically a sparse dynamic programming computation in which candidate lists are used to model the future contribution of all previously computed cells in distance matrix D to those yet to be computed. The cells in the list that can not affect the values of any cell to be computed are eliminated from the list. The key concept of this algorithm is the mapping of the original matrix D to another matrix in which each cell is indexed by the positions of the sites in the original sequences, and not by their positions in the maps. During the computation, this matrix is partitioned into intervals for which only a representative cell is used to compute the best alignment ending at each match in a given interval.

Here, we can not directly export this strategy, because, in contrast to the restriction enzyme maps which are points in the sequence, TFBSs are sequence intervals (having, thus, two dimensions). In addition, different TFBSs can start at the same point, but end at different positions. Since we explicitly prohibit overlapping between TFBSs in the alignments, the assignation of a cell representative within a given interval must not be irreversible.

```

Pre  $\equiv$  A, B: list of <factor,pos1,pos2,score>
(* Calculating the element i, j in S *)
for i = 0 to |A| - 1 do
  for j = 0 to |B| - 1 do
    if factor( $a_i$ ) = factor( $b_j$ ) then
      5:    $S(i, j) \leftarrow \text{ComputeInitialSimilarity}();$ 
       $x \leftarrow \alpha (\text{score}(a_i) + \text{score}(b_j));$ 
      (* Searching the best previous match in S *)
      for  $i' = 0$  to  $i - 1$  do
        for  $j' = 0$  to  $j - 1$  do
          10:   if  $\text{pos2}(a_{i'}) < \text{pos1}(a_i)$  and  $\text{pos2}(b_{j'}) < \text{pos1}(b_j)$  then
             $y \leftarrow \lambda((i - i' - 1) + (j - j' - 1));$ 
             $z \leftarrow \mu(|(\text{pos1}(a_i) - \text{pos1}(a_{i'})) - (\text{pos1}(b_j) - \text{pos1}(b_{j'}))|);$ 
             $\text{currentSim} \leftarrow S(i', j') + x - y - z;$ 
            if  $\text{currentSim} > S(i, j)$  then
              15:    $S(i, j) \leftarrow \text{currentSim};$ 

```

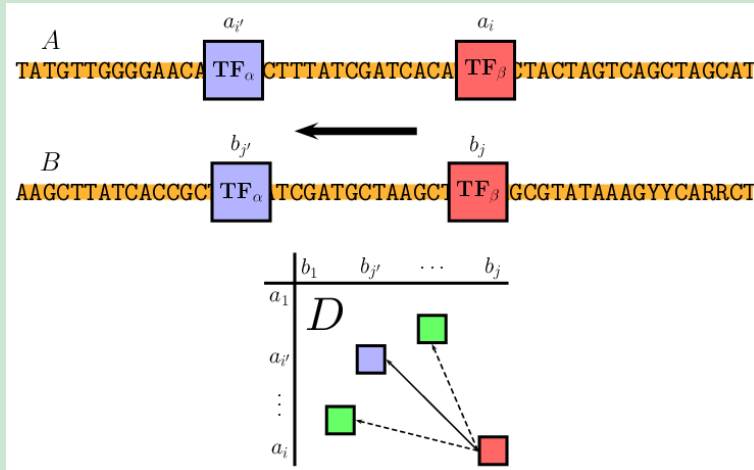


Figure 5.3 The Naive TF-map alignment algorithm. The whole matrix must be visited for each new match S_{ij}

However, we have still taken advantage of the extreme sparsity of the matrix S when aligning TF-maps (Blanco et al., 2006b). Note that, in general, the probability of matching two elements from two sequences of characters that follow a uniform random distribution is inversely proportional to the size of the character alphabet. For instance the probability of matching two nucleotides when comparing two random DNA sequences in the four letter alphabet is about 0.25. In an alphabet of about 100 characters –the order of magnitude of the alphabets of symbols denoting TFs that we are considering here– such a probability would be about 0.01. When aligning sequences in alphabets of such sizes, the matrix S above, that only takes values for match positions between A and B, becomes therefore extremely sparse. Indeed, Figure 5.4 displays the occupancy of the matrix S corresponding

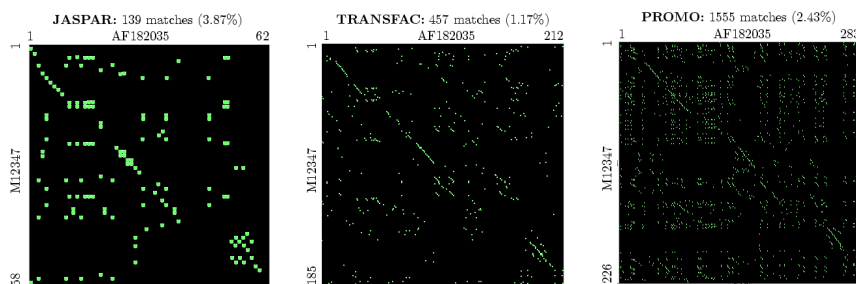


Figure 5.4 Graphical representation of the sparse dynamic programming matrix S . Matrices produced by the transcription factor map alignment between the human and mouse promoters of the *skeletal alpha-actin* gene (ACTA1, GenBank entries AF182035 and M12347), using different collections of position weight matrices for transcription factor binding sites (TFBSs). The axes of the matrix list the transcription factor labels of the predicted TFBSs in the human and mouse promoters. Despite the differences in the total number of predicted TFBSs depending on the collection, the occupancy of the matrix remains consistently low.

to the alignments of the TF-maps obtained on the human and mouse promoters of the *skeletal muscle α -actin* gene (ACTA1, GenBank entries AF182035 and M12347). We have used three different collections of PWMs for TFBSs (see next section) to obtain the TF-maps of both promoter sequences. In all cases, despite the differences in the lengths of the obtained maps, the occupancy of the matrix S is well under 5%.

In the algorithm presented in Figure 5.5, we substitute the two internal nested loops by a list L to register the coordinates of the match cells in the sparse matrix S . Each node of L is represented as structures p and n with the functions *abscissa* and *ordinate* returning the corresponding coordinates. Thus, to compute the optimal score at the cell S_{ij} , only the non-empty cells in S need to be accessed. In addition, we maintain the list sorted by optimal score, so that the cell scoring the maximum value is at the beginning of the list. Scanning the list from the beginning to the end implies that, in most cases, only a few nodes will need to be accessed before a critical node is reached beyond which the optimal score can not be improved.

While investigating the exact complexity of this algorithm is difficult –depending mostly on the size of the input maps and the sparsity of the resulting matrix S –, the expected time cost analysis can be performed. The $O(n^4)$ cost of the naive algorithm can be explained in terms of (a) a first quadratic term derived from the obligatory comparison between all of the TFBSs of both maps to detect the match cells and (b) a second quadratic term necessary to search for each match the best adjacent previous pair in the optimal TF-map alignment.

In this enhanced algorithm, the contribution (a) is inevitable so that the lower bound of the cost function is the number of matches between both TF-maps, that is $O(n^2)$. However, the substitution of the two inner loops for a list of cell matches sorted by optimal score does affect the contribution (b). Thus, such a term is now equivalent to the expected number of consulted elements of the ordered list L to compute each S_{ij} value. This expectation can be approximated to

```

Pre  $\equiv$  A, B: list of <factor,pos1,pos2,score>, L: list of <abscissa,ordinate>, L =  $\emptyset$ 
(* Calculating the element i, j in S *)
for i = 0 to |A| - 1 do
  for j = 0 to |B| - 1 do
    if factor(ai) = factor(bj) then
      5:   S(i, j)  $\leftarrow$  ComputeInitialSimilarity();
          x  $\leftarrow$   $\alpha$  (score(ai) + score(bj));
          (* Searching the best previous match in L *)
          p  $\leftarrow$  first(L);
          i'  $\leftarrow$  abscissa(p);
      10:  j'  $\leftarrow$  ordinate(p);
          while end(L) = FALSE and S(i', j') + x > S(i, j) do
            if pos2(ai') < pos1(ai) and pos2(bj') < pos1(bj) then
              y  $\leftarrow$   $\lambda((i - i' - 1) + (j - j' - 1))$ ;
              z  $\leftarrow$   $\mu(|(\text{pos1}(a_i) - \text{pos1}(a_{i'})) - (\text{pos1}(b_j) - \text{pos1}(b_{j'}))|)$ ;
      15:  currentSim  $\leftarrow$  S(i', j') + x - y - z;
          if currentSim > S(i, j) then
            S(i, j)  $\leftarrow$  currentSim;
            p  $\leftarrow$  next(L);
            i'  $\leftarrow$  abscissa(p);
      20:  j'  $\leftarrow$  ordinate(p);
          n  $\leftarrow$  CreateNewNode(i, j);
          InsertNode(n, L);

```

Figure 5.5 The Enhanced TF-map alignment algorithm.

$$O\left(\sum_{\alpha \in \mathcal{A}} (P(\alpha)^2 n^2)\right) \quad (5.4)$$

where \mathcal{A} is the set of symbols (in our case the alphabet of TFs) and $P(\alpha)$ is the probability to match the symbol α in a random trial (it is a particular case of the sequence comparison by hashing, see Theorem 8.1 in [Waterman \(1995\)](#)). Therefore, under the previous hypothesis of a comparison between two TF-maps in an alphabet of 100 characters that follows a uniform random random distribution ($P(\alpha) = 0.01$, only 1% of the matrix is occupied), the expected value of the contribution (b) is $O(0.01 n^2)$.

The empirical results obtained during the program training (see next section) confirmed such analysis ([Blanco et al., 2006b](#)). In average, on the order of 200 million elements were consulted by the naive algorithm during the optimization. In contrast, the enhanced algorithm only needed to access nearly two million elements to compute the same set of alignments (see Figure 5.6).

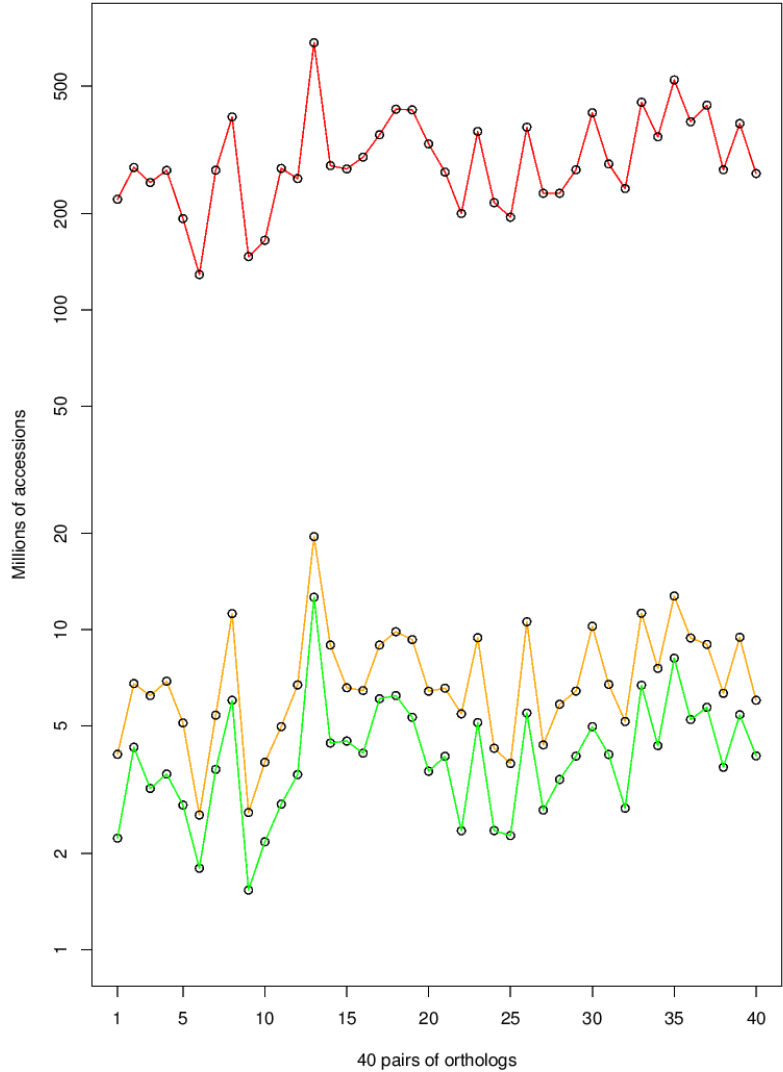


Figure 5.6 Number of accessions (in millions) to the matrix S. In red, the performance of the Naive algorithm; in orange, the performance of the Enhanced algorithm, with a normal list L; in green, the performance of the Enhanced algorithm, sorting the list L.

5.4 TF-map alignment training

The optimal alignment between two TF-maps is obviously dependant on the α , λ , and μ parameters. In principle, we want the optimal alignment between the maps derived from promoter sequences of two co-expressed genes to include most of the mapped TFBSs known to be involved in the regulation of the genes (high sensitivity), and few of the mapped TFBSs

not known to be involved in such regulation (high specificity). The implicit assumption here is that the TFBSs in the alignment are considered predictions of TFBSs on the underlying promoter sequences. It is also important to stress that two different TFBSs can be aligned if they correspond to the same TF.

The optimal parameter configuration, however, is likely to depend on the particular problem to be addressed: the genes to be compared (orthologous genes from different species or genes co-regulated after an expression microarray experiment, for instance), and the particular protocol to map the TFBSs into the original promoter sequences. Often the optimal configuration of parameters will be specific of the pair of gene promoters to be compared.

With these caveats in mind, since our focus here is on mammalian comparisons, we have estimated the parameters that are globally optimal when aligning a set of well annotated human-mouse orthologous promoter pairs (Blanco et al., 2006b). The underlying assumption is that these orthologous pairs are regulated in a similar way. We have estimated the optimal parameters separately in three different collections of PWMs for locating TFBSs, and in each case we have chosen the parameters such that the resulting global alignment achieved the maximum average sensitivity and specificity as defined below.

Datasets

From several landmark papers in the field (Wasserman and Fickett, 1998; Krivan and Wasserman, 2001; Blanchette and Tompa, 2002; Dermitzakis and Clark, 2002; Lenhard et al., 2003), we have gathered and manually curated a collection of 278 TFBSs (139 + 139 orthologous sites) that had been experimentally tested in 40 orthologous human and rodent genes. The transcription start site (TSS) of each entry in the literature was compared to the RefSeq (Pruitt et al., 2005) annotation of the corresponding genome to ensure that we were dealing with the actual proximal promoter. Because most (214 out of 278) of the annotated TFBSs are located in the 200 nucleotides immediately upstream of the TSS, we restricted to this region in our training and evaluation analysis, and considered only those cases for which the same pair of TFBSs had been annotated in this region for both species. This resulted in a collection of 202 sites (101 + 101) from 36 genes, to which we refer here as the HR SET.

We have estimated the optimal parameters in the HR SET for the JASPAR 1.0, PROMO 2.0 and TRANSFAC 6.3 collections. In the three cases, the original frequency coefficients of the matrices have been converted into log-likelihood ratios using the random equiprobability distribution as a background model. The log operation can not be directly performed on matrix positions containing null values (that is, 0 occurrences). We have instead estimated the value of the log-likelihood function for the null positions in a given matrix row, taking into account the values computed in that row for one and two occurrences. Let $y = f(x)$ be the log-likelihood function approached as a line that goes from the point $P = (x_1, y_1)$ to the point $Q = (x_2, y_2)$. If we consider $P = (x_1, 1)$ and $Q = (x_2, 2)$ which correspond to the cases in which one and two occurrences are present, the values x_1 and x_2 can be easily computed. Thus, the equation of the line that goes from Q to P can be inferred for each row of the matrix. In particular, the value of this line in the point $R = (x_0, 0)$ can be trivially calculated, being used as an estimation for the null values in that row of the matrix.

Let M be a PWM constructed from 33 TFBSs, where M_i and M_i^* denote the absolute and

relative frequency of each nucleotide at the position i , respectively. The conversion from M_i into a log-likelihood ratio matrix is explained in the following example (base-e logarithms):

	A	C	G	T
M_i	7	25	0	1
$M_i^* = \frac{M_i}{33}$	0.21	0.75	0	0.03
$\log \frac{M_i^*}{0.25}$	-0.164	1.109	?	-2.110
Estimation				-2.803

The resulting matrices were used to obtain the list of TFBSs matches along the 200 bases upstream of the TSS in each of the 36 pairs of promoter sequences from the HR SET. A prediction obtained with a given PWM was accepted if it had an score above the 50% (JASPAR), 70% (PROMO) and 55% (TRANSFAC) of the maximum possible score for such PWM. These values correspond in the three cases to the conventional 80% threshold when considering the original frequency matrices (Blanco et al., 2006b).

Those annotated TFBSs not included in the predictions for both orthologous pairs (either because no matrix exists in the collection for such TFBSs, or because the match is below the threshold) were discarded. This reduced the effective number of training gene pairs (those with at least one real predicted TFBS for both orthologous pairs) from 36 to 29 for the three collections considered here (Blanco et al., 2006b).

Table 5.1 shows for each collection the total number of matrices, and TFs to which they correspond, the number of genes for which at least one annotated TFBS is predicted on each ortholog after the search, and the number of real and predicted TFBSs (the total and the average per gene pair). As it is possible to see, slightly more than three conserved TFBSs were annotated per orthologous gene pair (Blanco et al., 2006b).

Collecting regulatory data

Information about the genomic coordinates and the sequence of experimentally identified transcription factor binding sites is found scattered under a variety of diverse formats. The availability of standard collections of such high-quality data is important to design, evaluate and improve novel computational approaches to identify binding motifs on promoter sequences from related genes.

Typically, computational methods to detect regulatory elements use their own training set of experimental annotated TFBSs. These annotations are usually collected from bibliography or from general repositories of gene regulation information, such as JASPAR (Sandelin et al., 2004) or TRANSFAC (Matys et al., 2003). However, each program establishes different criteria and formats to retrieve and display the data that forms the final training set, which makes the comparison between different methods very difficult. The construction of a good benchmark to evaluate the accuracy of several pattern discovery methods is therefore not a trivial procedure (Tompas et al., 2005).

To build the TF-map alignment training dataset, we gathered from the literature a collection of experimentally validated binding sites that are conserved in at least two orthologous vertebrate promoters. The sites and the promoter sequences were manually curated to ensure data consistency. The data is publicly available at the ABS database (see Web Glossary).

We annotated in ABS (Blanco et al., 2006a) up to 650 experimental binding sites from 68 transcription factors and 100 orthologous target genes in human, mouse, rat or chicken genome sequences. Computational predictions and promoter alignment information are also provided for each entry. In addition, we provided a web interface to interact and analyze the promoters and their binding sites (see Figure 5.7). We also included a customizable generator of artificial datasets and an evaluation tool to aid during the training of motif-finding programs (Blanco et al., 2006a).

Accuracy measures

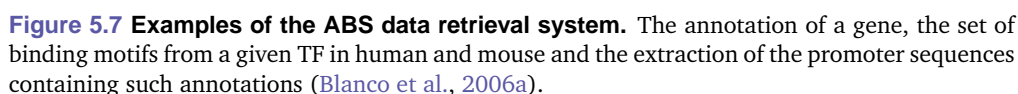
After the maps were obtained, we aligned them within each orthologous pair using the algorithm described in the previous section with different combinations of parameters. Each parameter was allowed to independently take values between 0.0 and 1.0, in incremental steps of 0.01. In total, thus, one million parameter configurations were evaluated for each collection of PWMs. For each configuration, the resulting optimal alignments on the pairs of orthologous promoters (that is, the predicted TFBSs) were compared to the annotated TFBSs in the promoters.

Two values were computed to measure the agreement between predicted and annotated TFBSs: sensitivity and specificity. Sensitivity is the number of correctly predicted TFBSs over the number of annotated TFBSs, and specificity is the number of correctly predicted TFBSs over the number of predicted TFBSs. We used here the term specificity as in the gene finding literature. However, the value that we compute here is more generally known as Positive Predictive Value. We considered an annotated TFBS to be correctly predicted when there was a predicted TFBS that overlapped it by at least 1 nucleotide in both human and mouse sequences, irrespectively of whether the TF label associated to the aligned TFBS matched that of the annotated TFBS. This is because TFBSs for different TFs often cluster at the same position when using PWMs (see Figure 5.8). If a similar cluster occurs in the two sequences to be aligned, our algorithm will inevitably choose to align the pair of TFBSs with the highest sum of match scores.

As an optimization measure we computed the average value of sensitivity and specificity. Table 5.1 lists the optimal combination of parameters with regard to this measure for each of the three collections of PMWs used here. Table 5.1 also lists sensitivity, specificity, their average, the average length of the optimal alignments (that is, the number of predicted TFBSs after the alignment), and the fraction of the promoter region covered by the predicted (aligned) TFBSs. In addition, for each optimal configuration we have also computed the same set of accuracy measures under the strict criterion of considering an annotated TFBS to be correctly predicted only when the TF label of the prediction matched that of the overlapped annotation. We also computed sensitivity and specificity at the nucleotide level. At this level, we compute the number of nucleotides in predicted TFBSs that are also in annotated TFBSs.

		JASPAR	PROMO	TRANSFAC	JASPAR _{TOP50}
PWMs	Number of Matrices	111	316	442	50
	Number of TFs	101	181	296	47
TF-MAPS	Number of Gene Pairs	29	29	29	17
	Number of Real TFBSs	93 (×2)	94 (×2)	94 (×2)	50 (×2)
	Number of Real TFBSs per Gene Pair	3.2 (×2)	3.2 (×2)	3.2 (×2)	2.9 (×2)
	Number of Predicted TFBSs	2683 × 2605	8322 × 8027	6644 × 6628	207 × 216
	Number of Predicted TFBSs per Gene Pair	93 × 90	287 × 277	229 × 229	12 × 13
	(NUCLEOTIDE)	Sensitivity, Specificity	0.97, 0.16	0.99, 0.14	0.99, 0.14
		Correlation Coefficient	0.10	0.04	0.03
		Coverage	88%	97%	98%
	(SITE)	Sensitivity, Specificity	1.00, 0.02	1.00, 0.00	1.00, 0.00
		Average	0.51	0.50	0.50
TF-MAP ALIGNMENTS	α, λ, μ	0.5, 0.1, 0.1	0.25, 0.1, 0.2	0.25, 0.1, 0.1	0.5, 0.1, 0.1
	Length	12.7 (×2)	23.5 (×2)	15.2 (×2)	3.4 (×2)
	(NUCLEOTIDE)	Sensitivity, Specificity	0.76, 0.23	0.72, 0.19	0.85, 0.21
		Correlation Coefficient	0.19	0.10	0.18
		Coverage	51%	62%	65%
	(SITE)	Sensitivity, Specificity	1.00, 0.25	0.94, 0.13	0.98, 0.21
		Average	0.63	0.53	0.59
	(SITE+LABEL)	Sensitivity, Specificity	0.57, 0.07	0.30, 0.03	0.29, 0.04
		Average	0.32	0.16	0.16

Table 5.1 TF-map alignment accuracy results on the HR SET. Parameters were estimated independently using three different collections of position weight matrices (PWMs) for transcription factor binding sites (TFBSs) to obtain the TF-maps of the promoter sequences. The table has three parts. On **top**, number of matrices in each of these collections, and the number of transcription factors (TFs) these matrices correspond to. In the **middle**, statistics of the resulting TF-maps: number of promoter pairs (out of 36) for which matches to at least one common TFBS was found in both the human and mouse orthologs (and for which, therefore, there exist a non-void TF-map alignment), total and average number of real TFBSs per promoter sequence, total and average number of predicted TFBSs per promoter sequence, and sensitivity and specificity at the nucleotide and site levels (see main text for definitions). The average sensitivity and specificity at the site level is the optimization measure when estimating the parameters of the algorithm. Coverage is the fraction of the sequence of the promoters covered by matches to TFBSs. At the **bottom**, results of the optimal TF-alignments: optimal parameters and average length (number of aligned elements in the optimal TF-map alignments), measures of sensitivity and specificity at the levels of nucleotide, site overlap, and site plus label match (see main text for definitions). Coverage is the fraction of the sequence of the promoters covered by matches to TFBSs.



This number over the total number of nucleotides in annotated TFBSs is the sensitivity, and over the total number of nucleotides in predicted TFBSs is the specificity. Finally, as a summary of these two numbers we compute the correlation coefficient. All the accuracy measures were also computed on the initial PWM predictions, prior to the alignments.

	SENSITIVITY	SPECIFICITY	CORRELATION COEFFICIENT	COVERAGE
BLASTN	0.70	0.19	0.16	54%
BLASTN _{WSIZE=7}	0.85	0.18	0.15	63%

Table 5.2 Results when using BLASTN to detect conservation between orthologous pairs.

Accuracy results

As it is possible to see, the main effect of the meta-alignment is the dramatic reduction in the number of predicted TFBSs that typically result after a PWM-based search (see also Figure 5.8). Taking, for instance, the popular TRANSFAC collection, the average number of TFBSs predicted per promoter in our dataset using this database is about 230. The TF-map alignment reduces this number approximately 15-fold, while the predicted TFBSs still covering essentially all annotated TFBSs (Blanco et al., 2006b). This gain in specificity is not simply due to the selection of an arbitrary set of non-overlapping TFBSs, since as a result of the map alignments the proportion of the promoter region covered by predicted TFBSs drops from 98% to 65% –a number which is more consistent with the estimated occupancy by TFs of the core promoter regions (Wray et al., 2003).

In this regard, we have compared the map alignments here with direct sequence alignments in their ability to identify TFBSs in the promoter regions of co-regulated genes. We have used NCBI-BLASTN (Altschul et al., 1990) to identify conserved blocks in the promoter region of the orthologous pairs in the HR SET. We have searched for local, instead of global alignments because we expect the TFBSs to distribute discretely along the promoter region –resulting in a patch of conserved and non-conserved fragments. In addition, local alignments are insensible to the relative rearrangements in the order of the TFBSs between the promoters sequences compared. This is an advantage over the map alignments, which require colinearity of the TFBSs in the sequences to be compared. Despite this, and the fact that promoter elements are usually embedded within well conserved sequences in human and mouse orthologous promoters, map alignments are comparable or outperform the BLASTN comparison when identifying TFBSs in them (Blanco et al., 2006b). The correlation coefficient between the sequences covered by the BLASTN alignments and the annotated TFBSs is 0.15, while the same measure when considering the sequences covered by the map alignments is 0.19 for JASPAR, 0.10 for PROMO and 0.18 for TRANSFAC. Table 5.2 lists these values, as well as the the values of sensitivity and specificity. To obtain these values, BLASTN was run with default parameters, but decreasing the word size to 7 (the minimum accepted value in NCBI-BLASTN). This allows for the detection of shorter and weaker alignments. The performance of BLASTN degrades if we increase the word size. We obtained similar results using the WU-BLASTN version, which allows for shorter word sizes (data not shown).

The values in Table 5.1 reflect differences between the three collections of matrices when used in the context of the map alignments. In this context, JASPAR appears to show the better balance between sensitivity and specificity. This can be partially explained because there is less matrix redundancy –which in turn implies less overprediction– in JASPAR than in the other collections. To further minimize overprediction, we have computed the information content of all JASPAR matrices and selected the most informative ones. Let P be a PWM where $P(x, i)$ denotes the probability of observing the nucleotide x in the position i of a

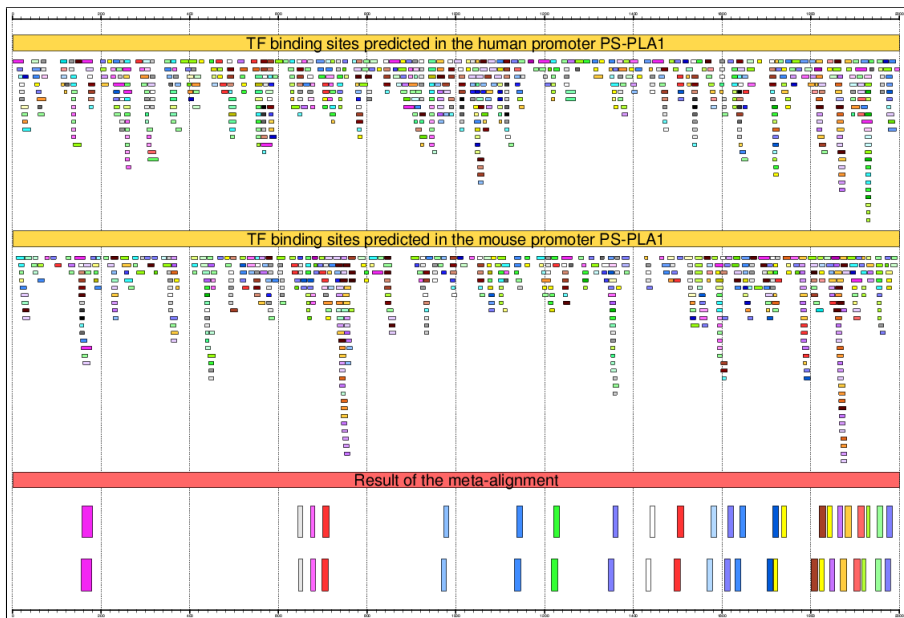


Figure 5.8 TF-map alignment of the human and mouse PLA1A gene. Results of the TF-alignment of the human and mouse promoters of the *phospholipase A1 member A* gene (PLA1A, RefSeq entries NM_015900, NM_134102). Here, the 2000 nucleotides upstream of the annotated transcription start site (TSS) have been considered (with position 1 corresponding to -2000). The TF-maps on these sequences were obtained using TRANSFAC 6.3 (Matys et al., 2003). These maps contained 676 predicted binding sites in human and 595 in mouse (threshold 85%), and they are represented graphically on the top right of the figure. Each box represents a different binding site and the color corresponds to the associated transcription factor (TF). The resulting TF-map alignment is also represented graphically at the bottom right. As it is possible to see, while the region proximal to the TSS is not more dense in predicted TFBSs than other regions, most of the aligned elements cluster near to the TSS. Indeed, more than half of the elements in the TF-map alignments are within 500 nucleotides of the TSS. The program GFF2PS (Abril and Guigo, 2000) has been used to obtain the graphical representation of input predictions and final alignment.

motif of length n . The amount of information R of the matrix P is defined as Schneider and Stephens (1990):

$$R(P) = \sum_{i=1 \dots n} \left(2 + \sum_{x \in A, C, G, T} P(x, i) \log P(x, i) \right). \quad (5.5)$$

When using the collection of the 50 JASPAR matrices with the highest R value (which we refer to as JASPAR_{TOP50}) to obtain the TF-maps, detection of TFBSs through map alignments improves over the entire set of JASPAR matrices: while there is some loss of sensitivity, there is a larger gain in specificity (see Table 5.1).

Finally, we have also performed a complementary test to measure the specificity of the TF-map alignments (Blanco et al., 2006b). As a negative control, we have shuffled the or-

thologous pairing in the HR SET to construct a pool of unrelated human-mouse gene pairs. Then, the corresponding TF-map alignments between these non-orthologous paired promoters were obtained using the parameters previously optimized. For the three collections of matrices, the TF-map alignments between pairs of unrelated promoters were significantly shorter with an average score about 50% smaller than TF-map alignments between “bona fide” orthologous promoters. For instance, the average length of the TF-map alignments between orthologous promoters when using the JASPAR collection was 12.7 TFBSs, with an average score of 55.2. In contrast, the length of the TF-map alignments between non-related promoters was 8.36 TFBSs, with an average score of 20.67. The sites in the alignments involving non-orthologous gene promoters may hypothetically correspond to general regulatory elements present in most core promoters. An alternative, more probable, hypothesis is that they reflect the poor specificity of most PWMs representing TFBSs. Indeed, when we perform the same test using the more informative JASPAR_{TOP50} collection, no TF-map alignments can be obtained between any pair of the non-related promoters.

