Universidade de Aveiro
2016

Departamento de Electrónica,
Telecomunicações e Informática
Programa de Doutoramento em Informática
das Universidades de Aveiro, Minho e Porto

Diogo Pratas

Compressão e análise de dados genómicos

Compression and analysis of genomic data

Universidade de Aveiro
2016

Departamento de Electrónica,
Telecomunicações e Informática
Programa de Doutoramento em Informática
das Universidades de Aveiro, Minho e Porto

Diogo Pratas

Compressão e análise de dados genómicos

Compression and analysis of genomic data

Tese apresentada às Universidades de Aveiro, Minho e Porto para cumprimento dos requisitos necessários à obtenção do grau de Doutor em Informática, realizada sob a orientação científica do Doutor Armando José Formoso de Pinho, Professor Associado com Agregação do Departamento de Eletrónica, Telecomunicações e Informática da Universidade de Aveiro.

Trabalho financiado pelas seguintes entidades:

o júri / the jury

presidente / president

Doutor João Carlos Matias Celestino Gomes da Rocha
Professor Catedrático, Universidade de Aveiro

vogais / examiners committee

Fernando Manuel Augusto da Silva
Professor Catedrático, Faculdade de Ciências, Universidade do Porto

Mário Alexandre Teles de Figueiredo
Professor Catedrático, Instituto Superior Técnico, Universidade de Lisboa

Paulo Jorge dos Santos Gonçalves Ferreira
Professor Catedrático, Universidade de Aveiro

Armando José Formoso de Pinho
Professor Associado com Agregação, Universidade de Aveiro

Susana de Almeida Mendes Vinga Martins
Investigadora Principal, Instituto Superior Técnico, Universidade de Lisboa

agradecimentos

Devo ao Armando J. Pinho um agradecimento muito especial, sobretudo pela amizade, ensinamentos, educação, dedicação, condições, inspiração e encorajamento de que soube sempre ser fonte ao longo destes anos, por intenções, palavras e acções. Em resumo *lossy*, existem professores que ensinam, outros que educam. O Prof. Armando J. Pinho é excelente em ambas.

Ao Paulo J.S.G. Ferreira, que contribuiu como constante fonte de informação, optimização, coragem, planeamento e boa disposição recorrente.

Aos actuais membros do grupo de biologia computacional, nomeadamente à Raquel M. Silva, João M. O. S. Rodrigues, Carlos A. C. Bastos, Sónia Gouveia, Vera Afreixo, Raquel Sebastião. Aos ex-membros do grupo: Sara P. Garcia, Luís M. O. Matos, Luísa Castro, Diana Rodrigues.

À Susana Brás, Ílidio Oliveira e José Maria Fernandes pelo apoio, amizade e espaço.

Aos que de uma forma específica ajudaram: Liliana Correia, Micael Pedrosa, José Luís Oliveira, Sónia Mendo, António Correia, Eduardo Dias, Anabela Viegas, António Neves, Tomás Silva, Nuno Oliveira, Ana Isabel Martins, Alina Trifan, Luís Bastião, Mário Antunes, José Serra, Joel Arrais, Susana Vinga, Mário Figueiredo, João Rocha, Paul Green, Fernando Silva, Rogério Reis, Samuel Silva, Miriam Lopes, Rui Mendes, Rui Martins, Maria Jorge Pratas, José Vieira, Miguel Hernandez, Cláudio Teixeira, Sérgio Tafula, Raúl Oliveira, António Carvalho, Henrique Neves, Cláudia Nunes, Catarina Marques, Nuno Farias, Cátia Pinho, Ana Rocha, José Nuno Oliveira, Margarida Ribeiro, Jarvis e Kiki.

Aos elementos do surf magnet que proporcionaram escapatórias intensivas: Rui Matos e Sara Sofia, André Ferreia e Ana Caetano, João Ribeiro, Tiago Matos, Érica Barge, Rui Nunes, entre outros.

À familia, especialmente, à minha mãe, pai e irmão. À minha avó, tio, tios, tias, primos, Chester, Chica e Preta.

À senhora do bar de ambiente.

Às entidades financiadoras, nomeadamente FCT e RDConnect.

Yeder.

Palavras-chave

compressão de sequências genómicas, métodos não supervisionados, métodos sem alinhamento, complexidade relativa, modelos de contexto-finito

Resumo

As sequências genómicas podem ser vistas como grandes mensagens codificadas, descrevendo a maior parte da estrutura de todos os organismos vivos. Desde a apresentação da primeira sequência, um enorme número de dados genómicos tem sido gerado, com diversas características, originando um sério problema de excesso de dados nos principais centros de genómica. Por esta razão, a maioria dos dados é descartada (quando possível), enquanto outros são comprimidos usando algoritmos genéricos, quase sempre obtendo resultados de compressão modestos.

Têm também sido propostos alguns algoritmos de compressão para sequências genómicas, mas infelizmente apenas alguns estão disponíveis como ferramentas eficientes e prontas para utilização. Destes, a maioria tem sido utilizada para propósitos específicos. Nesta tese, propomos um compressor para sequências genómicas de natureza múltipla, capaz de funcionar em modo referêncial ou sem referência. Além disso, é bastante flexível e pode lidar com diversas especificações de *hardware*. O compressor usa uma mistura de modelos de contexto-finito (FCMs) e FCMs estendidos. Os resultados mostram melhorias relativamente a compressores estado-de-arte.

Uma vez que o compressor pode ser visto como um método não-supervisionado, que não utiliza alinhamentos para estimar a complexidade algorítmica das sequências genómicas, ele é o candidato ideal para realizar análise de e entre sequências. Em conformidade, definimos uma maneira de aproximar directamente a distância de informação normalizada (NID), visando a identificação evolucionária de similaridades em intra e inter-espécies. Além disso, introduzimos um novo conceito, a compressão relativa normalizada (NRC), que é capaz de quantificar e inferir novas características nos dados, anteriormente indetectados por outros métodos. Investigamos também medidas locais, localizando eventos específicos, usando perfis de complexidade. Propomos e exploramos um novo método baseado em perfis de complexidade para detectar e visualizar rearranjos genómicos entre sequências, identificando algumas características da evolução genómica humana.

Por último, introduzimos um novo conceito de singularidade relativa e aplicamo-lo ao *Ebolavirus*, identificando três regiões presentes em todas as sequências do surto viral, mas ausentes do genoma humano. De facto, mostramos que as três sequências são suficientes para classificar diferentes sub-espécies. Também identificamos regiões nos cromossomas humanos que estão ausentes do ADN de primatas próximos, especificando novas características da singularidade humana.

Keywords genomic sequence compression, unsupervised methods, alignment-free methods, relative complexity, finite-context modeling

Abstract

Genomic sequences are large codified messages describing most of the structure of all known living organisms. Since the presentation of the first genomic sequence, a huge amount of genomics data have been generated, with diversified characteristics, rendering the data deluge phenomenon a serious problem in most genomics centers. As such, most of the data are discarded (when possible), while other are compressed using general purpose algorithms, often attaining modest data reduction results.

Several specific algorithms have been proposed for the compression of genomic data, but unfortunately only a few of them have been made available as usable and reliable compression tools. From those, most have been developed to some specific purpose. In this thesis, we propose a compressor for genomic sequences of multiple natures, able to function in a reference or reference-free mode. Besides, it is very flexible and can cope with diverse hardware specifications. It uses a mixture of finite-context models (FCMs) and eXtended FCMs. The results show improvements over state-of-the-art compressors.

Since the compressor can be seen as a unsupervised alignment-free method to estimate algorithmic complexity of genomic sequences, it is the ideal candidate to perform analysis of and between sequences. Accordingly, we define a way to approximate directly the Normalized Information Distance, aiming to identify evolutionary similarities in intra- and inter-species. Moreover, we introduce a new concept, the Normalized Relative Compression, that is able to quantify and infer new characteristics of the data, previously undetected by other methods. We also investigate local measures, being able to locate specific events, using complexity profiles. Furthermore, we present and explore a method based on complexity profiles to detect and visualize genomic rearrangements between sequences, identifying several insights of the genomic evolution of humans.

Finally, we introduce the concept of relative uniqueness and apply it to the *Ebolavirus*, identifying three regions that appear in all the virus sequences outbreak but nowhere in the human genome. In fact, we show that these sequences are sufficient to classify different sub-species. Also, we identify regions in human chromosomes that are absent from close primates DNA, specifying novel traits in human uniqueness.

Contents

List of Figures

List of Tables

"Houston, we have ~~had a~~ (many) problem(s) here ..."
James A. Lovell

1 Introduction

Science is a cumulative process that organizes knowledge in order to test predictions and explanations about the universe. Science was developed by many scientists, in multiple fields of knowledge, achieving multiple breakthroughs.

1.1 Overview

One of the most important breakthroughs was the invention of the first mechanical computer by Charles Babbage, around 1822, to calculate polynomials. The device was called the difference engine. Its development led to the analytical engine around 1837, encapsulating most of the elements of modern computers. The evolution of mechanical computing was so popular that in the following century, with the WW1 (World War One) and mostly with WW2, they started to be used as prediction systems and for communications purposes. Most of the prediction systems were made for military artillery, while in communications cryptography played a role towards the end of the WW2.

1.1.1 History marks on communications and computing

In order to understand the foundations of communications and its inherent human development, we have to go much further back. After the delayed evolutionary development of the language, which is still a very variable and unclear thematic, its evolution as been truly achieved with the ability of writing. The capacity of coding vocal sounds into symbols that expressed objects or actions boosted the capacity for evaluating the own method (how to describe an object) and, by consequence, evolve it.

With the introduction of the first dictionary, humans corrected and unified the alphabet. Moreover, the ability to search fast for a specific word became a reality given the lexicographic order or, in some cases, using other methods, such as by topology or meaning. Even several language variants, such as in Mandarin, could now be understood by symbolic communication without the necessity of similar vocal communication. At this point, the communication speed was only dependent on the channel that transported the message. For example, in the

Roman empire, the speed of communication was dependent on the bird, horse (or similar), person or boat that travel with the message. Along with the confidentiality, the need for the fastest deliver was actually the main concern. Accordingly, both encoding and fast delivery of messages started to be investigated and developed.

Bird message delivery, mostly by pigeons, was used for several centuries. However, for mobile armies it only allowed a direction in communication. The communication by visual events, such as flags, symbols, fire or smoke, have also been used. However, they were very dependent on weather conditions and day time. In Africa, the talking drums have been used as one of the most effective technologies taking into account the development environment, as an acoustic way. The addition of redundancy and repeaters (from tribe to tribe), enabled a very fast communication in a relatively large distance. The talking drums, although they have been used for several centuries, have only been truly understood in the 18th century by the Europeans.

Mostly funded by the French government, France got its first telegraph network system (fixed communication by event using network repeaters controlled by operators) in the end of the seventeen century. Its success led to the usage of the same technology in much other countries. With the development of the electric telegraphs and their inherent message codes (such as Morse code), the number of users growth exponentially. In fact, now it was possible to send messages without high importance or priority almost instantly by anyone. The massive usage shown a lucrative way for several companies that developed their private electric telegraphs. Cost was now a variable enhancing the development of coding alphabets based on word frequency or reference public books. At this point the need for efficient data compression and transmission was truly followed, namely as a sampling principle [1, 2, 3]. For foundations in telegraphy sampling see [4].

However, coding theory suffered from a massive unconcern with the commercialization of the telephony, an invention credited mostly to Alexander Bell. The ability to easily transmit and receive messages by sound drop out the necessity to develop better codes. The telephony suffered from a massive adhesion, mostly after the first alleged transcontinental call around the beginning of WW1. The motivation for coding was, at this point, essentially for hiding the true meaning of the message.

When the Nazi party took the power in Germany, the Nazis realized that they could transmit classified messages through air using a cipher produced by a machine, the Enigma. Accordingly, the cryptanalysis re-started as a deep military need, that led several Polish to crack it (break the code) seven years before the WW2. Yet, the Nazi Germans increased their code complexity leading to one of the most successful encryption machines.

With the beginning of the WW2, it was crucial to crack the Enigma to put end to war. Therefore, Alan Turing and his associates, using his computation ideas [5], discovering a flaw in Enigma usage (the messages started and/or ended with the same words) built a secret machine (Bombe) that could brute-force most possible keys of the enigma using times much shorter that several humans. Therefore, everyday the Bombe cracked the Enigma code. The following action was probably most sacrificing but crucial. The Nazi Germans could not suspect that the Enigma was being cracked everyday or they would increase Enigma complexity, and hence, Alan Turing and several colleagues started to transmit only the minimal number of messages that could led to the end of the war. It was a statistical game. They had the ultimate success in 1945.

The importance of the computing machines was well established with the end of WW2. The commercialization of the first computers started as a series of computers known by

Colossus. Communications purposes originated the development of computing machines, which led to the development of better communications, namely those who where concerned with the foundations of entropy.

1.1.2 Entropy

Nyquist and Hartley gave a basic definition of entropy (or information) for each symbol as the logarithm of the number of possible symbols

$$H = n \ \log \ s, \tag{1.1}$$

where n is the number of symbols on the message and s is the cardinality of the alphabet (number of possible symbols). In 1948, Claude Shannon proposed a more realistic approach, defined as

$$H = -\sum \ p_i \ \log_2 \ p_i, \tag{1.2}$$

where p_i is the probability of each message and $\log_2$ was convenient due to the binary scale [6]. At this point, the term information was new and a synonym of entropy, unpredictability and randomness, mostly, for stationary sources.

Fundamentally, the term entropy dates back, apparently, to the first thermodynamic experiments, namely by Rudolf Clausius around 1850. It has originated the three basic rules of thermodynamics [7]. The first law states that in an isolated system the energy is constant (energy conservation), and thus, it is possible to derive that several different forms of work can be converted. However, since conversion also costs work, in some way it excludes perpetual motion machines. The second law states that the quantity of entropy in a thermodynamic isolated system tends to increase with time until it reaches a maximum value. However, it is only a probabilistic rule. The third law states that the entropy of a thermodynamic isolated system approaches a constant value as the temperature approaches absolute zero, and hence, all the processes stop and the entropy becomes minimal. When joining the three laws we are able to see that if two systems are in thermal equilibrium respectively with a third system, they must be in thermal equilibrium with each other, where it helps to define the notion of temperature.

Supported by Clausius fundamentals, James Maxwell defined his demon as a hacker which could hypothetically violate the second law of thermodynamics [8]. Maxwell gave a case example where a tiny finite being was watching a small diaphragm which separated a gas box (thermodynamic isolated system). The creature can identify the molecules which are faster (hot) from the slower (cold), and hence, it can let them pass to the other side or not, altering the probabilities and creating two thermodynamic isolated systems with different temperatures. This experience shows the unsymmetrical property of thermodynamics, where to mixture two isolated systems in one is a easy way since they naturally tend to the thermal equilibrium over time. However, to reverse the action the problems start to appear and sometimes referred to as deterministic irreversible [9].

Analogous to the thermodynamics, Andrey Kolmogorov proposed his own demon to improve Shannon's description of information. In fact, he proposed three quantitative definitions of information: a combinatorial, a probabilistic (mostly refining Shannon's definitions) and an algorithmic [10]. The algorithmic approach became the standard and ultimate approach to quantify information. Moreover, a more appropriate term started to be used instead of information: complexity. Perhaps without knowing about Kolmogorov works, due

to language and political barriers, Gregory Chaitin studying Turing machines [5] and Kurt Gödel's incompleteness theorem [11], proposed, with a definition of the algorithmic information similar to Kolmogorov complexity, the Turing incomputability issue [12]. Moreover, Ray Solomonoff also achieved similar results, contributing mostly with probability theory perspectives [13, 14]. Therefore, the Kolmogorov complexity can be seen as an independent discovery of Kolmogorov, Chaitin and Solomonoff. Most of its applications address not only individual objects (or systems) but also relations between objects.

More recently, Charles Bennett gave several natural definitions of a universal information metric, based on the length of shortest programs (Kolmogorov complexity) for either ordinary computations or reversible computations [15]. Bennett and his mentor, Rolf Landauer, proposed that only an irreversible operation increases the entropy of a system and, as a logic operation, this is analogous to deletion, leading, in thermodynamics, to the dissipation of heat [16]. Accordingly, Bennett proposed a measure (universal, anti-symmetric, and transitive) for the thermodynamic work required to transform one object in another object by the most efficient process, where applied to algorithmic information it adds to Kolmogorov complexity the notion of time. Therefore, it can be seen as the time needed to transform a object into another given a pattern/cognitive similarity. This type of measure is known as the logical depth [15].

1.1.3 Incomputability

At its most fundamental, any information in a isolated system is a propagation of cause and effect within a system, described with a language, loaded by its inherent parameters, and approximated with the shortest, fastest and most economic energy descriptor (less work possible). The problem is to describe such descriptors, given the available data. In the case of algorithmic complexity the program needs to learn how the system works, which includes all the players and their relations, and this is a task that needs computational work. It is perhaps the hardest asymptotic (inverse) unsymmetrical problem. Therefore, given the Turing incomputability in the algorithmic complexity (in can only be approximated), the Gödel's incompleteness theorem in mathematical logic's, the Heisenberg uncertainty [17] in quantum mechanics, we realize that for these (analogous) related areas there is not a language that addresses perfectly the data content, and hence, be considered complete. Nevertheless, we share EPR belief [18] that such a language exists.

Although the existence of Turing incomputability, for many applications, computation gives actually a good approximation to objects nature. In fact, algorithmic complexity is currently the best known way to measure complexity in individual objects or between them, specially when they are digital or have reduced number of dimensions. Several applications involving algorithmic complexity description have been reported, for example, in genomics, virology, languages, literature, music, handwritten digits and astronomy [19]. Genomics is probably one of the best application fields, namely because genomic sequences (DNA sequences) are mostly objects with one dimension (as far as we know). For an introduction on the biological procedures, from DNA to Protein, and some foundations, see Chapter 2.

1.1.4 Finding descriptors

Genomic sequencing has had a major impact on life sciences since the wide scale adoption of the Sanger sequencing method [20], leading to consequent improvements of sequencing

methods and big amounts of data generation. Nowadays, the genomics sequencing centers and the scientific community are being flooded with genomic data [21]. In spite of the possibility that a transformative breakthrough in storage technology occurs in the following years, the $1000 genome milestone is most likely to arrive before the $100 petabyte hard disk, mainly because the cost of disk storage is steadily decreasing over time, not matching the dramatic change in the cost and volume of sequencing. Nowadays, it is common to find genomics sequencing projects having a larger fraction of the budget allocated to the computational infrastructure (including the storage component) than to the biological part. This was unthinkable, for example, when the first draft of the human genome was released. Therefore, compression descriptors, the ones who code and extract redundancy of the representation of the object play a key role in storage.