5.5 Using TF-map alignments to distinguish promoters from other genomic regions

Results in the previous section indicate that alignments of TF-maps can contribute –together with other tools, such as primary sequence alignments– to the characterization of the promoter region of co-regulated genes. This contribution is mostly obtained through the substantial reduction of the overwhelming number of candidate TFBSs that PWMs and other pattern based searches typically produce. The co-regulated genes in the test case of the previous section, however, were orthologous human-mouse pairs. The promoter regions of such pairs show substantial sequence conservation (Waterston et al., 2002). It can be argued that under such circumstances map alignments may not be much more informative than primary sequence alignments. Note that, in general, good alignments at the primary sequence level will inevitably result –given the low specificity of the PWM search– in good map alignments, although such map alignments may bear little relationship to the underlying conserved configurations of TFBSs. To assess to what extent good TF-map alignments are simply a reflection of underlying sequence conservation, we have compared the meta-alignments obtained using JASPAR_{TOP50}, in the 200 nucleotides of the promoter region of the 36 gene pairs from the HR SET, with the meta-alignments obtained in fragments of 200 nucleotides from intergenic (2000 nucleotides upstream of the TSS), 5'UTR (downstream of the TSS), coding (downstream of the translation start site and considering only coding DNA), intronic (downstream of the first intron junction), and downstream (downstream of the transcription termination site) sequences. The test is graphically represented In Figure 5.9.

We have computed the average score of the map alignments in each of the genomic regions and have identified, for each homologous pair, the genome regions in which the alignment produces the highest score (Blanco et al., 2006b). We have performed the same exercise using global pairwise sequence alignments, obtained with CLUSTALW (Thompson et al., 1994). Results appear in Table 5.3 (Top). As expected, nucleotide sequence alignments score the highest in the coding regions (in 26 out of 36 cases), followed by the alignments in the promoter (5 out of 36) and 5' UTR regions (4 out of 36). The scores

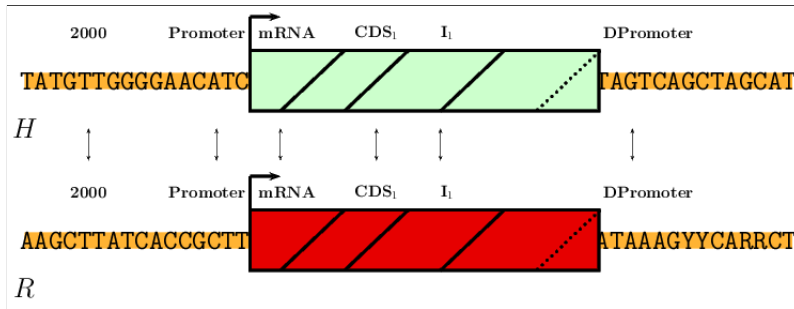


Figure 5.9 TF-map alignment on several genomic samples of two species.

of the sequence alignments show that promoter regions are less conserved than coding regions, and have a level of conservation similar to that observed in 5'UTRs. Despite this, TF-map alignments score the highest in the promoter regions (in 25 out of 36), where the average score of map alignments is almost twice as high as that of the coding regions. Only in 6 out of 36 cases the TF-map alignment scores the highest in coding regions. Interestingly, while intron sequences in the orthologous human-mouse pairs are much less conserved than 5'UTRs, TF-map alignments have a similar score in both regions. In fact, in 3 cases, TF-map alignments have the highest score in first introns, while only in 1 case in 5'UTRs. This is consistent with the fact that first introns are known to often contain regulatory motifs.

In order to measure the ability of TF-map alignments to detect conserved regulatory elements at larger evolutionary distances –at which the degree of sequence conservation may be negligible– we have carried out the same analysis on a set of human-chicken orthologous pairs derived from the HR SET. Using the RefSeq gene set as mapped into the UCSC genome browser, we have identified the chicken ortholog for 25 genes in the HR SET. We refer to the resulting set of human-chicken gene pairs as the HC SET (Blanco et al., 2006b). As before, we have compared promoter, intergenic, 5'UTR, coding, intronic and downstream sequences between the orthologous human-chicken genes using both TF-map alignments based on JASPAR_{TOP50} and sequence alignments using CLUSTALW. Results appear in Table 5.3 (Bottom). While, as expected, the scores of the alignments are, in both cases, clearly lower for human–chicken than for human–mouse comparisons, the same relative trends can be observed, with sequence alignments being most significant between coding regions, and TF-map alignments between promoter regions. However, while coding sequences are still distinctively conserved between human and chicken, similarity in promoter sequences degrades substantially. Indeed, in contrast with human-rodent comparisons, 5'UTRs are, for instance clearly more conserved than the promoters between human and chicken orthologous genes. Despite this lack of sequence similarity in the human-chicken promoter pairs and the fact that we trained our algorithm specifically on human and rodent genes, the TF-maps remarkably still score the highest in these regions (in 9 out of 25). Interestingly, TF-map alignments are able to score comparatively high in downstream regions even though they do not appear to exhibit sequence conservation; regulatory motifs have been occasionally reported on these regions. Overall, these results indicate that alignments of TF-maps are able to detect conservation of regulatory signals, which can not be detected by sequence similarity alone (Blanco et al., 2006b).

HR SET	TF-MAP ALIGNMENT		CLUSTALW	
	TOP1	Avg.Score	TOP1	Avg. score
CODING	6	10.86	26	1211.72
PROMOTER	25	20.45	5	979.27
5'UTR	1	4.56	4	958.50
DOWNSTREAM	1	2.31	1	395.38
INTRONIC	3	4.43	0	525.66
INTERGENIC	0	2.51	0	421.13
HC SET	TF-MAP ALIGNMENT		CLUSTALW	
	TOP1	Avg.Score	TOP1	Avg. score
CODING	2	1.66	21	820.92
PROMOTER	9	2.14	1	454.52
5'UTR	5	1.88	3	698.12
DOWNSTREAM	6	1.63	0	358.66
INTRONIC	3	1.49	0	384.52
INTERGENIC	0	1.55	0	368.04

Table 5.3 TF-map alignment results on several orthologous genomic samples (Top) Sequence and TF-map alignments of different genomic regions between the human and mouse orthologous pairs in the HR SET. (Bottom) Sequence and TF-map alignments of different genomic regions between the human and chicken orthologous pairs in the HC SET. TOP1 is the number of pairs in which the highest scoring alignment is found in a given genomic region.

Promoter identification with TF-map alignments

Promoter identification is still a difficult problem (reviewed in Chapter 4). TF-map alignments may be helpful in this problem. Using a set of 278 orthologous human-chicken gene pairs of another study (Abril et al., 2005), we have performed the following experiment.

We have extracted the human promoter of these genes (500 nucleotides) from the UCSC human genome distribution according to the RefSeq coordinates. For the chicken genes, we have extracted the mRNA from the chicken genome surrounded by 5,000 nucleotides upstream of the TSS and 5,000 downstream of the end of the transcript. Finally, we have extracted samples of 500 nucleotides from these long sequences, without overlapping between each contiguous windows. For each gene, the upstream promoter region, orthologous to that of human, is therefore located in the window between the positions 4,500 and 5,000 nucleotides (see Figure 5.10).

Next, we have used the 50 more informative matrices from TRANSFAC (Matys et al., 2003) as a mapping function to obtain the map of each sample in the chicken sequences. We have also used TRANSFAC for mapping the predicted TFBSs on the human promoters. The experiment consisted in performing the pairwise TF-map alignment between the human promoter and all of the samples in its chicken ortholog. Then, for each window we have counted in how many cases out of the 278 genes the TF-map alignment between the human promoter and that window sample scores the highest, among all of the windows. As shown in Table 5.4, the chicken gene fragment in which more genes hit the best was the 4,500 –

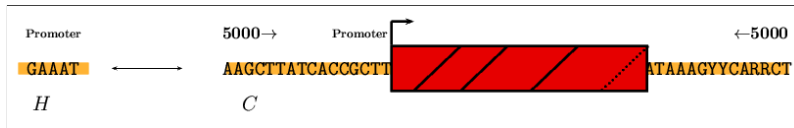


Figure 5.10 TF-map alignment in promoter detection.

5,000 sample (31%), which corresponds with the upstream promoter region according to the RefSeq annotations. In addition, 14% and 21% of the 278 gene pairs obtained the highest TF-map alignment score on the windows located at 4,000–4,500 and 5,000–5,500, respectively. This bias is not observed in the rest of the windows. These percentages agree well with the errors in the precise TSS annotation (Suzuki et al., 2004).

We also counted for each window in how many cases the meta-alignment between this sample and the human orthologous promoter scores among the TOP-10 best alignments. Despite the results are less significant, it is interesting to notice that in more than 200 gene pairs (76%), the TF-map alignment between the human promoter and the chicken sample in the window 4,500–5,000 was among the TOP-10. We repeated the test with the full collection of TRANSFAC 6.3 (442 matrices). The results, shown in Table 5.4, are slightly worse. This fact is probably related to the poor specificity of many matrices that are included in the full collection.

Again, we performed the same experiment with the program BLASTN, using the score of the best HSP on each alignment to rank the window comparisons. Table 5.4 lists the results. The sequence alignments can detect correctly the actual promoter pair in less than 16% of the 278 genes (31% among the best 10 alignments).

Future experiments should be conducted in a genome-wide mode to verify the accuracy of TF-map alignments in larger datasets. However, the meta-alignment, at least in this set of 278 gene pairs, was clearly superior to sequence alignment to detect the correct promoter region. In principle, we could be able with the TF-map alignments to accurately detect the promoter region in one species, scanning this genome with the orthologous promoter in the other informant genome.

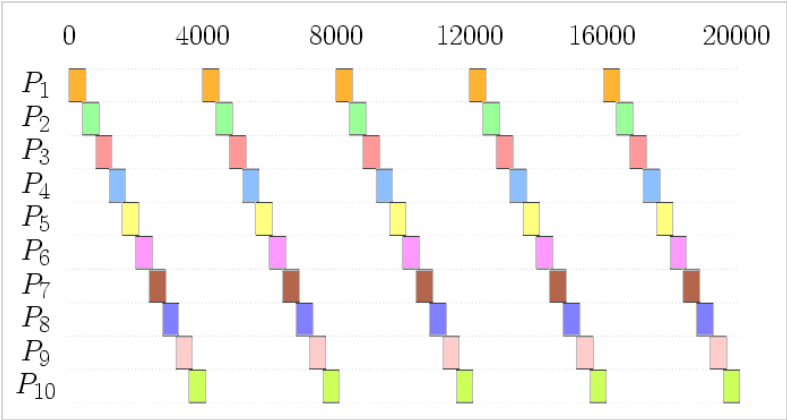
Parallel meta-alignment: PGWS

Let M be a long genomic region of m nucleotides. Let P be a short genomic sequence of p nucleotides, with $m \gg p$. The problem of mapping and aligning the sequence P to a contiguous set of windows in M must be carefully analyzed to obtain in a reasonable amount of time that window from M whose TF-map alignment to P reaches the highest value. If $p = 500$ bps, $m = 20,000$ bps and the windows are 500 bps with an overlap between adjacent windows of 100 bps then the number of windows (that matches the number of pairwise TF-map alignments to do) is 50. Obviously, if the test is repeated for hundreds of gene pairs, the computation of the best windows requires some improvement.

In fact, the calculation of the TF-map alignment between P and a given window from M is independent from the rest of alignments. Thus, the alignments can be easily dispatched to different processors to be performed in parallel. At the end of the process, the scores of

the alignments are ranked and displayed. Notice we are only interested in the score of the alignments to construct a ranking so the TFBSs that actually constitute them are logically not necessary in this case. Thus, we register the value calculated on each dynamic programming similarity matrix, but the paths of the alignments are not constructed.

Following with the same example: if there are 10 available processors, we can divide uniformly the list of windows (alignments) among them using any offset schema to ensure the load of each processor is similar. For instance, if we consider an offset of 4,000 bps between two windows that are processed by the same unit, we will assign the series of alignments $(M_{0-500}, M_{4000-4500}, M_{8000-8500}, M_{12000-12500}, M_{16000-16500})$ to the processor P_1 , the series $(M_{400-900}, M_{4400-4900}, M_{8400-8900}, M_{12400-12900}, M_{16400-16900})$ to the processor P_2 and so on. The chronograph of events associated to this parallel processing is:



In this case, we can divide the sequential time $T(n)$ by the number of processors so that the parallel time is $\frac{T(n)}{10}$. We can then compute 50 TF-map alignments with 10 computers using the same amount of time that is necessary for calculating 5 alignments in a single processor machine. As the same comparisons must be done for hundreds of genes, the save of time using this parallel version is considerable.

The program `pgws` (Promoter Genome-Wide Search) is a generalization of the schema presented here, in which the input consists of a list of probes $P = p_1, p_2 \dots p_{|P|}$ (gene promoters from species A) and a list of long genomic sequences $M = m_1, m_2 \dots m_{|M|}$ (chromosomes from species B). In an efficient parallel environment, the program `pgws` may be used, for instance, to locate the ortholog promoter of a chicken gene in the human genome.

5.6 Using TF-map alignments to characterize promoter regions of co-regulated genes

We expect, therefore, the map alignments to be particularly useful to characterize promoter regions of co-regulated genes in absence of sequence conservation. In such cases, the map alignments can help to recover conserved configurations of TFBSs that primary sequence

	1-500		4000-4500		4500-5000		5000-5500	
	TOP1	TOP 10	TOP1	TOP 10	TOP1	TOP 10	TOP1	TOP 10
50T	4%	30%	14%	70%	33%	76%	21%	62 %
TRANSFAC	2%	37%	12%	55%	23%	61%	17%	48%
BLASTN	0 %	12%	4%	17 %	16%	31%	5%	22%
	5500-6000		10000-10500		20000-20500		50000-50500	
	TOP1	TOP 10	TOP1	TOP 10	TOP1	TOP 10	TOP1	TOP 10
50T	9%	57%	2%	37%	1%	15%	0%	0%
TRANSFAC	7%	48%	5%	36%	1%	17%	0%	1%
BLASTN	1%	20%	2%	16%	1%	6%	0 %	1%

Table 5.4 Promoter identification with human-chicken TF-map alignments. The percentages are relative to the proportion of the 278 human-chicken promoter pairs that score the highest in each window (or within the TOP 10). The correct promoter window is 4,500 – 5,000. The 50T collection are the 50 more informative matrices from TRANSFAC.

comparisons would not. It is important to stress in this regard, that the match state in the alignment of TF-maps is defined based on the transcription factor label, and not based on the label of the specific binding site. Since a given TF can be associated to different binding sites (for instance, the approximately 90 TFBSs in the HR SET correspond only to about 30 TFs), an alignment of TF-maps can include the alignment of TFBSs that show no sequence conservation.

Many examples could be found in which map alignments produce a better characterization of the promoter region of co-regulated genes than that obtained through primary sequence alignments. We would like, however, to move beyond such an anecdotal evidence, and have a more exhaustive evaluation of the power of TF-map alignments to characterize promoter regions of co-regulated genes in absence of sequence similarity. Towards such a goal we have used the set of co-regulated genes in the CISRED database (Robertson et al., 2006). The CISRED database is primarily a collection of conserved regulatory sequence elements identified by a genome-scale computational system that uses pattern discovery, similarity, clustering, co-occurrence and co-expression calculations. CISRED includes, as well, a database of high-confidence co-expressed gene pairs (Griffith et al., 2005), obtained from cDNA microarray hybridization, SAGE and other experiments, as well as Gene Ontology (GO, The Gene Ontology Consortium (2000)) analysis. Version 1 of CISRED high confidence co-expression human set contains 60,912 co-expression gene pairs for 5562 genes. Because of the criteria to establish co-regulation within CISRED, we do not expect strong bias towards co-expression pairs sharing strong sequence similarity in their promoter regions.

We have, thus, performed the following experiment (graphically represented in Figure 5.11): we have compared the promoter region of each gene x in the CISRED set with the promoter regions of the genes co-regulated with x , $\text{coreg}(x)$, and with the promoter region of the genes no co-regulated with x , $\overline{\text{coreg}}(x)$. Even though the promoter of the gene x may not show stronger sequence similarity with the promoters of the genes in $\text{coreg}(x)$ than with the promoters of the genes in $\overline{\text{coreg}}(x)$, our assumption is that it will still share some common regulatory signal (maybe very weak) with the promoters of the (at least a fraction of) the genes in $\text{coreg}(x)$, whereas no common signal will be shared between the promoter of x and the promoters of the genes in $\overline{\text{coreg}}(x)$. Our hypothesis is therefore that alignments

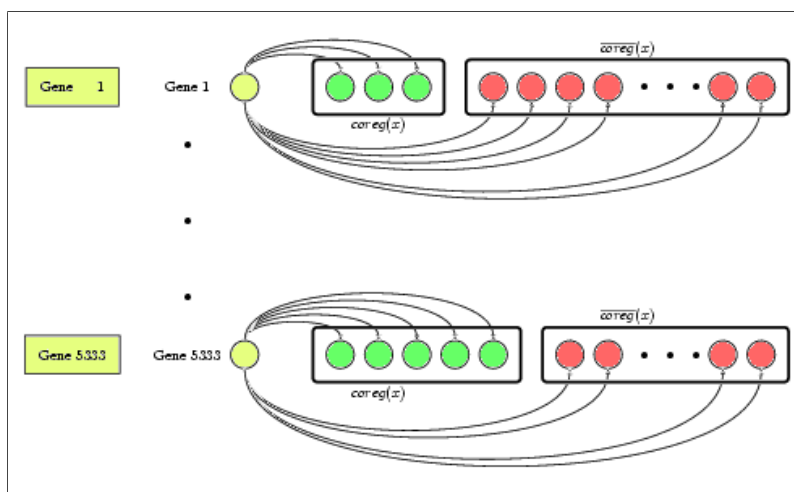


Figure 5.11 Alignment experiment with the CISRED genes.

of TF-maps will be superior in detecting such signals to alignments of the primary nucleotide sequence.

We have proceed in the following way: we have used ENSMART to extract 500 nucleotides upstream of each gene in CISRED according to genome coordinates in ENSEMBL. We have used 500 nucleotides upstream here, instead of 200 nucleotides as before, because of the intrinsic imprecision of ENSEMBL when annotating the coordinates of the TSS. We obtained such a sequence for 5333 out of 5562 CISRED genes and considered it the promoter region of the gene. For this set of 5333 genes, 56,632 co-expression gene pairs are described in CISRED. We have used next the collection of matrices in JASPAR_{TOP50} (see previous section) to obtain the TF-maps of each promoter region. Then for each gene x we have obtained the optimal map alignment with each gene in $\text{coreg}(x)$ and in $\overline{\text{coreg}}(x)$. We have used the enhanced TF-map alignment algorithm with the optimal parameters estimated in the training procedure. Finally, we have determined whether the scores of the map alignments between the promoter of gene x and the promoters of the genes in $\text{coreg}(x)$ were significantly higher than the scores of the map alignments between the promoter of gene x and the promoters of the genes in $\overline{\text{coreg}}(x)$. Because the scores of the optimal TF-maps alignments follow, as optimal sequence alignments, a Gumbel or extreme-value distribution (see Figure 5.12), we calculated the Wilcoxon test to assess this hypothesis. We obtained 42,756 non-void $\text{coreg}(x)$ alignments and 20,600,640 non-void $\overline{\text{coreg}}(x)$ alignments. 4,784 genes in CISRED had non-void alignments for both the $\text{coreg}(x)$ and the $\overline{\text{coreg}}(x)$ sets. The average score of the $\text{coreg}(x)$ alignments was 6.02, and the average length 2.13 sites. For the $\overline{\text{coreg}}(x)$ alignments, the values were 5.57 and 2.06, respectively. For 97 genes, the score of the $\text{coreg}(x)$ alignments was significantly higher than that of the $\overline{\text{coreg}}(x)$ alignments at a significance level of $p=0.01$. At a p -value of 0.001, the number was 23. Since CISRED is partially based on microarray experiments, one could argue that cross-hybridization with recently duplicated genes may artefactually bias these results. However, no duplicated copies of genes exist in the sets of co-regulated genes with the 97 positive cases above.

We performed the same experiment, using BLASTN (Altschul et al., 1990) instead to

compare the promoter region of each gene x in the CISRED set with the promoters of the genes in $\text{coreg}(x)$ and $\overline{\text{coreg}}(x)$. BLASTN was used with the parameters word size 7 and expectation value 10 so that short stretches of conservation could also be retrieved. In each comparison, we identified the score of the best HSP. We obtained 981 $\text{coreg}(x)$ alignments and 445,371 non-void $\overline{\text{coreg}}(x)$ alignments. 653 genes in CISRED had BLASTN alignments in both the $\text{coreg}(x)$ and the $\overline{\text{coreg}}(x)$ sets. The average score of the $\text{coreg}(x)$ alignments was 29.9, and the average length 51 nucleotides. For the $\overline{\text{coreg}}(x)$ alignments, the values were 24.3 and 40.5, respectively. For 11 genes, the score of the $\text{coreg}(x)$ alignments was significantly higher than that of the $\overline{\text{coreg}}(x)$ alignments at a significance level of $p=0.01$; there was only one gene for which the score of the $\text{coreg}(x)$ alignments was significantly higher than that of the $\overline{\text{coreg}}(x)$ alignments, at a significance level of $p=0.001$.

We have investigated whether differences in conservation of regulatory elements could be found between promoters associated to CpG islands (CpG+) and promoters not associated to them (CpG-). CpG- promoters have been linked to tissue-specific expression patterns (Smale and Kadonaga, 2003), and therefore they could be overrepresented in the set of co-expressed genes for which we have been able to identify conserved regulatory motifs. We computed for each gene the GC content and the CpG score as defined by Yamashita et al. (2005). The presence of a CpG island on a window (-100:+100) centered around the TSS of a gene is accepted when its GC content is greater than 0.5 and when its CpG score is greater than 0.6 (CpG+); otherwise they are classified as CpG negative genes (CpG-). Genes lacking CpG islands around their TSS have been shown to have a more tissue-specific expression pattern (Yamashita et al., 2005). Based on these considerations, 3844 out of the 5333 promoters (72%) were identified as CpG+ genes, while only 1489 (28%) were classified as CpG-. Among the 97 genes for which the score of the $\text{coreg}(x)$ TF-map alignments was significantly higher than that of the $\overline{\text{coreg}}(x)$ alignments at a significance level of $p=0.01$, 63 were CpG+ (65%). At a p-value of 0.001, the number of CpG+ genes was 13, out of a total of 23 (56%). It, thus, indeed appears that genes with CpG- promoters are slightly overrepresented in the set of co-regulated genes with conserved (specific) regulatory signals.

As it is possible to see, despite the general poor ability of both the sequence alignments and the TF-maps to uncover relationships between the promoters of the co-regulated genes in CISRED, it is clear that TF-map alignments are able to detect more relationships than BLASTN alignments (97 vs. 11 at a p-value < 0.01 , 23 vs. 1 at a p-value < 0.001). It can be argued that this is partially an artefact, resulting from BLASTN reporting only sequence alignments over a given threshold, while non void TF-map alignments are always produced, provided that the maps to align share at least one common element. In fact, given the number of genes for which valid alignments are obtained, at a p-value < 0.01 there are twice as many cases in which $\text{coreg}(x)$ scores are significantly higher than $\overline{\text{coreg}}(x)$ as expected if there was actually no difference in the distributions of scores, both using TF-map and sequence alignments. At a p-value < 0.001 , however, the number of cases in which $\text{coreg}(x)$ scores are significantly higher than $\overline{\text{coreg}}(x)$ coincides with the expected value using BLASTN, but it is five times the expected value, using TF-maps. We believe that this indicates that, even after taking into account the effect of the different number of total alignments reported, the TF-map alignment algorithm is superior to BLASTN in detecting relationships between the promoter regions of co-regulated genes. Indeed, among the 445,371 total BLASTN alignments obtained, there are 981 alignments between co-regulated genes, while the 445,371 top scoring TF-map alignments obtained include 1240 alignments between co-regulated genes. Interestingly, there are only 148 alignments in common between

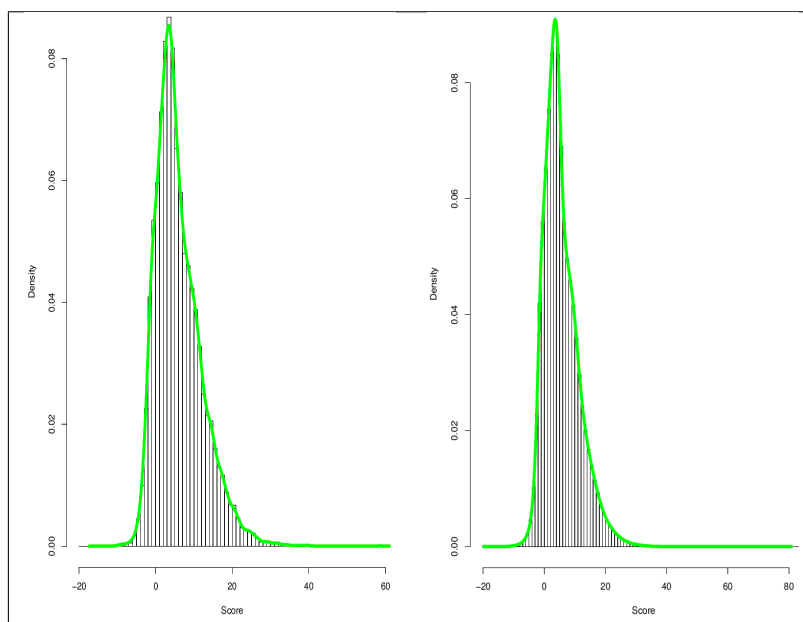


Figure 5.12 Score distribution of the CISRED TF-map alignments. (Left) Distribution of the $\text{coreg}(x)$ TF-map alignment scores. (Right) Distribution of the $\overline{\text{coreg}}(x)$ TF-map alignment scores.

both approaches, indicating that they could be used to complement each other.

It could be argued that the superiority of the TF-map over sequence alignments has little to do with the alignments and more to do with the maps. In other words, we would have obtained similar results if we were to simply score the proportion of TF labels common to the compared promoter regions –without the need for an alignment. Therefore, we have computed such a score for each pair of genes in CISRED: if p and q are the sets of elements in the TF-maps of the promoters to be compared, we have computed $|p \cap q|^2 / |p| \cdot |q|$, where $|p|$ is the size (cardinality) of the set p . Among the 445,371 top scoring comparisons, 1072 corresponded to co-regulated genes (with only 394 gene comparisons in common with the TF-map alignment approach), a value intermediate between that obtained with sequence and with TF-map alignments. This reflects that conservation of the relative position of the TFs along the primary sequence, and not only common presence, is indicative of gene co-regulation. Conservation of relative position can only be captured by TF-map alignments.

As an example, Table 5.5 summarizes the TF-map alignments obtained when aligning the promoter region of the *transthyretin* gene (TTR, ENSEMBL entry ENSG00000118271) with that of its co-regulated genes in CISRED. TTR is a serum carrier protein expressed in liver and brain. The regulatory regions that control the TTR expression in liver have been experimentally determined (Costa et al., 1989), and consist of a 100-nucleotide enhancer located at -2000 nucleotides upstream of the TSS and a proximal promoter region between -200 and -90 nucleotides upstream of the TSS (relative to the coordinates in the ENSEMBL entry). This proximal region is constituted of 6 binding sites (coordinates relative to TSS of the *transthyretin* gene as in the ENSEMBL database): HNF-1 (-137,-109), HNF-3 (-140,-

BEGIN	END	TF	FREQUENCY
-492	-477	HMG-IY	11
-486	-475	HNF-3beta	10
-406	-393	Broad-complex_1	9
-380	-367	Broad-complex_1	21
-364	-350	TBP	5
-362	-349	SQUA	9
-362	-347	HMG-IY	10
-312	-301	TEF-1	12
-307	-296	HFH-2	9
-273	-262	HNF-3beta	21
-271	-256	HMG-IY	6
-253	-238	HMG-IY	6
-251	-238	Broad-complex_1	9
-236	-225	HFH-3	9
-203	-194	RORalpha-1	18
-141	-130	HFH-3	17
-128	-115	HNF-1	6
-102	-91	HNF-3beta	22
-30	-16	TBP	21

Table 5.5 TF-map alignment reconstruction of the TTR gene promoter. Summary of the TF-map alignments obtained between the promoter of the *transthyretin* gene (TTR, ENSEMBL entry ENSG00000118271) and the promoters of the genes co-regulated with it according to the CISRED database. The table lists the predicted transcription factors on the promoter of *transthyretin*, which appear at least in five TF-map alignments with co-regulated genes. The experimentally verified sites are highlighted.

128 and -106,-91), HNF-4 (-151,-140), C/EBP binding (-195,-177 and -135,-112). The TATA box is located at -30. CISRED lists 105 genes co-regulated with TTR. Interestingly, while BLASTN is unable to detect any sequence similarity between the promoter of TTR and that of its co-regulated genes, TF-map alignments are obtained in 83 cases, and scored significantly (p -value < 0.001). We have reconstructed the structure of the TTR promoter from the elements that appear in the TF-map alignments. A total of 35 TFBSs were initially mapped with JASPAR_{TOP50} in the TTR promoter. For each predicted TF, Table 5.5 lists the number of TF-map alignments between TTR, and its co-regulated genes in which the TF appears. Only elements appearing in at least five alignments are reported. No matrices for the detection of C/EBP and HNF-4 were included in the JASPAR_{TOP50} collection that was used to perform the test. However, the meta-alignments were overrepresented in the other experimentally annotated sites, HNF-1, HNF-3 and TATA, exactly in the region where promoter activity has been reported (see Figure 5.13). The binding of HNF-3 to positions -140,-128 is not directly reported. The TF-map alignments, however, are highly enriched in the HFH-3 factor (HNF3/fork head homolog) at this region. In fact, both share a similar consensus binding sequence in TRANSFAC (Matys et al., 2003): TRTTTTRTTT for HFH-3 and TRTTTRYTT for HNF-3.

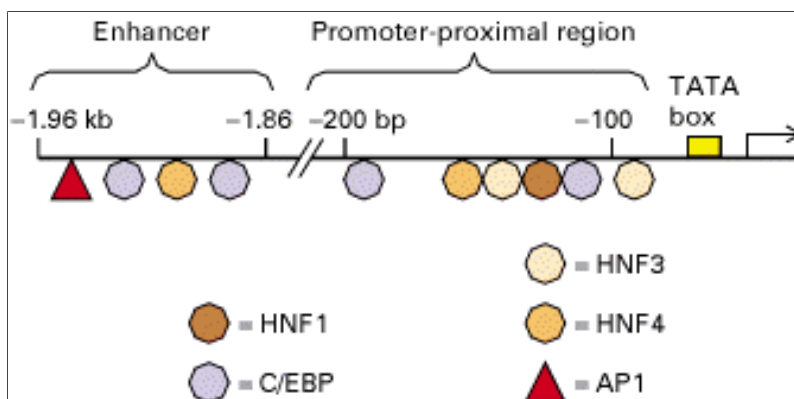


Figure 5.13 Experimental annotation of the TTR gene promoter. Binding sites for activators that control transcription of the mouse *transferrin* (TTR) promoter in hepatocytes are shown. Adapted from (Lodish et al., 2000).

Fast computing of all the CISRED TF-map alignments

For the results of this section, it was necessary to perform $5,333 \times 5,333 = 28,440,889$ pairwise TF-map alignments. These combinations can be represented into a similarity matrix that is addressed by the 5333×5333 CISRED gene promoter comparison indexes. As the similarity between two maps A and B is equal to the similarity between B and A , we only needed to compute $\frac{5333 \times 5333}{2}$ alignments (the other half of the matrix is symmetrical). The alignment between a gene and itself is also discarded. However, such a number of alignments is still too high to perform this test several times to evaluate different conditions in a reasonable amount of time.

Following the same strategy of the program `pgws` shown in the section before, we have divided the work load into different processors. Thus, we have assigned a part of the similarity matrix to each node taking. Let $G = (g_1, g_2 \dots g_{5333})$ be the CISRED collection of gene promoters. A possible planning of tasks based on dividing such a matrix by rows into several parts may be: the alignments between the genes $g_1 \dots g_{1000}$ and all of the genes for a first processor; the alignments between the genes $g_{1000} \dots g_{3000}$ and all of the genes for a second processor; the alignments between the genes $g_{3000} \dots g_{5333}$ and all of the genes for a third processor.

The number of assigned rows is different for each processor as the number of alignments that must be computed for a row is different depending on the part of the matrix is located. For a given row g_i in the matrix, only the alignments between such a gene and the genes $g_{i+1} \dots g_{5333}$ must be performed.

After this process, each alignment between two gene promoters g_i and g_j is classified into $\text{coreg}(g_i)$ or $\overline{\text{coreg}}(g_i)$ whether the pair (g_i, g_j) is co-regulated or not according to the CISRED collection.

5.7 TF-map alignments and matrix specificity

Throughout this chapter, we have used in many experiments smaller subsets of the full collections of matrices (e.g. JASPAR_{TOP50}). This fact was explained because of the poor specificity of many of these matrices in JASPAR or TRANSFAC. Several theoretical and practical studies have concluded there is a great amount of redundancy in these collections (Rahmann et al., 2003; Schones et al., 2005). In this section, we have numerically explored the specificity of current matrices, using the TF-map alignment to obtain similar conclusions.

Position Weight Matrices (PWMs, see Chapter 4 and Figure 5.14 for a review) have been traditionally used to characterize families of TFBSs. New sequences can be analyzed with this model in order to locate putative occurrences of the represented regulatory element. However, the ambiguous nature and the short length of the binding sites usually induce an overwhelming amount of false positive predictions in the searching process.

High conservation in certain positions of a PWM may be relevant for the activity of the site. Base frequencies may be proportional to the binding energy contribution of the bases. The information content of a PWM introduced in Chapter 4 can be used as a estimation of its specificity. However, this fact is not always true.

To determine the specificity of current weight matrix models in a genome-wide scale, we have used protein-coding sequences (CDS) as a negative control. No TFBSs are expected to be functional in the CDS regions. For the 21,538 genes in the UCSC hg17 human genome release, we have extracted 500 nucleotides upstream the TSS (PROMOTER samples) and 500 nucleotides downstream the Start Codon (CDS samples).

For each matrix x in JASPAR 1.0 and TRANSFAC 6.3, we obtained the number of predicted TFBSs in both sets of human samples (Threshold = 0.80): $f_{\text{PROM}}(x)$ and $f_{\text{CDS}}(x)$. Next, we define the function Q as the log-likelihood ratio between both numbers:

$$Q(x) = \log \frac{f_{\text{PROM}}(x)}{f_{\text{CDS}}(x)}. \quad (5.6)$$

In Figure 5.15, the distribution of the PWMs in JASPAR and TRANSFAC according to this measure is shown. Not surprisingly, 40% of the TRANSFAC matrices (37% in JASPAR) produced even more predictions in the CDS sequences than in the actual promoter regions (see Table 5.6). For different values of Q , more strict sets of matrices can be obtained, as shown in Table 5.6.

The test we performed on the HR SET (see Figure 5.9) showed that TF-map alignment could distinguish two orthologous promoters better than any other pair of orthologous genomic samples, even with lower sequence similarity (see Section 5.5 for further details). JASPAR_{TOP50} was used as a mapping function, because the 50 most informative matrices in JASPAR were supposed to be the more specific. In fact, we can now quantify the optimal number of matrices (and which matrices) to achieve the maximum discrimination power, using the Q -value function.

As we are going to align human-mouse pairs, we have also computed the Q -value using the mouse genome (17,213 genes, mm5) for the complete collection of matrices in JASPAR

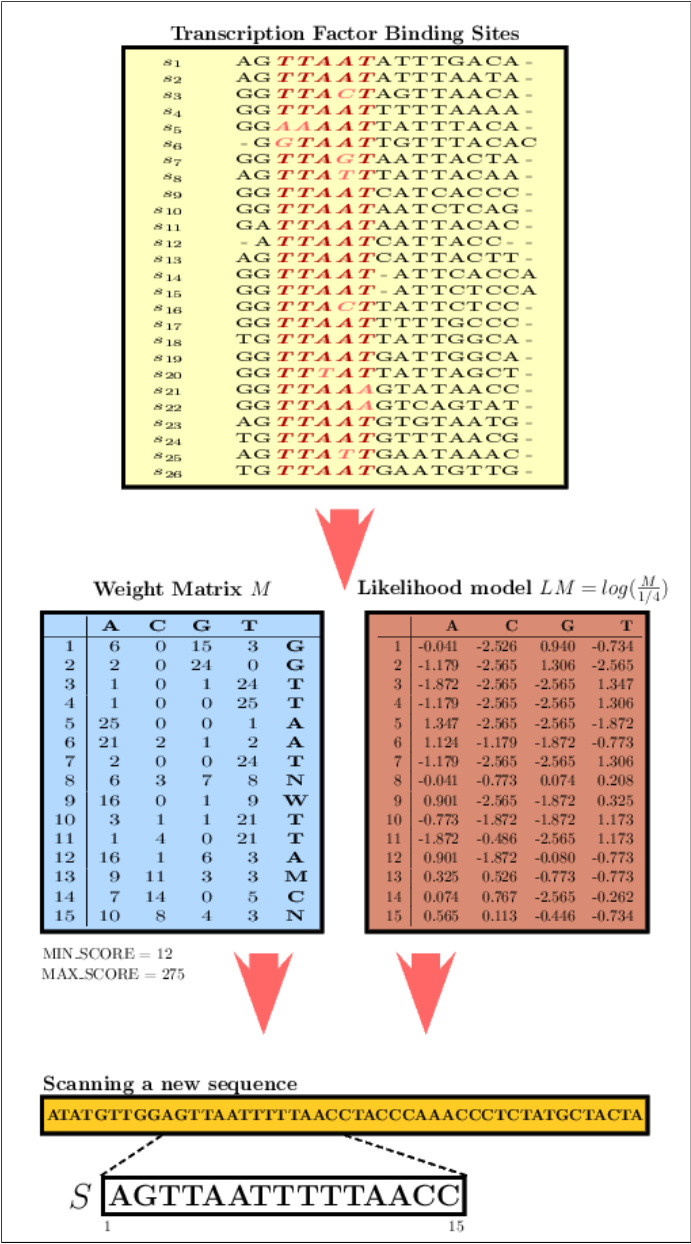


Figure 5.14 Construction and use of a PWM. (1) Collect a family of experimentally verified binding sites. (2) Align the sites to find conservations (anchored alignment). (3) Build a weight matrix representation of the alignment: Determine the optimal length; Define a Threshold value; Using a background model, construct the likelihood ratio matrix. (4) Search new occurrences of this signal in other sequences.

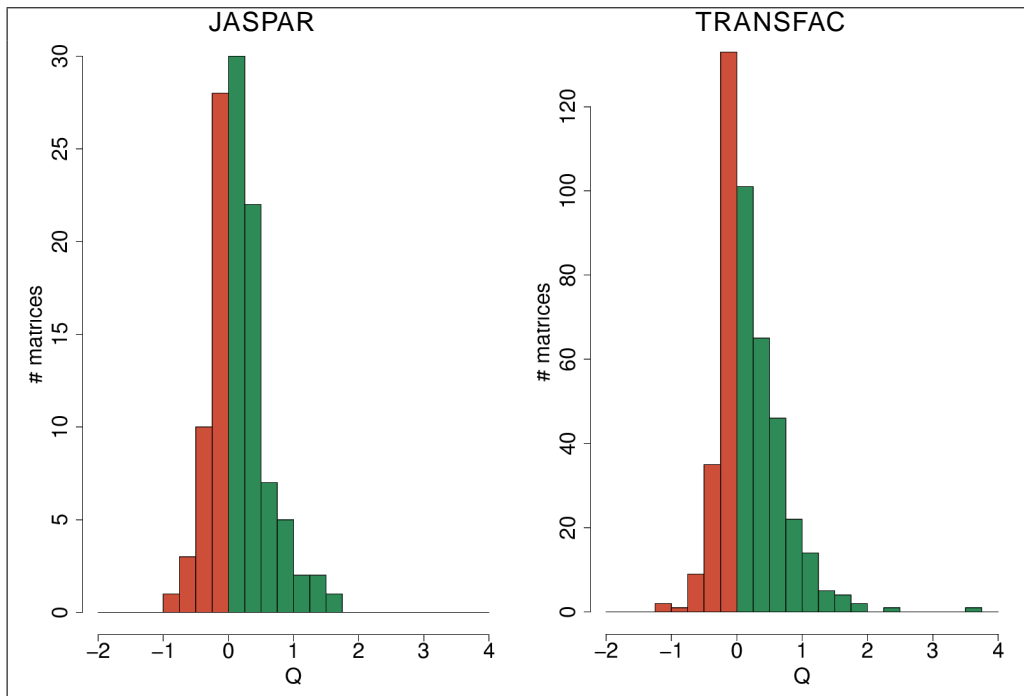


Figure 5.15 The Q-value distribution in JASPAR and TRANSFAC. In red, the matrices that produced more predictions in the CDSs; in green, the matrices that produced more predictions in the promoters.

and TRANSFAC, following the procedure explained above for the human genes. For each Q-value, we have intersected the subset of matrices according to the human and the mouse genomes. Then, we have repeated the test detailed in Section 5.5 using these different sets of matrices. The test with the full collections was also performed to compare against the smaller subsets.

Table 5.7 lists the number of times each genomic region (promoter, 5'UTR, CDS, intronic, intergenic, downstream) scores the highest in each gene of the HR SET using each subcollection of matrices. It is remarkable that the $Q \geq 0.5$ in JASPAR, with only 16 matrices, identified correctly 20 of the promoter pairs. Notice the poor performance when we used the full JASPAR collection. In fact, the results do not improve when we add or remove other matrices to the optimal subset of matrices. Similar results are obtained when we used TRANSFAC. The optimal collections are listed in Table 5.8. In both cases, the majority of the matrices are the most informative. Despite this, some significant matrices with a small information content are also included in both optimal sets (e.g. the SP1 matrix in JASPAR and TRANSFAC). As in the previous test, we performed the global alignment to show the sequence similarity of each sample pair with the program *needle* of the EMBOSS software (Olson, 2002).

Finally, it is important to mention that the subset of matrices that we arbitrarily selected in the original test (JASPAR_{TOP50}, see Section 5.5) obtained slightly better results than

COLLECTION	Size	Q < 0	Q ≥ 0	Q ≥ 0.5	Q ≥ 1
JASPAR 1.0	111	42 (37%)	69 (63%)	17 (15%)	5 (4%)
TRANSFAC 6.4	441	177 (40%)	264 (60%)	95 (21%)	27 (6%)

Table 5.6 Q-value and PWM matrix specificity in JASPAR and TRANSFAC.

the optimal set estimated with the Q-value method. This subset, however, only have 16 matrices, while JASPAR_{TOP50} is constituted of 50 matrices.

Several conclusions can be extracted, therefore, from this simple test:

1. Up to 40% of the matrices from popular matrices repositories are prone to predict the same number of TFBSs either in human promoters or in protein coding sequences. Therefore, analysis with these models must be very carefully evaluated.
2. Although a high information content normally implies better specificity of the matrices, there are cases in which both characteristics are not related.
3. The use of complete collections to analyze homologous promoters usually produces the recognition of artefactual sequence conservations as shown when the matrices are applied on protein coding regions or intron sequences.
4. To locate the actual common regulatory elements in a set of co-expressed sequences, it is advisable to restrict the search using smaller collections of matrices. A simple procedure to detect those matrices that consistently appear more frequently in a set of co-regulated genes than in a negative control set can provide interesting results.
5. Many of the numerous drawbacks of the weight matrices such as redundancy and low specificity are caused by the simplicity of the model. Therefore, the use of more complex models to incorporate additional information will obviously improve future predictions. However, we also suggest a more rational application of the current systems to enhance the advantages and to mask the inconveniences of these representations.



5.8 Local TF-map alignments

Local alignments are very useful to identify short stretches of a sequence that are conserved in another one, despite the rest of both sequences is probably different. Local comparisons are also interesting mechanisms to locate the location (if any) of a short composite (cluster of TFBSs, a super-pattern of TFBSs) in a long TF-map (see Figure 5.16).

Two alternative designs were presented in Chapter 3 (Section 3.4) to implement a sequence local alignment according to the scoring function: similarity or distance. Based on them, we present here two different implementations to identify local meta-alignments between two TF-maps.

JASPAR	Q≥1	Q≥0.75	Q≥0.5	Q≥0.25	Q≥0	FULL	needle
# MATRICES	4	9	16	32	63	111	-
PROMOTER	9	12	20	18	11	4	2
5'UTR	0	1	2	4	8	6	5
CDS	2	6	6	6	17	28	32
INTRON	6	9	5	6	1	0	0
DOWNSTREAM	8	6	3	4	2	1	1
INTERGENIC	8	6	4	2	1	1	0
NOALIGN	7	0	0	0	0	0	0

TRANSFAC	Q≥1.5	Q≥1.25	Q≥1	Q≥0.75	Q≥0	FULL	needle
# MATRICES	6	10	23	46	246	442	-
PROMOTER	18	19	24	21	2	1	2
5'UTR	1	1	1	2	4	2	5
CDS	5	6	6	11	32	35	32
INTRON	7	7	6	4	0	1	0
DOWNSTREAM	3	3	1	1	1	1	1
INTERGENIC	1	0	2	1	1	0	0
NOALIGN	5	4	0	0	0	0	0

Table 5.7 Matrix specificity in several subsets of JASPAR and TRANSFAC.

Local TF-map alignments using similarity

In a short communication, [Smith and Waterman \(1981\)](#) published a slight modification of the [Needleman and Wunsch](#) algorithm, as revisited by [Smith et al. \(1981\)](#), to deal with local alignments. The main objective is to find the pair of segments, one from each of two long sequences, such that there is no other pair of segments with greater similarity (homology).

The basic rationale of this strategy is the following: let $S(i, j)$ a position in the dynamic programming matrix. The best local alignment ending at $S(i, j)$ is computed according to the three adjacent values in the matrix S as long as the incorporation of one of these elements does not produce an alignment with negative homology. In that case, the score of the alignment ending at $S(i, j)$ is set to 0. The traceback procedure then starts from the matrix cell having the maximum similarity, constructing the best local alignment until a cell that contains a 0 is reached.

The application of this approach to the meta-alignment is trivial. We can rewrite the Equation 5.3 introducing the 0 in the appropriate place to produce the local alignment. Thus, the maximum local similarity S_{ij} between TF-maps $A = a_1 \dots a_i$ and $B = b_1 \dots b_j$ where the site a_i^f is equal to the site b_j^f , can be computed as:

$$\begin{aligned}
 S_{ij} \equiv S(a_i, b_j) = & \max\{0, \alpha(a_i^s + b_j^s) + \\
 & \max_{\substack{0 < i' < i \\ 0 < j' < j \\ a_i^{p_2} < a_i^{p_1} \\ b_j^{p_2} < b_j^{p_1}}} \{S_{i'j'} - \lambda(i - i' - 1 + j - j' - 1) \\
 & - \mu(|(a_i^{p_1} - a_i^{p_1}) - (b_j^{p_1} - b_j^{p_1})|)\}\}.
 \end{aligned}
 \tag{5.7}$$

JASPAR (Q ≥ 0.5)					TRANSFAC (Q ≥ 1)				
RANK	16 MATRICES	Q _H	Q _M	Bits	RANK	23 MATRICES	Q _H	Q _M	Bits
3	RREB-1	1.58	1.97	27.72	1	V\$HOGNESS B	3.52	3.05	49.11
5	Pax-4	1.32	1.72	26.04	18	V\$CAAT C	1.27	1.20	30.32
17	HNF-1	0.78	0.90	19.37	20	V\$TANTIGEN B	2.25	1.94	29.70
20	NFY	1.02	0.92	18.78	26	V\$STAF 01	1.51	1.45	27.21
27	Broad_complex_1	0.87	1.08	17.34	31	V\$MEF2_03	1.32	1.49	26.35
28	SQUA	0.54	0.68	17.18	36	V\$PAX4_04	1.85	1.76	25.85
31	MEF2	1.11	1.19	17.03	47	V\$MEF2_02	1.40	1.21	25.25
32	HMG-IY	0.86	1.06	16.99	49	V\$STAF_02	1.11	1.00	24.88
36	HFH-3	0.59	0.86	16.50	61	V\$RSRFC4_Q2	1.14	1.32	23.46
38	HFH-2	1.34	1.54	16.34	71	V\$RSRFC4_01	1.18	1.31	22.67
40	TBP	0.56	0.80	16.27	88	V\$RREB1_01	1.23	1.39	21.30
54	Broad_complex_4	0.65	0.92	14.79	89	V\$OCT1_04	1.12	1.68	21.29
56	CF2-II	0.61	1.00	13.75	102	V\$FOXJ2_01	1.11	1.43	20.65
60	Hunchback	0.75	0.90	13.35	107	V\$HFH4_01	1.33	1.90	20.42
68	SP1	0.81	0.70	12.87	140	V\$EGR1_01	1.13	1.10	19.47
71	MZF_5-13	0.63	0.51	12.65	144	V\$NGFIC_01	1.50	1.41	19.43
					150	V\$HNF1_01	1.00	1.06	19.32
					156	V\$EGR2_01	1.10	1.09	19.21
					173	V\$NFY_01	1.14	1.07	18.67
					195	V\$MAZR_01	1.60	1.76	17.84
					231	V\$GC_01	1.63	1.64	16.59
					253	V\$SP1_Q6	1.77	1.89	16.06
					345	V\$MAZ_Q6	1.18	1.07	13.43

Table 5.8 JASPAR and TRANSFAC specific subsets. In red, the matrices that are not among the most informative ones.

If we save the N positions in S that have the best score, we can report the best N local alignments or blocks between A and B. The cost of the algorithm is the same as in the global TF-map alignment algorithm, as no additional operations are necessary.

Local TF-map alignments using distance

Despite the solution to the problem of local meta-alignment using similarity is simple and clear, we also decided to investigate the form to produce local alignments under the original distance scheme framework (Waterman et al., 1984). We have taken advantage of this research to study in depth the distribution of the scores (distance) in the meta-alignments.

As reviewed in Chapter 3 (Section 3.4) the solution developed by Smith and Waterman (1981) to produce local alignments using a similarity scoring function can not be directly applied in the case of the distance metric. Goad and Kanehisa (1982) defined the mismatch density of the alignment between two segments as the ratio of the minimum distance D between both sequences and the length L of such an alignment. Thus, only those alignments with a mismatch density below a certain positive threshold T should be reported.

Formally, we are interested in those paths in the dynamic programming distance matrix such that the mismatch density on them is minimal. The length of these alignments is a priori unknown and can be variable. The value of the threshold T is different for each input, having a statistical and biological meaning at the same time.

This is the procedure we follow to obtain the local meta-alignment between two maps A and B (see Figure 5.17):

- ① Compute the global alignment of both maps (distance metrics), to fill the dynamic

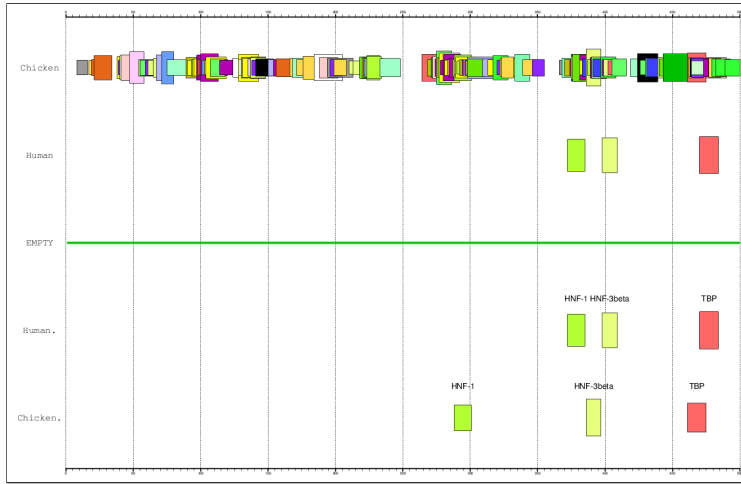


Figure 5.16 Using local meta-alignment to identify known patterns in orthologous sequences. (Top) TF-map obtained with JASPAR_{TOP50} on the chicken promoter of the TTR gene, and a second map of three experimentally verified TFBSs in the human ortholog. (Bottom) The local alignment between both maps identifies the putative location of the human sites in chicken.

programming matrix D in. Each position $D(i, j)$ contains the minimum distance in terms of a meta-alignment between the map $A = a_1 \dots a_i$ and the map $B = b_1 \dots b_j$.

- ② Compute the matrix ΔD from D . For each two consecutive nodes in the matrix $D(i, j)$ and $D(i', j')$ that are part of a path, we compute the increase of the distance value produced by adding the second match after the first one:

$$\Delta D(i, j) = D(i, j) - D(i', j') \text{ where } i' < i, j' < j. \quad (5.8)$$

- ③ Define the threshold T according to the ΔD values in the alignments of length $L = 2$ TFBSs. We define this threshold taking into account that the distribution of the distance in such alignments follows the Gumbel or extreme-value distribution (see Figure 5.18). The Gumbel function is defined as:

$$y = e^{-x - e^{-x}} \text{ where } P(x < 0) = 0.368, P(x > 0) = 0.632. \quad (5.9)$$

We are interested in defining T such that a small fraction of the smallest values is selected. The normalization of a Gumbel function is computed as:

$$z = \lambda(x - \mu) \text{ where } \lambda = \frac{1.285}{\sigma}, \mu = \bar{x} - 0.45\sigma. \quad (5.10)$$

\bar{x} and σ are the mean and the deviation of the distance values computed for the current set of paths, respectively. If we are considering the values $P(z \leq Z) = 0.05$, that is under 5% of the area covered by z , then:

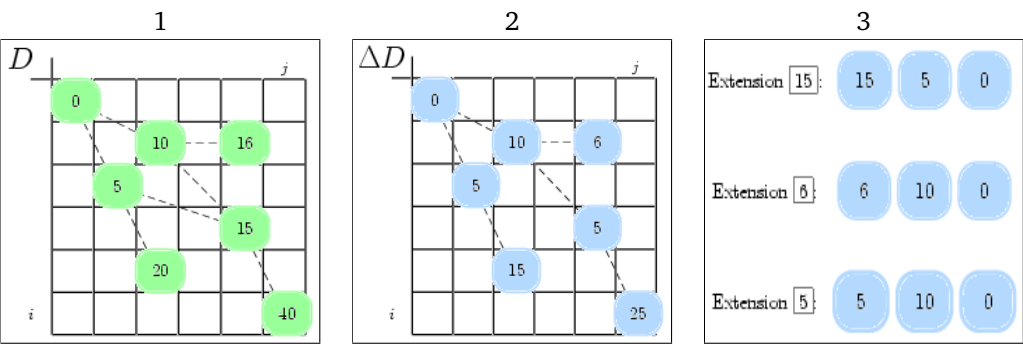


Figure 5.17 Local meta-alignment using the distance metric (1) Global alignment of both maps. (2) Compute the ΔD matrix for $L = 2$. (3) Extend the best local paths with the score below T .

$$z = \frac{1.285 \cdot x - 1.285 \cdot \bar{x} - 1.285 \cdot 0.45 \sigma}{\sigma}$$
$$G(P) = -\ln(\ln(\frac{1}{p})) \text{ where } p = 0.05, (\text{value of } z).$$

(5.11)

For each alignment input, we will have a different \bar{x} and σ values that, according to this equation, will provide a threshold T to obtain only the 5% of the minimal distance alignments of length 2.

- ④ Finally, trace back the paths ending at each match in the ΔD matrix. The rule to extend a local alignment takes into account a weighted version of the mismatch density value. A new match is added to the path if the accumulated distance is below T :

$$\frac{\Delta D(i, j)}{l} < T$$

(5.12)

where l is the length of the current local alignment path. Visited nodes are marked up to be skipped in future path extensions (avoid overlapping of the solutions).

5.9 Discussion

Much of the biology of the past decades has been based on the technological advances that have accelerated our ability to sequence DNA and proteins. It is certainly in the sequence of the genome where the biological traits of organisms are encoded. While we have a relatively good understanding of some of the basic mechanisms involved in the processing of the information encoded in the DNA sequence, it is in general very difficult to predict the biological traits –even at the molecular level– from the nucleotide sequence alone. Gene promoters are a case in point: while the sequence of the promoter is likely to contain most

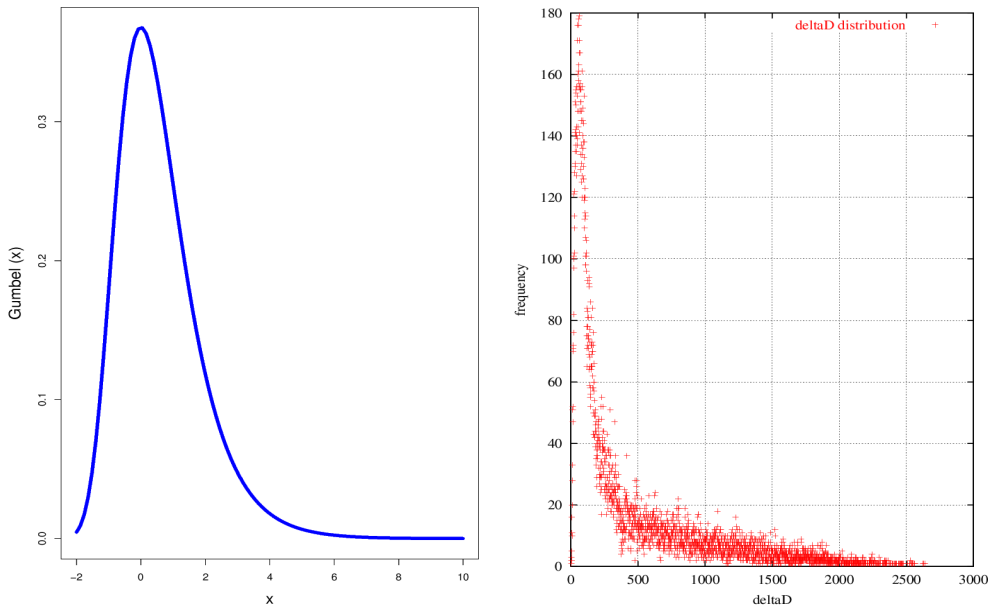


Figure 5.18 Gumbel distribution of local meta-alignments. (Left) The Gumbel generic function. (Right) TF-map alignment scores in a real pair of promoters.

of the information to control the expression of a gene, it is currently impossible to predict the expression pattern of a gene from the analysis of its promoter sequence alone.

While inferring function directly from sequence is thus far from trivial, it is still true, that because sequence encodes function, similar sequences often encode similar functions. Sequence comparisons, therefore, are an extraordinary tool to infer functional relationships: through sequence comparisons the function of known sequences can be extrapolated to newly obtained ones, and the specific sequence motifs can be identified responsible for the common functionality of a set of sequences. But sequence comparisons have limitations: often similar functions are encoded by diverse sequences. Again, gene promoters are a case in point: many TFs bind to sequence motifs which do not show sequence conservation. Thus, while through phylogenetic footprinting, conserved regulatory motifs have been in occasions uncovered in the promoters of orthologous genes (Blanchette and Tompa, 2002; Lenhard et al., 2003), searching for common patterns through the comparison of promoter sequences in sets of co-regulated genes –as, for instance, those resulting from microarray experiments– is usually a frustrating exercise.

Here, we have attempted to address this limitation implicit in sequence comparisons, by annotating the primary sequence with predicted functional domains, comparing the resulting annotations instead of the underlying primary sequence. If functional domains are encoded by diverse sequences, the comparison and alignment of the annotation may be more revealing of the functional relationships between sequences and of the specific domains involved in the common functionality than the comparison and alignment of the primary sequence. In particular, we have attempted this strategy for the comparison and characterization of promoter regions from genes with similar expression patterns. We have annotated

the sequence with predictions of TFBSs –using a variety of popular tools and databases– and identified the predicted sites with the labels of the corresponding TFs. We have then compared and aligned the resulting sequence of labels. Because TFs can bind to sites that show no sequence conservation, their labels can be aligned which correspond to domains that, while exhibiting similar functions, may not show sequence conservation.

Precedents of this approach can be found in the literature. (Quandt et al., 1996), for instance, distinguish explicitly between first-level analysis of promoters, in which the nucleotide sequence is directly interrogated for the presence of regulatory motifs, and second-level methods, in which basic higher order patterns can be defined from a number of correlated first-level units. This approach is further developed in (Frech et al., 1997) and (Klingenhoff et al., 1999), where more complex composite patterns are derived capturing the functional organization of individual regulatory elements, and are then used to identify and characterize related promoter regions in absence of sequence conservation. Here, we go one step further, and infer automatically the composite patterns by explicitly aligning the sequences of labels corresponding to TFs for which binding sites have been predicted in the compared promoters (the second-level annotation).

To align these sequences of labels—to which we refer as TF-maps– we have stated the problem as a restriction enzyme map alignment, and adapted a dynamic programming algorithm developed by Waterman et al. (1984). This algorithm, as well as ours, belong to a larger class of map alignments algorithms (see also, (Miller et al., 1990, 1991; Myers and Huang, 1992; Huang and Waterman, 1992)). In typical alignments, the sequences are of labels denoting either nucleotides or amino acids. In map alignments, the sequences are of pairs (label, integer), where the label denotes a predicted domain or site (possibly exhibiting some behavior or functionality), and the integer the position on the primary sequence where the domain or the site has been predicted. In global pairwise sequence alignments, the goal is to obtain the alignment that maximizes the sum of the scores of the aligned positions – given the score of the individual alignments of all possible pairs of labels. In contrast, in map alignments, only positions with identical labels can be aligned and the goal is to obtain the largest common subsequence constrained to minimize the differences in distances on the primary sequence between consecutive aligned positions. Sequence and map alignments can be generalized to a broader class of alignments that includes both.