Although, many other projects in other areas, such as physics, are feeling the same problems. One such high-profile project is the Large Hadron Collider (LHC) at CERN, which will generate an estimated 15 petabytes of data per year when fully active[1]. Moreover, astronomy data generation is also a concern where large volumes of galaxy redshift surveys and cosmic microwave background (CMB) data continue to be generated. Many people call this the big data era, where there are mostly two options: to compress the data (lossless) or to analyze *on the fly* the generated data and discard it, storing only lossy descriptors that might be enough for conclusion purposes.

In spite of the existence of similar concerns in different areas, genomics is probably one of the most important, because it enables to advance medicine (for life-quality and health treatment) and the notion of species similarity (i.e., to understand the differences and why do they exist). Mainly, the advances in genomics led to the unveiling of the first notion of cause/effect in a semi-isolated thermodynamic system, called body, and their environmental interactions. Accordingly, the development of models and/or methods that address the representability of a species genome and their interactions is a very important problem to understand the nature of the species puzzle. It is perhaps a puzzle of a lifetime.

1.2 Motivation

The data deluge phenomenon is becoming a serious problem in most genomics centers, as it can be see by the growing number of the fully sequenced and re-sequenced genomes from large-scale projects such as the 1000 Genomes Project [2], The Cancer Genome Atlas [3], The 10k Genomes [4], among many others. Moreover, the prizes that reward cheaper, faster, less prone to errors and higher throughput sequencing methodologies [5], help to increase this scenario. To alleviate it, general purpose tools, such as gzip, are used to compress the data. However, although pervasive and easy to use, these tools fall short when the intention is to reduce as much as possible the data, for example for medium and long term storage. In fact, for several genomic sequences they attain results worst than the 2 bits per base. To face this, a competition has been proposed for the achievement of better compression algorithms [6]. A number of algorithms have been proposed for the compression of genomics data, but

[1] http://public.web.cern.ch/public/en/LHC/Computing-en.html
[2] `http://www.1000genomes.org/`
[3] `http://cancergenome.nih.gov/`
[4] `http://genome10k.soe.ucsc.edu/`
[5] `http://genomics.xprize.org/`
[6] `http://www.pistoiaalliance.org/projects/sequence-squeeze/`

unfortunately only a few of them have been made available as usable and reliable compression tools. From these, they are mostly developed to some specific purpose within the genomic sequences, namely to a specific nature (for instance highly repetitive) or to cope with a specific file format.

On the other hand, the underlying model of compression can be used as a descriptor for the object. Therefore, the better the compression model is, the better the descriptor of the object is. Accordingly, it is a way to approximate the object nature. When we explore this principle between objects, we are approximating the space between that nature of objects. This is very important if our objective is to understand evolution, given the Darwin and Huxley theory of evolution [22]. Compression models are usually nonsupervised, therefore they are automatic learning programs that search for regularities. In this sense, improving the compression models is also a way to improve Artificial Intelligence systems. Moreover, the information contained in these models can led to the development of new diagnostics and therapeutic methodologies, besides the ability to look for a past window of millions of years in order to understand human nature.

1.3 Objectives

At a high level, the objectives are essentially the contribution with compressors for storage minimization purposes and the development of unsupervised algorithms that are able to determinate regularities (or absence of them) in and between genomic sequences.

At a lower level, we aim to create a universal genomic compressor (in the sense of any genomic sequence specific characteristic) that is able to obtain state-of-the-art results, using the minimal time and memory resources, but at the same time flexible enough in the sense of memory optimization for any computer hardware specifications. The compressor must be also prepared to perform analysis.

Another objective is to use the compressor in analysis, studying the effects and foundations of algorithmic complexity, associated to metrics or measures. As such, we aim to quantify and locate low and high regions of complexity. For the purpose, we analyze regularities in the genomic sequences using a triple approach: individual sequences, between sequences of the same species and between sequences that represent different species. Moreover, using algorithmic complexity, the objective is to build an unsupervised method to detect rearrangements (or regularities) between different sequences.

As complement, analyze unique features in genomic sequences that might classify or characterize species in order to detect insights into genomic evolution or novel traits.

1.4 Structure

The thesis is divided in seven chapters. The first chapter is an introduction. The second chapter gives a background in biology and compression. The third chapter describes techniques and methods for the compression of genomic sequences, namely presenting two compressors: a compressor for genomic collections and a universal genomic compressor. The fourth chapter depicts the usage of compression-based measures for analysis. It is constituted by three sections, namely a Kolmogorov complexity introduction, global measures and local measures. In global measures, we present a way to compute directly the Normalized Information Distance and introduce a new notion, the Normalized Relative Compression. In local

measures, we define complexity profiles and give several applications. The fifth chapter is an extension of the fourth, however for a specific purpose, the detection of genomic rearrangements. The sixth chapter addresses the complement: relative uniqueness. Accordingly, two applications are presented, namely the detection of Ebola virus specific signatures (relatively to human) and the detection of human specific regions according to primates. The seventh chapter finishes the thesis with conclusions and future work.

1.5 Contributions

In the context of this thesis several contributions have been made.
In journals:

1. Diogo Pratas, Raquel Silva, Armando J. Pinho, Paulo J S G Ferreira. An alignment-free method to find and visualise rearrangements between pairs of DNA sequences. Scientific Reports, vol. 5, p. 10203, May 2015 (IF 2014: 5.578);

2. Raquel Silva, Diogo Pratas, Luísa Castro, Armando J. Pinho, Paulo J S G Ferreira. Three minimal sequences found in Ebola virus genomes and absent from human DNA. Bioinformatics, April 2015 (IF 2014: 4.981);

3. Luís Matos, António J. R. Neves, Diogo Pratas, Armando J. Pinho. MAFCO: A compression tool for MAF files. PLoS ONE, vol. 10, no. 3, p. e0116082, March 2015 (IF 2013: 3.534);

4. Armando J. Pinho, Diogo Pratas. MFCompress: a compression tool for FASTA and multi-FASTA data. Bioinformatics, vol. 30, no. 1, p. 117-118, January 2014 (IF 2013: 4.621);

5. Diogo Pratas, Armando J. Pinho, João M. O. S. Rodrigues. XS: a FASTQ read simulator. BMC Research Notes, vol. 7, no. 40, January 2014;

6. Armando J. Pinho, Sara Pinto Garcia, Diogo Pratas, Paulo J S G Ferreira. DNA sequences at a glance. PLoS ONE, vol. 8, no. 11, p. e79922, November 2013 (IF 2013: 3.534);

7. Luís Matos, Diogo Pratas, Armando J. Pinho. A compression model for DNA multiple sequence alignment blocks. IEEE Transactions on Information Theory, vol. 59, no. 5, p. 3189-3198, May 2013 (IF 2013: 2.650);

8. Sara Pinto Garcia, João M. O. S. Rodrigues, Sérgio Santos, Diogo Pratas, Vera Afreixo, Carlos A C Bastos, Paulo J S G Ferreira, Armando J. Pinho. A genomic distance for assembly comparison based on compressed maximal exact matches. IEEE/ACM Transactions on Computational Biology and Bioinformatics, vol. 10, no. 3, p. 793-798, May 2013 (IF 2013: 1.536);

9. Armando J. Pinho, Diogo Pratas, Sara Pinto Garcia. GReEn: a tool for efficient compression of genome resequencing data. Nucleic Acids Research, vol. 40, no. 4, p. e27, February 2012 (IF 2012: 8.278) (IF 2013: 8.808);

Book chapters:

1. Armando J. Pinho, Diogo Pratas, Sara Pinto Garcia. Compressing resequencing data with GReEn. Deep Sequencing Data Analysis, Noam Shomron (Ed.), Humana Press, vol. 1038 (Methods in Molecular Biology), p. 27-37, July 2013;

In international conferences:

1. Diogo Pratas, Armando J. Pinho. Exploring deep Markov models in genomic data compression using sequence pre-analysis. Proc. of the 22nd European Signal Processing Conference, EUSIPCO-2014, Lisbon, Portugal, September 2014;

2. Diogo Pratas, Armando J. Pinho. A conditional compression distance that unveils insights of the genomic evolution. Proc. of the Data Compression Conference, DCC-2014, Snowbird, UT, p. 421, March 2014;

3. Armando J. Pinho, Diogo Pratas, Paulo J S G Ferreira. Information profiles for DNA pattern discovery. Proc. of the Data Compression Conference, DCC-2014, Snowbird, UT, p. 420, March 2014;

4. Sara Pinto Garcia, João M. O. S. Rodrigues, Diogo Pratas, Armando J. Pinho. Comparing maximal exact repeats in human genome assemblies using a normalized compression distance. 20th Annual International Conference on Intelligent Systems for Molecular Biology, Long Beach, California, USA, July 2012;

5. Luís Matos, Diogo Pratas, Armando J. Pinho. Compression of whole genome alignments using a mixture of finite-context models. Proceedings of 9th International Conference on Image Analysis and Recognition, ICIAR 2012, Aveiro, Portugal, vol. Aurélio Campilho and Mohamed Kamel (Eds.): Part I, LNCS 7324, p. 359-366, June 2012;

6. Diogo Pratas, Armando J. Pinho, Sara Pinto Garcia. Exon: A Web-Based Software Toolkit for DNA Sequence Analysis. Advances in Intelligent and Soft Computing, Proc. of the 6th Int. Conf. on Practical Applications of Computational Biology & Bioinformatics, PACBB 2012, Salamanca, Spain, vol. 154, p. 217-224, March 2012;

7. Diogo Pratas, Armando J. Pinho, Sara Pinto Garcia. Computation of the normalized compression distance of DNA sequences using a mixture of finite-context models. Bioinformatics 2012: International Conference on Bioinformatics Models, Methods and Algorithms, Vilamoura, Portugal, February 2012;

In national conferences:

1. Diogo Pratas, Raquel Silva, Armando J. Pinho, Paulo J S G Ferreira. Detection and visualisation of regions of human DNA not present in other primates. Proceedings of the 21th Portuguese Conference on Pattern Recognition, RecPad 2015, Faro, Portugal, October 2015;

2. Diogo Pratas, Raquel Silva, Armando J. Pinho. Large-scale inversions between human reference assemblies. Proceedings of the 20th Portuguese Conference on Pattern Recognition, RecPad 2014, Covilhã, Portugal, October 2014;

3. Raquel Silva, Luísa Castro, Diogo Pratas, Armando J. Pinho. Towards personalized medicine: ebola virus absent words in the human genome. Proceedings of the 20th Portuguese Conference on Pattern Recognition, RecPad 2014, Covilhã, Portugal, October 2014;

4. Diogo Pratas, Armando J. Pinho. Insights into primates genomic evolution using a compression distance. Proc. RecPad 2013, Lisbon, November 2013;

5. Diogo Pratas, Armando J. Pinho. On the compression of FASTQ quality-scores. Proceedings of the 18th Portuguese Conference on Pattern Recognition, Coimbra, Portugal, October 2012;

On the other hand, although in a way connected, the following contributions have been made simultaneously:

1. Armando J. Pinho, Diogo Pratas, Paulo J S G Ferreira. A new compressor for measuring distances among images. Proc. of the International Conference on Image Analysis and Recognition, ICIAR-2014, Vilamoura, Portugal, vol. LNCS 8814, p. 30-37, October 2014;

2. Diogo Pratas, Armando J. Pinho. On the Detection of Unknown Locally Repeating Patterns in Images. Proceedings of 9th International Conference on Image Analysis and Recognition, ICIAR 2012, Aveiro, Portugal, vol. Aurélio Campilho and Mohamed Kamel (Eds.): Part I, LNCS 7324, p. 158-165, June 2012;

"Follow the white rabbit."

2 Background

2.1 Biological Background

The biological processes, such as self-sustaining and signaling, defines the organisms nature, namely as the smallest and optimized contiguous units for a specific function over a certain period of time. Organisms can be classified as unicellular, such as *Valonia ventricosa*, or multi-cellular, such as *Homo sapiens*. In fact, humans are composed of many trillions of cells grouped into specialized tissues and organs. An organism may be either a prokaryote or eukaryote (eukarya). Having a higher factor of diversity, prokaryotes are represented by the Bacteria and Archaea domains [23], while eukaryotes are characterized by the presence of a membrane-bound cell nucleus and contains additional membrane-bound compartments (called organelles), namely mitochondria, Golgi apparatus or chloroplasts [24]. Animals, plants and fungis are well known examples of eukaryotic kingdoms.

Individual organisms are time limited [25, 26]. Therefore, to avoid species extinction, they must have the ability to create new individuals of the same kind, either asexually from a single parent, or sexually from two parent organisms. Ultimately, they can be seen as (metabolic) functions that accept parameters, namely through environment (energy and materials) and ancestors (parents), and produce a transformed outcome that is tightly related with its survival and, in some way, in species survival. Globally, these functions need to be constantly evolving, according to a process called the natural selection [22], using the smallest amount of energy that ensures efficiency. It is a survival game, as the major four mass extinctions in the marine fossil record exemplify [27].

In agreement with [28, 29], Fig. 2.1 depicts a proposed phylogenetic tree showing the separation of bacteria, archaea, and eukaryotes [30, 31]. Despite the visual similarity to bacteria, archaea possess genes and several metabolic pathways that are more similar to those of eukaryotes, namely the enzymes involved, in transcription and translation, and their reliance on ether lipids in their cell membranes. Moreover, they are known by using more energy sources than eukaryotes and by living in extreme environments.

Besides the three separated domains, there are the viruses. These have been defined as small infectious agents that replicate only inside the living cells of other organisms [32]. In

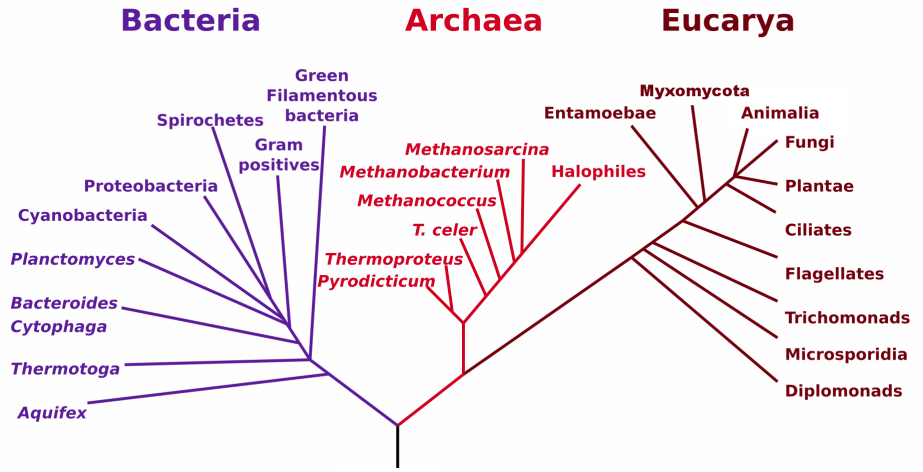


Figure 2.1: A proposed rooted tree for rRNA genes, showing major branches Bacteria, Archaea, and Eukaryote. Source: wikipedia.org.

---

fact, viruses can infect all types of life forms and in some cases integrate the hostage genomes [33, 34, 35]. Their classification is currently a controversial subject, mainly because many believe that the recent discovery of the giant DNA viruses, also known as megavirus [36], should represent a separated fourth domain of life in addition to the Bacteria, Archaea and Eukarya domains [37, 38]. Despite their association with other organisms death, they are considered one of the major factors of increasing the genetic diversity of organisms [39].

Despite metabolic and structural diversity between species, living organisms and several viruses have their instructions, also known as genetic information, embedded in the *Deoxyribonucleic acid*, or DNA for short, while other viruses use the *Ribonucleic acid*, or RNA for short. The DNA sequences are composed by four different elements (or bases): *adenine* (A), *cytosine* (C), *guanine* (G), and *thymine* (T). *Adenine* and *guanine* are classified as *purines*, in agreement with their chemical similarity, while *cytosine* and *thymine* as *pyrimidines*. The definition of *nucleotide* derives from the combination of a base with a sugar/phosphate. The DNA sequences are usually represented and analyzed as strings over a quaternary alphabet, that can be small as a virus or as large as a human. The process of unveiling DNA nucleotides is known as DNA sequencing [20].

Although in 2010 it has been proposed that a microbe (GFAJ-1) that, when starved of phosphorus, was capable of substituting arsenic for a small percentage of its phosphorus and sustain its growth [40], subsequent independent published studies found no detectable arsenate in the DNA of GFAJ-1. Moreover, they have demonstrated that GFAJ-1 is simply an arsenate-resistant phosphate-dependent organism [41]. Therefore, to the present all the (known) living organisms are based on phosphate.

Based on the image of diffraction X-rays by Rosalind Franklin, 1952, and a DNA image improved by Maurice Wilkins, followed by biochemical information by Erwin Chargaff, James Watson and Francis Crick discovered the double helix structure of DNA in 1953 [42], making evident that the total percentage of complementary nucleotides, namely *adenine- thymine* (A-T) and *cytosine-guanine* (C-G), in a double-stranded molecule should be equal. This property had been previously reported by Chargaff and it is defined as his first parity rule [43]. The detailed analysis of some bacterial genomes led to the formulation of Chargaff's

second parity rule, which asserts that the percentage of complementary nucleotides should also be equal in each of the two strands. This rule has been extensively confirmed in bacterial and eukaryote genomes, including recent results [44] that use global association measures and suggest that the symmetry holds for k-mers up to 10 nucleotides. Nevertheless, the universality of Chargaff's second parity rule has been questioned for organelle DNA and some viral genomes [45].