Map alignments have been mostly used to align restriction enzyme maps. In this case, the label denotes a restriction enzyme, and the integer the position on the primary sequence of the site recognized by the enzyme. Waterman et al. (1984) first established the concept of map alignment and provided an algorithm for computing the optimal alignment of two maps. Later Myers and Huang (1992) described an improved algorithm to efficiently find map alignments which relies on the extreme sparsity of the dynamic programming matrix in (Waterman et al., 1984) –the result of the match state being defined only between identical labels. Miller et al. (1990, 1991) introduced new algorithms that permitted the efficient search of a long map for the best matches to a shorter probe map. Huang and Waterman (1992) generalized these algorithms to deal with different map errors.

In our case, the label denotes a TF, and the integer the initial position on the primary sequence where a binding motif for the TF has been predicted. There are, however, two important differences between restriction enzyme maps and TF-maps. First, while prediction of restriction sites is deterministic, producing a binary output (“site”, “no site”), prediction of TFBSs is often probabilistic and predicted sites may have an associated score. The score can usually be related to the strength of the binding of the TF to the site (Stormo, 2000). Since, it

makes sense, therefore, to prefer in TF-map alignments higher scoring sites, the score of the TFBSs needs to be taking into account when building optimal TF-map alignments. Second, enzyme restriction sites are single-nucleotide positions on the primary sequence. TFBSs, in contrast, are sequence intervals, and have thus, in addition to position, an associated length. Because we explicitly prohibit overlap between aligned elements, we can not directly extrapolate the algorithm of Myers and Huang (1992). However, as in their approach, we have also taken advantage of the extreme sparsity of the dynamic programming matrix to implement an efficient algorithm that, in our experience, is comparable in efficiency. There is another important feature characteristic of our approach that, while it does not influence the algorithmic strategy, it is essential to its success. As we have already stressed, we do not label the site, but the function of the site. That is, we do not label the TFBSs, but the TFs that bind to the sites. This allows for significant functional alignments even in the absence of sequence conservation.

We have estimated the optimal parameters of the algorithm in a small, but well annotated, set of orthologous human-mouse genes. We used three popular collections of PWMs for TFBSs (JASPAR 1.0 (Sandelin et al., 2004), PROMO 2.0 (Farre et al., 2003) and TRANSFAC 6.3 (Matys et al., 2003)) to obtain the TF-maps of the promoter sequences. Results on this data set indicate that, by dramatically reducing the overwhelming number of spurious predictions of TFBSs produced using these collections, TF-map alignments are able to successfully uncover the few conserved functionally active regulatory domains. Differences can be observed between the performance of the different collections of TFBSs; alignments obtained using JASPAR –and, in particular, using a subset consisting of the 50 top most informative matrices– appear to show the optimal balance between sensitivity and specificity. The data set that we have used, however, is too small to infer general trends on the comparative behavior of these collections.

Interestingly, despite the stronger sequence conservation between protein-coding regions, TF-map alignments score the highest between promoter regions in the training set of orthologous human-mouse genes. This indicates that TF-map alignments are able to pick up regulatory signals that sequence alignments can not. Results in an independent larger data set of co-regulated genes from the CISRED database are also in support of this conclusion: we have been able to obtain more significant alignments between the TF-maps than between the nucleotide sequences of the promoters of co-regulated genes. Results in CISRED are certainly not extraordinary. Both sequence and TF-map alignments perform very poorly when detecting relationships between co-regulated genes in CISRED. Only in 97 out of 5333 gene representatives in CISRED (1.8%), TF-map alignments scored significantly higher for co-regulated than for non co-regulated genes. Using BLASTN, this number was only 11 (0.2%). Finding relationships between the promoters of the genes co-regulated in CISRED is a task as challenging as one can imagine. The CISRED collection of high-confidence co-expressed genes is not derived from overall conservation, or from co-occurrence of motifs, in the sequence of the gene promoters. CISRED co-expression is derived instead from cDNA microarray, SAGE and other high-throughput gene expression monitoring techniques. CISRED co-expression clusters are thus a mixture of directly and indirectly co-regulated genes and one would then expect only a few genes within each cluster –maybe in a few subsets– to share functionally equivalent motifs in their promoter sequences. The poor performance of TF-map alignments, however, could also be reflecting the incompleteness of the current collections of TFBSs, and how little we know of the molecular rules governing the expression of human genes.

On the other hand, while building global pairwise alignments maybe appropriate to compare promoter sequences of orthologous human-mouse genes, to compare sequences from multiple genes weakly co-regulated –such as those in CISRED– multiple and/or local alignments may be more effective in capturing the functional motifs underlying co-expression. Indeed, from a multiple TF-map alignment of promoters of a set of co-regulated genes, a “transcriptional regulatory super-pattern” can be derived capturing those elements conferring expression specificity. Using a local alignment search algorithm, the super-pattern can then be used to identify additional genes or transcripts belonging to the same expression class (see other approaches in (Knight and Myers, 1995)).

Even more appropriate to the analysis of sets of weakly co-expressed genes (that is, including genes both directly and indirectly co-regulated), such as those in the CISRED clusters, would be the extension of the unsupervised pattern recognition techniques usually applied to motif discovery in DNA sequences (in programs such as MEME (Bailey and Elkan, 1994), AlignAce (Roth et al., 1998) and others (see (Tompa et al., 2005), for a recent comparative evaluation) to motif discovery in TF-maps. This would allow for the identification within a co-expression cluster of different “transcriptional regulatory super-patterns”. These super-patterns, in turn, and the subclusters they induce, could contribute to sort out direct vs. indirect co-regulation effects within the cluster. These and other extensions to the TF-map alignments (for instance, those allowing to deal with non-colinear arrangements of TFBSs that have been indeed observed in orthologous genes, see next chapter) are all feasible, and will certainly contribute to the discriminatory power of TF-map comparisons and alignments.

In summary, our results suggest that comparisons of annotations of higher order domains can, in occasions, be more meaningful to characterize the underlying functionality of sequences, than direct comparisons at the very primary sequence level. Here we have explored these strategies for the characterization of the promoter regions of co-regulated genes, and we have annotated the primary sequence of them with predictions of TFs. Moreover, we have also used the discriminative power of TF-maps for a better identification of orthologous promoter regions along large genomic sequences (e.g. chromosomes). In addition, we measured the specificity of PWMs in protein coding sequences and promoters.

However, we can imagine similar strategies to address many other problems in sequence analysis. One can imagine, for instance, annotating protein sequences with PFAM domains (Bateman et al., 2004), and compare the resulting annotations to detect distant functional relationships between proteins and protein families. Or annotating genome sequences with the Gene Ontology (GO, (The Gene Ontology Consortium, 2000)) labels of the genes encoded in these sequences, and aligning the GO labels to detect clusters of conserved functions across genomes. In fact, the annotation of the primary sequence with higher order domains to improve alignments has been often explored. For instance, to compare protein secondary structures, or to anchor whole genome alignments (Batzoglou et al., 2000), or even alignments of promoter regions (Berezikov et al., 2004). In all these cases, however, the ultimate goal is to obtain an optimal sequence alignment either between the original primary sequences, or between the 1-1 mappings of the primary sequence into a reduced alphabet (for instance, denoting secondary structure elements). We believe that, as the molecular functionality of the primary sequence becomes better understood, comparisons between higher order annotations, such as those performed here, in which the primary sequence is completely abstracted, may become increasingly relevant.

Bibliography

- J. F. Abril, R. Castelo, and R. Guigo. Comparison of splice sites in mammals and chicken. *Genome Research*, 15:111–119, 2005.
- J. F. Abril and R. Guigo. gff2ps: visualizing genomic annotations. *Bioinformatics*, 8:743–744, 2000.
- S.F. Altschul, W. Gish, W. Miller, E.W. Myers, and D.J. Lipman. Basic local alignment search tool. *Journal of Molecular Biology*, 215:403–10, 1990.
- T. L. Bailey and C. Elkan. Fitting a mixture model by expectation maximization to discover motifs in biopolymers. *Proceedings of the 2nd International Conference on Intelligent Systems for Molecular Biology (ISMB)*, pages 28–36, 1994.
- A. Bateman, L. Coin, R. Durbin, R.D. Finn, V. Hollich, S. Griffiths-Jones, A. Khanna, M. Marshall, S. Moxon, E.L.L. Sonnhammer, D.J. Studholme, C. Yeats, and S.R. Eddy. The Pfam protein families database. *Nucleic Acids Research*, pages D138–D141, 2004.
- S. Batzoglou, L. Pachter, J.P. Mesirov, B. Berger, and E.S. Lander. Human and mouse gene structure: comparative analysis and application to exon prediction. *Genome Research*, 10:950–958, 2000.
- E. Berezikov, V. Guryev, R. H. A. Plasterk, and E. Cuppen. Conreal: conserved regulatory elements anchored alignment algorithm for identification of transcription factor binding sites by phylogenetic footprinting. *Genome Research*, 14:170–178, 2004.
- M. Blanchette and M. Tompa. Discovery of regulatory elements by a computational method for phylogenetic footprinting. *Genome Research*, 12:739–748, 2002.
- E. Blanco, D. Farre, M. Alba, X. Messeguer, and R. Guigó. ABS: a database of annotated regulatory binding sites from orthologous promoters. *Nucleic Acids Research*, 34:D63–D67, 2006a.
- E. Blanco, X. Messeguer, T.F. Smith, and R. Guigó. Transcription factor map alignments of promoter regions. *PLoS Computational Biology*, 2:e49, 2006b.
- R. H. Costa, D. R. Grayson, and J. E. Darnell. Multiple hepatocyte-enriched nuclear factors function in the regulation of transthyretin and $\alpha 1$ -antitrypsin genes. *Molecular and Cellular Biology*, 9:1415–1425, 1989.
- E. T. Dermitzakis and A. G. Clark. Evolution of transcription factor binding sites in mammalian gene regulatory regions: conservation and turnover. *Molecular Biology and Evolution*, 7:1114–1121, 2002.
- D. Farre, R. Roset, M. Huerta, J. E. Adsuara, L.L. Rosello, M. Alba, and X. Messeguer. Identification of patterns in biological sequences at the ALGGEN server: PROMO and MALGEN. *Nucleic Acids Research*, 31:3651–3653, 2003.
- K. Frech, J. Danescu-Mayer, and T. Werner. A novel method to develop highly specific models for regulatory units detects a new LTR in genbank which contains a functional promoter. *Journal of Molecular Biology*, 270:674–687, 1997.
- W.B. Goad and M.I. Kanehisa. Pattern recognition in nucleic acid sequences i. a general method for finding local homologies and symmetries. *Nucleic Acids Research*, 10:247–278, 1982.
- O. L. Griffith, E. D. Pleasance, D. L. Fulton, M. Oveisi, M. Ester, A. Sidiqui, and S. J. M. Jones. Assessment and integration of publicly available SAGE, cDNA microarray, and oligonucleotide microarray expression data for global coexpression analyses. *Genomics*, 86:476–488, 2005.

- L.W. Hillier, W. Miller, E. Birney, W. Warren, R.C. Hardison, C.P. Ponting, P. Bork, D.W. Burt, M.A. Groenen, M.E. Delany, J.B. Dodgson, G. Fingerprint Map Sequence, Assembly, A.T. Chinwalla, P.F. Cliften, S.W. Clifton, and others (International Chicken Genome Sequencing Consortium, ICGSC). Sequence and comparative analysis of the chicken genome provide unique perspectives on vertebrate evolution. *Nature*, 432:695–716, 2004.
- X. Huang and M. S. Waterman. Dynamic programming algorithms for restriction map comparison. *Bioinformatics*, 8:511–520, 1992.
- A. Klingenhoff, K. Frech, K. Quandt, and T. Werner. Functional promoter modules can be detected by formal models independent of overall nucleotide sequence similarity. *Bioinformatics*, 15:180–186, 1999.
- J. R. Knight and E. W. Myers. Super-pattern matching. *Algorithmica*, 13:211–243, 1995.
- W. Krivan and W. W. Wasserman. A predictive model for regulatory sequences detecting liver-specific transcription. *Genome Research*, 11:1559–1566, 2001.
- B. Lenhard, A. Sandelin, L. Mendoza, P. Engstrom, N. Jareborg, and W. W. Wasserman. Identification of conserved regulatory elements by comparative genome analysis. *Journal of Biology*, 2:13, 2003.
- H. Lodish, A. Berk, L. Zipursky, P. Matsudaira, D. Baltimore, and J. Darnell. *Molecular Cell Biology*. W.H. Freeman, fourth edition, 2000. ISBN 0-7167-3706-X.
- V. Matys et al. TRANSFAC: transcriptional regulation, from patterns to profiles. *Nucleic Acids Research*, 31:374–378, 2003.
- W. Miller, J. Barr, and K.E. Rudd. Improved algorithms for searching restriction maps. *CABIOS*, 7: 447–456, 1991.
- W. Miller, J. Ostell, and K.E. Rudd. An algorithm for searching restriction maps. *CABIOS*, 3:247–252, 1990.
- E.W. Myers and X. Huang. An $o(n^2 \log n)$ restriction map comparison and search algorithm. *Bull. Math. Biol.*, 54:599–618, 1992.
- S. B. Needleman and C. D. Wunsch. A general method to search for similarities in the amino acid sequence of two proteins. *Journal of molecular biology*, 48:443–453, 1970.
- S.A. Olson. EMBOSS opens up sequence analysis. European Molecular Biology Open Software Suite. *Briefings in Bioinformatics*, 3:87–91, 2002.
- K.D. Pruitt, T. Tatusova, and D.R. Maglott. NCBI Reference Sequence (REFSEQ): a curated non-redundant sequence database of genomes, transcripts and proteins. *Nucleic Acids Research*, 33 Database Issue:D501–D504, 2005.
- K. Quandt, K. Grote, and T. Werner. GenomeInspector: a new approach to detect correlation patterns of elements on genomic sequences. *CABIOS*, 12:404–413, 1996.
- S. Rahmann, T. Muller, and M. Vingron. On the power of profiles for transcription factor binding site detection. *Statistical Applications in Genetics and Molecular Biology*, 2:7, 2003.
- A.G. Robertson, M. Bilenky, K. Lin, A. He, W.Yuen, et al. cisRED: a database system for genome-scale computational discovery of regulatory elements. *Nucleic Acids Research*, 34:D68–D73, 2006.
- F.R. Roth, J.D. Hughes, P.E. Estep, and G.M. Church. Finding dna regulatory motifs within unaligned non-coding sequences clustered by whole-genome mrna quantitation. *Nature Biotechnology*, 16: 939–945, 1998.

- A. Sandelin, W. Alkema, P. Engstrom, W. W. Wasserman, and B. Lenhard. JASPAR: an open-access database for eukaryotic transcription factor binding profiles. *Nucleic Acids Research*, 32:D91–D94, 2004.
- T.D. Schneider and R.M. Stephens. Sequence logos: a new way to display consensus sequences. *Nucleic Acids Research*, 18:6097–6100, 1990.
- D. E. Schones, P. Sumazin, and M. Q. Zhang. Similarity of position frequency matrices for transcription factor binding sites. *Bioinformatics*, 21:307–313, 2005.
- S.T. Smale and J.T. Kadonaga. The RNA polymerase II core promoter. *Annu. Rev. Biochem*, 72:449–479, 2003.
- T.F. Smith and M.S. Waterman. Identification of common molecular subsequences. *Journal of Molecular Biology*, 147:195–197, 1981.
- T.F. Smith, M.S. Waterman, and W.M. Fitch. Comparative biosequence metrics. *Journal of Molecular Evolution*, 18:38–46, 1981.
- G.D. Stormo. Gene-finding approaches for eukaryotes. *Genome Research*, 10:394–397, 2000.
- T. Strachan and A.P. Read. *Human Molecular Genetics 2*. John Wiley & Sons, Inc. (New York, USA), 1999. ISBN 0471330612.
- Y. Suzuki, R. Yamashita, S. Sugano, and K. Nakai. Dbtss: Database of transcriptional start sites: progress report 2004. *Nucleic Acids Research*, 32:D78 – D81, 2004.
- The Gene Ontology Consortium. Gene ontology: tool for the unification of biology. *Nature Genetics*, 25:25–29, 2000.
- J. D. Thompson, D. G. Higgins, and T. J. Gibson. Clustalw: improving the sensitivity of progressive multiple sequence alignment through sequence weighting, position-specific gap penalties and weight matrix choice. *Nucleic Acids Research*, 22:4673–4680, 1994.
- M. Tompa et al. Assessing computational tools for the discovery of transcription factor binding sites. *Nature Biotechnology*, 23:137–144, 2005.
- J.C. Venter, M.D. Adams, E.W. Myers, P.W. Li, R.J. Mural, G.G. Sutton, H.O. Smith, M. Yandell, C.A. Evans, R.A. Holt, J.D. Gocayne, P. Amanatides, R.M. Ballew, D.H. Huson, J.R. Wortman, et al. The sequence of the human genome. *Science*, 291:1304–1351, 2001.
- W. W. Wasserman and J. W. Fickett. Identification of regulatory regions which confer muscle-specific gene expression. *Journal of Molecular Biology*, 278:167–181, 1998.
- M. S. Waterman, T. F. Smith, and H. L. Katcher. Algorithms for restriction map comparisons. *Nucleic acids research*, 12:237–242, 1984.
- M.S. Waterman. General methods of sequence comparison. *Bulletin of mathematical biology*, 46: 473–500, 1984.
- M.S. Waterman. *Introduction to computational biology*. Chapman and Hall, UK, 1995. ISBN 0-412-99391-0.
- R.H. Waterston, K. Lindblad-Toh, E. Birney, J. Rogers, J.F. Abril, P. Agarwal, R. Agarwala, R. Ainscough, M. Alexandersson, P. An, S.E. Antonarakis, J. Attwood, R. Baertsch, J. Bailey, K. Barlow, and others (International Mouse Genome Sequencing Consortium, IMGSC). Initial sequencing and comparative analysis of the mouse genome. *Nature*, 420:520–562, 2002.

- G.A. Wray, M.W. Hahn, E. Abouheif, J.P. Balhoff, M. Pizer, M.V. Rockman, and L.A. Romano. The evolution of transcriptional regulation in eukaryotes. *Molecular Biology and Evolution*, 20:1377–1419, 2003.
- R. Yamashita, Y. Suzuki, S. Sugano, and K. Nakai. Genome-wide analysis reveals strong correlation between CpG islands with nearby transcription start sites of genes and their tissue specificity. *Gene*, 350:129–136, 2005.

Chapter 6

Multiple Non-Collinear TF-map Alignment

Summary

The generalization of the pairwise TF-map alignment is presented here. First, the formal definition of a multiple map alignment and how to compute the optimal score is provided. Next, we use a progressive approach to build up a multiple alignment in a stepwise manner. Then, we have studied how to break the non-collinearity property inherent to the alignments produced by dynamic programming techniques. Results on biological data indicate that multiple TF-map alignments are able to locate regulatory elements in several promoters that are not conserved at sequence level.

6.1	The need for multiple TF-map alignment	172
6.2	Basic definitions	174
6.3	The algorithms	176
6.4	Non-collinear TF-map alignments	181
6.5	Biological results	184

6.1 The need for multiple TF-map alignment

SEQUENCE COMPARISONS ARE ONE OF THE MOST IMPORTANT COMPUTATIONAL TOOLS in molecular biology. Sequences are good symbolic representations of biological molecules that encode relevant information about their structure, function and history. From the analysis of several related sequences, biologically significant facts can be inferred. For instance, genomic sequence comparisons are performed in order to identify genes or regulatory sites across different genomes, as these functional elements tend to exhibit consensual patterns different from those observed in regions that are not functional.

In attempt to allow for multiple sequence comparisons, the basic dynamic programming recurrences introduced in the 1970s to align efficiently two sequences of n symbols in $O(n^2)$ (Needleman and Wunsch, 1970; Sellers, 1974), can be naturally extended for k sequences, with an exponential cost $O(n^k)$ (Waterman et al., 1976). As this cost is unaffordable in practice, many heuristics have appeared to provide acceptable solutions with a minor cost. The most popular of them is the hierarchical or clustering method (Feng and Doolittle, 1987; Thompson et al., 1994).

This procedure, also called progressive alignment, is a greedy algorithm that runs in $O(k^2n^2)$ time. In a first step, this method performs all of the pairwise alignments to build an evolutionary tree. In a second step, an initial alignment is constructed from the two closest sequences, incorporating then the rest to the profile following the guide tree. Such a procedure does not guarantee to find the optimal solution in mathematical terms. However, the results are generally in good agreement with the biological problem of aligning correctly bases of homologous functional elements. See Chapter 3 Section 3.5 for a comprehensive review of this topic.

Progressive alignment has also commonly used in the genome-wide alignment methods that perform rapid multiple genomic alignments to identify conserved biological features between distant species. Basically, these algorithms identify local similarities between two genomes that are then used as anchors to align the interleaving regions (Delcher et al., 1999). The progressive technique is then combined with these genome pairwise aligners to build up the multiple genome alignment (Brudno et al., 2003; Bray and Patcher, 2004).

These comparisons at the sequence level have limitations however. Although similar sequences do tend to play similar biological functions, the opposite is not necessarily true. Often similar functions are encoded in higher order sequence elements that are not necessarily conserved at the sequence level. As a result, similar functions are frequently encoded by diverse sequences which are undetectable by conventional sequence alignment methods.

Gene promoter regions are a good example. The information that governs the RNA synthesis is mostly encoded in the gene promoter, a region normally 200 to 2,000 nucleotides long upstream of the transcription start site of the gene (TSS). Transcription factors (TFs) bind to sequence specific motifs (the TF binding sites, TFBSs) within the promoters. TFBSs are 5–8 nucleotides long and one promoter region contains on the order of 10 to 50 of them (Wray et al., 2003). Such motifs appear to be arranged in specific configurations that define the temporal and spatial transcriptional pattern program of each gene. Genes presenting similar expression patterns are assumed to share similar configurations of TFBSs in their promoters. However, TFBSs associated to the same TF are known to contain sequence

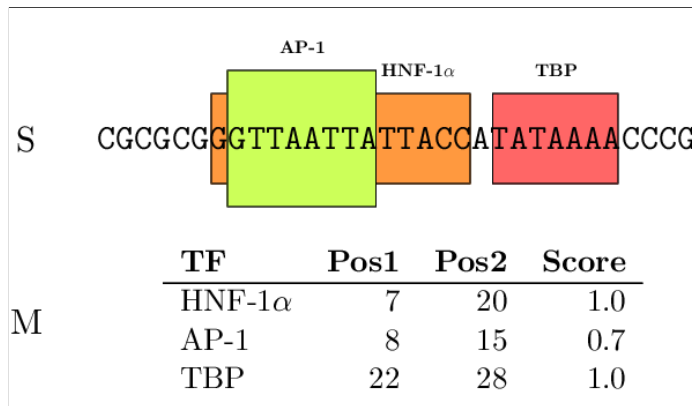


Figure 6.1 TF-mapping in a simple example.

substitutions, being in many cases completely different. Promoter regions of genes with similar expression pattern may not be similar at the sequence level, even though they may be co-regulated.

In the previous chapter (Blanco et al., 2006b), we suggested the existence of regulatory information conserved between related promoters that could not be detected at the sequence level. Let Σ_{TF} be the alphabet of TFs denoting symbols. We initially defined the process of mapping a nucleotide sequence into a sequence in Σ_{TF} (the TF-maps). Then, we developed an efficient algorithm to obtain the global pairwise alignment between two TF-maps (Blanco et al., 2006b). Finally, we showed the TF-map alignments were more accurate than conventional sequence alignment to distinguish pairwise gene co-expression in a collection of microarray results (Blanco et al., 2006b).

In this chapter, we present an efficient implementation of the multiple TF-map alignment based in the progressive alignment paradigm. We have introduced some modifications in the pairwise global TF-map alignment algorithm to align two clusters of TF-maps, eventually allowing non-collinear arrangements of TFBSs in the results without additional cost. Most dynamic programming global alignments rarely cope with the presence of rearrangements observed in the DNA, being only partially identified by combining global and local alignment strategies (Brudno et al., 2004; Darling et al., 2004). This problem is particularly relevant in the case of the regulatory regions, where non-collinear configurations of TFBSs are prone to be conserved (Nix and Eisen, 2005).

The structure of the chapter is the following: first, we briefly reviewed the concept of mapping functions and provide the formal definition of a multiple TF-map alignment. Then, we introduce the main algorithm that performs the progressive alignment of multiple TF-maps. Next, we detail the algorithm to compute the optimal pairwise alignment of two clusters of maps. Later, we define formally a non-collinear alignment, introducing some modifications in the pairwise algorithm to allow the detection of these cases. Finally, we systematically estimate the optimal parameters of the alignment to distinguish promoters from other gene regions in a set of well characterized human-rodent gene pairs and their corresponding orthologs in chicken and zebrafish. These results are compared to those obtained by conventional sequence alignment methods, showing the validness of our ap-

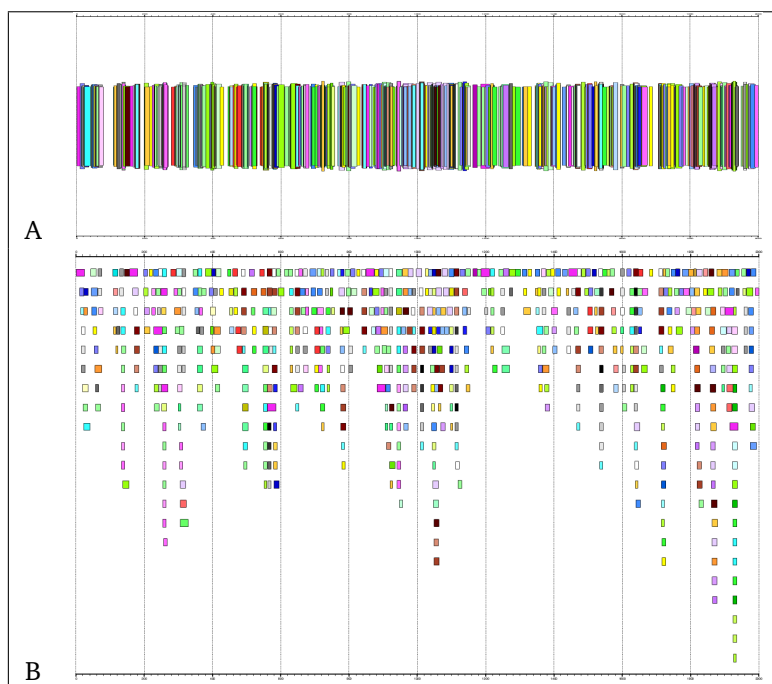


Figure 6.2 TF-mapping of the human promoter NM_015900 (500 nucleotides). (A) Condensed representation of the TRANSFAC predictions. (B) The same set of predictions displayed in a non-overlapping format.

proach. Several particular examples are presented in which multiple TF-map alignments characterize conserved regulatory elements that are otherwise imperceptible in sequence-level comparisons.

6.2 Basic definitions

Mapping a promoter sequence into a TF-map

Let Σ_{DNA} be the alphabet of four nucleotides. Let Σ_{TF} be the alphabet of TFs denoting symbols. In a previous work (Blanco et al., 2006b), we defined a mapping function as a procedure to translate a promoter region $S = s_1 s_2 \dots s_k$ where each nucleotide $s_i \in \Sigma_{\text{DNA}}$, into a sequence of TF-tuples $M = m_1 m_2 \dots m_n$ where each TF-tuple $m_i = \langle m_i^f, m_i^{p1}, m_i^{p2}, m_i^s \rangle$ denotes the match of a binding site for the TF $m_i^f \in \Sigma_{\text{TF}}$ occurring between the position m_i^{p1} and the position m_i^{p2} over the sequence S with score m_i^s . Different mapping functions can be used to obtain the translation from S to M such as a collection of weight matrices representing TFBSs (JASPAR (Vlieghe et al., 2006), PROMO (Farre et al., 2003) or TRANSFAC (Matys et al., 2006)). For each match over a given threshold, we register a new TF-tuple

in M defined by the label (m_i^f) of the TF associated to the PWM, the positions (m_i^{p1}, m_i^{p2}) and the score (m_i^s) of the match (see Figure 6.1, for an example). Other mapping functions can be used instead, such as pattern discovery programs that identify a set of unknown motifs conserved in several promoters (e.g. MEME (Bailey and Elkan, 1994)).

Matches are annotated at a given location irrespective of their orientation in which they occur. This translation preserves the order of S in M , that is if $i < j$ in M then ($m_i^{p1} < m_j^{p1}$). Matches to different TFs may possibly occur at the same position, being false positives in most cases (see a real example in Figure 6.2). We refer to the resulting sequence of TF-tuples M as a Transcription Factor Map, or simply a TF-map.

Multiple alignment of TF-maps

Let M_1, M_2, \dots, M_k be a set of TF-maps where each map is denoted as $M_i = m_{i,1} m_{i,2} \dots m_{i,|M_i|}$ and each TFBS is denoted as $m_{i,j}^f \in \Sigma_{TF}$. Let $M_1^*, M_2^*, \dots, M_k^*$ be the extended set of TF-maps where each map is denoted as $M_i^* = m_{i,1}^* m_{i,2}^* \dots m_{i,|M_i^*|}^*$, and each TFBS is denoted as $m_{i,j}^{*f} \in \Sigma_{TF} \cup \{-\}$. The symbol $'-'$ indicates a gap, which can be considered as a particular TF-tuple $\langle '- ', \cdot, \cdot, \gamma \rangle$ where the value \cdot is a null value and γ is the penalty for introducing a gap in a column of the alignment.

The alignment of k maps M_1, M_2, \dots, M_k is then a correspondence T , maybe empty, among the extended maps $M_1^*, M_2^*, \dots, M_k^*$ such that:

1. The extended maps have the same length.
2. If the gaps are removed from each M_i^* , we recover M_i .
3. At least one element in a column is different from a gap.
4. The elements that are aligned in a column correspond to the same TF.
5. No overlap in the primary sequence is permitted between adjacent sites in the alignment.

Note that the first three conditions define the classical multiple alignment of sequences. Last two conditions, however, introduce two new constraints that are related to the match state and the non-overlapping property, according to the notion of pairwise TF-map alignment provided in (Blanco et al., 2006b).

The score of a multiple alignment of TF-maps

A multiple TF-map alignment –or simply, a multiple map alignment (MMA), in contrast to a multiple sequence alignment (MSA)– can be also represented as a rectangular array:

$$T = \begin{pmatrix} m_{1,1}^* & m_{1,2}^* & \dots & m_{1,t}^* \\ m_{2,1}^* & m_{2,2}^* & \dots & m_{2,t}^* \\ \dots & \dots & \dots & \dots \\ m_{k,1}^* & m_{k,2}^* & \dots & m_{k,t}^* \end{pmatrix}, \quad (6.1)$$

where each column $T(i) = (m_{1,i}^*, m_{2,i}^*, \dots, m_{k,i}^*)$ is the multiple match among the TF-tuples in position i from $M_1^*, M_2^* \dots M_k^*$. Given the multiple alignment T , we compute the score $s(T)$ of the MMA as:

$$s(T) = - \frac{\alpha \sum_{i=1}^t \sum_{j=1}^k m_{j,i}^{*s}}{\lambda(g)} - \mu \sum_{\forall i,i'} f(m_{1,i}^{*p1} - m_{1,i'}^{*p1}, m_{2,i}^{*p1} - m_{2,i'}^{*p1}, \dots, m_{k,i}^{*p1} - m_{k,i'}^{*p1}) \quad (6.2)$$

where $\alpha, \lambda, \mu > 0$, g is the number of columns with only one element different from a gap in the MMA (unaligned elements), and f is a function that measures the conservation of distance between the sites of every map in two consecutive columns (i, i') with more than one aligned element in the MMA. That is, the score of the alignment increases with the score of the aligned elements and the penalty of the gaps (α), and decreases with the number of unaligned elements (λ), and with the difference in the distance between adjacent aligned elements (μ). See the previous chapter and [Blanco et al. \(2006b\)](#) for further details about the TF-map alignment parameters.

6.3 The algorithms

There are many possible alignments between a group of TF-maps. The optimal alignment is the one scoring the maximum among all possible alignments. In a previous work ([Blanco et al., 2006b](#)), we implemented a dynamic programming algorithm to obtain such an alignment efficiently for the case of two TF-maps. The optimal multiple sequence alignment problem (and therefore also the multiple alignment of maps) is, however, much more difficult, being formally a NP-complete problem ([Wang and Jiang, 1994](#)).

Here, we propose to adapt the popular progressive alignment strategy to the TF-map alignment. The solutions obtained by this method are not guaranteed to be optimal. However, multiple progressive alignments usually have an underlying biological explanation ([Thompson et al., 1994](#)). We have also introduced some changes in the basic pairwise TF-map alignment algorithm developed in ([Blanco et al., 2006b](#)), in order to deal now with two clusters of MMAs instead of two single TF-maps.

Progressive MMA algorithm

Let $(G_1 \dots G_k)$ be the initial list of k TF-map groups, where each group contains a single TF-map. Let S be the similarity matrix where $S(G_i, G_j)$ denotes the similarity between the TF-map groups G_i and G_j .

The progressive MMA algorithm shown in Figure 6.3 builds up a multiple TF-map alignment in a stepwise manner. In a first step, all pairwise TF-map alignments are performed. The initial multiple alignment is created with the two most similar ones. Both maps are substituted for a new group that contains their alignment. The similarity between this new cluster and the rest of the TF-maps is then estimated, updating the S matrix (see Implementation).


```

Pre  $\equiv$   $G$ : list of TF-map groups ( $G_1 \dots G_k$ )

(* Initial Step: pairwise alignment all Vs all *)
maxSim  $\leftarrow -\infty$ 
for  $i = 1$  to  $k$  do
5:   for  $j = i + 1$  to  $k$  do
        $S(G_i, G_j) \leftarrow \text{ComputePairwiseSimilarity}(G_i, G_j)$ ;
       (* Select the pair with maximum similarity *)
       maxSim  $\leftarrow \max(\text{maxSim}, S(G_i, G_j))$ ;
       (* Create a new group: estimate the similarity to others *)
10:   $G_{iSim-jSim} \leftarrow \text{MergeGroups}(G_{iSim}, G_{jSim})$ ;

(* Progressive Step: cluster the two most similar groups *)
while  $|G| > 1$  do
    maxSim  $\leftarrow -\infty$ 
15:   for  $i = 1$  to  $|G|$  do
       for  $j = i + 1$  to  $|G|$  do
           (* Select the pair with maximum similarity *)
           maxSim  $\leftarrow \max(\text{maxSim}, S(G_i, G_j))$ ;
           (* Create a new group: estimate the similarity to others *)
20:    $G_{iSim-jSim} \leftarrow \text{MergeGroups}(G_{iSim}, G_{jSim})$ ;

```

Figure 6.3 Progressive multiple map alignment algorithm.

In a second step, an iterative procedure selects at each round the pair of clusters that are more similar from the pool of available groups. These two groups are aligned and merged again into a new TF-map cluster, estimating the similarity to the remaining ones. At the end of the process, there is only one group that contains the progressive alignment of the input TF-maps.

The cost of the progressive MMA can be expressed in terms of the number of pairwise TF-map alignments that must be computed. Let k be the number of maps to be aligned and n be the length of each map. The initial round performs $O(k^2)$ pairwise alignments. Next, the progressive round performs $O(k)$ alignments involving two groups. Let $P(n)$ be the cost of each pairwise operation, then the cost of the progressive alignment algorithm is $O(k^2 \cdot P(n))$. The expected value of $P(n)$ is calculated in the next section.

Implementation

In the progressive MMA algorithm shown in Figure 6.3, the variable *maxSim* saves the maximum score so far computed at each round. The group identifiers of such a score can easily be retrieved using a supplementary pair of variables *iSim*, *jSim*.

The pairwise TF-map alignment algorithm called *ComputePairwiseSimilarity* (Blanco et al., 2006b) has been slightly modified to accomodate the alignment of two TF-maps groups, as explained in the next section. The optimal pairwise alignments between the

input TF-maps in the initial round are saved, as they could be required during the iterative procedure.

Once a new TF-map group is created from the two most similar ones, their binding sites must be merged (function *MergeGroups*). The order of the TFBSs in the new group must take into account the position of the binding sites in their primary promoter sequences. In the approach here, we do not create a profile of each MMA. Instead, all of the TFBSs of each group are always available for subsequent TF-map alignments.

The alignments between this new TF-map group and each one of the rest of the groups are not explicitly computed. The similarity among them is instead estimated with the WPGMA method (Weighted Pair Group Method with Arithmetic Mean) according to the previous similarity between the groups G_{iSim} and G_{jSim} to the others. If an alignment between two groups whose similarity was estimated before is identified as the most similar during the progressive step, the MMA must be explicitly computed before merging both TF-map groups.

The alignment of two clusters of MMAs

Let $G_x = m_{x,1}m_{x,2} \dots m_{x,|G_x|}$ and $G_y = m_{y,1}m_{y,2} \dots m_{y,|G_y|}$ be the two most similar groups of TF-maps in the current round of the progressive alignment. Let S be the scoring dynamic programming matrix where $S(i,j) = S(m_{x,i}, m_{y,j})$ denotes the similarity of the best TF-map alignment of the groups $G_x = m_{x,1} \dots m_{x,i}$ and $G_y = m_{y,1} \dots m_{y,j}$, according to the scoring function in Equation 6.2. The *ComputePairwiseSimilarity* algorithm explained here is a generalization of that developed in (Blanco et al., 2006b) to align two TF-maps that computes the optimal pairwise TF-map alignment between G_x and G_y .

This algorithm basically searches the the maps of both groups to find matches between one site in one group and one site in the other. Once a new match is identified, the previous matches must be evaluated in order to construct the optimal alignment ending at this one (see Figure 6.4). Because this class of scoring matrices are highly sparse, we register the coordinates in S of the matches computed previously. Thus, to compute the optimal score at the cell $S(i,j)$, only the non-empty cells in S that are visible for the current match need to be accessed. In addition, we maintain the list sorted by optimal score, so that the cell scoring the maximum value is at the beginning of the list and, in most cases, only a few nodes will need to be accessed before a critical node is reached beyond which the optimal score can not be improved (Blanco et al., 2006b).

The number of computations $P(n)$ in this algorithm is very similar to that obtained in the conventional pairwise TF-map alignment algorithm (Blanco et al., 2006b). The exact complexity of this algorithm is difficult to be studied –depending mostly on the size of the input maps and the sparsity of the resulting matrix S . An expected time cost analysis reveals that the cost function can be explained in terms of (a) a first quadratic term derived from the obligatory comparison between all of the TFBSs of both maps to detect the match cells and (b) a second quadratic term necessary to search for each match the best adjacent previous pair in the optimal TF-map alignment. In (Blanco et al., 2006b), we studied the contribution of using a list of non-empty cells in S that reduces the second component to an expected cost of $O(p \cdot n^2)$, where p is the percentage of the matrix that is occupied. This value was estimated to be below 5% of occupancy for the pairwise TF-map promoter comparisons.

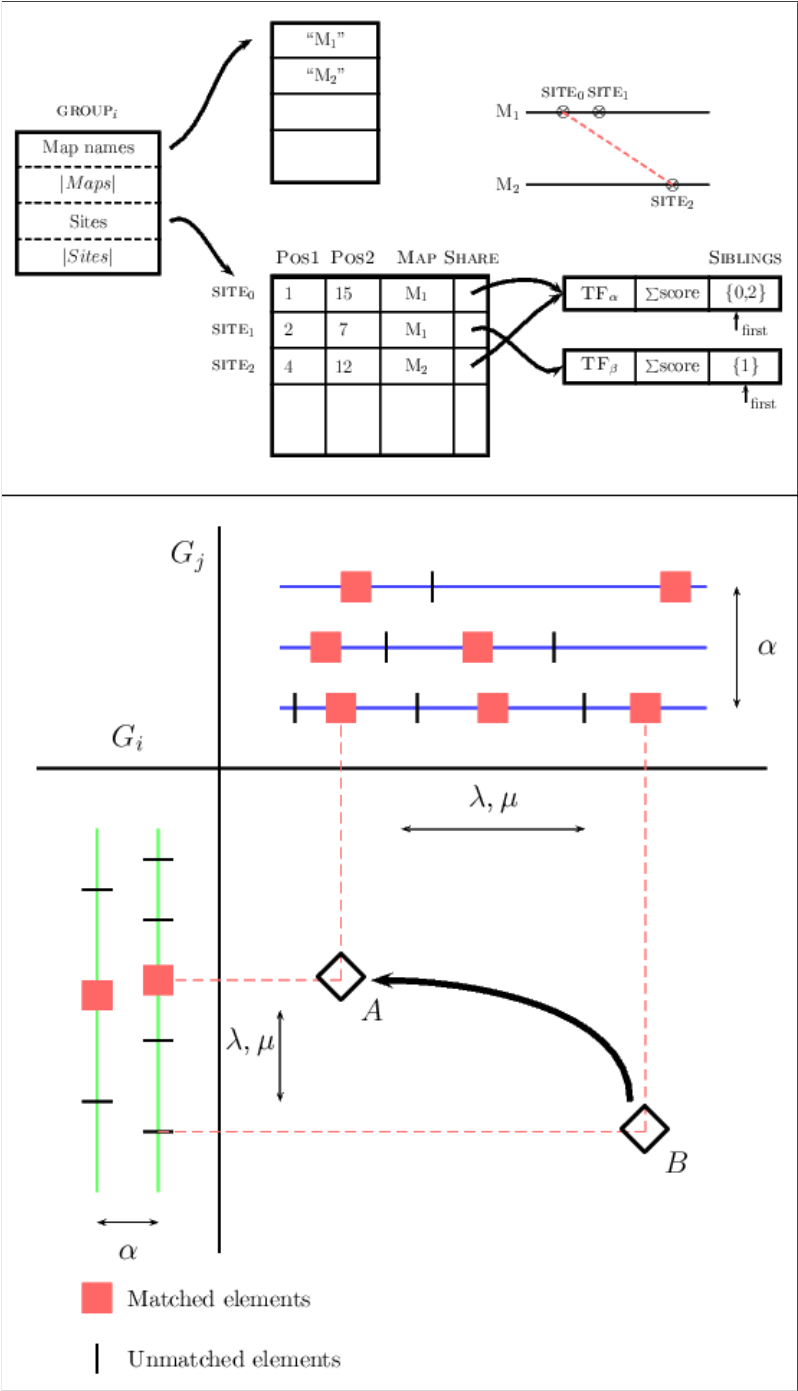


Figure 6.4 MMA algorithm: data structures and similarity matrix.

```

Pre  $\equiv G_x, G_y$ : TF-map groups, L: list of <abscissa,ordinate>, L =  $\emptyset$ 
(* Calculating the element i, j in S *)
for i = 0 to  $|G_x| - 1$  do
  for j = 0 to  $|G_y| - 1$  do
    if factor( $m_{x,i}$ ) = factor( $m_{y,j}$ ) then
5:      S(i, j)  $\leftarrow$  ComputeInitialSimilarity( $m_{x,i}, m_{y,j}$ );
      x  $\leftarrow$   $\alpha$  (score( $m_i$ ) + score( $m_j$ ));
      (* Searching the best previous match in L *)
      p  $\leftarrow$  first(L);
      i'  $\leftarrow$  abscissa(p);
10:     j'  $\leftarrow$  ordinate(p);
      while end(L) = FALSE and S(i', j') + x > S(i, j) do
        (* Compute the  $\mu$  value and check overlap *)
        (D1, D2, overlap)  $\leftarrow$  ComputeOverlap(i, i, j, j', Gx, Gy);
        if overlap = FALSE then
15:         y  $\leftarrow$   $\lambda$  (ComputeLambda(i, i, j, j'));
         z  $\leftarrow$   $\mu$ (|D1 - D2|);
         maxSim  $\leftarrow$  S(i', j') + x - y - z;
         if maxSim > S(i, j) then
           S(i, j)  $\leftarrow$  maxSim;
20:         p  $\leftarrow$  next(L);
         i'  $\leftarrow$  abscissa(p);
         j'  $\leftarrow$  ordinate(p);
      n  $\leftarrow$  CreateNewNode(i, j);
      InsertNode(n, L);

```

Figure 6.5 Pairwise alignment of two clusters of TF-maps.

Implementation

In the pseudocode in Figure 6.5, the groups G_x and G_y are represented as two arrays of sites sorted by the position in their promoters, where each site corresponds to an input TFBS. The multiple TF-map alignment of a cluster is internally encoded with pointers among the sites that form each match. Gaps here are not explicitly represented.

Each site $m_{x,i}$ is a structure as described above with the functions *factor*, *pos1*, *pos2* and *score* returning the values of the corresponding fields. The variable *maxSim* stores the optimal score so far computed. The sites in the optimal TF-map alignment can be easily retrieved using a supplementary structure *path(i,j)* that points to the previous cell in the optimal path leading to cell S(i, j). In addition, the function *ComputeInitialSimilarity* calculates for each match S(i, j) the initial score of a hypothetical alignment that includes only the sites $m_{x,i}$ and $m_{y,j}$.

Once the match between two sites $m_{x,i}$ and $m_{y,j}$ has been identified, the best previous match between two other sites $m_{x,i'}$ and $m_{y,j'}$ is used to construct the new alignment (see the matches A and B in Figure 6.4). The list L is used to locate the non empty positions in S. Each node of the list L is represented as structures p and n with the functions *abscissa*

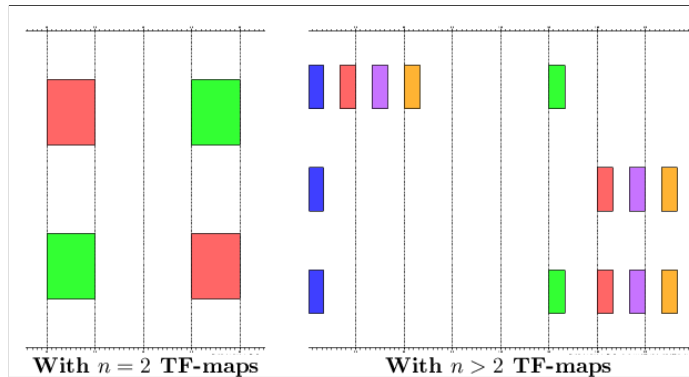


Figure 6.6 Two examples of non-collinear MMAs. (Left) A pairwise non-collinear TF-map alignment. (Right) A non-collinear MMA.

and *ordinate* returning the corresponding coordinates in S of each previous match.

The score of the new match between $m_{x,i}$ and $m_{y,j}$ is the sum of the scores of the columns in which both elements were aligned in their respective MMAs. Unaligned sites are scored with the gap penalty γ . The function *ComputeLambda* counts the number of sites in each group that are not included in the alignment, taking into account the size of each group. The function *ComputeOverlap* calculates the average distances $D1$ and $D2$ between any pair of consecutive matches in the maps of both groups, verifying the absence of physical overlap in their promoters. The function $|D1 - D2|$ scores the conservation of distance between the sites of every map in two consecutive columns in the MMA (function f , see Equation 6.2).

6.4 Non-collinear TF-map alignments

The existence of regulatory elements that are conserved in different order between related promoter regions is documented, specially in enhancers (Nix and Eisen, 2005). Even at the sequence level, the identification of these DNA rearrangements is very difficult. We have here introduced some subtle changes in the pairwise TF-map alignment algorithm shown before to deal with non-collinear alignments. The aligned TFBSs in such MMAs are therefore not necessarily located in the same relative order in every map.

Definition

Let T be an alignment between two TF-maps M_1 and M_2 formally defined as a correspondence $T = \{(m_{1,i_1}, m_{2,j_1}), \dots, (m_{1,i_t}, m_{2,j_t})\}$. Let $(m_{1,i}, m_{2,j})$ and $(m_{1,k}, m_{2,l})$ two matches in T , not necessarily contiguous, with $i < k$. Then, we define the collinearity or non-collinearity of T in terms of the ordering between j and l , for all the match pairs of T as:

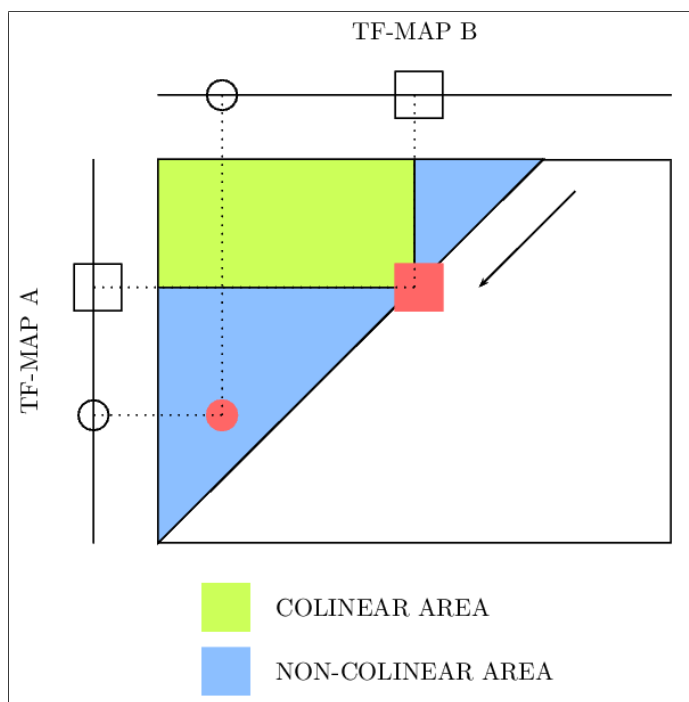


Figure 6.7 Diagonal filling of the alignment matrix.

1. If $j < l$ then T is a collinear alignment
2. If $j > l$ then T is a non-collinear alignment (see example shown in Figure 6.6 (Left)).

The generalization of this definition for $k > 2$ TF-maps is immediate (see the example of a non-collinear MMA for $k = 3$ TF-maps in Figure 6.6 (Right)).

The algorithm

The non-collinear matches shown in Figure 6.6 can not be detected in the basic pairwise TF-map alignment algorithm. Let A and B be two TF-maps in which two matches could form a non collinear alignment (represented as a circle and a square in Figure 6.7). The normal implementation fills in the matrix row by row, from top to bottom (or column by column, from left to right). According to this, when the first match is being processed (red square), the second one (red circle) is not still available (green area). On the contrary, when the second match is processed, the first one is not accessible as the basic algorithm only allows the search for best previous aligned elements in the list of computed values that are in the area delimited by the current match.

To overcome such a limitation, we propose to compute the optimal values of the matrix S following a different order, to allow the visibility of one of these elements (circle) by the

other (square). For instance, the top-bottom diagonal filling of the matrix depicted in Figure 6.7 may process in first position the element that was not visible before (circle) for the other element (square) that will be computed later in the next diagonal (square). While this strategy still produces the same alignments obtained with the ordinary implementation, non-collinear alignments produced by new combinations of matches can also be formed.

Adjusting the non-collinearity

Non-collinear conservation of regulatory elements is documented in very specific cases (Nix and Eisen, 2005). Most upstream promoter regions, however, are constituted of collinear arrangements of TFBSs. Because of the poor specificity of the collections of PWMs (Schones et al., 2005), many non-collinear alignments produced with the algorithm described above are simply artifacts.

Thus, we have designed a simple mechanism to adjust the frequency of non-collinear aligned sites in the output. As the function *ComputeOverlap* in the algorithm above needed to be redefined in order to detect non-overlap between non-collinear matches as well, we have introduced an additional parameter c to weight those alignments involving non-collinearity.

The following example is graphically presented in Figure 6.8 (Left). Let A and B be two TF-maps in which a previous match has been identified (represented as a circle). Then, a second match between an element in A and another in B is being processed (the squares). The dotted lines indicate that such a site in B can be located either on the left or on the right of the circle site in the same map. In the first case, a non-collinear alignment is produced; in the second case, a normal collinear alignment is constructed.

The algorithm to align two clusters of TF-maps must be slightly modified to accommodate the non-collinearity parameter c (the case in which the non-collinear match occurs in A can be similarly defined):

$$z = \begin{cases} \text{if } (D_2 < 0) & \rightarrow \mu|D_1 - c \cdot D_2|, c \geq 1 \\ \text{if } (D_2 \geq 0) & \rightarrow \mu|D_1 - D_2| \end{cases} \quad (6.3)$$

The optimal positional conservation between both matches occurs when $d_1 = d_2$. However, the parameter c is used into the μ penalty to punish only those matches that do not respect the collinearity of the current alignment (the square site is on the left of the circle site in B , see Figure 6.8).

Informally, if $c = 1$ then both collinear and non-collinear matches are indistinctly combined into the resulting MMA. High values of c , however, produce a higher amount of collinear matches into the results. In order to establish formally the behaviour of this parameter, we have counted the number of non-collinear matches in the TF-map alignment of the human and mouse promoters (500 nucleotides) of the MMP13 gene (REFSEQ entries NM_002427 and NM_008607). In Figure 6.8, there is a clear correspondence between the amount of inversions in the MMA and the value of c . No inversions are produced for large values of c .

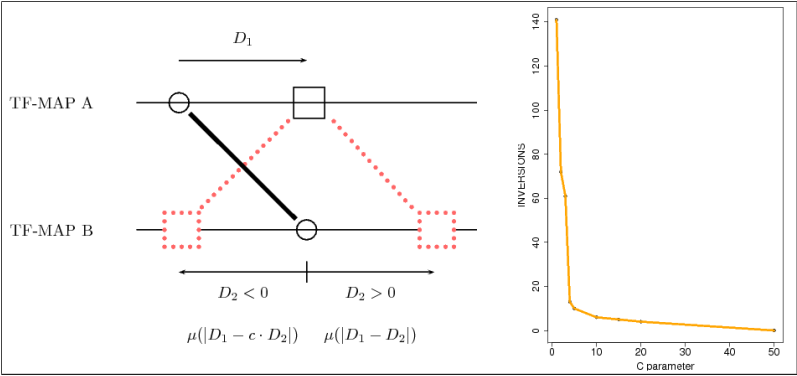


Figure 6.8 The non-collinearity parameter.

Identification of non-collinear configurations of TFBSs in regulatory regions is poorly known. We recommend, therefore, to use this option very carefully. In addition, we also suggest the use of a small set of matrices to perform the mapping, which can reduce the number of artifacts in the resulting non-collinear MMA.

6.5 Biological results

The optimal MMA of a set of TF-maps is obviously dependant on the values of the $\alpha, \lambda, \mu, \gamma$ and c parameters. In addition, the optimal parameter configuration is likely to depend on the particular problem to be addressed (orthologous genes or co-regulated genes in microarray experiments), and the particular protocol to map the TFBSs on the sequences.

Results in the previous chapter (Blanco et al., 2006b), indicated that TF-maps alignments are able to characterize promoter regions of co-regulated genes in absence of sequence similarity. Thus, TF-map alignments were shown to detect high-order regulatory signals conserved in a collection of related promoters that were undetectable for current sequence alignment methods. It is important to mention that two different TFBSs can be aligned if they correspond to the same TF, irrespectively of their sequence motifs.

Here we have conducted a similar systematic training over an extended set of orthologous promoters for obtaining the optima configuration. In order to verify the ability of MMA to identify regulatory elements that are rarely detected in conventional comparisons, we have compared the results to those obtained by global sequence alignment methods. In addition, we have focused on three specific examples to show the abilities of MMA in the characterization of co-regulated gene promoters. In all of the cases, we have only constructed collinear map alignments as non-collinear regulatory rearrangements have not been reported on them.

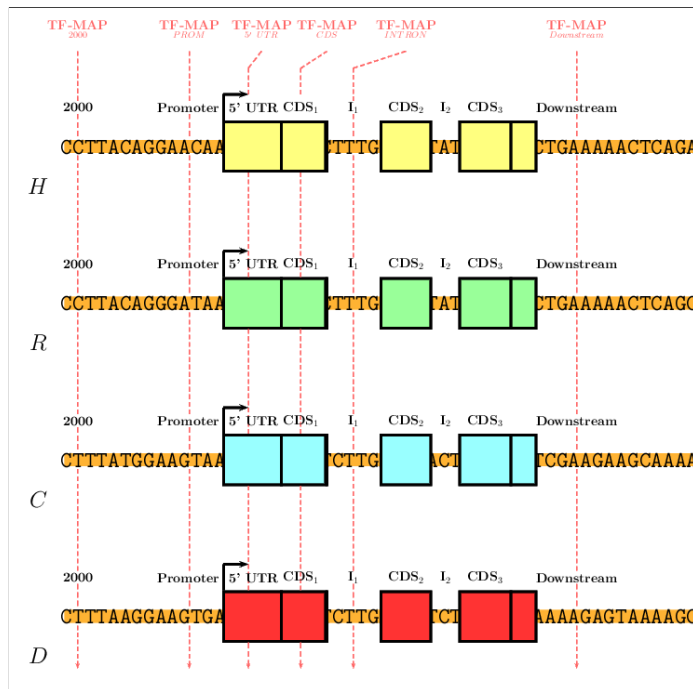


Figure 6.9 Distinguishing promoters from other genomic regions.

Multiple TF-map training

For the pairwise TF-map alignment, we estimated the optimal parameters in a set of experimentally characterized human and rodent gene promoters (Blanco et al., 2006b). Here we have extended such a dataset by searching the corresponding orthologs in chicken and zebrafish as well. Using the REFSEQ (Pruitt et al., 2005) gene set as mapped into the UCSC genome browser, we have correctly identified the ortholog in both species, if available. We refer to the resulting set of human-mouse-chicken-zebrafish homologous genes as the HRCZ SET. This dataset contains 18 human-rodent-chicken-zebrafish orthologs, 7 human-rodent-chicken orthologs, 4 human-rodent-zebrafish orthologs, and 7 human-rodent orthologs.

The lack of available collections of experimentally verified TFBSs is an important limitation for the evaluation and the training of phylogenetic footprinting systems. Despite several databases of annotations and promoter sequences have recently appeared (Blanco et al., 2006a; Xuan et al., 2005), there is not a minimum amount of regulatory information conserved among species other than human and mouse to train the MMA on them.

Thus, we can not repeat the training procedure used in (Blanco et al., 2006b) to evaluate the ability of MMA to detect conserved regulatory elements at larger evolutionary distances –at which the degree of conservation may be negligible. However, we can use another method, also presented in (Blanco et al., 2006b), to show that MMAs are much more informative than primary multiple sequence alignments.

We first have mapped the TFBSs occurrences in the promoter sequences using the collec-

HRCZ SET	Multiple TF-map alignment		CLUSTALW	
	TOP1	Avg.Score	TOP1	Avg. score
CODING	9	18.61	28	3706.72
5'UTR	2	11.80	4	2671.78
PROMOTER	21	27.81	4	2005.67
INTRONIC	3	9.75	0	1359.19
DOWNSTREAM	1	10.53	0	1174.28
INTERGENIC	0	7.84	0	1052.92

Table 6.1 Results when distinguishing promoters with MMAs.

tion of 50 most informatives matrices in JASPAR 1.0 (Sandelin et al., 2004), to which we refer as JASPAR_{TOP50} (Blanco et al., 2006b).

Then, we have compared the MMAs obtained in the 200 nucleotides of the promoter region of the 36 gene pairs from the HRCZ SET, with the MMAs obtained in fragments of 200 nucleotides from intergenic (2,000 nucleotides upstream of the TSS), 5'UTR (downstream of the TSS), coding (downstream of the translation start site and considering only coding DNA), intronic (downstream of the first intron junction), and downstream (downstream of the transcription termination site) sequences (see Figure 6.9 for a graphical representation of the test). We have computed the average score of the MMA on each one of the genomic regions and have identified, for each orthologous set, the genome regions in which the alignment produces the highest score. We have performed the same exercise using global pairwise sequence alignments (obtained with CLUSTALW, (Thompson et al., 1994)).

We have repeated this test using different combinations of parameters. Systematically, the parameters α, λ and μ were allowed to independently take values between 0.0 and 1.0, in incremental steps of 0.1. At the same time, the parameter γ (gap penalty) was tested between 0 and -10 . The optimal parameter configuration is considered to be that set of parameter values that better discriminate between promoters and the rest of genomic regions.

Results appear in Table 6.1. As expected, nucleotide sequence alignments score the highest in the coding regions (in 28 out of 36 cases), followed by the alignments in the 5' UTR regions (4 out of 36) and in the promoters (4 out of 36). The scores of the sequence alignments show that promoter regions are less conserved than coding regions, and 5'UTRs. Despite this, the optimal MMA configuration in the collinear configuration ($\alpha = 1, \lambda = 0.1, \mu = 0.1, \gamma = -2$) scores the highest in the promoter regions (in 21 out of 36, see Table 6.1). In addition, the average score of map alignments is notably higher than that of the coding regions. Only in 9 out of 36 cases the TF-map alignments score the highest in coding regions. Interestingly, while intron sequences in the human-mouse-chicken-zebrafish orthologs are much less conserved than 5'UTRs, MMAs score the highest in intronic regions in 3 cases whereas they only score the best in 5'UTRs in 2 cases. This is consistent with the fact that first introns are known to often contain regulatory motifs.

Finally, we have also performed a complementary test to measure the specificity of the TF-map alignments. As a negative control, we have shuffled the orthologous associations in the HRCZ SET to construct a pool of unrelated human-mouse-chicken-zebrafish 36 gene entries. Then, the corresponding multiple TF-map alignments of these non-orthologous paired promoters were obtained using the parameters previously optimized. The TF-map alignments of the unrelated promoters of each entry were significantly worse with an average

score more than 50% smaller than TF-map alignments that involved “bona fide” orthologous promoters. For instance, the average score of the TF-map alignments among orthologous promoters when using the JASPAR_{TOP50} collection was 27.81. In contrast, the score of the TF-map alignments between non-related promoters was 12.51. The sites in the alignments involving non-orthologous gene promoters may hypothetically correspond to general regulatory elements present in most core promoters. An alternative, more probable, hypothesis is that they reflect the poor specificity of most PWMs representing TFBSs.

Promoter characterization

We have selected three examples to show the ability of MMAs to characterize promoter regions in the absence of sequence conservation. In the three cases, we have compared the multiple TF-map alignment against the corresponding multiple sequence alignment produced by CLUSTALW, as in the section above.

All of the cases are graphically represented as pictures in which the input TF-maps are displayed on the upper part of the picture and the resulting MMAs are displayed on the lower part of the picture, using the `gff2ps` program (Abril and Guigo, 2000).