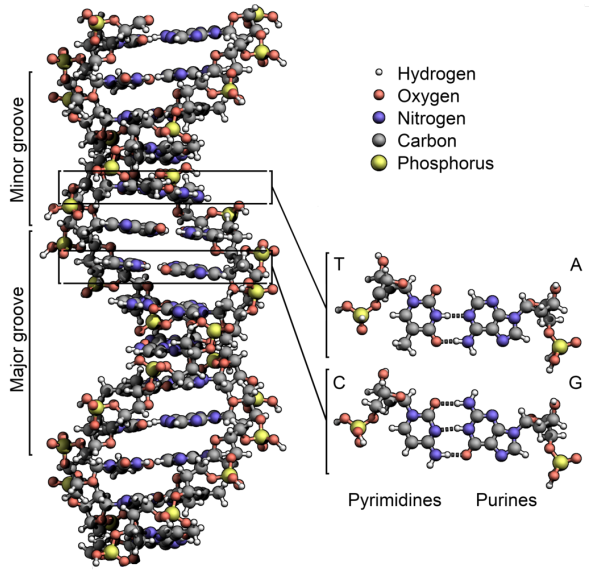


Figure 2.2: The structure of the DNA double helix. The atoms in the structure are color-coded by element and the detailed structure of two base pairs are shown in the bottom right. Source: wikipedia.org.

Fig. 2.2 depicts a biochemistry representation of the double helix. Accordingly, the base pairs are separated 0.34 nm in the double helix, and a complete turn of one chain over the other materializes 3.4 nm, making 10 pair bases by rotation. The chains are not uniformly spaced on the double helix, forming cavities of different sizes, namely a large (major groove) and a thin (minor groove). Both chains are anti-parallel, which means that they follow from $5'$ to $3'$ in an inverted sense. These chains may contain many instructions which can be fundamental to the development and survival of the organisms and regularly called genes.

According to Fig. 2.3, a chromatid is composed by many genes, regulatory elements and other non-coding DNA/RNA, all packaged and organized, namely by the nucleosomes and histones [46]. A chromatid that joined to the other copy by a single *centromere* constitute a chromosome. The tip of chromosomes are called *telomers* and are normally associated with aging [47]. Some organisms have multiple copies of chromosomes: diploid, triploid, tetraploid and so on. As an example, humans are diploid, while salmons are tetraploid. A set of chromosomes forms a genome.

The human genome, as Matt Ridley metaphorized [48], can be seen as a book that wrote itself, continually adding, deleting and amending over four billion years. The book contains twenty-three chapters, called chromosomes. Each chapter contains several thousand stories, called genes. Each story is made up of paragraphs, called exons, which are interrupted by advertisements called introns. Each paragraph is made up of words, called codons. Each

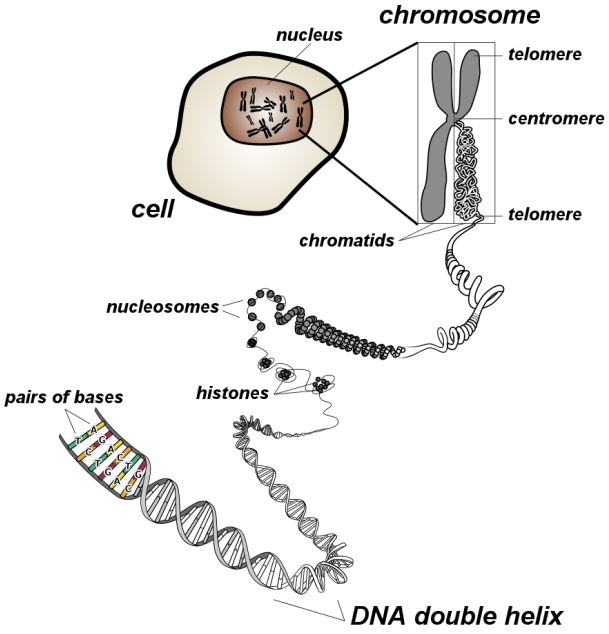


Figure 2.3: Progressive visualization from a cell to DNA. Image adapted from wikipedia.org.

word is written in letters called bases.

The human genome is determined by approximately 3 000 million base pairs [49]. This means that being the DNA a language written with an alphabet of four different symbols, it takes approximately 750 MBytes to represent the human genome using $\log_2 4 = 2$ bits per symbol (bps).

Analogous to the OSI model in communications, genomic information has its own layer model, namely from DNA to RNA and to Protein. According to Fig. 2.4 there are two main steps to produce Proteins from DNA. The first one is transcription, that is to make RNA from DNA. RNA (Ribonucleic Acid) is synthesized in the nucleus and is very similar to DNA. The synthesis of RNA also involves the use of bases. However, in RNA synthesis no thymine (T) is used, but uracil (U) instead. The sequence of RNA corresponds to the sequence of DNA from which the RNA is synthesized.

The second one, is known to be the translation, making Proteins from RNA. It all begins in the RNA strand, where the protein will be synthesized. A protein is made from amino acids. The protein strand, although in the Fig. 2.4 is being presented as a line, has complex interactions between amino acids leading to three dimensional spatial forms that are essential for the functioning of the protein. In the translation of RNA to protein one amino acid is added to the protein strand for every three bases in the RNA. So, a RNA sequence of 24 bases codes for a protein strand of eight amino acids. On the left of Fig. 2.4 there is a table with the encoding correspondent to the translation, where the combination of three bases gives always the same amino acid, for example "CAT" corresponds to "gln" (glutamine).

As described previously, sequencing is the process to unveil the genomic information (individual content and relative order), into digital format. This enables the possibility to analyze the data. However, the sequencing process is not yet perfect. In fact, currently it is only possible to sequence fragments. As such, it is the same as to find the order and the content of a few pieces in a huge puzzle that we need to assemble and, after, perform an analysis,

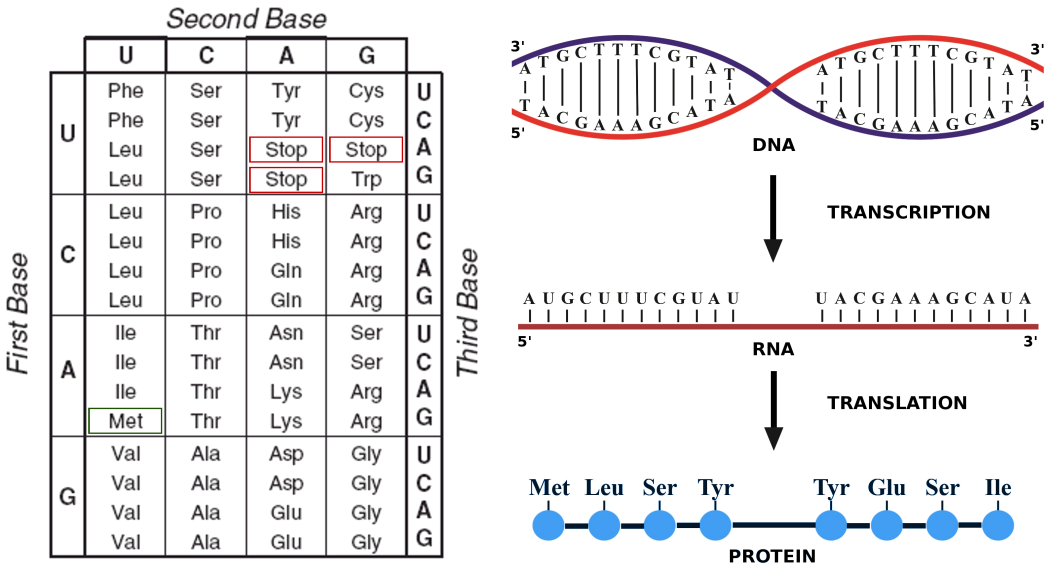


Figure 2.4: Stages from DNA to Protein: transcription (DNA to RNA) and translation (RNA to amino acid). Several amino acids form a Protein. Translation is done according to the table on the left.

sometimes comparative (between puzzles), sometimes individual (within the same puzzle). To challenge more this quest, the tiny fragments, namely between 25 to 300 bases, might have several errors (it is a probabilistic label attribution). Therefore, the analysis of genomic sequences needs to be addressed with methods/tools that are aware of these difficulties [50].

There are many file formats to store the data after sequencing, such as FASTQ, BAM, SAM, VCF, FASTA, among others. The FASTQ is by default the current output format from sequencers. FASTQ is made of multiple reads. Each read has three information channels: headers (information relative to each read), a small sub-sequence containing DNA bases (normally between 25 and 300 bases) and quality-scores containing, for each DNA base, the corresponding sequencing quality. This is a highly redundant format that led to the development of the SAM format. SAM explores the redundancy according to a reference sequence, and hence, for high similarity between sequences (such has multiple human genomes) it enables large space savings. Besides the headers, this format only stores information of deviations relatively to the reference sequence (both in DNA bases and quality-scores). BAM is the binary version of SAM, reducing even more the space, although it needs a simple transformation for analysis access purposes. The VCF file format does not store quality scores neither headers (identifiers), since it only maps the differences (substitutions, deletions and insertions) according to an assembled reference sequence.

Probably the most popular and used file format is FASTA. This format has two channels of information: headers and DNA bases. It can be used after and before assembling the sequences. If it is assembled, without the headers, can be considered a genomic sequence, containing only the bases: A, C, G, T. In several cases, there are other symbols, such as N, that represent mostly sequencing or assembling uncertainties and, therefore, since they are caused by faults in the accuracy of sequencing or human comprehension, in most cases, we ignore or uniformly generated random bases for its replacement.

2.2 Compression

Compression is a lossless or lossy technique that identifies and eliminates redundancy of data using a source coding method called compressor [51, 52]. Lossless compression does not loose information. This means that the compressed data can be decompressed to exactly its original value. On the other hand, lossy compression looses information, making (perhaps) *impossible* to obtain the original value in the decompression process.

Most successful compression techniques fall into three categories: variable-length coding, dictionary-based and arithmetic coding.

Generically, a variable-length code can be seen as a code that maps a source of symbols to a variable number of bits, such as the well known Shannon-Fano [6], Huffman [53] and Golomb [54] codes. One of the most successful applications is bzip2, based on the Burrows-Wheeler transform (BWT) [55], that divides the input into fixed size blocks and uses the BWT for each block. The sort order is the lexicographical order of the string to which it refers, wrapping around to the beginning of the block when necessary. After the transformation, the data is encoded with the Huffman technique.

Dictionary-based compressors, also known as substitutional compressors, replace an occurrence at a particular offset with a reference to a previous occurrence, such as the well known gzip, based on the Lempel-Ziv method [56]. Given that gzip uses the self sequence to match repeated occurrences, the memory increases in proportion to the size of the sliding window (by default, gzip uses a fixed-size window to avoid high memory usage).

The most recent compressors are based on arithmetic coding [57, 58]. According to Fig. 2.5, this technique leds to a new concept, the full separation of modeling from coding [59], mostly given by the flexibility of arithmetic coding that can be fed by any statistical model.

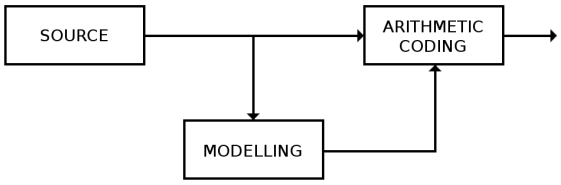


Figure 2.5: Arithmetic coding scheme

The idea of arithmetic coding enables the possibility to have *fractions of bits*, because the entire message is represented using a single code-word: a number in $[0, 1)$. This idea exceeds variable-length codes, namely because, unlike arithmetic coding, they are optimum only if the probabilities of the symbols are powers of 2 [60].

If we consider the binary sequence `0110` ($P_0 = 0.6$, $P_1 = 0.4$), the encoded message is a number in the interval $[0.504, 0.5616)$ as depicted in Fig. 2.6. Moreover, Fig. 2.6 also depicts the process of decoding of the previous binary example, considering $x = 0.54$. It is important to say that, in both processes, probabilities can never be zero, or this will harm the arithmetic coding process [61]. Therefore, estimators that assume always the possibility of an event to happen play a key role.

Most of the updates in arithmetic coding are related to trades between precision and processing speed [62], because the efficiency of an arithmetic coder is very close to the entropy of the model. Therefore, modeling is a key factor.

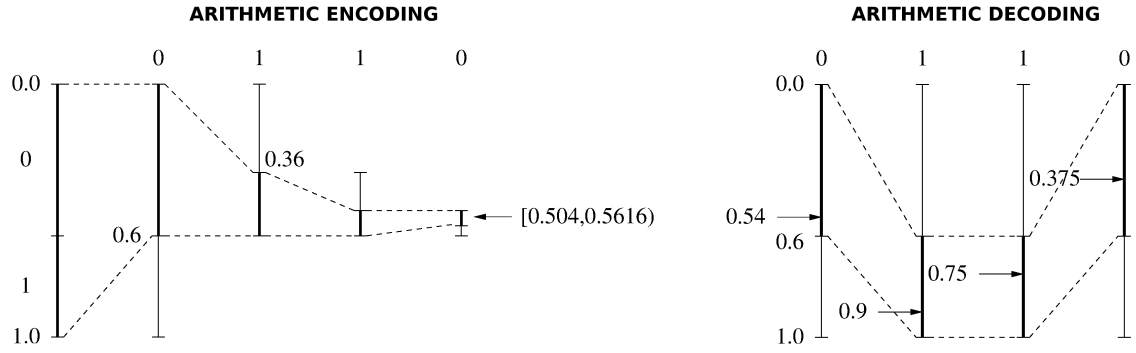


Figure 2.6: Arithmetic encoding (left) and decoding (right) processes given the binary string 0110 assuming static probabilities with $P_0 = 0.6$ and $P_1 = 0.4$.

For example, a statistical compressor, such as prediction by partial match (PPM) [63], explicitly estimates the probability distribution of each symbol. Statistical compression algorithms depend on assumptions about how the sequence is generated to calculate the distribution. These assumptions are the model of the sequence. If the model gives a high probability to the actual value of the next symbol, good compression is obtained. A model that produces good compression makes good predictions and is a good description of the data. More recently, the PAQ compressors [64] are the most successful and popular statistical applications (see `http://mattmahoney.net/dc/` for more).

2.2.1 Finite-context models

Finite-context modeling (FCM), a statistical method assuming the Markov property, has been used in several areas, such as in text and image. Although FCM generally has good performance, they are limited by the number of symbols of the alphabet (memory grows exponentially). For data sources with restricted alphabet size, normally less than 60, it seems to attain top compression ratios.

Consider an information source that generates symbols, s, from a finite alphabet $\Theta = \{s_1, s_2, \ldots, s_{|\Theta|}\}$, where $|\Theta|$ denotes the size of the alphabet. In the case of DNA data, $\Theta = \{\mathrm{A}, \mathrm{C}, \mathrm{G}, \mathrm{T}\}$ and, therefore, $|\Theta| = 4$. Also, consider that the information source has already generated the sequence of n symbols $x^n = x_1 x_2 \ldots x_n,\ x_i \in \Theta$. A subsequence of x^n, from position i to j, is denoted as x_i^j.

A FCM of an information source (see Fig. 2.7) assigns probability estimates to the symbols of the alphabet, according to a conditioning context computed over a finite and fixed number, M, of past outcomes (order-M FCM) [59, 65, 66]. At time n, these conditioning outcomes are represented by $c = x_{n-M+1}, \ldots, x_{n-1}, x_n$. The number of conditioning states of the model is $|\Theta|^M$, dictating the model complexity or cost. In the case of DNA, an order-M model implies 4^M conditioning states.

Previously, FCMs have been used with low-orders. Currently, and in the case of genomic sequences, high orders have also been considered. In fact, high orders are the most significant in terms of compression capabilities, however they also require more space/time resources.

The probability estimates, $P(s|c), \forall_{s \in \Theta}$, are usually calculated using symbol counts that are accumulated while the sequence is being processed, which makes them dependent not only of the past M symbols, but also of n. In other words, these probability estimates are

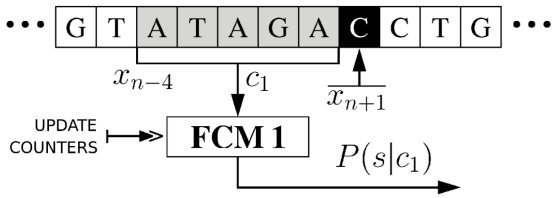


Figure 2.7: FCM: the probability of the next outcome, x_{n+1}, is conditioned by the M previous outcomes. In this example, $\Theta = \{\mathrm{A}, \mathrm{C}, \mathrm{G}, \mathrm{T}\}$ and $M = 5$.

Table 2.1: Simple example illustrating how statistical data are typically collected in FCMs. Each row of the table represents a probability model at a given instant n. In this example, the particular model that is chosen for encoding a symbol depends on the last five processed symbols (order-5 context).

Context, c	n_{A}^c	n_{C}^c	n_{G}^c	n_{T}^c	$n^c = \sum_{a\in\Theta} n_a^c$
AAAAA	23	41	3	12	79
⋮	⋮	⋮	⋮	⋮	⋮
ATAGA	16	6	21	15	58
⋮	⋮	⋮	⋮	⋮	⋮
GTCTA	19	30	0	4	53
⋮	⋮	⋮	⋮	⋮	⋮
TTTTT	8	2	18	11	39

generally time varying.

Table 2.1 shows an example of how statistical data are usually collected in FCMs. In this example, an order-5 FCM is presented (as that of Fig 2.7). Each row represents a probability model that is used to represent a given symbol according to the last processed symbols (five in this example). The counters are updated each time a symbol is processed.

The theoretical per symbol information content average provided by the FCM, after having processed n symbols, is given by

$$H_n = -\frac{1}{n}\sum_{i=1}^{n} \log_2 P(x_i|c) \quad \text{bpb}, \tag{2.1}$$

where "bpb" stands for "bits per base". Recall that the entropy of any sequence of four symbols is limited to two bits per symbol, a value that is obtained when the symbols are independent and equally likely.

The probabilities are estimated using the parameter α. Accordingly, we address the information content estimation process under the form

$$P_\alpha(s|c) = \frac{n_s^c + \alpha}{n^c + \alpha|\Theta|}, \tag{2.2}$$

where n_s^c represents the number of times that, in the past, the information source generated symbol s having c as the conditioning context and where n^c is the total number of events

that has occurred so far in association with context c. Parameter α allows balancing between the maximum likelihood estimator and an uniform distribution. Note that when the total number of events, n, is large, the estimator behaves as a maximum likelihood estimator. For $\alpha = 1$, (2.2) is the Laplace estimator.

The combination of multiple FCMs can be mainly addressed using two schemes: competition and cooperation (or mixture). These are explained in the following subsections.

2.2.1.1 Competition

DNA sequence data are non-stationary. In fact, one of the reasons why most DNA encoding algorithms uses a mixture of two methods, one based on repetitions and the other relying on low-order FCMs, is to try to cope with the non-stationary nature of the data. We also follow this line of reasoning, i.e., that of using different models along the sequence. However, unlike the other approaches, we use exclusively the finite-context paradigm for modeling the data, changing only the order of the model as the characteristics of the data change. More precisely, we explore an approach based on multiple FCMs of different orders that compete for encoding the data.