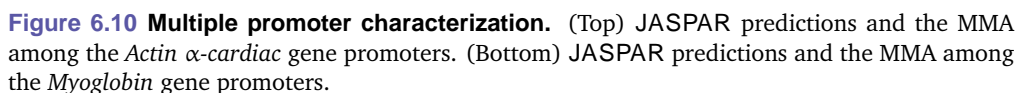
As it is possible to see, the main effect of the MMA is the dramatic reduction in the number of predicted TFBSs that typically result after a PWM-based search (see Figure 6.10 and Figure 6.11). For instance, we aligned 157 human sites to 197 mouse sites, 229 chicken sites and 167 zebrafish sites mapped in the respective Actin α -cardiac gene promoter orthologs (see next section). The resulting multiple TF-map alignment only contained 14 TFBSs, which approximately represents a 13-fold reduction. Graphically, this reduction is noticed in the smaller density of aligned sites in the resulting MMAs picture.

In addition to this, most aligned sites in the MMAs are concentrated in the proximal promoter region of each gene (200 nucleotides upstream of the TSS). This gain in specificity is not simply due to the selection of an arbitrary set of non-overlapping TFBSs, as many experimentally annotated TFBSs on these promoters are successfully covered by the MMAs.

Actin α -cardiac gene

Actins are highly conserved proteins that are involved in various types of cell motility. The alpha actins are found in muscle tissues and are a major constituent of the contractile apparatus. The Actin α -cardiac gene has been identified in many kinds of cells including muscle, where it is a major constituent of the thin filament, and platelets.

The promoter of the human and mouse Actin α -cardiac genes (ACTC, GENBANK entries M13483 and M26773) have been extensively characterized by experimental means (Wasserman and Fickett, 1998). In the ABS database (Blanco et al., 2006a), the entry A0028 informs about the known orthologous binding sites in the respective human and mouse promoters (500 nucleotides, the position +501 is the TSS). The human ACTC promoter is constituted of three SRF sites (+301, +352, +392), a SP1 site (+418), a MYOD site (+445) and a TATA box (+469). Using the REFSEQ gene annotations, we have also identified the corresponding orthologous promoters in chicken and zebrafish (REFSEQ entries NM_001031229 and NM_214784).



We have then aligned the four promoters and compared the resulting MMA with the functional annotations detailed above. In general terms, the multiple TF-map alignment of the four orthologous promoters of ACTC contains many of the functional sites in human and mouse, detecting as well the corresponding orthologs in the other species. The output

coverage is, however, smaller than 50% of the promoter nucleotides.

The MMA of the ACTC promoters is shown in Figure 6.10 (Top). While the region proximal to the TSS is not more dense in predicted TFBSs than other regions, most of the aligned elements cluster near to the TSS. In addition, the alignment agrees well with the functional annotation available in human and mouse, providing novel orthologous sites in chicken and zebrafish:

1. The second SRF binding site is correctly identified in human, mouse and also in zebrafish.
2. A RREB-1 site that overlaps the SP-1 active site is identified in the MMA. RREB-1 and SP-1 are both members of the zinc finger protein families (Vlieghe et al., 2006).
3. A SQUA site that overlaps the third SRF active site is identified in the MMA. SQUA and SRF are both members of the MADS family (Vlieghe et al., 2006).
4. A novel fourth SRF binding site is located immediately upstream of the experimental first one at the four species.
5. The TATA box is correctly detected in human, mouse and zebrafish as well.

No significant conservation among the sequences was, however, detected in the CLUSTALW multiple alignment of the four ACTC promoters (data not shown).

Myoglobin gene

The *Myoglobin* gene is a member of the globin superfamily and is expressed in skeletal and cardiac muscles. The encoded protein is a haemoprotein contributing to intracellular oxygen storage and transcellular facilitated diffusion of oxygen.

The promoter of the *Myoglobin* gene in human (MB, GENBANK entry X00371) and in mouse (REFSEQ entry NM_013593) have been experimentally characterized (Bassel-Duby et al., 1992; Wasserman and Fickett, 1998). In the ABS database (Blanco et al., 2006a), the entry A0037 informs about the known orthologous binding sites in the respective human and mouse promoters (500 nucleotides, the position +501 is the TSS). The human MB promoter is constituted of a CCAC box (+272), a MEF-2 site (+335) with two surrounding E-boxes (+326, +348) and a TATA box (+469). Using the REFSEQ gene annotations, we have also identified the corresponding orthologous promoters in chicken and zebrafish (REFSEQ entries NM_203377 and NM_200586).

We have then aligned the four promoters and compared the resulting MMA with the functional annotations detailed above. The multiple TF-map alignment of the four orthologous promoters of MB contains several of the functional sites in human and mouse, detecting some of the orthologs in the other two species. The output coverage is again very small.

The MMA of the MB promoters is shown in Figure 6.10 (Bottom). Most of the aligned elements are present near to the TSS, while this spatial trend is not observable at the predictions at each promoter. The alignment also contains several of the functional human and mouse sites, providing their counterparts in chicken and zebrafish:

1. A RREB-1 site that overlaps the functional CCAC box is identified in the MMA. In fact, the RREB-1 matrix consensus in JASPAR represents an A/C rich area that contains the CCAC motif (Vlieghe et al., 2006).
2. The TATA box is correctly detected in the four species.

The CLUSTALW multiple alignment of the four MP promoters did not reveal any significant conservation (data not shown).

Collagenase-3 gene (MMP13)

The two previous examples have been extracted from the HRCZ SET. We have now focused on another gene with a more complete set of identified orthologous promoters to test the ability of the MMAs to elucidate high-level conservation even at more phylogenetically distant sequences.

The *Collagenase-3* (MMP13) gene is a member of the matrix metalloproteinase family. MMP13 plays a major role in normal tissue remodeling processes, being abnormally expressed in breast carcinomas and in cartilage from arthritic patients (Pendás et al., 1997). Many experimental studies have confirmed the presence of several functional binding sites for known TFs in human and mice (Pendás et al., 1997; Benbow and Brinckerhoff, 1997; Jiménez et al., 1999; Sun et al., 2000; Hess et al., 2001; Benderdour et al., 2002; Wu et al., 2002).

Here, we have analyzed the proximal promoter regions of MMP13 in human, chimp, mouse, rat, cow, dog, chicken, zebrafish and *Xenopus* (Ortín et al., personal communication). As the 5'UTR of this gene is very small in most cases, we have considered the region 500 bps immediately upstream the ATG (Translation Start Codon) as the proximal promoter.

We performed the multiple TF-map alignment of the nine MMP13 promoters with the optimal configuration calculated in the previous section for four species, increasing the μ parameter to 0.75 to highlight only those regulatory elements that can be aligned in similar positions in most promoters. We also performed the multiple sequence alignment of the nine promoters with the program CLUSTALW. The MMA and the CLUSTALW alignments are both shown in Figure 6.11.

The comparison between the the resulting MMA shown in Figure 6.11 (Top) and experimental annotations on MMP13 gene promoter reveals interesting results. Up to four TFBSs that have been experimentally reported to be functional in human and mouse are remarkably included in such a MMA:

1. The AML-1 binding site included in the resulting MMA (position 330 in human promoter; alternative names: CBFA-1, OSE-2, OSF-2) (Pendás et al., 1997; Jiménez et al., 1999; Hess et al., 2001).
2. The FREAC-4 binding site (position 370 in human promoter; alternative names: FREAC, p53) (Sun et al., 2000).
3. The SPI-1 binding site (position 391 in human promoter; alternative names: AP-1, ETS, PEA-3) (Pendás et al., 1997; Benbow and Brinckerhoff, 1997; Wu et al., 2002).

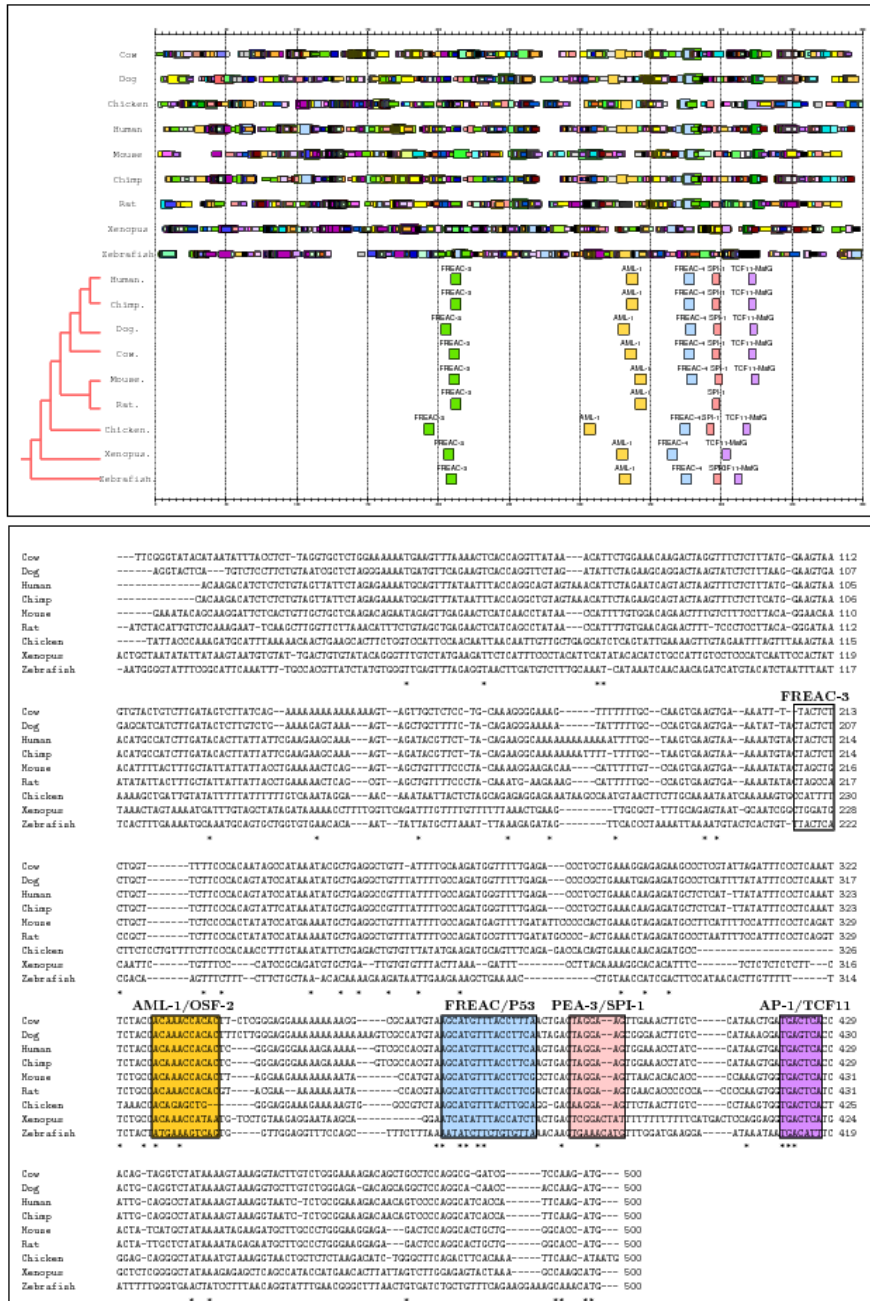


Figure 6.11 MMA of the MMP13 promoter in 9 species. (Top) JASPAR predictions and the resulting multiple TF-map alignment. (Bottom) The CLUSTALW multiple sequence alignment of the 9 promoters.

The SPI-1 transcription factors are distant related members of the Ets family (Ray-Gallet et al., 1995).

4. The TCF11-MafG binding site (position 420 in human promoter, alternative names: AP-1) (Pendás et al., 1997; Benbow and Brinckerhoff, 1997; Wu et al., 2002). The human transcription factor TCF11 is known to bind to a subclass of AP1-sites (Johnsen et al., 1998).

We have not only detected the human and mouse experimental binding sites but we have also identified with the MMA the putative novel site of each TF in most orthologs of the other species, including the most distant ones. The first aligned TF in the MMA (FREAC-3), which has not been experimentally detected so far, presents a similar positional conservation in all of the orthologs. In addition, the resulting phylogenetic tree constructed from the progressive multiple TF-map alignment (shown in red, left) correlates well with the real phylogeny of these nine species.

Accurate inspection of the the global sequence alignment by CLUSTALW in Figure 6.11 (Bottom) only reveals some weak conservation blocks that could partially contain any of the functional TFBSs detected by the multiple TF-map alignment. We also tested several configurations of CLUSTALW (adjusting the gap open and gap extension penalties). However, we did not found any parameter combination that was able to clearly detect all of the four functional sites.

Bibliography

- J. F. Abril and R. Guigo. gff2ps: visualizing genomic annotations. *Bioinformatics*, 8:743–744, 2000.
- T. L. Bailey and C. Elkan. Fitting a mixture model by expectation maximization to discover motifs in biopolymers. *Proceedings of the 2nd International Conference on Intelligent Systems for Molecular Biology (ISMB)*, pages 28–36, 1994.
- R. Bassel-Duby, M.D. Hernandez, M.A. Gonzalez, J.K. Krueger, and R.S. Williams. A 40-kilodalton protein binds specifically to an upstream sequence element essential for muscle-specific transcription of the human myoglobin promoter. *Molecular and Cellular Biology*, 12:5024–5032, 1992.
- U. Benbow and C.E. Brinckerhoff. The ap-1 site and mmp gene regulation: what is all the fuss about? *Matrix Biology*, 15:519–526, 1997.
- M. Benderdour, G. Tardif, J. Pelletier, M. Dupuis, C. Geng, and J. Martel-Pelletier. A novel negative regulatory element in the human collagenase-3 proximal promoter region. *Biochemical and Biophysical Research Communications*, 291:1151–1159, 2002.
- E. Blanco, D. Farre, M. Alba, X. Messeguer, and R. Guigó. ABS: a database of annotated regulatory binding sites from orthologous promoters. *Nucleic Acids Research*, 34:D63–D67, 2006a.
- E. Blanco, X. Messeguer, T.F. Smith, and R. Guigó. Transcription factor map alignments of promoter regions. *PLoS Computational Biology*, 2:e49, 2006b.
- N. Bray and L. Patcher. Mavid: constrained ancestral alignment of multiple sequences. *Genome Research*, 14:693–699, 2004.



Figure 6.12 Using MEME as a mapping function. (Top) The MEME motifs and the resulting MMA in the Actin α -cardiac orthologous promoters. (Bottom) The MEME motifs and the resulting MMA in the Myoglobin orthologous promoters.

M. Brudno, B.D. Chuong, G.M. Cooper, M.F. Kim, E. Davydov, NISC CSB E.D. Green, A. Sidow, and S. Batzoglou. Lagan and multi-lagan: efficient tools for large-scale multiple alignment of genomic dna. *Genome Research*, 13:721–731, 2003.

M. Brudno, S. Malde, A. Poliakov, B.D. Chuong, O. Couronne, I. Dubchak, and S. Batzoglou. Glocal alignment: finding rearrangements during alignment. *Bioinformatics*, 19:i54–i62, 2004.

A.C.E. Darling, B. Mau, F.R. Blattner, and N.T. Perna. Mauve: multiple alignment of conserved genomic sequence with rearrangements. *Genome Research*, 14:1394–1403, 2004.

A.L. Delcher, S. Kasif, R.D. Fleischmann, J. Peterson, O. White, and S.L. Salzberg. Alignment of whole genomes. *Nucleic Acids Research*, 27:2369–2376, 1999.

D. Farre, R. Roset, M. Huerta, J. E. Adsuara, LL. Rosello, M. Alba, and X. Messeguer. Identification of patterns in biological sequences at the ALGGEN server: PROMO and MALGEN. *Nucleic Acids Research*, 31:3651–3653, 2003.

- D. Feng and R.F. Doolittle. Progressive sequence alignment as a prerequisite to correct phylogenetic trees. *Journal of Molecular Evolution*, 25:351–360, 1987.
- J. Hess, D. Porte, C. Munz, and P. Angel. Ap-1 and cbfa/run1 physically interact and regulate parathyroid hormone-dependent mmp13 expression in osteoblasts through a new osteoblast-specific element 2/ap1 composite element. *The Journal of Biological Chemistry*, 276:20029–20038, 2001.
- M.J. G. Jiménez, M. Balbín, J.M. López, J. Álvarez, T. Komori, and C. López-Otín. Collagenase 3 is a target of cbfa1, a transcription factor of the runt gene family involved in bone formation. *Molecular and Cellular Biology*, 19:4431–4442, 1999.
- O. Johnsen, P. Murphy, H. Prydz, and A.B. Kolsto. Interaction of the CNC-bZIP factor TCF11/LCR-F1/Nrf1 with MafG: binding-site selection and regulation of transcription. *Nucleic Acids Research*, 26:512–520, 1998.
- V. Matys et al. TRANSFAC and its module TRANSCOMP: transcriptional gene regulation in eukaryotes. *Nucleic Acids Research*, 34:D108–D110, 2006.
- S. B. Needleman and C. D. Wunsch. A general method to search for similarities in the amino acid sequence of two proteins. *Journal of molecular biology*, 48:443–453, 1970.
- D.A. Nix and M.B. Eisen. Gata: a graphic alignment tool for comparative sequence analysis. *BMC Bioinformatics*, 6:9, 2005.
- A.M. Pendás, M. Balbín, E. Llano, M.G. Jiménez, and C. López-Otín. Structural analysis and promoter characterization of the human collagenase-3 gene (mmp13). *Genomics*, 40:222–233, 1997.
- K.D. Pruitt, T. Tatusova, and D.R. Maglott. NCBI Reference Sequence (REFSEQ): a curated non-redundant sequence database of genomes, transcripts and proteins. *Nucleic Acids Research*, 33 Database Issue:D501–D504, 2005.
- D. Ray-Gallet, C. Mao, A. Tavittian, and F. Moreau-Gachelin. DNA binding specificities of Spi-1/PU.1 and Spi-B transcription factors and identification of a Spi-1/Spi-B binding site in the c-fes/c-fps promoter. *Oncogene*, 11:303–313, 1995.
- A. Sandelin, W. Alkema, P. Engstrom, W. W. Wasserman, and B. Lenhard. JASPAR: an open-access database for eukaryotic transcription factor binding profiles. *Nucleic Acids Research*, 32:D91–D94, 2004.
- D. E. Schones, P. Sumazin, and M. Q. Zhang. Similarity of position frequency matrices for transcription factor binding sites. *Bioinformatics*, 21:307–313, 2005.
- P. Sellers. On the theory and computation of evolutionary distances. *SIAM Journal of applied Mathematics*, 26:787–793, 1974.
- Y. Sun, J.M. Cheung, J. Martel-Pelletier, J.P. Pelletier, L. Wenger, R.D. Altman, D.S. Howell, and H.S. Cheung. Wild type and mutant p53 differentially regulate the gene expression of human collagenase-3 (hmm13). *The Journal of Biological Chemistry*, 275:11327–11332, 2000.
- J. D. Thompson, D. G. Higgins, and T. J. Gibson. Clustalw: improving the sensitivity of progressive multiple sequence alignment through sequence weighting, position-specific gap penalties and weight matrix choice. *Nucleic Acids Research*, 22:4673–4680, 1994.
- D. Vlieghe, A. Sandelin, P.J. De Bleser, K. Vleminckx, W.W. Wasserman, and B. Lenhard. A new generation of JASPAR, the open-access repository for transcription factor binding site profiles. *Nucleic Acids Research*, 34:D95–D97, 2006.

- L. Wang and T. Jiang. On the complexity of multiple sequence alignment. *Journal of Computational Biology*, 337:337–348, 1994.
- W. W. Wasserman and J. W. Fickett. Identification of regulatory regions which confer muscle-specific gene expression. *Journal of Molecular Biology*, 278:167–181, 1998.
- M.S. Waterman, T.F. Smith, and W.A. Beyer. Some biological sequence metrics. *Advances in Mathematics*, 20:367–387, 1976.
- G.A. Wray, M.W. Hahn, E. Abouheif, J.P. Balhoff, M. Pizer, M.V. Rockman, and L.A. Romano. The evolution of transcriptional regulation in eukaryotes. *Molecular Biology and Evolution*, 20:1377–1419, 2003.
- N. Wu, S. Opalenik, J. Liu, E.D. Jansen, M.G. Giro, and J.M. Davidson. Real-time visualization of mmp-13 promoter activity in transgenic mice. *Matrix Biology*, 21:149–161, 2002.
- Z. Xuan, F. Zhao, J. Wang, G. Chen, and M.Q. Zhang. Genome-wide promoter extraction and analysis in human, mouse, and rat. *Genome Biology*, 6:R72, 2005.

Conclusions

THE TF-MAP ALIGNMENTS CAN BE VERY USEFUL to efficiently perform searches of promoter elements that might be conserved in different species. In short, the research presented here has contributed to improve the computational characterization of gene transcription regulatory regions in the following aspects:

- ① We have designed a new family of algorithms, which are named TF-map alignments or simply meta-alignments, to detect conserved high-order configurations of functional elements that do not show discernible sequence conservation. The meta-alignment algorithm does not directly compare the primary sequences. Instead, the algorithm aligns the map of high-level elements obtained with an external mapping function over the original sequences, taking into account their position, the element class and the mapping score.
- ② We have generalized the pairwise meta-alignment algorithm to deal with multiple maps. We followed a progressive approach in which the multiple meta-alignment is build up in a stepwise manner: a first multiple alignment is created with the two most similar maps, and the rest of maps or groups of maps are then aligned to this initial multiple meta-alignment following a guide tree.
- ③ We have investigated the structure and the shape of the resulting meta-alignments. We have incorporated some modifications in the basic algorithm in order to detect non-collinear configurations in the alignments without additional computational cost.
- ④ We have successfully applied the meta-alignment algorithms on the biological problem of eukaryotical promoter characterization. First, we have manually curated a collection of orthologous transcription factor binding sites from the literature, that are experimentally verified in human, mouse, rat or chicken. Next, we have trained the meta-alignment program on a subset of well characterized human-mouse promoters, extracted from this collection. Then, we have shown the TF-map alignments are more accurate than conventional sequence alignment to distinguish pairwise gene co-expression in a large collection of microarray results.

- ⑤ We have also used the meta-alignment approach to distinguish promoters from other gene regions in a set of well characterized human-rodent gene pairs and their corresponding orthologs in chicken and zebrafish. In this particular problem, the multiple meta-alignment identified correctly most orthologous promoter regions, even when comparing to protein coding regions that presented a stronger sequence conservation.
- ⑥ We have comprehensively reviewed the topic of sequence alignment, specially focusing on the pioneering algorithms that have mostly contributed to the field. In addition, we have also contributed to extend our expertise in the areas of computational gene finding and promoter characterization, within the field of bioinformatics.

PART IV

Appendices

Curriculum Vitae

PERSONAL DATA

Name: Enrique Blanco García
Birthplace and birthdate: Barcelona, January 12th. 1976
Working Address: Centre de Regulació Genòmica
Passeig de la Barceloneta 37-49
Barcelona
Telephone number: +34 93 224 08 91
E-mail: eblanco@imim.es
Web page: <http://genome.imim.es/~eblanco>

ACADEMIC CURRICULUM

- ENGINEER IN COMPUTER SCIENCE (*Ingeniero superior en Informática*). Facultat d'informàtica de Barcelona. Universitat Politècnica de Catalunya, Spain (June 2000). [Mark: 7.40/10, PFC: MH]
- DEA IN ALGORITHMICS (*Diploma de Estudios Avanzados, Research Sufficiency*). Departament de Llenguatges i Sistemes Informàtics. Facultat d'informàtica de Barcelona. Universitat Politècnica de Catalunya, Spain (June 2002).
- AQU CERTIFICATE: Professorat Col.laborador (teaching staff), 25 November 2005.

Language Skills

- English : ADVANCED LEVEL (CERTIFICAT D' APTITUD) (LEVEL C), Official School of Languages, Barcelona (EOIBD), Spain.

- Italian : ELEMENTARY LEVEL (CERTIFICAT ELEMENTAL) (LEVEL B), Official School of Languages, Barcelona (EOIBD), Spain.
- Catalan and Spanish : mother tongues.

RESEARCH CURRICULUM

- 2001 - 2006. PhD student (Software program, Universitat Politècnica de Catalunya) at Genome Informatics Research Lab, IMIM, Barcelona.
PhD supervisors:
 - Dr. Xavier Messeguer - peypoch@lsi.upc.edu
(Facultat d'informàtica de Barcelona. Universitat Politècnica de Catalunya)
 - Dr. Roderic Guigó - rguigo@imim.es
(Genome Informatics Research Lab, Research Group of Medical Informatics. IMIM-UPF-CRG).
- 1999 - 2000. Programmer in Genome Informatics Research Lab, Research Group of Medical Informatics, at IMIM, Barcelona.

Research areas

1. Bioinformatics (algorithmics)
 - Sequence analysis
 - Sequence and map alignments
 - Multiple alignments
 - Representation of biological signals
2. Bioinformatics (computational biology)
 - Characterization of gene regulatory regions
 - Gene expression
 - Comparative genomics
 - Microarray analysis
 - Computational gene prediction
3. Computer Science
 - Algorithmics
 - Artificial intelligence
 - Parallelism and supercomputation
 - Internet applications

Computer Skills

- Programming languages: Perl, C, C++, Java, LISP, Pascal, Modula, Ada, PVM, Prolog, GAWK
- Document edition: \LaTeX , pdf \LaTeX
- Web design: XML, HTML, JavaScript, CGI-scripts (web servers), Macromedia Flash, CSSs
- Operating systems: Linux, MAC OS X, Irix, Solaris, Windows 95/98/00/XP
- Office: Word, PowerPoint, Excel, Access

Publications

- **E. Blanco**, X. Messeguer, T.F. Smith and R. Guigó. Transcription Factor Map Alignment of Promoter Regions. *PLOS Computational Biology*, 2(5):e49(2006).
- **E. Blanco**, D. Farre, M. Albà, X. Messeguer, and R. Guigó. ABS: a database of Annotated regulatory Binding Sites from orthologous promoters. *Nucleic Acids Research*, 34:D63-D67 (2006).
- **E. Blanco** and R. Guigó. Predictive Methods Using DNA Sequences. In A. D. Baxevanis and B. F. Francis Ouellette, chief editors: *Bioinformatics: A Practical Guide to the Analysis of Genes and Proteins, Third Edition*. John Wiley & Sons Inc., New York (2005). ISBN: 0-471-47878-4.
- S. Castellano, S.V. Novoselov, G.V. Kryukov, A. Lescure, **E. Blanco**, A. Krol. V.N. Gladyshev and R. Guigó. Reconsidering the evolution of eukaryotic selenoproteins: a novel non-mammalian family with scattered phylogenetic distribution. *EMBO reports*, 5(1):71-77 (2004).
- S. Beltran, **E. Blanco**, F. Serras, B. Perez-Villamil, R. Guigó, S. Artavanis-Tsakonas and M. Corominas. Microarray analysis of the transcriptional network controlled by the trithorax group gene ash2 in *Drosophila melanogaster*, *PNAS*, 100: 3293-3298, (2003).
- **E. Blanco**, G. Parra and R. Guigó. Using geneid to Identify Genes. In A. Baxevanis and D.B. Davidson, chief editors: *Current Protocols in Bioinformatics*. Volume 1, Unit 4.3 (1-26). John Wiley & Sons Inc., New York, (2002). ISBN: 0-471-25093-7.
- G. Parra, **E. Blanco**, and R. Guigó. geneid in *Drosophila*. *Genome Research*, 10: 511-515, (2000).

Posters

- **E. Blanco**, M. Pignatelli, X. Messeguer and R. Guigó. “Deconstructing the position weight matrices to detect regulatory elements. Systems Biology meeting: global regulation of gene expression”. *Cold Spring Harbor: global regulation of gene expression*. (March 2005, New York, USA).
- **E. Blanco**, X. Messeguer and R. Guigó. “Novel computational methods to chracterize regulatory regions. Systems Biology meeting: genomic approaches to transcriptional regulation”. *Cold Spring Harbor: genomic approaches to transcriptional regulation*. (March 2004, New York, USA).
- **E. Blanco**, X. Messeguer and R. Guigó. “Alignment of Promoter Regions by Mapping Nucleotide Sequences into Arrays of Transcription Factor Binding Motifs”. *Seventh annual internation conference on computational biology-RECOMB*. (April 2003, Berlin, Germany).
- **E. Blanco**, G. Parra, S. Castellano, J.F. Abril, M. Burset, X. Fustero, X. Messeguer and R. Guigó. “Gene prediction in the post-genomic era”. *9-th international conference on Intelligent Systems in Molecular Biology*. (July 2001, Copenhagen, Denmark).
- J.F. Abril, **E. Blanco**, M. Burset, S. Castellano, X. Fustero, G. Parra and R. Guigó; “Genome Informatics Research Laboratory: Main Research Topics.” *I Jornadas de Bioinformática* (June 2000, Cartagena, Spain).

Grants

- Predoctoral fellowship. Formacion de Personal Investigador (FPI). Ministerio de Educacion y Ciencia (Spain), 2001-2004.
- Predoctoral fellowship. Institut Municipal d’Investigacio Medica (Spain), 2005-2006.

Participation in Research Projects

- Plan Nacional I+D (2003-2006), ref. BIO2003-05073, Ministerio de Ciencia y Tecnologia (Spain). Principal investigator: Dr. R. Guigó i Serra.
- Plan Nacional I+D (2000-2003), ref. BIO2000-1358-C02-02 Ministerio de Ciencia y Tecnologia (Spain). Principal investigator: Dr. R. Guigó i Serra.