Using several models with different orders allows a better handling of DNA regions with diverse characteristics. Although these multiple models are continuously updated, only the best one is used for encoding a given region. For convenience, the DNA sequence is partitioned into non-overlapping blocks of fixed size, which are encoded by one (the best one) of the FCMs. Figure 2.8 shows an example where two competing FCMs are used. In this example, each model collects statistical information from a context of depth $k_1 = 5$ and $k_2 = 11$, respectively. At time n, the two conditioning contexts are $c_1 = x_{n-k_1+1} \dots x_{n-1} x_n$ and $c_2 = x_{n-k_2+1} \dots x_{n-1} x_n$.

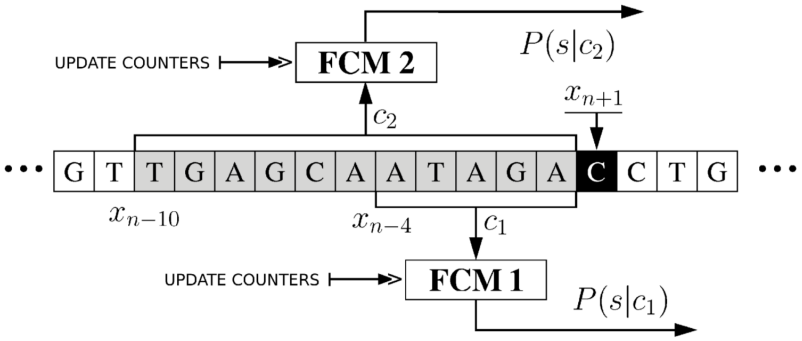


Figure 2.8: Example of the use of multiple FCMs for encoding DNA data. In this case, two models are used, one with a depth-5 context and the other using an order-11 context.

2.2.1.2 Mixture

In a mixture of FCMs, instead of having competition, each model cooperates according to weights attributed. The weights are continuously adapted during compression, depending on the performance of each individual probabilistic model. After seeing the first n symbols

of x, x^n, the average number of bits generated by an order-k FCM is[1]

$$H_{k,n} = -\frac{1}{n}\sum_{i=1}^{n} \log P(x_i|x_{i-k}^{i-1}), \tag{2.3}$$

where, for simplicity, we assume the convention that $x_i^i = x_i, i \leq 0$ is known to both the encoder and decoder. When several models are used simultaneously, $H_{k,n}$ can be viewed as a measure of the average performance of model k [2] until position n. Therefore, the overall probability estimated for position $n+1$ is given by the weighted average of the probabilities provided by each model, according to their individual performance, i.e.,

$$P(x_{n+1}) = \sum_{k\in\mathcal{K}} P(x_{n+1}|x_{n-k+1}^{n})\ w_{k,n}, \tag{2.4}$$

where $\mathcal{K}$ denotes the set of $K = |\mathcal{K}|$ models involved in the mixture, and

$$w_{k,n} = P(k|x^n), \tag{2.5}$$

i.e., these weights are equal to the probabilities that each model has generated x^n. Hence, we have

$$w_{k,n} = P(k|x^n) \propto P(x^n|k)P(k), \tag{2.6}$$

where $P(x^n|k)$ denotes the likelihood of sequence x^n being generated by model k and $P(k)$ denotes the prior probability of model k. Assuming

$$P(k) = \frac{1}{K}, \tag{2.7}$$

we also obtain

$$w_{k,n} \propto P(x^n|k). \tag{2.8}$$

Calculating the negative logarithm of this probability we get

$$-\log P(x^n|k) = -\log\prod_{i=1}^{n} P(x_i|k, x^{i-1}) = -\sum_{i=1}^{n} \log P(x_i|k, x^{i-1}), \tag{2.9}$$

which is the number of bits that would be required by model k to represent the sequence x^n. It is, therefore, the accumulated measure of the performance of model k until position n.

To facilitate faster adaptation to non-stationarities of the data, instead of using the whole accumulated performance of the model, we adopt a progressive forgetting mechanism. The idea is to let each model to progressively forget the distant past and, consequently, to give more importance to recent performance results. To accommodate this, we first write (2.9) as

$$\log P(x^n|k) = \sum_{i=1}^{n-1} \log P(x_i|k, x^{i-1}) + \log P(x_n|k, x^{n-1}) \tag{2.10}$$

and then

$$\log p_{k,n} = \gamma \log p_{k,n-1} + \log P(x_n|k, x^{n-1}), \tag{2.11}$$

[1] For now on, we consider base-2 logarithms.

[2] For simplicity of notation, although incurring in some loss of generality, we identify a FCM by its order. However, as explained below, we may have more that one model of the same order in the mixture.

where $\gamma \in [0, 1)$ is the forgetting factor and $\log p_{k,n}$ represents the estimated number of bits that would be required by model k for representing the sequence x^n (we set $p_{k,0} = 1$), taking into account the forgetting mechanism. Removing the logarithms, we rewrite (2.11) as

$$p_{k,n} = p_{k,n-1}^{\gamma} P(x_n|k, x^{n-1}) \tag{2.12}$$

and, finally, set the weights to

$$w_{k,n} = \frac{p_{k,n}}{\sum_{k \in \mathcal{K}} p_{k,n}}. \tag{2.13}$$

2.2.2 Genomic compression

Although the success of the general purpose algorithms in many types of data, in genomic data they seem to be, in several aspects, behind specific purpose algorithms (whose existence is already an indicator). This happens because of the nature of the data, which, besides being very heterogeneous and non-stationary, has specific properties, such as inverted repeats [67, 68] (see subsection 2.2.2.3 for a definition and how we address their modeling).

The field of genomic sequence compression can be mainly divided in two areas:

1. Individual compression (pure compression);
2. Reference compression (conditional compression).

2.2.2.1 Individual compression

The individual sequence compression has arrived with the first sequenced genomes, where the purpose was initially to decrease the space of the transmitted data over the internet (very low speed connections at that time). Currently, it is used due to space constrains and efficiency in data manipulations. Nonetheless, the advances in the core of these models (and also in reference sequence compression) are always a way to approximate the optimal function (or program) that represents the object (or distance between objects).

In the literature, since Biocompress [69], several individual genomic compression algorithms have been proposed [70, 71, 72, 73, 74, 75, 76, 77, 78, 79, 80, 81, 82, 83, 84, 85, 86, 87, 88] (for reviews see [21, 89, 90]).

These algorithms have a common idea, which arises from the non-stationary nature of the genomic sequences. Typically, there are at least two compression methods, one based on Lempel-Ziv substitutional procedures [56] and another based on low-order FCM arithmetic coding. According to the substitutional paradigm, repeated regions of the genomic sequence are represented by a reference to a past occurrence of the repetition and the length of the repeating sequence. Both exact and approximate repetitions have been explored, as well as their inverted complements. In the case of approximate repetitions it is necessary to indicate where and how the sequences differ.

The substitutional approach is usually the main encoding method, with the low-order FCM assuming the role of a fallback, secondary choice. When the substitutional method is unable to provide satisfactory performance, the corresponding region of the genomic sequence is represented by a low-order FCM. This scheme for representing genomic sequences have been significantly improved by Tabus and Korodi *et al.*, based on the normalized maximum

likelihood (NML) algorithm [77, 79, 81], and mostly by Cao *et al.* [82], using the eXpert Model (XM).

The most recent version of the NML-based approach [81] is an evolution of the normalized maximum likelihood model introduced in [77] and improved in [79]. This new version, NML-1, aims at finding the best regressor block, i.e., an approximate repetition, using first-order dependencies. One of the drawbacks of the substitutional approaches is the associated computational complexity. In fact, most of the CPU time required by these encoding techniques is spent on finding good exact/approximate repeats or inverted complements. The authors of NML-1 reported that it took about 42 hours to compress a human genome (considering a single machine).

The XM statistical method comprises three types of experts: order-2 FCMs (see Subsection 2.2.1); order-1 local FCMs (typically using information only from the 512 previous symbols); the copy expert, that considers the next symbol as part of a copied region from a particular offset. The probabilities provided by the set of experts are combined using Bayesian averaging and then directed to an arithmetic encoder. In general, XM has the best space compression results. However, it uses high memory and time resources to compress larger sequences (such as a chromosome or human genome).

To changing the paradigm of the compression methods, we have proposed the exclusive usage of FCMs [91, 92, 84], where, first, using a competing and, after, a mixing approach. The better version emerged as FCM-Mx [84]. The FCM-Mx, a pure statistical method, relies on a mixture of multiple order-k FCMs with a special technique to deal with inverted repeats (see Subsection 2.2.2.3), where the probabilities are weighted and then directed to an arithmetic encoder. In general, the FCM-Mx presents competitive or even superior results, for example in bacterial genomes [84, 91], than XM, using significantly less computational time and memory.

2.2.2.2 Reference compression

The dramatic increase of sequenced genomes [93], given the dramatically reduced sequencing costs, with the high redundancy characteristics, led to the development of genomic reference sequence compression. Some models for storing and communicating redundant genomic data have already been presented, based on, for example, single nucleotide polymorphism (SNP) databases [94], or insert and delete operations [95].

Wang *et al.* proposed a compression tool, GRS, that is able to compress a sequence using another one as reference, without requiring any additional information about those sequences, such as a reference SNP map [96]. RLZ, a compressor able to perform relative Lempel-Ziv compression of DNA sequences was proposed by Kuruppu *et al.* [97].

Other approaches propose encoding the sequence reads that are output by massively parallel sequencing experiments (e.g., [98, 99, 100, 101]), which is also a very important problem and which shares some common points with the problem being addressed here. However, the compression of short reads needs to cope with other requirements, such as, for example, the efficient representation of base calling quality information.

We have proposed GReEn [102], a compression tool based on arithmetic coding that handles arbitrary alphabets. Its running time and memory depends on the size of the sequence being compressed. The model is mostly inspired in the copy expert [82]. The probability distribution, $P_n(c)$, can be provided by two different sources: (a) an adaptive model (the copy model) which assumes that the characters of the target sequence are an exact copy of (parts of) the reference sequence; (b) a static model that relies on the frequencies of the

Table 2.2: Table 2.1 updated after processing symbol "C" according to context "ATAGA" (see example of Fig. 2.7) and taking the inverted repeats property into account.

Context, c	n_{A}^c	n_{C}^c	n_{G}^c	n_{T}^c	$n^c = \sum_{a \in \Theta} n_a^c$
AAAAA	23	41	3	12	79
⋮	⋮	⋮	⋮	⋮	⋮
ATAGA	16	**7**	21	15	59
⋮	⋮	⋮	⋮	⋮	⋮
GTCTA	19	30	10	**5**	64
⋮	⋮	⋮	⋮	⋮	⋮
TTTTT	8	2	18	11	39

characters in the target sequence. The adaptive model is the main statistical model, as it allows a high compression rate of the target sequence, particularly in areas where the target and reference sequences are highly similar. The static model will act as a fallback mechanism, feeding the arithmetic coder with the required probability distribution.

For highly similar sequences, Wandelt *et al.* proposed an algorithm based on dictionary schemes [103], while, for large collections, Deorowicz *et al.* proposed GDC [99, 104]. Finally, Ochoa *et al.* proposed iDoComp [105], using mostly a suffix-array and dictionary scheme. Generally, this method seems to have top compression results.

2.2.2.3 Updating the inverted complements

Frequently, DNA sequences contain sub-sequences that are reversed and complemented copies of some other sub-sequences. These sub-sequences are named "inverted repeats". As mentioned before, this particularity of DNA sequence data is used by most of the DNA encoding methods that have been proposed and that rely on the sliding window searching paradigm.

For exploring the inverted repeats of a DNA sequence in the FCMs, besides updating the corresponding counter after encoding a symbol, we also update another counter that we determine in the following way [106]. Consider the example given in Table 2.1, where the context is the string "ATAGA" and the symbol to encode is "C". Reversing the string obtained by concatenating the context string and the symbol, i.e., "ATAGAC", we obtain the string "CAGATA". Complementing this string (A↔T, C↔G), we get "GTCTAT". Now we consider the prefix "GTCTA" as the context and the suffix "T" as the symbol that determines which counter should be updated. Therefore, according to this procedure, we take into consideration the inverted repeats if, after encoding symbol "C" of the example in Table 2.1, the counters are updated according to Table 2.2. As shown in [106], this provides additional modeling performance.

2.3 Conclusions

In this Chapter, we have introduced the basics of the biological processes, namely from DNA to protein. Also, we have explained the problems of the representation of genomic sequences in digital format and its most used file formats.

Moreover, we have described the Finite-Context Models (FCMs) and its relations: competition and cooperation (or mixture). These are the building blocks of most of the algorithms that we developed during this work.

Finally, we have presented the state-of-the-art in genomic sequence compression, namely describing reference and reference-free methods.

"Who is uncompressing all this stuff?"

Jarvis

3

Genomic sequence compression for storage

In this chapter we address the efficient compression of genomic sequences for storage purposes. We explore a specific reference-free compressor, introducing an algorithmic entropy filter. On the other hand, we introduce a general, in multiple purpose sense, genomic compressor integrating several new research topics, such as cache-hashes, extended FCMs and mixing different classes.

Genomic sequences, besides being very heterogeneous, non-stationary [91], with specific properties (such as inverted repeats [68]), are nowadays, very large [21]. To detect similar blocks of information very apart (such as in collections or complete genomes) common compressors load into memory the full sequence. This means that, if the sequence is very large, then the compressor will spend all available memory resources.

On the other hand, to run with controlled memory, most compressors create a buffered internal model of the data with limited size. However, most of the very far away blocks of information will be treated as if they were new (that will result in a inefficient compression).

As introduced in Section 2.2, XM and FCM-Mx are the current state-of-the-art compression methods for genomic sequences (compression capability, running time and memory used). Unfortunately, in average laptop computers they can not be used without splitting the sequences into smaller sub-sequences. One has to cut the sequences into smaller ones to ensure minimal RAM memory, specially with XM. This has a strong impact in the compression capability, namely in complete genomes (such as the human).

To deal with these issues and to improve the genomic compressor, we have studied, used and created several techniques to support and extend the usage of deep context models, namely an algorithmic entropy filter, cache-hash and extended FCMs, that we describe in the following subsections (from 3.1 to 3.2.3).

3.1 An algorithmic entropy filter

DNAEnc3 [85, 91], the competitive FCMs version (instead of the cooperation between FCMs as in FCM-Mx), is a top genomic compressor that presents reasonable balance between

running resources (time and memory) and compression results, being able to easily compress sequences with less than 200 MB in a few minutes and using up to 3 GB of memory. The method is based on multiple Markov models, with variable context orders, that compete to encode every block of the genomic sequence. DNAEnc3 is limited to contexts up to 16, where FCMs above contexts of size 14 use a linear hash table.

It is known that when the size of the sequences and the redundancy increases, the context need to be larger for a decent compression performance [91]. However this might be problematic in terms of memory, since the precision of the key in a hash table increases. By precision, we mean the number of bits needed to describe the interval of numbers (indexes) up to the hash size. On the other hand, the causality problem in data compression does not allow to know in a first time if a sub-sequence shares information with other, but only after seeing it a second time.

To tackle this we need to develop a data structure that filters the sequence in terms of information content and only updates the hash table with homologous regions, at the expense of some more time. Moreover, we are aware that the need to be fast and use low memory is on the side of the decompression, mainly because the compression of these datasets is normally done in computers with high hardware capabilities (and only once), while the decompression is done by very different (and sometimes modest) computers (and many times). Usually it is done one compression for thousands of decompressions.

We follow the line of DNAEnc3 [85, 91], exploring two competing Markov models with a low and high context order. However, unlike DNAEnc3, the proposed approach rely on deep context orders and on a preprocessing analysis to identify low complexity regions of the data, and hence the algorithmic entropy filter. This strategy allows the reduction of memory usage and, consequently, allows to use deeper contexts that positively impact the compression gain.

3.1.1 Multiple finite-context models

As mentioned, one of the reasons why most DNA encoding algorithms use a mixture of two methods, one based on repetitions and the other relying on low-order FCM, is to try to cope with the non-stationary nature of the data. We also follow this line of reasoning, using a low order FCM, typically 4, and a high order FCM that can be, in the case of DNA data, up to 32, where the high order context size depends on the size and repetitiveness of the sequence as shown in the results section.

As such, we explore an approach based on two FCMs of different orders (low and high) that compete for encoding the data, allowing a better handling of DNA regions with diverse characteristics. The competition between the two FCMs is held in the evaluation process for non-overlapping blocks of fixed size, such as 100 symbol blocks, which are then encoded by the best estimated FCM. The binary stream (side information) with the information of the respective FCM used in the compression of each block is encoded using an adaptive order-0 model followed by arithmetic coding.

3.1.2 Exploring high-order models using pre-analysis

Although the symbol counters for the low order FCM are constantly accumulated in a table with 16 bits of precision, the high order FCM requires two key approaches to maintain reasonable memory resources and the possibility of exploring deeper orders that might provide a better compression.

The first key approach is to use *sequence pre-analysis*, in order to classify the data blocks into low or high information content, reducing the number of them that otherwise would "pollute" the hash or counter table with incorrect statistics. Therefore, by applying this block-by-block analysis to the sequence, we are able to determine the blocks with low and high information content, and hence only update the high order model in the low information content blocks. As such, in the compression process, we spend more memory than in the decompression process. If we do not have the resources on the compression side, it is also possible to estimate these low and high information content blocks by using a smaller context order, only marginally reducing the performance. However, we recall that the importance of resources is generally on the side of the decompression phase and this is ensured to be light on memory and time.

The second key approach, enhanced by pre-analysis, is to use high-order hash-tables (indexes up to 2^{64}), mainly because when implementing the FCM using simple tables the memory requirements grow exponentially with k. For DNA data, and considering 16 bits counters, this would imply about 39.4 zettabytes of memory for implementing an order-32 model. However, this table would also be very sparse, because the maximum number of different words of size k that can be found in a sequence of length n is upper bounded by n. Therefore, using hash-tables, it is possible to explore large order FCMs having an approximate increase of memory proportional to the size of the sequence if data are random. Repetitive data mitigate the memory consumption, which is usually the case for genomic data, given its repetitive nature (due to homologous genes, transposons, centromeres, telomeres, among others).

3.1.3 The encoding process

The symbolic sequence is processed from left to right (LR), in order to create a binary sequence representing the blocks that correspond to low information regions, as depicted in Fig. 3.1. After that, from right to left (RL), the block sequence is updated only when a block is marked as a low information region and when the block sequence had the previous index marked in the LR as a high information block. This corresponds to the maximum of the LR and RL cases.

Finally, the sequence block information is used to compress the sequence with the high-order FCM, which is associated to the lower information content region blocks, and the low-order FCM, that is associated to the higher information content region blocks. In practice, the high-order FCM is used to compress regions that can be evaluated as low information content regions by orders lower or equal to its own.

The *inverted repeats* (IR) particularity of DNA sequence data is explored by most of the DNA specific compression methods, given the additional modeling performance (specially in high orders). Therefore, the IR are also explored in the high FCM (in the analysis and compression stages), where after encoding a symbol the respective sub-sequence IR counters are also updated in the same model. Specifically, the high FCM is constituted by two complete chains, the *regular* chain and the IR chain.

3.1.4 Software availability

The tool (HighFCM), written in C language, with the implementation of the method is available at `http://bioinformatics.ua.pt/software/highfcm`, under GPL-2, and can be applied to any textual sequence data. We have adopted a variable input alphabet, allowing

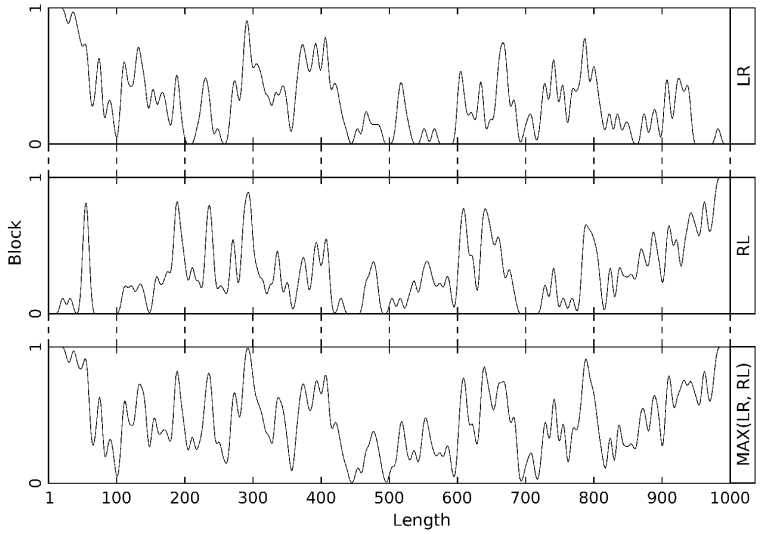


Figure 3.1: Block sequence profiles from 100 KBases of the human chromosome 22, processed in left-to-right mode (LR) and right-to-left mode (RL), and computing the maximum of LR and RL. Block value 0 indicates a low-order FCM (order-4) whereas 1 indicates a high-order FCM (order-16). The block size used was 100. Filtering: Blackman window of size 10.

this compressor to run in sequences with alphabets up to 256 symbols. Moreover, it is very flexible, since it allows a variable multi-thread approach, defined by the user, as well as the possibility of compressing with hash-tables and decompressing with regular tables (or vice versa).

3.1.5 Results

The experiments have been performed on a Linux server running Ubuntu with 16 Intel(R) Xeon(R) CPU E7320 at 2.13 GHz and with 256 GB of RAM. Three datasets have been used: Escherichia and Salmonella (bacterial) collections from NCBI (`ftp://ftp.ncbi.nlm.nih.gov/genomes/`) and a collection of 20 human chromosomes 22 (very repetitive) from the 1000 genomes project (`http://www.1000genomes.org/data`).

We have estimated only the best high order, as shown in Fig. 3.2, where the compression ratios using different context orders of the dataset are presented. As depicted, the high context orders have a fundamental role in the compression, namely in the larger sequence, since the best compression ratio is achieved with the higher order (supported by the current implementation).

Relatively to benchmark results, as shown in Table 3.1, in the second and third sequences, the proposed approach has the best compression ratio, compared to other existing techniques, while it stands out in the last sequence (eukaryotic genomic collections) with almost 50% reduction relatively to DNAEnc3. Moreover, the memory spent and time usage, in all sequences, seems to be reasonable since the maximum is only slightly higher than 3 GB. General purpose algorithms (Gzip, Bzip2, lzma) are not capable of handling efficiently this type of sequences, while specific FASTA tools (MFCompress and Deliminate) seem to be between general purpose and pure reference-free algorithms (DNAEnc3, XM and HighFCM). Although the XM method had the best compression result in the first sequence (the smallest one), it spent huge

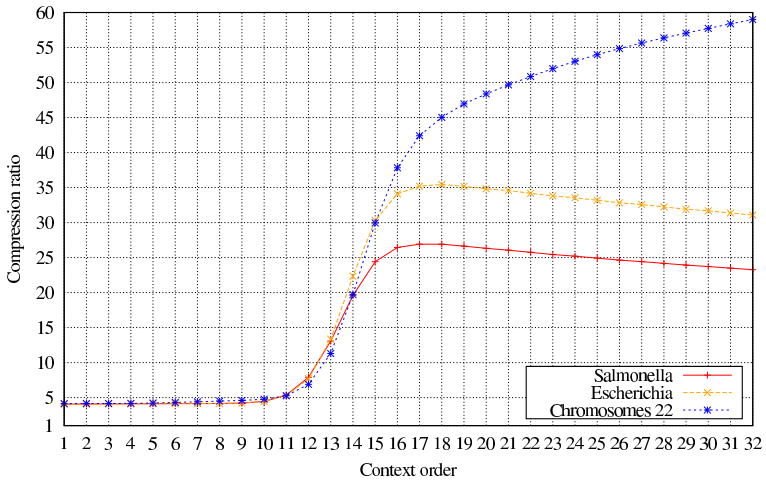


Figure 3.2: Compression ratios as a function of the variation of the order of the deeper context, for all dataset sequences (static low order: 4).

amounts of memory and time (both in compression and decompression). Moreover, it was not able to handle larger sequences, returning a runtime error.

To conclude, we have seen that applying preprocessing analysis techniques before compression can substantially improve the savings in memory resources, particularly in the decompression process. Moreover, these savings, together with appropriate high-order hash-tables, yield tremendous improvements in the compression ratio, specially in highly repetitive genomic sequences.

3.2 Universal genomic sequence compressor

3.2.1 Cache-hash

Although the success on the usage of the algorithmic entropy filter, it provides an asymmetrical scheme with compression times 3x higher than the time used for decompression. For storage purposes this is an advantage, since we want to minimize memory and time resources on the decompression side. However, for analysis, we only rely on the compression part method. During this work, one of the objectives was to create a genomic compressor that was able to obtain state-of-the-art results, using the minimal time and memory resources, but at the same time flexible in the sense of memory optimization for any computer hardware specification.

Accordingly, we have developed a cache-hash approach that keeps only the last hashed entries in memory, rendering a flexible and predictable quantification of the memory necessary to run in any sequence (memory does not blow with the size of the sequence). Depicted in Fig. 3.3, the cache-hash is able to store each index primarly by exploring precision splitting. For example, if the context order size can be up to 20 and the hash table has size 24 bits (specifically the next prime after 2^{24}), each hashed entry uses 24 bits for the "INDEX A" and 16 bits for the "INDEX B". This entry will be added or updated (only if already exists) according to the "POSITION" (circular buffer having the "MAXIMUM COLLISIONS" as size). An advantage is that we only need to store "INDEX B" since the full index can be

Table 3.1: Compression and decompression benchmarks. The memory (called "Mem" and expressed in MBytes) has been estimated with *valgrind*, using *massif*, while running time (in seconds) with the *time* Linux program. It was not possible to obtain two results using XM, due to a program error. "MFC" stands for MFCompress, while "DELIM" for DELIMINATE.

Dataset	Method	Mode	Compression			Decompression	
			Size (B)	Time	Mem	Time	Mem
Collection of	Gzip	-9 (best)	38,026,737	190	**6**	8	**6**
Salmonella	Bzip2	-9 (best)	36,707,571	39	15	24	11
	Lzma	-9 (best)	6,285,003	364	383	**7**	47
size: 130 MB	MFC	-3 (best)	5,572,546	184	2,329	126	2,329
	DELIM	-	11,199,769	**15**	8	19	8
	XM	500 experts	**3,480,673**	12,413	12,632	13,149	12,632
	DNAEnc3	4, 16 1/20, -ir	5,461,468	325	1,144	234	1,144
	HighFCM	4, 17 1/100 -ir	5,051,170	679	2,106	243	1,792
Collection of	Gzip	-9 (best)	85,356,853	399	**6**	**9**	**6**
Escherichia	Bzip2	-9 (best)	82,325,477	70	15	50	11
	Lzma	-9 (best)	14,240,205	2,150	383	12	47
size: 291 MB	MFC	-3 (best)	10,247,079	920	2,329	651	2,329
	DELI	-	19,156,665	**47**	656	45	68
	XM	500 experts	#	#	#	#	#
	DNAEnc3	4, 16 1/20, -ir	9,560,314	689	1,378	647	1,378
	HighFCM	4, 18 1/100 -ir	**8,615,784**	1,688	3,166	586	2,105
Collection of	Gzip	-9 (best)	183,918,347	875	**6**	**18**	**6**
20x human	Bzip2	-9 (best)	175,272,623	**138**	15	112	11
chromosomes	Lzma	-9 (best)	151,390,802	2,594	107	35	46
G22	MFC	-3 (best)	29,983,772	973	2,329	639	2,329
	DELIM	-	39,775,033	531	711	95	67
size: 664 MB	XM	500 experts	#	#	#	#	#
	DNAEnc3	4, 16 1/20, -ir	22,232,663	1,848	1,550	1,324	1,550
	HighFCM	4, 32 1/1000 -ir	**11,330,125**	1,821	3,087	625	1,770

disambiguated by the position where is it inserted or updated. For each block according to each "POSITION" we have the "INDEX B" (already described) and the "COUNTERS". The latter stores the counters with 2 bits precision in a unsigned char (4 symbols and 2 bits per symbol gives the 8 bits). Each time a counter reaches the value 3, all the counters (for the 4 symbols) are normalized, and hence, divided by two (only integer division). This would be problematic if the size of the context was small. A context of 20 is considered very large, resulting in a very sparse table, if represented in that sense. As such, the cache-hash in fact simulates a structure that can be seen as a middle point between a probabilistic and dictionary model.

The cache-hash uses a fixed hash function, based on a well known hash family of functions [107], defined as

$$h_{a,b}(x) = (ax + b) \mod p \tag{3.1}$$

where a and b are integers modulo p (p must be a prime) with $a \neq 0$. Specifically, for the average hashed entries and DNA nature and according to the lower probability of collisions (results reported in `http://planetmath.org/goodhashtableprimes`), we have chosen the

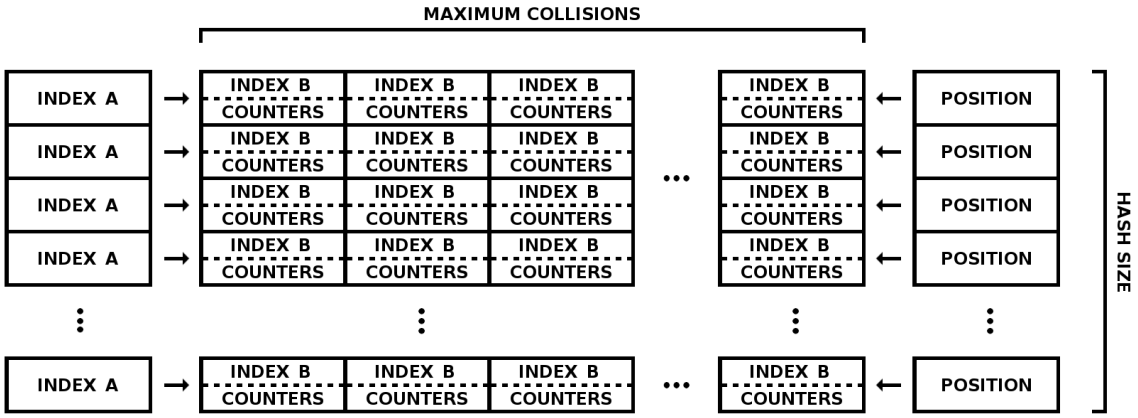


Figure 3.3: Cache-hash scheme. The "POSITION" indicates the position of the last edited block relatively to the hash index. Each hash index is shaped by both "INDEX A" and "INDEX B". The "COUNTERS" store the counts for each base. These are packed using a 2 bits per base approach that on overflow normalize all by half.

---

following parameters: $a = 786433, b = 196613, p = 68719476735$.

For searching in the cache-hash we need to compare each key, formed by "INDEX A" and "INDEX B", with the actual key, given by the context. This is a costly task, although needed. To minimize the processing time, we start the search from the previous position relatively to the current position, given by "POSITION", and search from newest to oldest entries. This is based on the characteristics of the genomic sequences that similar regions tend to be grouped or near, giving to the model faster searching times.

3.2.2 Extended FCMs

An extended FCM (XFCM) uses the memory from a FCM with the same context-order size although assigning a probability estimate that differs on the conditioning context that is assumed to be seen. As such, for a conditioning context it is considered that s has always been the most probable symbol, and hence the estimator

$$P(s|x'^{n}_{n-k+1}) = \frac{N(s|x'^{n}_{n-k+1}) + \alpha}{N(x'^{n}_{n-k+1}) + \alpha|\Theta|}, \tag{3.2}$$

where x' is a copy of x that is edited according to

$$x'_{n+1} = \underset{\forall s \in \Theta}{\operatorname{argmax}} P(s|x'^{n}_{n-k+1}). \tag{3.3}$$

This strategy enables to modify the context assumed to be seen without modifying the main model memory. Since these models only make sense in low complexity regions, we have created a way to turn them on or off, saving some time in the computation.

For the purpose, we permit t substitutions on the conditioning context k without being discarded and, hence, turned off. For example, consider that $s_0 = AGATATAGAGA$ and the past symbol occurrences are according to $A = 3, C = 0, G = 0, T = 0$, if the symbol that is being compressed is T (contradicting the probabilistic model), in a *regular* FCM we now will have a $s_1 = GATATAGAGAT$. On the other hand, the XFCM will assign a s_1 according to

the most probable outcome ($s_1 = GATATAGAGAA$). As such, the next probabilistic model will be dependent on the past context assumed to be seen and, hence, it assumes that the symbol that has been compressed is A.

The XFCM works well when using high orders. This also creates sparse occurrences that can be efficiently supported by a cache-hash memory and, therefore, from now on we will use cache-hashes when we use the XFCMs.

The XM algorithm [82] uses a copy-model that gives top compression genomic results although at expense of huge amounts of memory and time. Our intention with the XFCM is to approximate the copy-model using low and controllable memory, and to be as fast as possible.

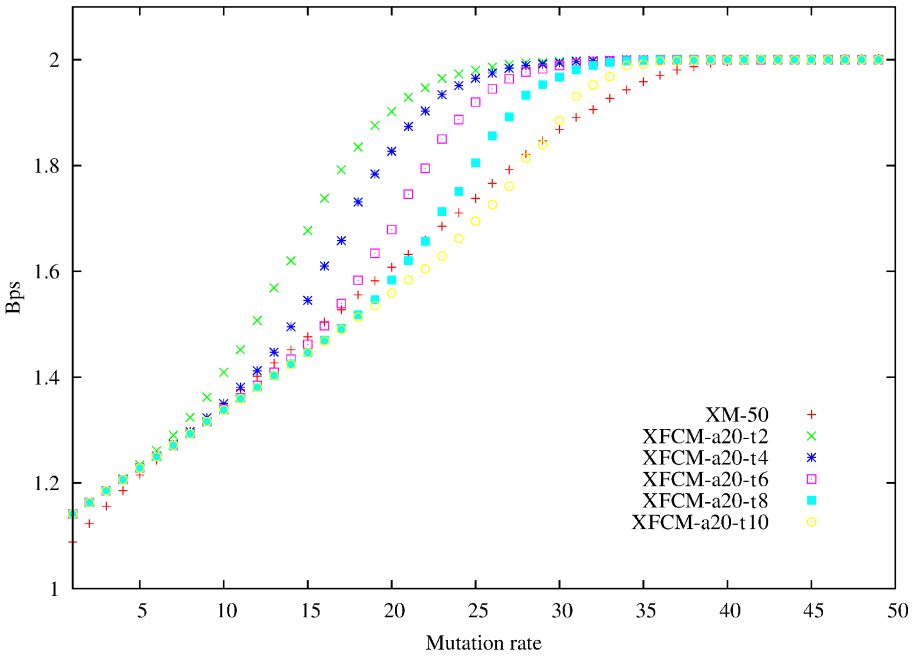


Figure 3.4: Extended FCM performance, varying the substitution threshold, over mutated data and using the original as reference. The "a" represents the denominator of the parameter of the estimator, $\alpha = 1/a$, while "t" represents the t permitted substitutions without being discarded. The "XM-50" stands for the XM compression model using 50 experts.

For simplicity, we have generated a synthetic sequence (using XS [108]), and mutated the sequence with a defined substitution rate. Our intention is to simulate genomic sequences given several degrees of mutations (and removing the self-redundancy of the sequence) and compress it using the original as reference. In Fig. 3.4 we show the performance of the XFCMs, compared with the XM model, using several values for parameter t. It can be seen that for a high value of t the model adjusts better to the nature of the data.

In Fig. 3.5 it is possible to see that a *regular* FCM is not able to deal with mutations as XFCMs and XM does. This was a disadvantage on our past models. Accordingly, when we mix FCM and XFCM ("FCM-Mix") we are able to get results even better that XM. Moreover, we do not have two memory models, since they share the same cache-hash (because they have the same context order). In fact, for the same statistics we have different predictors that cooperate according to a mixture.