TEACHING CURRICULUM

Topics

- Sequence alignment
- Dynamic programming

- Data structures
- Bioinformatics
- Weight matrices
- Likelihood ratios
- Pattern discovery (EM)
- Computational gene prediction
- Promoter characterization
- Genome browsers on internet
- Artificial neural nets
- Markov models
- Hidden Markov models
- The Human Genome Project
- DNA computing
- Introduction to UNIX

Teaching Activities

■2006

- Participation in the master *Tecnologie bioinformatiche applicate alla medicina personalizzata* (Genefinding: a primer). Consorzio21/Polaris - parco scientifico e tecnologico della Sardegna. Pula (Italy). [Master, 20h]
- January-March. Participation in the course *Bioinformatica* at Facultat de Ciències de la Salut i de la Vida. Universitat Pompeu Fabra. Barcelona (Spain). [University degree, 60h]

■2005

- Participation in the course *Bioinformatica* at Facultat de Ciències de la Salut i de la Vida. Universitat Pompeu Fabra. Barcelona (Spain). [University degree, 60h]
- Participation in the PhD course *Eines informatiques per a genètica molecular* (Computational Gene Prediction). PhD program in Genetics. Facultat de Biologia. Universitat de Barcelona. Barcelona (Spain). [PhD program, 5h]

- Participation in the summer course *Bioinformatica per a tothom* (Genome analysis). Universitat d'Estiu de la Universitat Rovira i Virgili. Reus (Spain). [Summer course, 10h]
- Participation in the summer course *Bioinformatica* (Computational Gene Prediction). Universidad Complutense de Madrid. Madrid (Spain). [Summer course, 6h]
- Participation in the master *Bioinformatics for health sciences* (Introduction to the UNIX environment). Universitat Pompeu Fabra. Barcelona (Spain). [Master, 10h]

■ 2004

- Participation in the course *Bioinformatica* at Facultat de Ciències de la Salut i de la Vida. Universitat Pompeu Fabra. Barcelona (Spain). [University degree, 60h]
- Participation in the Phd course *Eines informàtiques per a genètica molecular* (Computational Gene Prediction). PhD program in Genetics. Facultat de Biologia. Universitat de Barcelona. Barcelona (Spain). [PhD program, 5h]
- Participation in the summer course *Bioinformatica* (Computational Gene Prediction). Universidad Complutense de Madrid. Madrid (Spain). [Summer course, 5h]
- Participation in the master *Bioinformatics for health sciences* (Introduction to the UNIX environment). Universitat Pompeu Fabra. Barcelona (Spain). [Master, 10h]
- Participation in the workshop on *Computational genome analysis* at Cosmocaixa, Fundació La Caixa. Barcelona (Spain). [Workshop, 4h]
- Participation in the *Postgraduate programme in Bioinformatics* (Computational Gene Prediction). Universidade de Lisboa / Gulbenkian Institute. Lisbon (Portugal). [Master, 40h]

■ 2003

- Participation in the course *Bioinformatica* at Facultat de Ciències de la Salut i de la Vida. Universitat Pompeu Fabra. Barcelona (Spain). [University degree, 60h]
- Participation in the Phd course *Eines informàtiques per a genètica molecular* (Computational Gene Prediction). PhD program in Genetics. Facultat de Biologia. Universitat de Barcelona. Barcelona (Spain). [PhD program, 5h]
- Participation in the master *Bioinformatica y biología computacional* (Computational Gene Prediction). Universidad Complutense de Madrid. Madrid (Spain). [Master, 4h]

■ 2002

- Participation in the course *Bioinformatica* at Facultat de Ciències de la Salut i de la Vida. Universitat Pompeu Fabra. Barcelona (Spain). [University degree, 60h]

- Participation in the course *Bioinformatica* (Genome analysis) at ALMA bioinformatics. Madrid (Spain). [Course, 8h]

■2001

- Participation in the EMBL course *Bioinformatics for comparative and functional genomics* (Computational analysis of promoter regions). Universitat Pompeu Fabra. Barcelona (Spain). [Course, 2h]

■2000

- Participation in the EMB-net course *Bioinformatics* (Computational gene identification). Gulbenkian Institute. Lisbon (Portugal). [Course, 20h]

Attended conferences

- Cold Spring Harbor Labs: global regulation of gene expression. (March 2005, New York, USA).
- Cold Spring Harbor Labs: genomic approaches to transcriptional regulation. (March 2004, New York, USA).
- IV Jornadas de Bioinformática Españolas (September 2003, A Coruña, Spain).
- Seventh annual international conference on computational biology-RECOMB. (April 2003, Berlin, Germany).
- Workshop sobre bioinformatica y biologia computacional. Fundacion BBVA. (April 2002, Madrid, Spain).
- 9-th international conference on Intelligent Systems in Molecular Biology. (July 2001, Copenhagen, Denmark).
- I Jornadas de Bioinformática Españolas (June 2000, Cartagena, Spain).
- Jornada Catalana de Supercomputación. Parque tecnológico de la Universidad de Barcelona (October 1999, Barcelona).
- Segunda jornada científica sobre análisis computacional de biomoléculas. IMIM-UPF (October 1999, Barcelona).

Software

TF-map alignments

- ➔ **Programs:** <http://genome.imim.es/software/meta/index.html>
- ➔ **Web server:** <http://genome.imim.es/software/meta/meta.html>
- ➔ **Datasets:** <http://genome.imim.es/datasets/meta2005/index.html>

Multiple TF-map alignments

- ➔ **Programs:** <http://genome.imim.es/software/mmeta/index.html>
- ➔ **Web server:** <http://genome.imim.es/software/mmeta/mmeta.html>
- ➔ **Datasets:** <http://genome.imim.es/datasets/mmeta2006/index.html>

The ABS database of annotated promoters

- ➔ **Data:** <http://genome.imim.es/datasets/abs2005/index.html>
- ➔ **Constructor:**
<http://genome.imim.es/datasets/abs2005/constructor.html>
- ➔ **Evaluator:**
<http://genome.imim.es/datasets/abs2005/evaluator.html>

The geneid program

- **Program:** <http://genome.imim.es/software/geneid/index.html>
- **Web server:** <http://genome.imim.es/software/geneid/geneid.html>
- **Annotations:** <http://genome.imim.es/genepredictions/index.html>

List of Publications

Papers



E. Blanco, X. Messeguer, T.F. Smith and R. Guigó.
“Transcription factor map alignment of promoter regions.”
PLoS Computational Biology, 2: e49:403–416, 2006.



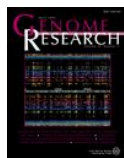
E. Blanco, D. Farré, M. Albà, X. Messeguer and R. Guigó.
“ABS: a database of Annotated regulatory Binding Sites from orthologous promoters.”
Nucleic Acids Research, 34:D63–D67, 2006.



S. Castellano, S.V. Novoselov, G.V. Kryukov, A. Lescure,
E. Blanco, A. Krol. V.N. Gladyshev and R. Guigó.
“Reconsidering the evolution of eukaryotic selenoproteins:
a novel non-mammalian family with scattered phylogenetic
distribution.”
EMBO Reports, 5:71–77, 2004.

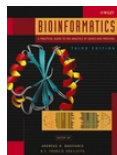


S. Beltran, E. Blanco, F. Serras, B. Perez-Villamil, R. Guigó,
S. Artavanis-Tsakonas and M. Corominas.
“Transcriptional network controlled by the trithorax-group
gene *ash2* in *Drosophila melanogaster*.”
Proc. Nat. Acad. Sci., 100:3293–3298, 2003.



G. Parra, E. Blanco and R. Guigó.
 “Geneid in Drosophila.”
Genome Research, 10:511–515, 2000.

Book Chapters



E. Blanco and R. Guigó.
 “Predictive Methods Using DNA Sequences.”
 In A. D. Baxeavanis and B. F. Francis Ouellette, chief editors:
**Bioinformatics: A Practical Guide to the Analysis of Genes
 and Proteins, Third Edition.**
 John Wiley & Sons Inc., New York, 2005. ISBN: 0–471–47878–4.



E. Blanco, G. Parra and R. Guigó.
 “Using geneid to Identify Genes.”
 In A. D. Baxeavanis and D. B. Davison, chief editors:
Current Protocols in Bioinformatics. Volume 1.
 John Wiley & Sons Inc., New York, 2002. ISBN: 0–471–25093–7.

Posters

E. Blanco, M. Pigantelli, X. Messeguer and R. Guigó.
 “Deconstructing the position weight matrices to detect regulatory elements.”
 Global regulation of gene expression, Cold Spring Harbor, USA (2005)

E. Blanco, X. Messeguer and R. Guigó.
 “Novel computational methods to chracterize regulatory regions.”
 Genomic approaches to transcriptional regulation, Cold Spring Harbor, USA (2004)

E. Blanco, X. Messeguer and R. Guigó.
 “Alignment of promoter regions by mapping nucleotide sequences into arrays
 of transcription factor binding motifs.”
 VIIth RECOMB, Berlin, Germany (2003)

E. Blanco, G. Parra, S. Castellano, J.F. Abril, M. Burset,
 X. Fustero, X. Messeguer and R. Guigó.
 “Gene Prediction in the Post-Genomic Era.”
 IXth ISMB, Copenhagen, Denmark (2001)

J.F. Abril, M. Albà, E. Blanco, M. Burset, F. Câmara, S. Castellano,
 R. Castelo, O. Gonzalez, G. Parra and R. Guigó.
 “Understanding the Eukaryotic Genome Sequence.”
 Inaugural Symposium of the Center for Genomic Regulation, Barcelona, Spain (2002)

E. Blanco, G. Parra, S. Castellano, J.F. Abril, M. Burset, X. Fustero,
X. Messeguer and R. Guigó.

“Gene Prediction in the Post-Genomic Era.”

IXth ISMB, Copenhagen, Denmark (2001)

J.F. Abril, E. Blanco, M. Burset, S. Castellano, X. Fustero, G. Parra and R. Guigó.

“Genome Informatics Research Laboratory: Main Research Topics.”

Ist Jornadas de Bioinformática, Cartagena, Spain (2000)

Publications

 Blanco *et al.*, PLoS Comput Biol 2(5): e49, 2006

Transcription Factor Map Alignment of Promoter Regions

Enrique Blanco^{1,2}, Xavier Messeguer², Temple F. Smith³, Roderic Guigó^{1,4*}

1 Research Group in Biomedical Informatics, Institut Municipal d'Investigació Mèdica/Universitat Pompeu Fabra, Barcelona, Catalonia, Spain, **2** Grup d'Algorismica i Genètica, Departament de Llenguatges i Sistemes Informàtics, Universitat Politècnica de Catalunya, Barcelona, Catalonia, Spain, **3** Biomolecular Engineering Research Center, Boston University, Boston, Massachusetts, United States of America, **4** Bioinformatics and Genomics Program, Centre de Regulació Genòmica, Barcelona, Catalonia, Spain

We address the problem of comparing and characterizing the promoter regions of genes with similar expression patterns. This remains a challenging problem in sequence analysis, because often the promoter regions of co-expressed genes do not show discernible sequence conservation. In our approach, thus, we have not directly compared the nucleotide sequence of promoters. Instead, we have obtained predictions of transcription factor binding sites, annotated the predicted sites with the labels of the corresponding binding factors, and aligned the resulting sequences of labels—to which we refer here as transcription factor maps (TF-maps). To obtain the global pairwise alignment of two TF-maps, we have adapted an algorithm initially developed to align restriction enzyme maps. We have optimized the parameters of the algorithm in a small, but well-curated, collection of human–mouse orthologous gene pairs. Results in this dataset, as well as in an independent much larger dataset from the CISRED database, indicate that TF-map alignments are able to uncover conserved regulatory elements, which cannot be detected by the typical sequence alignments.

Citation: Blanco E, Messeguer X, Smith TF, Guigó R (2006) Transcription factor map alignment of promoter regions. *PLoS Comput Biol* 2(5): e49. DOI: 10.1371/journal.pcbi.0020049

Introduction

Sequence comparisons are among the most useful computational techniques in molecular biology. Sequences of characters in the four-letter nucleotide alphabet and in the 20-letter amino acid alphabet are extremely good symbolic representations of the underlying DNA and protein molecules, and encode substantial information on their structure, function, and history.

Primary sequence comparisons, however, have limitations. Although similar sequences do tend to play similar functions, the opposite is not necessarily true. Often similar functions are encoded in higher order sequence elements—such as, for instance, structural motifs in amino acid sequences—and the relation between these and the underlying primary sequence may not be univocal. As a result, similar functions are frequently encoded by diverse sequences.

Promoter regions controlling eukaryotic gene expression are a case in point. The information for the control of the initiation of the RNA synthesis by the RNA polymerase II is mostly contained in the gene promoter, a region usually 200 to 2,000 nucleotides long upstream of the transcription start site (TSS) of the gene. Transcription factors (TFs) interact in these regions with sequence-specific elements or motifs (the TF binding sites (TFBSs)). TFBSs are typically 5–8 nucleotides long, and one promoter region usually contains many of them to harbor different TFs [1]. The interplay between these factors is not well understood, but the motifs appear to be arranged in specific configurations that confer on each gene an individualized spatial and temporal transcription program [1]. It is assumed, in consequence, that genes exhibiting similar expression patterns would also share similar configurations of TFs in their promoter.

However, TFBSs associated to the same TF are known to tolerate sequence substitutions without losing functionality,

and are often not conserved. Consequently, promoter regions of genes with similar expression patterns may not show sequence similarity, even though they may be regulated by similar configurations of TFs. For instance, only about 30% to 40% of the promoter regions are conserved between human and chicken orthologous genes [2], and the conservation of human–mouse orthologous promoter regions is only slightly higher than that observed in intergenic regions [3]. Indeed, despite the recent progress due to the development of techniques based on so-called phylogenetic footprinting [4], lack of nucleotide sequence conservation between functionally related promoter regions may partially explain the still limited success of current available computational methods for promoter characterization (see [5] and [6] for further information).

In the approach described here, we attempt to overcome this limitation by abstracting the nucleotide sequence, and representing a promoter region by a sequence in a new alphabet in which the different symbols denote different TFs. Using an external mapping function, for instance, a look-up table or a collection of position weight matrices (PWMs) that associates each TF to the nucleotide sequence motifs the factor is known to bind, we can translate the nucleotide sequence of the promoter into a sequence in this new

Editor: Philip Bourne, University of California San Diego, United States of America

Received: October 31, 2005; **Accepted:** March 31, 2006; **Published:** May 26, 2006

DOI: 10.1371/journal.pcbi.0020049

Copyright: © 2006 Blanco et al. This is an open-access article distributed under the terms of the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

Abbreviations: PWMs, position weight matrices; TF, transcription factors; TF-maps, transcription factor maps; TFBSs, TF binding sites; TSS, transcription start site

* To whom correspondence should be addressed. E-mail: rguigo@imim.es

 Blanco *et al.*, NAR 34:D63–D67, 2006

Nucleic Acids Research, 2006, Vol. 34, Database issue **D63–D67**
doi:10.1093/nar/gkj116

ABS: a database of Annotated regulatory Binding Sites from orthologous promoters

Enrique Blanco^{1,2,*}, Domènec Farré^{1,2}, M. Mar Albà¹, Xavier Messeguer² and Roderic Guigó¹

¹Grup de Recerca en Informàtica Biomèdica, Institut Municipal d'Investigació Mèdica/Universitat Pompeu Fabra/Centre de Regulació Genòmica, C/Doctor Aiguader 80, 08003 Barcelona, Spain and

²Grup d'algorísmica i genètica, Departament de Llenguatges i Sistemes Informàtics, Universitat Politècnica de Catalunya, C/Jordi Girona 1-3, 08034 Barcelona, Spain

Received August 1, 2005; Revised September 19, 2005; Accepted October 18, 2005

ABSTRACT

Information about the genomic coordinates and the sequence of experimentally identified transcription factor binding sites is found scattered under a variety of diverse formats. The availability of standard collections of such high-quality data is important to design, evaluate and improve novel computational approaches to identify binding motifs on promoter sequences from related genes. ABS (<http://genome.imim.es/datasets/abs2005/index.html>) is a public database of known binding sites identified in promoters of orthologous vertebrate genes that have been manually curated from bibliography. We have annotated 650 experimental binding sites from 68 transcription factors and 100 orthologous target genes in human, mouse, rat or chicken genome sequences. Computational predictions and promoter alignment information are also provided for each entry. A simple and easy-to-use web interface facilitates data retrieval allowing different views of the information. In addition, the release 1.0 of ABS includes a customizable generator of artificial datasets based on the known sites contained in the collection and an evaluation tool to aid during the training and the assessment of motif-finding programs.

INTRODUCTION

Expression of genes is regulated at many different levels, transcription of DNA being one of the most critical stages. Specific configurations of transcription factors (TFs) that interact with gene promoter regions are recruited to activate or modulate the production of a given transcript. Many of these TFs possess the ability to recognize a small set of genomic sequence footprints called TF-binding sites (TFBSs). These

motifs are typically 6–15 bp long and in some cases, they show a high degree of variability. In addition, many motifs may ambiguously be recognized by members of different TF families. Because of these flexible binding rules, computational methods for the identification of regulatory elements in a promoter sequence tend to produce an overwhelming amount of false positives. However, the identification of conserved regulatory elements present in orthologous gene promoters (also called phylogenetic footprinting) has proved to be more effective to characterize such sequences (1–3). In fact, the ever-growing availability of more genomes and the constant improvement of bioinformatics algorithms hold great promise for unveiling the overall network of gene interactions of each organism (4).

Typically, computational methods to detect regulatory elements use their own training set of experimental annotated TFBSs. These annotations are usually collected from bibliography or from general repositories of gene regulation information, such as JASPAR (5) and TRANSFAC (6). However, each program establishes different criteria and formats to retrieve and display the data that forms the final training set, which makes the comparison between different methods very difficult. The construction of a good benchmark to evaluate the accuracy of several pattern discovery methods is therefore not a trivial procedure (7).

Although important efforts are being carried out to standardize the construction of collections of promoter regions (8) or the presentation of experimental data (9), there is a clear necessity to provide stable and common datasets for future algorithmic developments. In this direction, we present here the release 1.0 of the ABS database constructed from literature annotations that have been experimentally verified in human, mouse, rat or chicken.

DATABASE CONSTRUCTION

We have gathered from the literature a collection of experimentally validated binding sites that are conserved in at least

*To whom correspondence should be addressed. Tel: +34 93 2240891; Fax: +34 93 2240875; Email: eblanco@imim.es

© The Author 2006. Published by Oxford University Press. All rights reserved.

The online version of this article has been published under an open access model. Users are entitled to use, reproduce, disseminate, or display the open access version of this article for non-commercial purposes provided that: the original authorship is properly and fully attributed; the Journal and Oxford University Press are attributed as the original place of publication with the correct citation details given; if an article is subsequently reproduced or disseminated not in its entirety but only in part or as a derivative work this must be clearly indicated. For commercial re-use, please contact permissions@oxfordjournals.org



Castellano *et al.*, EMBO Reports 5:71–77, 2004

Reconsidering the evolution of eukaryotic selenoproteins: a novel nonmammalian family with scattered phylogenetic distribution

Sergi Castellano¹, Sergey V. Novoselov², Gregory V. Kryukov², Alain Lescure³, Enrique Blanco¹, Alain Krol³, Vadim N. Gladyshev² & Roderic Guigó^{1,4*}

¹Grup de Recerca en Informàtica Biomèdica, Institut Municipal d'Investigació Mèdica, Universitat Pompeu Fabra, Barcelona, Catalonia, Spain, ²Department of Biochemistry, University of Nebraska, Lincoln, Nebraska, USA, ³UPR 9002 du CNRS, Institut de Biologie Moléculaire et Cellulaire, Strasbourg, France, and ⁴Programa de Bioinformàtica i Genòmica, Centre de Regulació Genòmica, Barcelona, Catalonia, Spain

While the genome sequence and gene content are available for an increasing number of organisms, eukaryotic selenoproteins remain poorly characterized. The dual role of the UGA codon confounds the identification of novel selenoprotein genes. Here, we describe a comparative genomics approach that relies on the genome-wide prediction of genes with in-frame TGA codons, and the subsequent comparison of predictions from different genomes, wherein conservation in regions flanking the TGA codon suggests selenocysteine coding function. Application of this method to human and fugu genomes identified a novel selenoprotein family, named SelU, in the puffer fish. The selenocysteine-containing form also occurred in other fish, chicken, sea urchin, green algae and diatoms. In contrast, mammals, worms and land plants contained cysteine homologues. We demonstrated selenium incorporation into chicken SelU and characterized the SelU expression pattern in zebrafish embryos. Our data indicate a scattered evolutionary distribution of selenoproteins in eukaryotes, and suggest that, contrary to the picture emerging from data available so far, other taxa-specific selenoproteins probably exist.

EMBO reports (2004) 5, 71–77. doi:10.1038/sj.embor.7400036

INTRODUCTION

Selenium is a micronutrient found in proteins in the eubacterial, archaeal and eukaryotic domains of life. It is present in selenoproteins in the form of selenocysteine (Sec), the 21st amino acid. Sec is inserted co-translationally in response to UGA codons, a stop signal in the canonical genetic code. The alternative decoding of UGA depends on several *cis*- and *trans*-acting factors. In eukaryotes, the main *cis*-factor is an mRNA element, the selenocysteine insertion sequence (SECIS), located in the 3'UTR of selenoprotein genes (Walczak *et al*, 1998; Grundner-Culemann *et al*, 1999). About 25 Sec-containing proteins have been identified in eukaryotes (Kryukov *et al*, 2003), but distribution among taxa varies greatly. For instance, no selenoproteins have been found in yeast and land plants, only one in worms and three in flies. The majority of selenoproteins have homologues in which Sec is replaced by cysteine (Cys), even in genomes lacking the Sec-containing gene.

Because of the dual role of the UGA codon, identification of novel selenoproteins in eukaryotes is very difficult. The more direct approach is to search for occurrences of the SECIS structural pattern. Although this approach has been successfully applied in expressed sequence tag (EST) and other cDNA sequences (Kryukov *et al*, 1999; Lescure *et al*, 1999), the low specificity of SECIS searches produces a large number of predictions when applied to eukaryotic genomes. Thus, for the analysis of *Drosophila melanogaster* (Castellano *et al*, 2001; Martin-Romero *et al*, 2001), we devised a strategy that coordinated SECIS identification with prediction of genes with in-frame TGA codons. Again, while this strategy efficiently identified novel selenoproteins in the fly, it resulted in a large number of potential selenoprotein candidates when applied to larger and more complex vertebrate genomes.

Here, we describe a comparative genomics strategy to target bona fide selenoproteins in such complex genomes. Underlying comparative genome methods is the assumption that conservation

¹Grup de Recerca en Informàtica Biomèdica, Institut Municipal d'Investigació Mèdica, Universitat Pompeu Fabra, Dr. Aiguader 80, 08003 Barcelona, Catalonia, Spain

²Department of Biochemistry, University of Nebraska, Lincoln, Nebraska 68588, USA
³UPR 9002 du CNRS, Institut de Biologie Moléculaire et Cellulaire, 15 Rue René Descartes, 67084 Strasbourg Cedex, France

⁴Programa de Bioinformàtica i Genòmica, Centre de Regulació Genòmica, Barcelona, Catalonia, Spain

*Corresponding author. Tel: +34 93 224 0877; Fax: +34 93 224 0875; E-mail: rguido@imim.es

Received 28 August 2003; revised 15 October 2003; accepted 15 October 2003; published online 19 December 2003

 Beltran *et al.*, PNAS 100:3293–3298, 2003

Transcriptional network controlled by the trithorax-group gene *ash2* in *Drosophila melanogaster*

Sergi Beltran*, Enrique Blanco†, Florenci Serras*, Beatriz Pérez-Villamil‡, Roderic Guigó†, Spyros Artavanis-Tsakonas‡, and Montserrat Corominas*§

*Departament de Genètica, Universitat de Barcelona, Diagonal 645, 08028 Barcelona, Spain; †Grup de Recerca en Informàtica Biomèdica, Institut Municipal d'Investigació Mèdica, Universitat Pompeu Fabra, Centre de Regulació Genòmica, Dr. Aiguader 80, 08003 Barcelona, Spain; and ‡Massachusetts General Hospital Cancer Center, Harvard Medical School, BI 149, 13th Street, Charlestown, MA 02129

Communicated by Walter J. Gehring, University of Basel, Basel, Switzerland, January 10, 2003 (received for review July 20, 2002)

The transcription factor *absent*, *small*, or *homeotic discs 2* (*ash2*) gene is a member of the trithorax group of positive regulators of homeotic genes. Mutant alleles for *ash2* are larval/pupal lethals and display imaginal disc and brain abnormalities. The allele used in this study is a true mutant for the trithorax function and lacks the longest transcript present in wild-type flies. In an attempt to identify gene targets of *ash2*, we have performed an expression analysis by using cDNA microarrays. Genes involved in cell cycle, cell proliferation, and cell adhesion are among these targets, and some of them are validated by functional and expression studies. Even though trithorax proteins act by modulating chromatin structure at particular chromosomal locations, evidence of physical aggregation of *ash2*-regulated genes has not been found. This work represents the first microarray analysis, to our knowledge, of a trithorax-group gene.

The trithorax group (trx-G) of activators and the Polycomb group (Pc-G) of repressors maintain the correct expression of several key developmental regulators, including the homeotic genes. Pc-G mutants exhibit posterior transformations in embryos and adults caused by derepression of homeotic loci in flies (1) and vertebrates (2). In contrast, proteins of the trx-G are required for the maintenance of activation of homeotic loci (3). Pc-G and trx-G proteins function in distinct multiprotein complexes that are believed to control transcription by changing the structure of chromatin, organizing it into either a "closed" or an "open" conformation (ref. 4 and references therein). It is thought that Pc-G and trx-G regulate many targets in addition to homeotic genes, indicating that epigenetic maintenance of activated or repressed states might represent a fundamental developmental mechanism (5).

The *ash2* (*absent*, *small*, or *homeotic discs 2*) gene is a member of the trx-G discovered, together with *ash1*, in a screen for late larval/early pupal lethals that had imaginal discs abnormalities (6–9). The ASH2 protein has a proline-, glutamic acid-, serine-, and threonine-rich region sequence characteristic of short-lived proteins, a putative double zinc-finger domain, a bipartite nuclear localization signal, and a SPRY domain (10). Biochemical studies have shown that ASH1 and ASH2 are subunits of distinct protein complexes and that ASH2 elutes in fractions with an apparent native molecular mass of 500 kDa (11). More recently it has been reported that the *Saccharomyces cerevisiae* SET1 complex includes two putative ASH2 homologues as well as a protein (SET1) with high similarity to TRX. This complex methylates histone 3 lysine 4, reinforcing the notion that methylation is important for regulating the transcriptional accessibility of chromatin (12–14).

Mutations in *ash2* cause the homeotic transformations expected for genes in this group in addition to a variety of additional pattern formation defects. *ash2* mutant hemizygotes that are able to survive until eclosion include supernumerary legs, duplication of thoracic bristles, and transformation of

campaniform sensilla to bristles (15). The line *l(3)112411* was isolated from a collection of *P-lacW* element insertional mutagenesis in the third chromosome (16) and corresponds to a new *ash2* allele. The few homozygous flies that reach the adult stage are sterile and display anomalous patterns of appendage differentiation. Clonal analysis in adult wings of homozygous cells for the stronger allele *ash2¹¹* reveals a role in vein–intervein patterning, because a reduction of intervein tissue and an increase of vein tissue are observed autonomously and nonautonomously in the clones (17). Moreover, a failure to form joints or fusion of several fragments leads to shortened legs when big clones are generated. Taken together, the pleiotropic phenotypes observed could not be explained only by changes in homeotic gene expression; therefore, more genes should be responding to the loss of *ash2* function.

In this work, we have applied cDNA microarray technology to analyze the transcription profile of *ash2¹¹* mutant larvae in comparison with WT, in an attempt to delineate the transcriptional consequences of lack of *ash2* function and to identify genes that may fulfill the criteria of *ash2* targets. Microarrays have been used to study a variety of biological processes, from differential gene expression in yeast sporulation (18) to human tumors (19). In the case of *Drosophila*, they were initially applied to analyze development during metamorphosis (20) and more recently for analyzing patterns of transcription under different situations or mutant conditions (21–26). The microarray analysis presented here represents the first approach, to our knowledge, to monitoring the genome wide-expression profile from a mutant of the trx-G. The regulated genes have been automatically classified and clustered according to the functional criteria in the Gene Ontology (GO) database (27), with the aim of finding a differential distribution among the regulated genes.

Materials and Methods

Canton-S and *ash2¹¹/TM6C* strains were maintained on standard medium and experiments performed at 25°C. Details of mitotic clone generation, 5'-rapid amplification of cDNA ends, Northern blot, and RT-PCR are published as *Supporting Materials and Methods* on the PNAS web site, www.pnas.org.

Microarray Analysis. One to three micrograms of poly(A) RNA from WT or mutant larvae were labeled by reverse transcription incorporation of Amino-allyl dUTP and coupling to cyanine dye (Cy3- or Cy5-NHS esters, Amersham Biosciences) and hybridized to cDNA microarrays constructed by using PCR products directly amplified from the DROSOPHILA gene collection 1.0 (www.fruitfly.org/dgc/index.html). GENEPIX 3.0 (Axon Instru-

Abbreviations: GO, Gene Ontology; SAM, significance analysis of microarrays; MF, molecular function; BP, biological process; CC, cellular component; trx-G, trithorax group; UBX, ultrabithorax; FLP-FRT, flipase–flipase recombination target.

§To whom correspondence should be addressed. E-mail: mcorminas@ub.edu.

 Parra *et al.*, GenRes 10:511–515, 2000

Methods

GeneID in *Drosophila*

Genís Parra, Enrique Blanco, and Roderic Guigó¹

Grup de Recerca en Informàtica Mèdica, Institut Municipal d'Investigació Mèdica (IMIM), Universitat Pompeu Fabra, E-08003 Barcelona, Spain

GeneID is a program to predict genes in anonymous genomic sequences designed with a hierarchical structure. In the first step, splice sites, and start and stop codons are predicted and scored along the sequence using position weight matrices (PWMs). In the second step, exons are built from the sites. Exons are scored as the sum of the scores of the defining sites, plus the log-likelihood ratio of a Markov model for coding DNA. In the last step, from the set of predicted exons, the gene structure is assembled, maximizing the sum of the scores of the assembled exons. In this paper we describe the obtention of PWMs for sites, and the Markov model of coding DNA in *Drosophila melanogaster*. We also compare other models of coding DNA with the Markov model. Finally, we present and discuss the results obtained when GeneID is used to predict genes in the *Adh* region. These results show that the accuracy of GeneID predictions compares currently with that of other existing tools but that GeneID is likely to be more efficient in terms of speed and memory usage. GeneID is available at <http://www.imim.es/~eblanco/Genelid>.

GeneID (Guigó et al. 1992) was one of the first programs to predict full exonic structures of vertebrate genes in anonymous DNA sequences. GeneID was designed with a hierarchical structure: First, gene-defining signals (splice sites and start and stop codons) were predicted along the query DNA sequence. Next, potential exons were constructed from these sites, and finally the optimal scoring gene prediction was assembled from the exons. In the original GeneID the scoring function to optimize was rather heuristic: The sequence sites were predicted and scored using position weight matrices (PWMs), a number of coding statistics were computed on the predicted exons, and each exon was scored as a function of the scores of the exon defining sites and of the coding statistics. To estimate the coefficients of this function a neural network was used. An exhaustive search of the space of possible gene assemblies was performed to rank predicted genes according with an score obtained through a complex function of the scores of the assembled exons.

During recent years GeneID had some usage, mostly through a now nonfunctional e-mail server at Boston University (geneid@darwin.bu.edu) and through a WWW server at the IMIM (<http://www1.imim.es/geneid.html>). During this period, however, there have been substantial developments in the field of computational gene identification (for recent reviews, see Claverie 1997; Burge and Karlin 1998; Haussler 1998), and the original GeneID has become clearly inferior to other existing tools. Therefore, some time ago we began developing an improved version of the GeneID program, which is at least as accurate as

other existing tools but much more efficient at handling very large genomic sequences, both in terms of speed and usage of memory. This new version maintains the hierarchical structure (signal to exon to gene) in the original GeneID, but we have simplified the scoring schema and furnished it with a probabilistic meaning: Scores for both exon-defining signals and protein-coding potential are computed as log-likelihood ratios, which for a given predicted exon are summed up into the exon score, in consequence also a log-likelihood ratio. Then, a dynamic programming algorithm (Guigó 1998) is used to search the space of predicted exons to assemble the gene structure (in the general case, multiple genes in both strands) maximizing the sum of the scores of the assembled exons, which can also be assumed to be a log-likelihood ratio. Execution time in this new version of GeneID grows linearly with the size of the input sequence, currently at ~2 Mb per minute in a Pentium III (500 MHz) running linux. The amount of memory required is also proportional to the length of the sequence, ~1 megabyte (MB)/Mb plus a constant amount of ~15 MB, irrespective of the length of the sequence. Thus, GeneID is able to analyze sequences of virtually any length, for instance, chromosome size sequences.