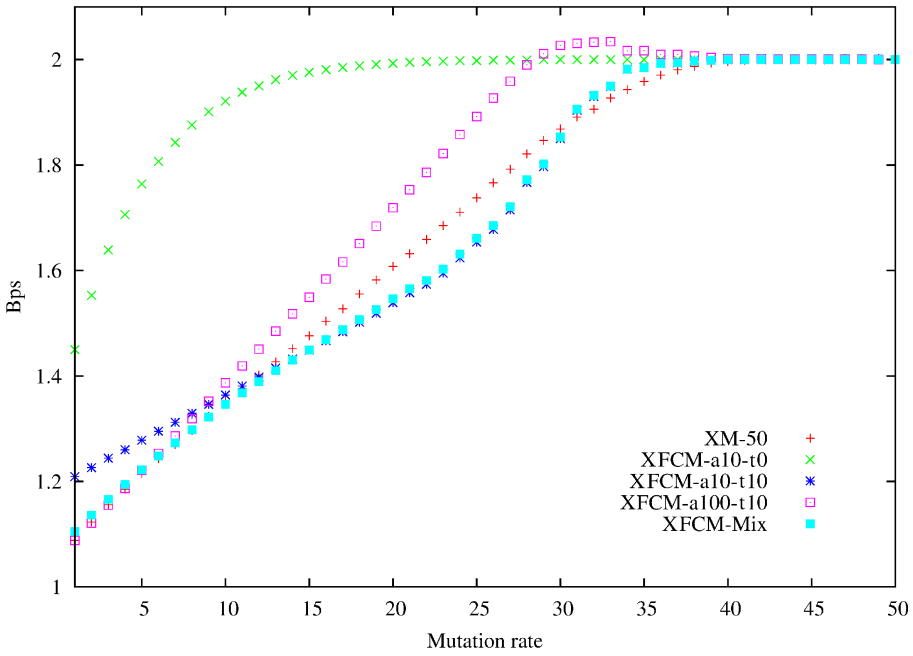


Figure 3.5: Extended FCM performance, varying the substitution threshold, over mutated data and using the original as reference. The "a" represents the denominator of the parameter of the estimator, $\alpha = 1/a$, while "t" represents the t permitted substitutions without being discarded. The "XM-50" stands for the XM compression model using 50 experts. The "FCM-MIX" uses a XFCM mixed with a FCM (the same as having $t = 0$) of the same context order.

3.2.3 Mixture of classes

The mixture is based on two model classes, those belonging to what we call reference set, $\mathcal{R}$, and those in the target set, $\mathcal{T}$. The reference set contains the FCMs or XFCMs responsible for modeling the conditioning string, i.e., the y of $C(x|y)$, whereas the target set of FCMs or XFCMs is used to represent x, when required. The $C(x|y)$ represents the number of bits when compressing object x given y and, thus, it can be seen as conditional compression.

Basically, the probability of the next symbol, x_{n+1}, is given by

$$P(x_{n+1}) = \sum_{k \in \mathcal{R}} P_r(x_{n+1}|x_{n-k+1}^n)\ w_{k,n}^r + \sum_{k \in \mathcal{T}} P_t(x_{n+1}|x_{n-k+1}^n)\ w_{k,n}^t, \tag{3.4}$$

where $P_r(x_{n+1}|x_{n-k+1}^n)$ and $P_t(x_{n+1}|x_{n-k+1}^n)$ are, respectively, the probability assigned to the next symbol by a model from the reference set and from the target set, and where $w_{k,n}^r$ and $w_{k,n}^t$ denote the corresponding weighting factors, with

$$w_{k,n}^r \propto (w_{k,n-1}^r)^\gamma P_r(x_n|c_{k,n-1}) \tag{3.5}$$

and

$$w_{k,n}^t \propto (w_{k,n-1}^t)^\gamma P_t(x_n|c_{k,n-1}), \tag{3.6}$$

constrained to

$$\sum_{k \in \mathcal{R}} w_{k,n}^r + \sum_{k \in \mathcal{T}} w_{k,n}^t = 1. \tag{3.7}$$

To compute $C(x|y)$, the compression is performed in two phases. In the first phase, the $\mathcal{R}$ class of models accumulates the counts regarding the y object. After the entire y object was processed, the models are kept frozen and, hence, the second phase starts. At this point, the x object starts to be compressed using the $\mathcal{R}$ models computed during the first phase, in cooperation with the set of models of the $\mathcal{T}$ class, that dynamically accumulate the counts only from x.

3.2.4 Software availability

The tool (GeCo), written in C language, with the implementation of the method is available at `http://bioinformatics.ua.pt/software/geco`, under GPL-2, and can be applied to any genomic sequence data. GeCo can also be used to sequence analysis, with capability to determine absolute measures, namely for many distance computations, and local measures, such as the information content contained in each element, providing a way to quantify and locate specific genomic events. GeCo can handle individual compression and referential compression.

3.2.5 Results

The experiments have been performed on a Linux server running Ubuntu with 16 Intel(R) Xeon(R) CPU E7320 at 2.13 GHz and with 256 GB of RAM.

We have used seven datasets. Three datasets were used from the recent sequenced birds project [109], namely duck[1], peregrine falcon[2] and cuckoo[3]. All of the birds datasets were not assembled (in contig state), totaling more than 3.2 GB, and thus, they can be seen as multi-FASTA derived format. On the other hand, the fourth dataset was the assembled human genome GRC-b38 [4] (with 24 sequences/chromosomes) totaling an approximation size of 2.9 GB. The fifth dataset contains a FASTQ file used in [110] [5], while the sixth can be achieved through [6]. For fair comparison, the datasets have been filtered and transformed into equivalent files in order to compare only genomic sequences (ACGT), using Goose framework `https://github.com/pratas/goose/`. The final dataset, for reference compression, has been downloaded from NCBI, including the assembled genomic sequences of human, chimpanzee, gorilla and orangutan (chromosomes 5,11 and 18).

As it can be seen in Table 3.2, Gzip was unable to compress the human genome below 2 bits per base. On the other hand, the specific methods were able to compress with success. DNACompact [88] used a very small amount of memory but at the expense of more computational time. The proposed algorithm, known as GeCo, provides a substantial compression improvement to our previous algorithm (DNAEnc3 [85]) using much less resources. In fact, GeCo is able to compress the human genome in less than 550 MB using memory equivalent to a laptop computer and much faster than the previous approaches. Moreover, memory will not explode with the size of the sequence, unlike DNAEnc3 and XM [82]. XM was unable to compress two of the birds datasets, due to a processing error.

[1] `ftp://climb.genomics.cn/pub/10.5524/101001_102000/101001/duck.scafSeq.gapFilled.noMito`
[2] `ftp://climb.genomics.cn/pub/10.5524/101001_102000/101006/peregrine.FG.2011.0223_sca.bk.fa`
[3] `ftp://climb.genomics.cn/pub/10.5524/101001_102000/101009/Cuculus_canorus.fa.gz`
[4] `ftp://ftp.ncbi.nlm.nih.gov/genomes/H_sapiens/Assembled_chromosomes/seq/`
[5] `ftp://ftp.ddbj.nig.ac.jp/ddbj_database/dra/fastq/SRA001/SRA001546/SRX000706/`
[6] `ftp://ftp.sra.ebi.ac.uk/vol1/fastq/SRR494/SRR494099/SRR494099.fastq.gz`

Table 3.3 depicts the benchmarks of two datasets using several tools, namely two state-of-the-art FASTA dedicated tools: Deliminate [111] and MFCompress [112]. Moreover, two state-of-the-art FASTQ tools are also used: faz_comp [110] and Orcom [113]. As it can be seen, GeCo provides substantial compression capabilities although at expense of more computational time. The tool Orcom, an efficient disk-based tool, was not able to address with much success these files, perhaps because it is more suitable for much larger files.

Table 3.4 includes several reference-based compressed tools, namely GReEn [102], iDoComp [105], and GeCo (proposed). We have also ran GRS [96], however, as similar as in [103], the sequences have some degree of dissimilarity and therefore the programs are not suitable for this purposes (GRS even suggests for not to be used in these cases). Moreover, GDC versions [99, 104] are suitable for large collections, since most of the cases they reported a compression value above two bits per base. Notwithstanding, we can see that on average GeCo outperforms the specific reference compressors at the expense of some more time and memory. In fact, we have only used the first reference mode (-l 11) and, therefore, the compression factor might increase although at the expense of more space/time resources.

3.3 Conclusions

In this chapter, we have introduced a new compressor based on an algorithmic entropy filter. The compressor uses multiple models that compete to represent a specific block of symbols. However, it applies a preprocessing analysis technique, separating the high regions of complexity from the low. This computation before compression can substantially improve the savings in memory resources, particularly in the decompression process. Moreover, these savings, together with appropriate high-order hash-tables, yield tremendous improvements in the compression ratio, specially in highly repetitive genomic sequences, such as genomic collections.

Despite the success on the compressor, it provides an asymmetrical compression and decompression time. For storage purposes, this is an advantage, since we want to minimize memory and time resources on the decompression side. However, for analysis, we only rely on the compression part. Accordingly, we have presented a universal genomic compressor that can be applied successfully to any genomic sequence both for non-referential and referential compression. For this purpose, we used a mixture of Finite-Context Models of several orders given by two models classes (reference and target). For high orders we have created a cache-hash to ensure flexibility given hardware specifications. Moreover, we have introduced the eXtended Finite-Context Models, which can be seen as very flexible context models (fault tolerant). Finally, we have shown the very good adaptability of the compressor to multiple types and characteristics of genomic sequences.

Table 3.2: Compression benchmarks for state-of-the-art pure genomic compression tools. Time is in minutes, while maximum memory peak is in MBytes. With the exception of Gzip, the compressors are symmetric (time/memory compression and decompression are approximately the same. Symbol "*" means that the compressor processed the dataset by parts because of memory, time or testing purposes.

Dataset	Method	Mode	Compression		
			Bits p/ base	Time (m)	Memory (MB)
Duck size: 1.0 GB SEQ	Gzip	-9 (best)	2.2010	26	6
	DNACompact	default	1.8998	1,656	1,348
	DNAEnc3	standard best	1.8676	118	10,668
	XM	50 experts	1.8601	1,131	32,879
	XM	150 experts	**1.8505**	1,384	33,634
	GeCo	-l 6	1.8570	52	4,800
	GeCo	-l 7	1.8520	64	5,800
Cuckoo size: 1.1 GB SEQ	Gzip	-9 (best)	2.1789	24	6
	DNACompact	default	1.9051	1,720	1,348
	DNAEnc3	standard best	1.8462	128	11,071
	GeCo	-l 6	1.8250	59	4,800
	GeCo	-l 7	**1.8200**	70	5,800
Peregrine Falcon size: 1.1 GB SEQ	Gzip	-9 (best)	2.2074	25	6
	DNACompact	default	1.9038	1,818	1,455
	DNAEnc3	standard best	1.8889	120	11,322
	GeCo	-l 6	1.8790	58	4,800
	GeCo	-l 7	**1.8740**	71	5,800
Assembled human genome (GRC), size: 2.9 GB SEQ	Gzip	-9 (best)	2.1108	70	6
	DNACompact*	default	1.7173	4,764	1,348
	DNAEnc3*	standard best	1.6597	427	4,379
	DNAEnc3	standard best	1.6216	489	14,839
	XM*	50 experts	1.6044	1,170	20,759
	XM*	150 experts	1.5832	1,594	22,295
	GeCo	-l 6	1.5750	131	4,800
	GeCo	-l 7	1.5710	148	5,800
	GeCo	-l 8	1.5690	155	6,400
	GeCo	-l 9	**1.5680**	156	7,800

Table 3.3: Compression benchmarks for state-of-the-art compression tools derived from FASTQ formats. Time is in minutes, while maximum memory peak in MBytes.

Dataset	Method	Mode	Compression		
			Bits p/ base	Time (m)	Memory (MB)
SRR003168 361 MB (only bases) FASTQ derived	Gzip	-9 (best)	2.1927	8	6
	fqz_comp	default	1.8029	1	79
	fqz_comp	-e -b -s5+	1.7652	1	199
	fqz_comp	-e -b -s6+	1.7607	1	583
	fqz_comp	-e -b -s7+	1.7602	2	2,070
	fqz_comp	-e -b -s8+	1.7660	2	8,263
	Orcom	-t4 -b256 -p6 -s6	2.1809	1	1,180
	Deliminate	a	1.7381	1	780
	MFCompress	-1	1.7413	3	514
	MFCompress	-2	1.7012	4	514
	MFCompress	-3	1.6405	6	2,322
	GeCo	-l 2	1.5500	17	4,800
	GeCo	-l 4	1.5491	16	3,900
	GeCo	-l 6	1.5344	18	4,800
	GeCo	-l 8	**1.5322**	20	6,400
SRR494099 486 MB (only bases) FASTQ derived	Gzip	-9 (best)	2.2136	11	6
	fqz_comp	default	1.8064	1	79
	fqz_comp	-e -b -s5+	1.7714	2	199
	fqz_comp	-e -b -s6+	1.7496	2	583
	fqz_comp	-e -b -s7+	1.7390	2	2,070
	fqz_comp	-e -b -s8+	1.7418	2	8,263
	Orcom	-t4 -b256 -p6 -s6	1.9495	1	1,252
	Deliminate	a	1.7995	1	780
	MFCompress	-1	1.8810	5	514
	MFCompress	-2	1.8459	5	514
	MFCompress	-3	1.8344	8	2,322
	GeCo	-l 2	1.6670	23	4,800
	GeCo	-l 4	1.6789	22	3,900
	GeCo	-l 6	**1.6662**	25	4,800
	GeCo	-l 8	1.6856	28	6,400

Table 3.4: Benchmarks for state-of-the-art genomic reference compressors using several references and targets. Time is in minutes, while maximum memory peak in MBytes. The prefix HS, PT, GG, PA, represent respectively human, chimpanzee, gorilla and orangutan. The suffix with the numbers represent the chromosome number.

Reference seq	Target seq	Method	Mode	Compression		
				Bits p/ base	Time (m)	Memory (MB)
HS18	PT18	GReEnc	-	1.2224	2	826
77 MB	71 MB	iDoComp	-	**0.2408**	2	599
SEQ		GeCo	-l 11	0.3176	5	3,938
	GG18	GReEnc	-	0.9800	2	826
	72 MB	iDoComp	-	**0.3568**	2	599
		GeCo	-l 11	0.3672	5	3,938
	PA18	GReEnc	-	1.7056	2	826
	71 MB	iDoComp	-	0.8224	2	599
		GeCo	-l 11	**0.5992**	5	3,938
PA11	HS11	GReEnc	-	1.8784	4	1,112
119 MB	129 MB	iDoComp	-	1.2816	3	1,114
SEQ		GeCo	-l 11	**0.6552**	8	3,938
	PT11	GReEnc	-	1.5752	4	1,112
	118 MB	iDoComp	-	1.1352	3	1,114
		GeCo	-l 11	**0.6024**	8	3,938
	GG11	GReEnc	-	1.5704	4	1,112
	118 MB	iDoComp	-	1.2784	3	1,114
		GeCo	-l 11	**0.6752**	8	3,938
HS5	PT5	GReEnc	-	1.3944	5	1,430
173 MB	167 MB	iDoComp	-	0.9352	4	1,420
SEQ		GeCo	-l 11	**0.3568**	10	3,938
	GG5	GReEnc	-	1.9040	5	1,430
	147 MB	iDoComp	-	0.9200	4	1,420
		GeCo	-l 11	**0.8632**	10	3,938
	PA5	GReEnc	-	1.4632	5	1,430
	165 MB	iDoComp	-	**0.5640**	4	1,420
		GeCo	-l 11	0.6344	11	3,938

"if we conceive of a being whose faculties are so sharpened that he can follow every molecule in its course, such a being, whose attributes are as essentially finite as our own, would be able to do what is impossible to us."

J. C. Maxwell

4

Compression-based measures

4.1 Kolmogorov complexity

Consider a binary information source that generates symbols from $\Theta = \{0, 1\}$ and, therefore, $|\Theta| = 2$. Also, consider that the information source has already generated the sequence of n symbols, $x^n = x_1 x_2 \dots x_n,\ x_i \in \Theta$. Now consider a second information source that generates symbols, Y, from the same alphabet Θ. Moreover, Y has already generated the sequence of p symbols, $y^p = y_1 y_2 \dots y_p,\ y_j \in \Theta$. For convenience, assume that $p \approx n$.

Consider the problem of quantifying the amount of information that an object x has. In 1965, Andrey Kolmogorov defined three approaches for this problem: combinatorial, probabilistic and algorithmic [10]. The last one became the most used nowadays in this field and is known as the Kolmogorov complexity or algorithmic entropy [13, 14, 10, 12, 114, 115]. For an historical introduction see Chapter 1.

The Kolmogorov complexity of x, $K(x)$ (sometimes called as the self-information of x), is given by the length of a shortest binary program that computes x on a universal computer, such as a universal Turing machine, and halts [5]. The conditional Kolmogorov complexity of x given y, $K(x|y)$, denotes the length of a shortest binary program that, having y furnished as an auxiliary input, computes x and halts. For $y = \lambda$, where λ denotes an empty object, $K(x|\lambda) = K(x)$. The conjoint Kolmogorov complexity of x and y, $K(x, y)$, defines the length of a shortest binary program that, without auxiliary information to the computation, computes both x and y (and how to separate them) and halts.

From now on we will disregard correcting terms, that asymptotically become irrelevant, although for these kind of details see [116]. Accordingly, the three functions in Kolmogorov complexity are related by the chain rule [116]

$$K(x, y) = K(x) + K(y|x). \tag{4.1}$$

Given the symmetric property of the conjoint Kolmogorov complexity, $K(x, y) = K(y, x)$,

and the chain rule (Eq. 4.1), we can define the algorithmic mutual information, $I(x : y)$, as

$$I(y : x) = K(x) - K(x|y), \tag{4.2a}$$
$$I(x : y) = K(y) - K(y|x), \tag{4.2b}$$
$$I(x : y) = K(x) + K(y) - K(x, y), \tag{4.2c}$$
$$I(x : y) = K(x, y) - K(x|y) - K(y|x), \tag{4.2d}$$
$$2I(x : y) = K(x) + K(y) - K(x|y) - K(y|x), \tag{4.2e}$$
$$I(x : y) = I(y : x). \tag{4.2f}$$

Fig. 4.1 depicts an illustration that is consistent with the formulations mentioned above and with a certain degree of similarity relatively to the Shannon entropy. For inequalities and bounds between both see [117].

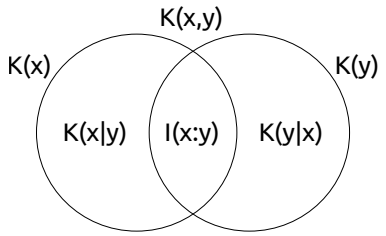


Figure 4.1: Relation of the algorithmic mutual information, $I(x : y)$, self information, $K(x)$ and $K(y)$, conditional information, $K(x|y)$ and $K(y|x)$, and conjoint information, $K(x, y)$, of objects x and y.

An extensive treatment of this topic can be found in [116].

4.1.1 A distance of information

A distance can be seen as a function D in R_0^+, defined on the Cartesian product $\Omega \times \Omega$ of a non empty set Ω. It is called a metric on Ω if for every $x, y, z \in \Omega$ respects the identity property

$$D(x, y) = 0 \iff x = y, \tag{4.3}$$

the symmetry property

$$D(x, y) = D(y, x), \tag{4.4}$$

and triangle inequality

$$D(x, y) + D(y, z) \geq D(x, z). \tag{4.5}$$

4.1.1.1 Information distance

The foundations of the Information Distance (ID) are built upon the Kolmogorov notion of complexity [10, 118, 116]. Bennett introduced the ID [15], defined as the length of the shortest binary program for the reference universal prefix Turing machine that, with input x, computes y, as well as with input y computes x. Formally, it is defined as

$$\mathrm{ID}(x, y) = \max\{K(x|y), K(y|x)\}, \tag{4.6}$$

up to an additive logarithmic term. The normalized version of $\mathrm{ID}(x, y)$, called the Normalized Information Distance (NID) [119], is formally defined as

$$\mathrm{NID}(x, y) = \frac{\max\{K(x|y), K(y|x)\}}{\max\{K(x), K(y)\}}, \tag{4.7}$$

and it is universal in the sense that minimizes, up to an additive error term, all normalized admissible distances in the class considered in [119]. However, the Kolmogorov complexity is non-computable in the Turing sense [12]. Therefore, an admissible approximation is made using (lossless) compression algorithms, denoted by C.

4.1.1.2 Normalized Compression Distance

Although the ID (defined in 4.6) is theoretically appealing, it is almost impractical since it is not computable. Inspired by the ID, a computable metric emerged, the (Conditional) Compression Distance (CD), defined as

$$\mathrm{CD}(x, y) = \max\{C(x|y), C(y|x)\}, \tag{4.8}$$

where $C(x|y)$ denotes the number of bits needed by the (lossless) compression program to represent x given object y as an auxiliary input to the computation, while $C(y|x)$ uses the same scheme but changing y by x.

However, the conditional information content can not be handled by most of the existing compressors. Therefore, using the chain rule (4.1), the following (conjoint) CD analog has been proposed

$$\mathrm{CD}(x, y) = \max\{C(x, y) - C(x), C(y, x) - C(y)\}, \tag{4.9}$$

up to an additive logarithmic term. The $C(x)$ and $C(y)$ denote, respectively, the number of bits needed by the (lossless) compression program to represent x and y, and $C(x, y)$ denotes the number of bits required to compress the conjoint information content of x and y (concatenation of x and y) and the information of how to split them.

The CD is related with the algorithmic mutual information content, $I(x : y)$, that can be obtained by both conditional and/or conjoint compressions, as illustrated in Fig. 4.1. The normalized version, known as the Normalized Compression Distance (NCD) [19], makes the role of a quasi-universal distance, which was primarily defined as

$$\mathrm{NCD}(x, y) = \frac{\max\{C(x, y) - C(x), C(y, x) - C(y)\}}{\max\{C(x), C(y)\}}, \tag{4.10}$$

and finally as

$$\mathrm{NCD}(x, y) = \frac{C(xy) - \min\{C(x), C(y)\}}{\max\{C(x), C(y)\}}, \tag{4.11}$$

up to an additive logarithmic term.

The NCD generates a non negative value in the interval $0 \leq \mathrm{NCD}(x, y) \leq 1$. Distances near 1 indicate dissimilarity, while distances near 0 indicate similarity, as it can be seen with the extreme example of similarity, $x = y$ (equality), where $\mathrm{NCD}(x, x) = \{C(xx) - C(x)\}/C(x)$ and finally, recalling the idempotency property (see below), $\mathrm{NCD}(x, x) \approx 0$.

In order to achieve an accurate and admissible NCD, the compressor, besides the need of having the best possible model to represent the nature of the data, needs to be *normal*. A compressor is *normal* if it satisfies the following conditions:

1. Idempotency: $C(xx) = C(x)$ and $C(\lambda) = 0$, where λ represents the empty object,
2. Monotonicity: $C(xy) \geq C(x)$,
3. Symmetry: $C(xy) = C(yx)$,
4. Distributivity: $C(xy) + C(z) \leq C(xz) + C(yz)$,

up to an additive $O(\log n)$ term, with n the maximal length of an element involved in the (in)equality concerned. For common pitfalls in compressor settings to calculate the NCD see [120], while for a generic framework to compute the NCD see `http://complearn.org/`.

4.2 Global measures

There are many methods to estimate information or distance between digital objects, although when we rely on sequences with low cardinality alphabets, such as genomic sequences, the most popular are Kullback-Leibler divergence [121], the Hamming distance[122], Levenshtein distance [123] and compression-based metrics (such as the NCD).

The Kullback-Leibler divergence measures the dissimilarity between two probability distributions, however is non-symmetric, and, therefore, is not a distance. The Hamming distance can only be applied when the sequences are aligned with precision and have the same size, requirements hardly found in large genomic sequences. The Levenshtein distance explores transformations between the sequences, namely insertions, deletions and substitutions. Although quite successful, its computational time is prohibitive for large sequences (the fastest known implementation runs with time complexity $O(n^2/\log n)$).

Compression-based approaches emerged as a natural way for measuring complexity, because, together with the appropriate decoder, the bitstream produced by a lossless compression algorithm allows the reconstruction of the original data and, therefore, can be seen as an upper bound of the algorithmic entropy of the sequence. Several approaches have been proposed (e.g., [124, 19, 125, 126, 127]) showing very good adaptability to diverse problems, such as in clustering and classification.

A compression-based distance computes the distance between two (digital) objects using the number of bits needed to describe one of them when a description of the other is available, as well as the number of bits required to describe each of them.

We have introduced the Kolmogorov complexity and its most popular related distance (NCD). In the following subsection we introduce the Normalized Conditional Compression Distance, that can be seen as a direct computation of the NID (defined in 4.7). After, we introduce the Normalized Relative Compression.

4.2.1 Normalized Conditional Compression Distance

A direct substitution of K by C in (4.7) would require the availability of compressors that are able to produce conditional compression, i.e., $C(x|y)$ and $C(y|x)$. Most compressors do not have this functionality and, therefore, the NCD avoids it by using suitable manipulations of the NID (4.7) [19]. Instead of $C(x|y)$ and $C(y|x)$, a term corresponding to the conjoint compression of x and y, $C(x, y)$, was preferred. Usually, this $C(x, y)$ term is interpreted as the compression of the concatenation of x and y, but, in fact, it could be any other form of

combination between x and y. Concatenation is often used because it is easy to obtain, but in fact its use may hamper the efficiency of the measure [120].

To overcome several limitations, we propose to use the direct form of conditional compression. Independently, the same strategy was used in image distortion studies [128]. According, the following expresses the Normalized Conditional Compression Distance (NCCD),

$$\text{NCCD}(x,y) = \frac{\max\{C(x|y), C(y|x)\}}{\max\{C(x), C(y)\}}, \tag{4.12}$$

where "Conditional" means that the compressor C needs to be able to perform conditional compression as described earlier. In fact, knowing the individual algorithmic complexities of objects x and y provides a way to compute only one conditional complexity, according to

$$\text{NCCD}(x,y) = \begin{cases} \dfrac{C(x|y)}{C(x)}, & C(x) \geq C(y) \\ \dfrac{C(y|x)}{C(y)}, & C(x) < C(y) \end{cases}. \tag{4.13}$$

This strategy enables the computation process of $C(x|y)$ to be faster than $C(x,y)$, mainly because the conditional method only needs to load models of y (much faster than compressing) while the conjoint needs to compress also y. In a *big data* scenario, involving many computations, this strategy has a deep impact.

4.2.1.1 Parameterization, assessment and concerns

We have used a fixed setup of five reference models and three target models mixed using a set of weights adapted by a forgetting mechanism with $\gamma = 0.9$ (see subsection 2.2.1.2). From our experience, we have verified that $\gamma = 0.99$ maximizes the compression gain for bacterial genomes, while for eukaryotic genomes $\gamma = 0.9$ seems to be the best choice. The reference models context orders used were: 4, 6, 8, 10 and 15. The first four assumed a Laplacian estimator ($\alpha = 1$), whereas the last used $\alpha = 0.001$. Usually, a small α is only important in high orders (above ten). Moreover, the high order used (15) ensures $C(xx) \approx C(x)$, as Fig. 4.2 depicts.

The lossy approach, assuming always the best FCM, shows that up to 32k size the curve is due to the adaptation of the method when not enough data is present, inferring that a very small sequence size may harm the identity property, as well as in very big sequence sizes. Nevertheless, the last may be overcome using higher FCM context orders, sacrificing computational memory.

On the other hand, the three target model context orders used were: 4, 10 and 15, where the first two used $\alpha = 1$ and the last one $\alpha = 0.05$. In the last two models the inverted repeats technique has been used. The maximum counters used in each reference models were, respectively, 2^9, 2^{12} and 2^{12}. This acts also as a forgetting mechanism, since every counter overflow is divided by two, which after some counter updating makes the older counters losing importance. More details about FCM parameterization can be found in [91, 84].

The DNA sequences are products of sequencing techniques, which have a sequencing quality, coverage and assembly technique associated [129]. Although these external factors intuitively may constitute a problem, we believe that generally they are dissipated and overcome by the compressor and metric. For a study on noise resistance, using the NCD, see [130].

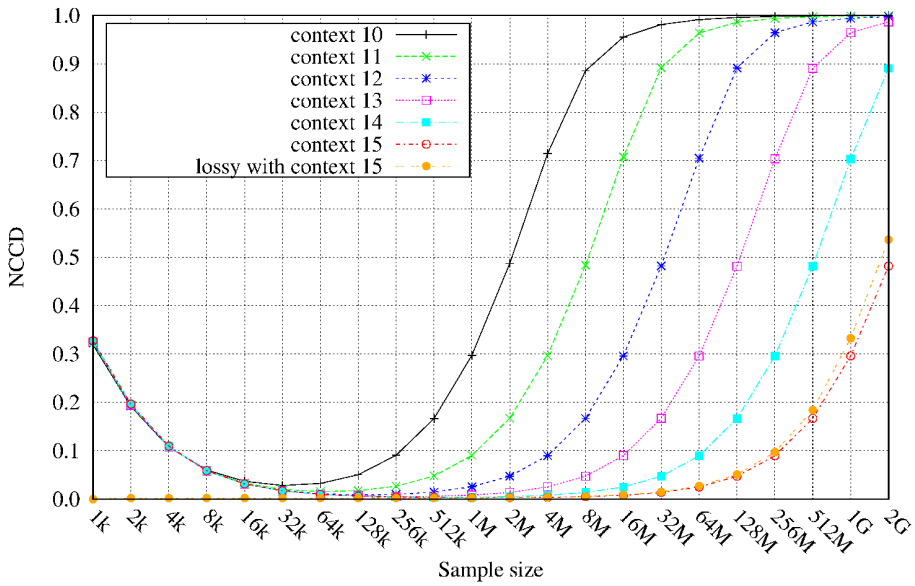


Figure 4.2: NCCD performance, on uniform stochastic DNA (synthetic) sequences with custom sizes, on testing several high orders.

Nevertheless, since we use a metric based on conditionals, we have assessed the impact of uniform distributed mutations, namely substitutions, insertions and deletions, over 50 MB of *real* (first 50 MB of chromosome 1 from *H. sapiens*) and synthetic (simulated using XS from Exon [131]) genomic data, as it can be seen in Fig. 4.3.

Specifically, substitutions seem to be slightly the most difficult mutation type to be handled by the compressor and, hence, the NCCD. Nevertheless, it is clear that the method is still reporting reasonable distances for sequences with 10 % of mutations, both in *real* and synthetic sequences.

Finally, we have assessed the importance of sequence completeness using progressive missing data, as Fig. 4.4 depicts. As expected, it is characterized by an approximately linear behavior. However, there is a gap between *real* and synthetic sequences, specially when there are lower missing rates. This is due to the nature of the sequences, namely self-similarity, since the beginning of the *real* sequence is composed by a telomeric zone (highly-repetitive). On the other hand, the synthetic sequence does not have a precise zero NCCD value when the missing rate is zero, because it has been simulated with several approximated repeating zones. Once more, this concern may be overcome with high FCM orders, using more computational memory.

4.2.1.2 Materials

The experiments were performed in a Linux server running Ubuntu with 16 Intel(R) Xeon(R) CPU E7320 at 2.13 GHz and with 256 GB of RAM. The NCCD values have been computed using an implementation in C programming language of the method referred in Section 3.2 with the parameters described in Section 4.2.1.1. The software application is freely available for non-commercial usage at `http://bioinformatics.ua.pt/software/nccd`. The evolutionary tree has been processed using T-REX [132], TreeDyn [133] web servers and

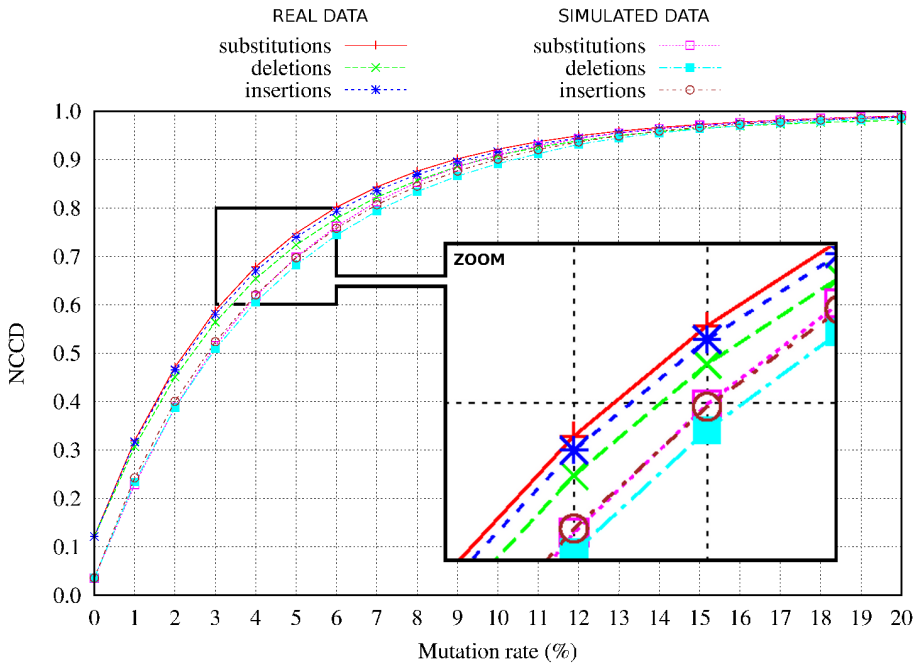


Figure 4.3: NCCD performance on synthetic and *real* 50 Mb of genomic mutated data.

further customized.

The data set is composed by 15 genomes, described in Table 4.1, downloaded from NCBI (`ftp://ftp.ncbi.nlm.nih.gov/genomes`).

4.2.1.3 Results

Using the materials described in Section 4.2.1.2, we have conducted several NCCD measures in order to unveil important relations of data, namely chromosomal and genomic similarities, where the first is further subdivided in the relations between chromosomes of the same species (intra-genomics) and across different species (inter-genomics).

The NCCD has been used to measure the distance between chromosomes of the same species, namely in *H. sapiens*, *P. troglodytes*, *G. gorilla*, *P. abelii*, *M. musculus*, *R. norvegicus*, *G. gallus* and *M. gallopavo* genomes, respectively described in Fig 4.5. At glance, the approach *all with all* highlights the identity property, since the *diagonals* have distances near zero. However, looking into the self-relations of chromosomes Y in *M. musculus* and chromosome 32 in *G. gallus*, the distance seems away from zero (not preserving the identity). Nevertheless, according to [120], this is due to the size of the sequence and/or high self-similarity (small complexity). In fact, we have verified that *M. musculus* chromosome Y has in fact a very high self-similarity (99.9% according to [134]) and *G. gallus* chromosome 32 has slightly more than 1000 bases (very small). These are particular characteristics of the data for calculating the NCCD.

Biologically, as expected in both species, the mitochondria sequence (M) is very different from the rest of the sequences [135], confirmed by the largest NCCD values. On the other hand, primates chromosome X has a small distance relatively to several large chromosomes,

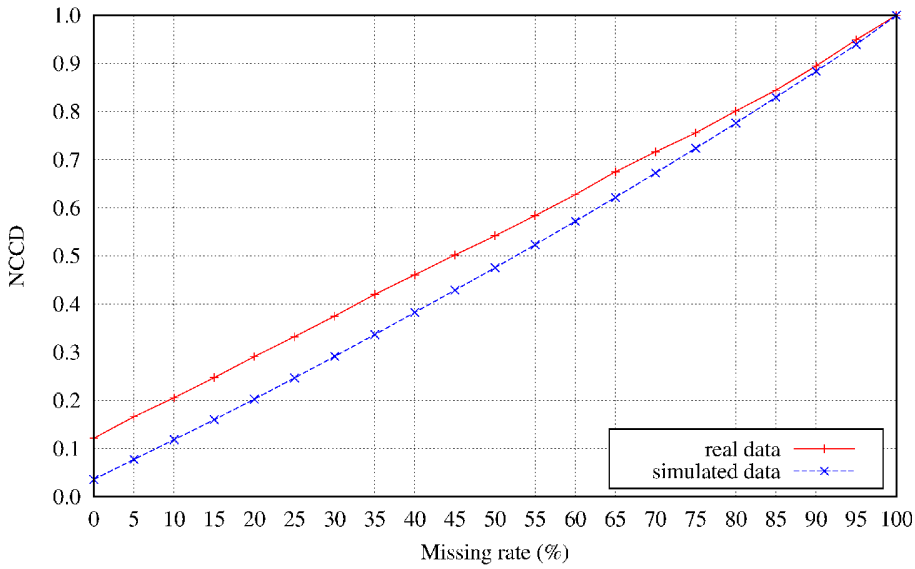


Figure 4.4: NCCD performance on progressive block missing data in *real* and synthetic sequences.

---

while *M. musculus* has a small distance relatively to average size and several small chromosomes. Moreover, overall and according to [136], *M. musculus* seems to have a curious different pattern compared to *R. norvegicus*, a pattern that is normally opposite to common species such as those from primates and turkey-chicken relations.