In this paper we describe the "training" of GeneID to predict genes in the genome of *Drosophila melanogaster*. In the context of GeneID training means essentially computing PWMs for splice sites and start codons, and deriving a model of coding DNA, which, in this case, is a Markov model of order 5, similar to the models introduced by Borodovsky and McIninch (1993). Therefore, in the following sections, we describe the training data set used, particularly our attempt to recreate a more realistic scenario to train and test GeneID by generating semiartificial large genomic

¹Corresponding author.
E-MAIL rguigo@imim.es; FAX 34-93-221-3237.

 Blanco and Guigó, in Baxevanis and Ouellette,
2005

CHAPTER FIVE

Predictive Methods using DNA Sequences

ENRIQUE BLANCO

RODERIC GUIGÓ

5.1 Introduction	116
5.2 Gene Prediction Methods	117
5.3 Gene Prediction Programs	120
5.4 How Well Do the Methods Work?	126
5.5 Promoter Analysis: Characterization and Prediction	128
5.6 Strategies and Considerations	132
5.7 Visualization and Integration Tools	135
BOX 5.1 Markov Models	118
BOX 5.2 Hidden Markov Models in Gene Prediction	119
BOX 5.3 Discriminant Analysis in Gene Prediction	124

Bioinformatics: A Practical Guide to the Analysis of Genes and Proteins, Third Edition, edited by
Andreas D. Baxevasis and B.E Francis Ouellette.
ISBN 0-471-47878-4 Copyright © 2005 John Wiley & Sons, Inc.

 Blanco *et al.*, in Baxevanis *et al.*, 2002

Using geneid to Identify Genes

UNIT 4.3

The gene-prediction program *geneid* is based on a simple hierarchical design: (1) search splicing signals, start codons, and stop codons, (2) build and score candidate exons, and (3) assemble genes (Guigó et al., 1992; Parra et al., 2000). An early version of *geneid* was available as an E-mail server in 1991. In 1999 a completely rewritten version was released (*geneid* v1.0). This version, while having an accuracy comparable to the most accurate gene-prediction programs, is very efficient at handling large genomic sequences in terms of memory and speed. *geneid* is able to analyze chromosome-size sequences in a few minutes on a standard workstation, and has a rich set of output options, which allow for a detailed analysis of gene features in genomic sequences. Both a Web server interface and a stand-alone distribution exist. A new, more powerful version of *geneid* (*geneid* v1.1) with parameter configurations for a larger number of species was released in May 2002.

This unit describes how to use the *geneid* Unix application to predict genes along genomic sequences (see Basic Protocol 1). These can be multiple genes on both strands of large genome sequences, or partial genes or exon signals in small genomic fragments. Basic Protocol 1 describes the default behavior of *geneid*, and introduces the basic options for configuring its output. Next, options for visualizing the output are described (see Basic Protocol 2). A third protocol describes how to use *geneid* together with experimental evidence (or evidence coming from other sources) to reannotate sequences whose genomic features have been partially annotated (see Basic Protocol 3). Use of the Web server version of *geneid* is described in the Alternate Protocol. The Support Protocol describes how to download the *geneid* software, which is in the public domain under a GNU-GPL license (<http://www.gnu.org/>). Complete, up-to-date documentation is provided with the *geneid* distribution, and can also be accessed through the *geneid* Web page (see Support Protocol).

USING THE *geneid* UNIX APPLICATION TO PREDICT GENES

BASIC PROTOCOL 1

geneid can be used in two different ways: via a Web server (see Alternate Protocol), or as a Unix application. The best way to take full advantage of the different options available in *geneid* is by running the stand-alone program on a Unix workstation.

In both cases, the user provides an input DNA sequence as a FASTA file (APPENDIX 1B), and selects a suitable model of parameters depending on the species (or taxonomic group) from which the sequence originates. A number of options are available to configure *geneid* actions and output. Although this option is not directly available in the stand-alone Unix application, *geneid* output can be directly plugged into a number of publicly available visualization tools (see Basic Protocol 2).

This protocol describes the use of *geneid* as a stand-alone Unix application. For use of the *geneid* Web server as an alternative, see Alternate Protocol.

Necessary Resources

Hardware

Unix/Linux workstation with at least 256 Mb RAM (recommended)

Software

geneid v1.1 full distribution (see Support Protocol)

Finding Genes

4.3.1

Posters

 Blanco *et al.*, Cold Spring Harbor, 2005

 Blanco *et al.*, Cold Spring Harbor, 2004

 Blanco *et al.*, RECOMB, 2003

Alignment of Promoter Regions by Mapping Nucleotide Sequences into Arrays of Transcription Factor Binding Motifs

BLANCO, E.^{1,2,*}

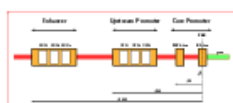
¹ Grupo de Procesos de Información Biomédica
IDIBI - Universidad Pompeu Fabra -
Carles de Calvo de Guzmán 1
Barcelona (08035)

MESSEGUER, X.¹

² Grupo de Algoritmos y Gestión de
Experimentos de Laboratorio Biomédico
Laboratorio de Biología de Celular y
Biomoléculas (IDIBI)

GARCÍA, R.¹

1. Promoter regions



The first step is to identify the TATA box in the promoter region. This is done by searching for the sequence TATA in the promoter region. The TATA box is a sequence of approximately 25 bp, the Initiator is approximately 10 bp, and the Core Promoter is approximately 100 bp.

- From our TFs:
- Location of the TATA box in the promoter region
 - The TATA box is a sequence of approximately 25 bp, the Initiator is approximately 10 bp, and the Core Promoter is approximately 100 bp.
 - The TATA box is a sequence of approximately 25 bp, the Initiator is approximately 10 bp, and the Core Promoter is approximately 100 bp.
 - The TATA box is a sequence of approximately 25 bp, the Initiator is approximately 10 bp, and the Core Promoter is approximately 100 bp.

Summary

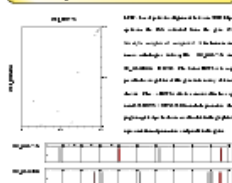
We address the problem of comparing promoter regions from genes with similar expression patterns (e.g. homologous genes). Similarly in gene expression data, the relative position of the promoter regions is not reflected in the sequence similarity in promoter regions which partially nulls the alignment of the promoter regions. This paper presents a new method for promoter prediction.

In the approach described here (based on an existing algorithm to align nucleotide sequences), we attempt to overcome such a limitation representing a promoter region as a sequence in a new alphabet in which the different symbols denote different Transcription Factors. Sequences in this new alphabet can be aligned. If the scoring model takes into account not only the presence/absence of a given symbol but its relative position on the putative sequence, then the optimal global alignment between the promoter regions of two similarly expressed genes will reflect the underlying common configuration of binding motifs.

The ability of this new method to reduce the noise produced by the large number of false positives is shown in a real case involving the promoter regions of two homologous genes.

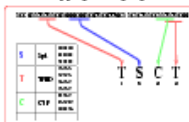
2. Comparative approaches

Given the same regulatory path, it is possible to generate different sets of genes. This is the case of the same regulatory path, but with different sets of genes. This is the case of the same regulatory path, but with different sets of genes.



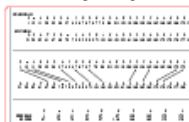
3. A new strategy: meta-alignments

- Mapping of languages -



Based on the mapping of languages, we can generate a new set of genes. This is the case of the same regulatory path, but with different sets of genes. This is the case of the same regulatory path, but with different sets of genes.

- A simple example -



Based on the mapping of languages, we can generate a new set of genes. This is the case of the same regulatory path, but with different sets of genes. This is the case of the same regulatory path, but with different sets of genes.

4. Algorithm and parameters

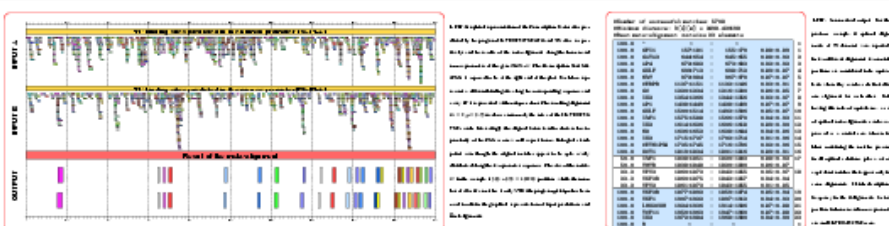
- Dynamic programming -

Dynamic programming is a method for solving complex problems by breaking them down into simpler subproblems. It is a method for solving complex problems by breaking them down into simpler subproblems. It is a method for solving complex problems by breaking them down into simpler subproblems.

- Alignment parameters -

Alignment parameters are used to control the alignment process. They include the gap penalty, the mismatch penalty, and the match bonus. They are used to control the alignment process. They include the gap penalty, the mismatch penalty, and the match bonus.

5. Using the meta-alignments to reduce the noise in a real example



Extensions

- Implementation of the algorithm in a more efficient way
- Extension of the algorithm to other types of sequences
- Extension of the algorithm to other types of sequences
- Extension of the algorithm to other types of sequences
- Extension of the algorithm to other types of sequences

The algorithm is implemented in a more efficient way. It is implemented in a more efficient way. It is implemented in a more efficient way.

References

- [1] Messegue, X., et al. (2004) A new method for promoter prediction. *Bioinformatics*, 20(12), 1551-1560.
- [2] Messegue, X., et al. (2005) A new method for promoter prediction. *Bioinformatics*, 21(12), 1351-1360.
- [3] Messegue, X., et al. (2006) A new method for promoter prediction. *Bioinformatics*, 22(12), 1451-1460.
- [4] Messegue, X., et al. (2007) A new method for promoter prediction. *Bioinformatics*, 23(12), 1551-1560.
- [5] Messegue, X., et al. (2008) A new method for promoter prediction. *Bioinformatics*, 24(12), 1651-1660.

Acknowledgements

This work was supported by the Spanish Ministry of Health and Consumer Affairs. The authors thank the Spanish Ministry of Health and Consumer Affairs for their support. The authors thank the Spanish Ministry of Health and Consumer Affairs for their support.



Blanco *et al.*, ISMB, 2001

Miscellanea

This thesis layout is largely derived from the \LaTeX template created by Robert Castelo in 2002¹. His templates were extended by Sergi Castellano and Genís Parra for their theses. Josep Francesc Abril substantially improved those files, creating an excellent automatical framework that produces a variety of different formats and layouts. Here, I provide some comments on his version and the modifications I incorporated to, and the source code for download.

Technical comments

This book was typeset with GNU `emacs` 21.3.1 in \LaTeX mode and converted to PDF with `pdflatex` 3.14159-1.10b (Web2C 7.4.5). All running on a linux box with Red Hat Fedora Core 2 and kernel 2.6.9-1.6. \LaTeX is a document preparation system, powerful, robust and able to achieve professional results (Lamport, 1994). However, the learning curve may be stiff.

The main document, `thesis.tex`, depends on several \LaTeX files—including each chapter, the tables and few POSTSCRIPT figures—, but it also depends on other files—such as style files, hacked \LaTeX packages, several bitmaps and the PDF files for the attached papers. Furthermore, `pdflatex` had to be run several times, together with `BIB \LaTeX` (to produce the bibliography chapter), `makeindex` (to build the index and the web glossary), `thumbpdf` (to generate the main PDF document thumbnails), and few `perl` scripts. A `Makefile` was written to automatize the compilation process of the whole document. In fact, the `Makefile` was extended to produce four versions of the main document. The “*draft*” version does not include figures and the PDF files for the papers, displaying crop marks and boxes around several elements (such as the area reserved for the pictures). The “*proofs*”, where everything is included but crop marks and boxes are kept, and different hyperlink types use different colors. The “*pdf*” version is the electronic version in which all the hyperlinks are marked in blue color, crop marks are disabled. Finally, the “*press*” version is very similar to the “*pdf*” one, currently the only difference is that all the hyperlinks are

¹R. Castelo, April 2002.

”The Discrete Acyclic Digraph Markov Model in Data Mining”
Faculteit Wiskunde en Informatica, Universiteit Utrecht

black. The Makefile also includes a rule to build the final book “cover”, which recycles the `abstract.tex` file and takes some customization from the same style file as the main `thesis.tex` file.

The compilation of a complete version of this document takes about 600 seconds—of course, the “*draft*” version takes much less—with an AMD Athlon 64 processor 3200+, with 512KB of RAM. This is mainly due to the several steps required to ensure that every reference, index and so on, is in place. The basic build series of commands is the following: an initial `pdflatex`, a `BIBTEX` run to produce the bibliography, a second run of `pdflatex` to include it, one call to `makeindex` (for the Web Glossary), a third run of `pdflatex` to include the glossary, another call to `makeindex` (to generate the final index) and to `pdflatex`, then `makeindex` and `pdflatex` are run again, an extra run of `pdflatex` is followed by `thumpdf`, and a final `pdflatex` to obtain the finished document. If any problem was found, like missing references, an extra round of `pdflatex`, `BIBTEX` and `pdflatex` is performed by the Makefile.

Here you can find the version of some of the programs refereed above: `BIBTEX` version 0.99c (Web2C 7.4.5), `thumpdf` version 3.2 (2002/05/26), and `makeindex` version 2.14 (2002/10/02).

L^AT_EX Packages

As there are four versions of the document, the `ifthen` package was used to define version specific parameters, as well as to include different files. The package `geometry` facilitates the definition of the page layout. The current document original dimensions for both, the electronic and printed versions, are 170 mm width by 240 mm height. The “cover” requires `calc` to calculate automatically the total width for the page layout, which includes the front and the back covers and the spine width. The main document basic font size is the default value for the “book” document class, 10 pt.

The `crop` package is useful to define the trimming marks for the “*draft*” and “*proofs*” versions of this document. It distinguishes between the logical page, the page sizes defined by the user, and the physical page, the page size for the hardcopy. The `layout` package is used in the “*draft*” version to show on the first page the `TEX` variable settings controlling the page layout. Another useful package has been `nextpage`, which provides additional “`clear...page`” commands that ensure to get empty even pages at the end of chapters—and of course, to ensure that all chapters begin at odd pages—, even with automatically generated sections like the Bibliography and the Index.

The `babel` package provides a set of options that allow the user to choose the language(s) in which the document will be typeset, for instance language-specific hyphenation patterns. The default language was set to “english”, while “catalan” and “spanish” were also loaded for using them for the corresponding translations of the ABSTRACT.

When working with `pdflatex` there are three unvaluable packages: `pdfpages`, which makes it easy to embed external PDF documents, such as the attached publications; `thumpdf`, it must be included in files for which a user wants to generate thumbnails (which are created by the `thumpdf` program); and `hyperref`, which extends the functionality of all the `TEX` cross-referencing commands to produce `special` commands which a driver

can turn into hypertext links. To protect URL characters we must load the `url` package, unless we have already provided `hyperref`. This package has its own version of the `url` macro, enhanced to provide clickable URLs.

To include POSTSCRIPT figures one needs `graphics` and/or `graphicx`. Those packages are modified by `pdflatex` so that they are able to include bitmaps (PNGs, JPEGs, and so on) and PDF files into the document. `color` facilitates the specification of user-defined colors (such as the cover green shades). Figures generated with \LaTeX can use any of the following packages: `pstricks`, `pstcol`, `multido`.

The bibliography was produced with \LaTeX . The package `natbib` (NATural sciences BIBliography) provides both author-year and numerical citations; it makes possible to define different citation styles. We have set the following options: “`round`”, to put citations within parenthesis; “`colon`”, to separate multiple citations with colons; “`authoryear`” to show author and year citations (instead of numerical citations); and the option “`sectionbib`” to use the package `chapterbib`. The style “`plainnat`” was then applied to format the bibliography. The package `chapterbib` allows to include a bibliography for each chapter. The package `minitoc` creates a mini table of contents for each chapter as well.

`makeidx` provides the macros required to make a subject index. To show the capital letter section headings, few variables were redefined on an auxiliary file (`header.ist`). One glossary was generated for this document: the web references. The package `glossary` allowed us to customize the format of this section.

We also defined a style file named `mythesis.sty`. It loads the following font packages: `fontenc` (with “`T1`” option), to set extended font encoding (accents and so on); `textcomp`, to include some extra symbols, such as the Euro symbol for instance; `pifont`, for `SYMBOL` and `ZAPF DINGBATS` fonts; `charter`, with which roman family is set to `BITSTREAM-CHARTER`; `helvet`, with which sans-serif family is set to `HELVETICA`; `euler`, with which formulas are set to `EULER`; and `courier`, to set typewriter family to `COURIER`. Other packages that were loaded are: `fancyhdr`, to produce nice headings; `fancyvrb`, to extend the `verbatim` environment; `comment`, to hide parts of the original \LaTeX files; `rotating`, to rotate boxes of text; and `multirow`, to get `multirow` cells within the `tabular` environment.

Getting the template files

You are free to copy, modify and distribute the template files of this thesis, under the terms of the GNU Free Documentation License as published by the Free Software Foundation. Any script bundled in this distribution, including the `Makefile`, is under the terms of the GNU General Public License. The template for this thesis as well as the DVD related files are available from:

<http://genome.imim.es/~eblanco/MyThesis/>

Bibliography

L. Lamport. *\LaTeX A Document Preparation System*. Addison Wesley, second edition, 1994. ISBN 0201529831.

WebSite References

ABS

ABS is a public database of experimentally verified orthologous transcription factor binding sites (TFBSs). Annotations have been collected from the literature and are manually curated. For each gene, TFBSs conserved in orthologous sequences from at least two different species must be available. For each regulatory site, the position, the motif and the sequence in which the site is present are available in a very simple format.

<http://genome.imim.es/datasets/abs2005/index.html>

CSHL MAMMALIAN PROMOTER DATABASE

Cold Spring Harbor Laboratory mammalian promoter database (CSHLmpd) used all known transcripts, integrating with predicted transcripts, to construct the gene set of human, mouse and rat genomes. For promoter information, they collected known promoter information from multiple resources, together with predicted ones. These promoters were mapped to genome, and linked to related genes. They also compared promoters of orthologous gene groups to detect the sequence conservation in promoter regions.

<http://rulai.cshl.edu/cshlmpd/index.html>

dbSNP

The NCBI database of SNPs.

<http://www.ncbi.nlm.nih.gov/SNP/>

DOE The Human Genome Project and Beyond

Genome programs of the U.S. Department of Energy Office of Science.

<http://www.doegenomes.org/>

EUROPEAN MOLECULAR BIOLOGY LABORATORY (EMBL)

EMBL-nucleotide sequence database.

<http://www.ebi.ac.uk/embl/>

ENSEMBL

Ensembl is a joint project between EMBL - EBI and the Sanger Institute to develop a software system which produces and maintains automatic annotation on selected eukaryotic genomes.

<http://www.ensembl.org/>

EPD

The Eukaryotic Promoter Database (EPD) is an annotated non-redundant collection of eukaryotic polymerase II promoters for which the TSS has been determined experimentally.

<http://www.epd.isb-sib.ch>

GENBANK

Overview about the content of GENBANK.

<http://www.ncbi.nlm.nih.gov/Web/GenBank/genbankstats.html>

GENBANK

GenBank is the NIH genetic sequence database, an annotated collection of all publicly available DNA sequences.

<http://www.ncbi.nlm.nih.gov/Genbank/index.html>

GENE ONTOLOGY

The Gene Ontology (GO) project is a collaborative effort to address the need for consistent descriptions of gene products in different databases. The Gene Ontology project provides a controlled vocabulary to describe gene and gene product attributes in any organism.

<http://www.geneontology.org>

Genetics GSK report: Genes and diseases

GlaxoSmithKline educational resource.

<http://genetics.gsk.com/link.htm>

JASPAR

JASPAR is a collection of transcription factor DNA-binding preferences, modelled as matrices. These can be converted into Position Weight Matrices (PWMs or PSSMs), used for scanning genomic sequences. JASPAR is the only database with this scope

where the data can be used with no restrictions (open-source).

http://mordor.cgb.ki.se/cgi-bin/jaspar2005/jaspar_db.pl

NCBI A Science Primer (bioinformatics)

A Basic Introduction to the Science Underlying NCBI Resources.

<http://www.ncbi.nlm.nih.gov/About/primer/bioinformatics.html>

NCBI A Science Primer (genomics)

A Basic Introduction to the Science Underlying NCBI Resources.

http://www.ncbi.nlm.nih.gov/About/primer/genetics_genome.html

NCBI A Science Primer (pharmacogenomics)

A Basic Introduction to the Science Underlying NCBI Resources.

<http://www.ncbi.nlm.nih.gov/About/primer/pharm.html>

NCBI MAP VIEWER

The Entrez Map Viewer is a software component of Entrez Genomes. It allows you to view an organism's complete genome, integrated maps (when available) for each chromosome, and sequence data for a region of interest.

<http://www.ncbi.nlm.nih.gov/mapview/>

NHGRI/NIH report: Genetics, the Future of Medicine

National Human Genome Research Institute.

www.nhgri.nih.gov

PROMO

PROMO is a virtual laboratory for the identification of putative transcription factor binding sites (TFBS) in DNA sequences from a species or groups of species of interest. TFBS defined in the TRANSFAC database are used to construct specific binding site weight matrices for TFBS prediction. The user can inspect the result of the search through a graphical interface and downloadable text files.

<http://alggen.lsi.upc.es/>

THE REFERENCE COLLECTION (RefSeq)

The Reference Sequence (RefSeq) collection aims to provide a comprehensive, integrated, non-redundant set of sequences, including genomic DNA, transcript (RNA), and protein products, for major research organisms.

<http://www.ncbi.nlm.nih.gov/RefSeq/>

TRANSFAC

TRANSFAC is a database on eukaryotic cis-acting regulatory DNA elements and trans-acting factors. It covers the whole range from yeast to human.

<http://www.gene-regulation.com/pub/databases.html#transfac>

UBC BIOMEDIA IMAGE AND MOVIE DATABASE

The Biomedica database is designed to provide Cell Biology students with a large number of images and movies of cell structure from a wide variety of cell types. The images and movies have been generated using high quality light microscopes, transmission electron microscopes (TEM) and scanning electron microscopes (SEM), such as the ones found in the UBC BioImaging Facility.

<https://www.biomedica.cellbiology.ubc.ca/cellbiol/default.php>

UCSC GENOME BROWSER

This site contains the reference sequence and working draft assemblies for a large collection of genomes. It also provides a portal to the ENCODE project.

<http://genome.ucsc.edu/>

Index

A

ABS, 96, 139
acceptor, 89
algorithms
 BLAST, 67
 Carrillo and Lipman, 69
 CLUSTALW, 70
 FASTA, 66
 Gotoh, 57
 Hirschberg, 46
 Myers and Huang, 76
 Needleman and Wunsch, 40
 Needleman and Wunsch revisited, 51
 Sellers, 42
 Smith and Waterman, 61
 TF-map alignment, 132
 Waterman, Smith and Katcher, 74
alignment, 36
 mismatch density, 64
 databases searches, 65
 BLAST, 67
 Carrillo and Lipman, 69
 classes, 39
 CLUSTALW, 70
 changes, 37
 example, 37
 FASTA, 66
 global, 39
 Gotoh, 57
 Hirschberg, 46
 local, 39
 meta-alignments, 128
 multiple meta-alignments, 175
 multiple TF-map alignments, 175
 multiple, 40, 69

 number of, 38
 Needleman and Wunsch, 40
 Needleman and Wunsch revisited, 51
 pairwise, 40
 progressive, 70
 scoring function, 37
 Sellers, 42
 sequences and TF-maps, 131
 Smith and Waterman, 61
 TF-map alignments, 128
alphabet, 35
 IUPAC alphabet, 35
alternative splicing, 90

B

bacteria, 10
binding sites, 92
Bioinformatics, 17
BLAST, 67

C

cancer, 25
cell, 10
 cell cycle, 12
 primitive cells, 10
 multicellular organisms, 11
 cell mutations, 12
chromatin, 15, 93
chromosomes, 15
CISRED, 149
CLUSTALW, 70
coding statistic, 101
codon
 codon bias, 101
codons, 14
comparative gene prediction, 104

comparative genomics, 103
comparative promoter prediction, 104
consensus, 97
CpG island, 94

D

databases searches, 65
distance, 38
 distance and similarity, 53
DNA, 10
 binding sites, 92
 chromatin, 93
 complementation, 12
 double helix, 15
 intergenic, 15
 DNA and RNA, 10
 structure, 15
 histones, 93
 methylation, 94
 microarrays, 105
 nucleosomes, 93
 nucleotides, 12
 signals, 96
 strands, 12
donor, 89

E

EMBL, 19
enhancer, 93
Ensembl, 20
EPD, 97
ESTs, 22
eukaryotes, 11
evolution, 10
exon
 classes, 89
 initial, 89
 intronless gene, 90
 internal, 89
 exon-defining signals, 88
 terminal, 89
exons, 14

F

FASTA, 66
 format, 19
first exon, 89

G

gap model
 affine, 58
 concave, 60
 general, 55
GenBank, 19
gene, 12
 catalogue, 88
 protein-coding regions, 101
 gene expression, 91
 genefinding, 95
 cancer, 25
 alleles, 12
 genes and illness, 25
 promoters, 92
 signals, 88
 transcription, 14
 homology, 35
 intronless gene, 90
 CpG islands, 94
 orthology, 35
 paralogy, 35
 selenoproteins, 90
 silencing, 93
 structure, 88
 gene regulation, 92
 translation, 14
Gene Ontology (GO), 22
genefinding, 95
 state of the art, 107
geneid, 109
genetic code, 14
genome, 15
 databases, 20
 landscape, 15
 complexity, 15
 projects, 20
 human genome, 126
genomic mapping, 73, 126
genotype, 12
GNU-GPL, 239

H

haplotype, 26
histones, 93
homology, 35

I

information content, 100

initial exon, 89
internal exon, 89
introns, 14

J

JASPAR, 96, 137
JASPAR_{TOP50}, 143

L

log-likelihood ratio, 99

M

map alignment, 73
 example, 74
 Myers and Huang, 76
 Waterman, Smith and Katcher, 74
maps, 72, 126
 alignments, 73
 TF-maps, 128
meta-alignments, 128
 accuracy, 142
 in CISRED, 148
 score distribution, 150
 training, 136
 parallel, 147
microarray, 105
multiple TF-map alignments, 175
 alignment of two clusters, 178
 non-collinear alignments, 181
 progressive alignment, 176
 training, 185

N

nucleosomes, 93

O

orthology, 35

P

paralogy, 35
pattern discovery, 107
pattern-driven methods, 96
PGWS, 147
pharmacogenomics, 25
phenotype, 12
phylogenetic footprinting, 105
position weight matrices, 98
 JASPAR, 96
 PROMO, 96

 specificity, 155
 TRANSFAC, 96
progressive alignment, 70
prokaryotes, 10
PROMO, 137
promoter, 92
 enhancers, 93
 characterization, 95
 identification, 146
 TSS, 102
promoter characterization
 state of the art, 111
protein synthesis, 14
protein-coding regions, 101
proteins, 15
pseudogene, 90

R

reading frames, 14
RefSeq, 20
restriction enzymes, 72
restriction map, 72
RNA, 10
 nucleotides, 14
 messenger, 14
 splicing, 14
 types, 14

S

search
 by content, 101
 by homology, 103
 by signal, 96
selenoproteins, 90, 110
sequence, 36
 alignment, 36
 consensus, 97
 distance, 38
 evolution, 35
 sequence comparison, 35, 103, 126
 databases, 19
 signals, 96
 similarity, 37
sequence-driven methods, 107
signals, 96
 collections, 96
 representation, 97
similarity, 37, 103

- similarity and distance, 53
- sites, 96
 - representation, 97
- SNP
 - classes, 26
 - distribution, 26
- SNPs, 15
- software
 - typesetting
 - BibTeX , 237
 - \LaTeX , 237–239
 - `pdflatex`, 237
 - `thumbpdf`, 237
- splicing, 14
 - acceptor site, 89
 - alternative splicing, 90
 - donor site, 89
 - non-canonical splicing, 90
- start codon, 88
- stop codon, 88
- subsequence, 36
- super-pattern, 166
- synteny, 103

T

- terminal exon, 89
- TF-map alignments, 128
 - accuracy, 142
 - in CISRED, 148
 - score distribution, 150
 - enhanced algorithm, 132
 - local, 158
 - multiple TF-map alignments, 175
 - naive algorithm, 132
 - non-collinear alignments, 181
 - promoter identification, 146
 - sequence alignments, 131
 - training, 136
 - training datasets, 137
- TF-maps, 128
 - alignments, 128
- thesis
 - chronology, 5
 - conclusions, 197
 - general objectives, 4
 - objectives, 4
 - outline, 7
- transcription factor, 92

- binding sites, 92
- transcriptional regulation, 92
- TRANSFAC, 96, 137
- TSS, 102

U

- UCSC genome browser, 20

W

- weight matrices, 98

Notes

Titles in the GBL Dissertation Series

- 2002-01** Moisés Burset.
Estudi computacional de l'especificació dels llocs d'splicing.
[Computational analysis of the splice sites definition.]
Departament de Genètica, Universitat de Barcelona.
- 2004-01** Sergi Castellano.
Towards the characterization of the eukaryotic selenoproteome: a computational approach.
Departament de Ciències Experimentals i de la Salut, Universitat Pompeu Fabra.
- 2004-02** Genís Parra.
Computational identification of genes: "ab initio" and comparative approaches.
Departament de Ciències Experimentals i de la Salut, Universitat Pompeu Fabra.
- 2005-01** Josep F. Abril.
Comparative Analysis of Eukaryotic Gene Sequence Features.
Departament de Ciències Experimentals i de la Salut, Universitat Pompeu Fabra.
- 2006-01** Enrique Blanco.
Meta-Alignment of Biological Sequences.
Departament de Llenguatges i Sistemes Informàtics, Universitat Politècnica de Catalunya.

META-ALIGNMENT OF BIOLOGICAL SEQUENCES

Enrique Blanco García

The sequences are very versatile data structures. In a straightforward manner, a sequence of symbols can store any type of information. Systematic analysis of sequences is a very rich area of algorithmics, with lots of successful applications. The comparison by sequence alignment is a very powerful analysis tool. Dynamic programming is one of the most popular and efficient approaches to align two sequences. However, despite their utility, alignments are not always the best option for characterizing the function of two sequences. Sequences often encode information in different levels of organization (meta-information). In these cases, direct sequence comparison is not able to unveil those higher-order structures that can actually explain the relationship between the sequences.

We have contributed with the work presented here to improve the way in which two sequences can be compared, developing a new family of algorithms that align high level information encoded in biological sequences (meta-alignment). Initially, we have redesigned an existent algorithm, based in dynamic programming, to align two sequences of meta-information, introducing later several improvements for a better performance. Next, we have developed a multiple meta-alignment algorithm, by combining the general algorithm with the progressive schema. In addition, we have studied the properties of the resulting meta-alignments, modifying the algorithm to identify non-collinear or permuted configurations.

Molecular life is a great example of the sequence versatility. Comparative genomics provide the identification of numerous biologically functional elements. The nucleotide sequence of many genes, for example, is relatively well conserved between different species. In contrast, the sequences that regulate the gene expression are shorter and weaker. Thus, the simultaneous activation of a set of genes only can be explained in terms of conservation between configurations of higher-order regulatory elements, that can not be detected at the sequence level. We, therefore, have trained our meta-alignment programs in several datasets of regulatory regions collected from the literature. Then, we have tested the accuracy of our approximation to successfully characterize the promoter regions of human genes and their orthologs in other species.

GBL Dissertation Series

Universitat Politècnica de Catalunya