Although the existence of other relevant lower NCCD values that express chromosome similarity, the NCCD between X and Y, in the *H. sapiens* genome, stands immediately out, while in the *P. troglodytes* and *M. musculus* genomes can be seen with much higher distances. According to a recent study [137], this insight is probably due to the genetic information exchange between X and Y chromosomes in the recombination process.

More than a century ago, Huxley [138] and Darwin [139] launched the hypothesis of a common ancestor between humans and great apes, while modern molecular studies confirmed and extended those predictions. Using an automatized approach, that can be used on every chromosome in every sequenced species we quantify the distance, as depicted by the intensities in Fig. 4.6.

Accordingly, the approach *all with all* in primates shows a direct correlation (*diagonal*) with respect to the same chromosomal number/identification, except for humans on chromosome 2 with 2A and 2B. This is justified by a hypothetical chromosomal fusion that made human evolve from previous ancestors [140]. Moreover, we found a high distance, compared to the other distances in the *diagonal*, between orangutan chromosome 1 and human-chimpanzee-gorilla chromosomes 1.

Supported by the results from this section, we can conclude that the *H. sapiens* Y chromosome has a large distance comparing to the *P. troglodytes* Y chromosome, which agrees with a surprising recent study [137]. Consequently, there is an increasing distance relatively to the deviation of the primates species between chromosome X and human chromosome Y.

Although the existence of other relevant unveiled lower NCCD values that express chromosome similarity, mainly by duplicated segments that have been shown to be associated

Table 4.1: Data set table. The number of expected chromosome pairs for each species is represented by 'Exp', while 'Missing' denotes a non-existent sequence and Mb represents the approximated size in Mega bases.

Organism	Build	Exp	Missing	Mb
Homo sapiens	37.p10	23	-	2,861
Pan troglodytes	2.1.4	24	-	2,756
Gorilla gorilla	r100	24	Y	2,719
Pongo abelii	1.3	24	Y	3,028
Macaca mulatta	2.1	21	Y	2,725
Callithrix jacchus	1.2	23	M	2,664
Mus musculus	38.p1	20	-	2,716
Rattus norvegicus	5.1	21	Y	2,443
Bos taurus	6.1	30	-	2,679
Equus caballus	3.1	32	Y	2,335
Canis familiaris	3.1	39	Y	2,318
Gallus gallus	r102	39	29-31, 33-38	999
Danio rerio	5.1	25	-	1,355
Meleagris gallopavo	1.1	40	31-39	917
Felis catus	r100	19	Y	2,353

with rapid gene innovation and chromosomal rearrangement in primates genomes[141], there is a linkage present in all primates, namely in chromosomes 13 and 14. More relevant, there is a very low distance between *G. gorilla* and *H. sapiens* 5 and 17 chromosomes [142], justified by a translocation of part of chromosome 5 to 17 in the *G. gorilla* genome, that can only be detected in homologous species.

Relatively to *M. musculus*, there is an obvious similarity with *R. norvegicus*, although smaller than *P. maniculatus* / *M. norvegicus* correlations [136]. When compared with human and chimpanzee, no important similarities are found (in a genomic level), specially in human/chimpanzee chromosomes 19 and 22. Moreover, it seems that only the mithocondrial sequences achieve some level of similarity. Nevertheless, *M. musculus* (MM) and *R. norvegicus* (RN) *diagonal* is very dissipated for such a low distance depicted in the mithocondrial sequence. In fact, only chromosome (chr) 18 and X seem to be bias homologous (in the *diagonal*). Subsequent analysis show strong similarity between MM chr2 / RN chr3, MM chr9 / RN chr8 and MM chr11 / RN chr10, and considerable similarity between MM chr4 / RN chr5, MM chr6 / RN chr4, MM chr12 / RN chr6 and MM chr14 / RN chr15, without detracting other important patterns.

The last map demonstrates that most chicken (*G. gallus*, GG) chromosomes appear to correspond to single orthologous turkey (*M. gallopavo*, MG) chromosomes with a few exceptions, namely GG chr2 shares high similarity with MG chr3 and MG chr6, while GG chr4 is similar to MG chr4 and MG chr9. These phenomenons have been proposed as centric fission events on turkey lineage [143]. Moreover, MG chr8 and chr2 are, respectively, homologous to GG chr6 and chr3. From MG chr10 with GG chr8 orthologous, a straight linear homology (*diagonal*) is present, with exception for GG chr32 and GG/MG chrW. As explained in the previous subsection GG chr32 is very small and GG/MG chrW are have high self-repetitive nature [144].

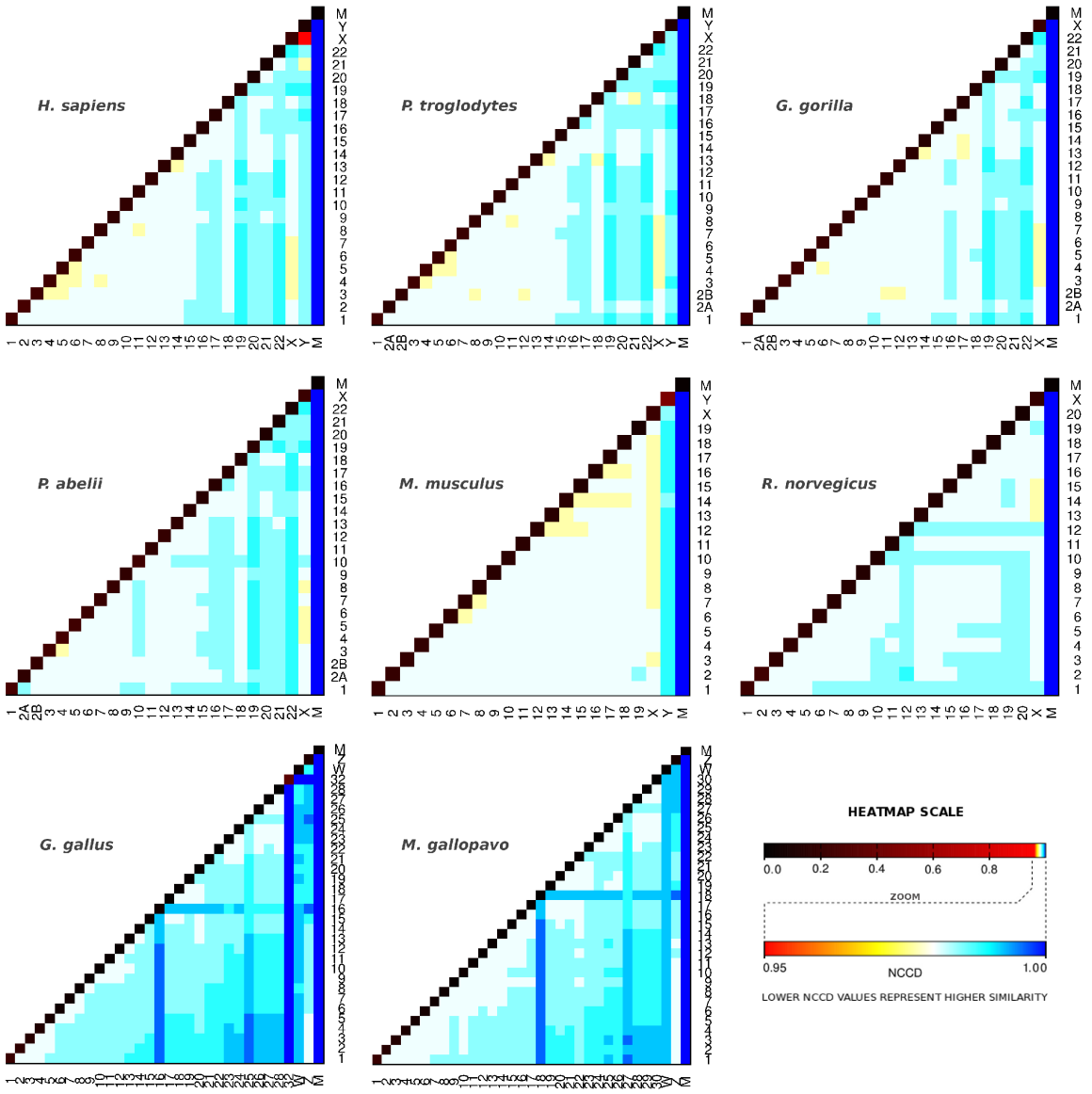


Figure 4.5: Intra-genomics chromosomal NCCD heatmaps for *H. sapiens*, *P. troglodytes*, *G. gorilla*, *P. abelii*, *M. musculus*, *R. norvegicus*, *G. gallus* and *M. gallopavo*. The heatmap scale quantifies the NCCD distance.

In Fig. 4.7 are presented the chromosomal distances of *P. troglodytes*, *G. gorilla* and *P. abelii* (chromosomes 2A and 2B have been concatenated) according to *H. sapiens* chromosomes order, while above are the differences of sizes according to *H. sapiens*. At glance, *P. troglodytes* has the lowest distance relatively to *H. sapiens* [145], and after *G. gorilla* [146] and *P. abelii* [147], respectively (for most of the chromosomes). Specifically, *G. gorilla* chromosomes 5 and 17 have large distances because of the previous mentioned translocation, while *P. abelii* seems to have a very different chromosome 1 besides other relevant dissimilarities. Mitochondria sequences, as expected, show that *P. troglodytes* is the nearest *H. sapiens* species,

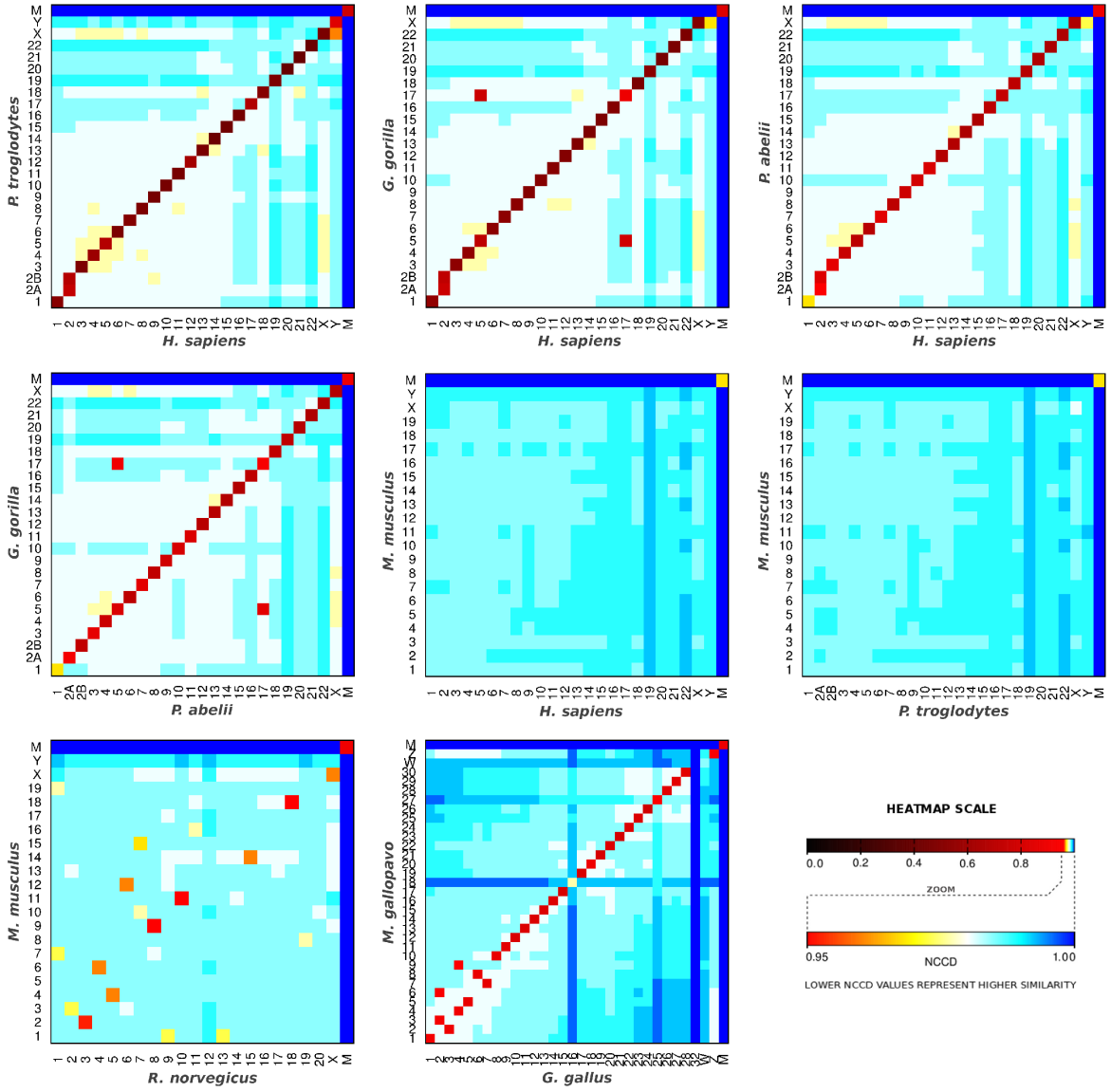


Figure 4.6: Inter-genomics chromosomal NCCD heatmaps between several species, namely *H. sapiens* and *P. troglodytes*, *H. sapiens* and *G. gorilla*, *H. sapiens* and *P. abelii*, *G. gorilla* and *P. abelii*, *M. musculus* and *H. sapiens*, *M. musculus* and *P. troglodytes*, *M. musculus* and *R. norvegicus*, *M. gallopavo* and *G. gallus*.

---

followed by the *G. gorilla* and, lastly, by *P. abelii*, although this is further explored in the next section.

Finally, we have detected *G. gorilla* chromosomes 4, 12 and 18 with distances lower to *H. sapiens* than to the respective *P. troglodytes* chromosomes, while *G. gorilla* chromosomes 5 and 17 have higher distances than *P. abelii* chromosomes. Measuring these distances may also be important in the sense that metagenomics high compression gains can be achieved with reference-based compression, both in chromosomes and genomes, using the lowest distance

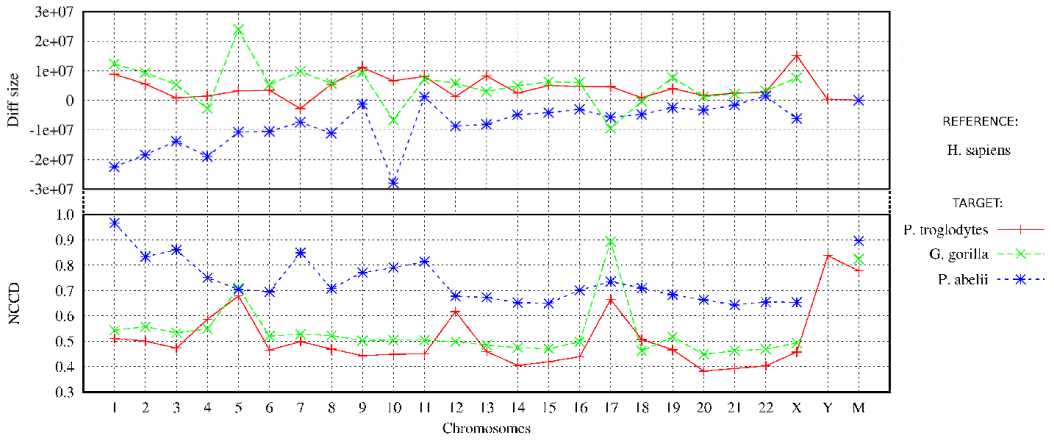


Figure 4.7: *Homo sapiens*, *Pan troglodytes* and *Pongo abelii* related chromosomal normalized conditional compression distance values.

---

inter-genomic sequences.

Further analysis of the computation of the NCCD on related objects, such as chromosomes or entire genomes, can led to the development of evolutionary trees. Accordingly, we used chromosomes 18 of human, chimpanzee, gorilla, orangutan and rhesus to give an example. As it can be seen in Fig. 4.8, the NCCD values in line with the two most recent evolutionary theories [148, 149]. The computation of the tree has been made with the whole chromosome sequences in a common laptop in about 30 minutes. Moreover, there is no limitation to the input, since it can deal with whole genomes.

4.2.2 Normalized Relative Compression

The relative compression can be seen as the number of bits used to represent an object x having exclusively the information from object y, and hence, is defined as $C(x\|y)$. As such, we are interested in data compressors that comply with relations

1. $C(x\|y) \approx 0$ iff y contains x;
2. $C(x\|y) \approx |x|$ iff $C(x|y) \approx C(x)$,

based on which we define the *Normalized Relative Compression* on x given exclusively y as

$$\mathrm{NRC}(x, y) = \frac{C(x\|y)}{|x|}. \tag{4.14}$$

Note that in the usual conditional compression, denoted by $C(x|y)$, we should have $C(x|y) \approx C(x)$ iff $K(x|y) \approx K(x)$ (i.e., when x and y are totally unrelated), and $K(x|y) \approx |x|$ iff $C(x|y) \approx |x|$ (i.e., when x and y are totally unrelated and x is incompressible).

We call $C(x\|y)$ a *relative compressor*, because it has y and only y available to represent x. In other words, this compressor cannot use self-similarities that might occur in x. A relative compressor starts by building an appropriate model of y (in the limit, it can retain the whole y). Then, it represents x using *only* the information from the model of y. Hence, in essence

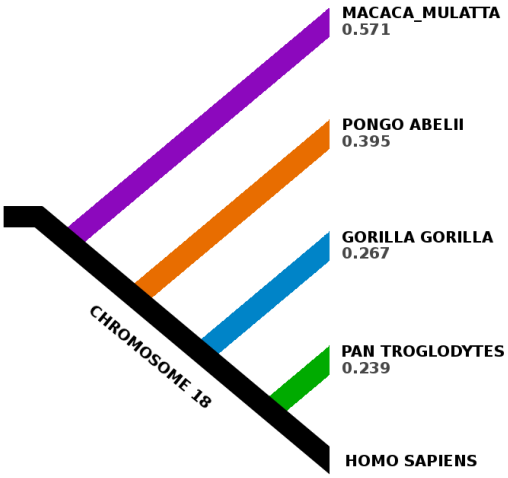


Figure 4.8: Distance Tree given by the Normalized Conditional Compression Distance on chromosome 18 sequences of human, chimpanzee, gorilla, orangutan and rhesus species. Distances have been computed relatively to human. The NCCD has been computed using GeCo with mode 14 for conditional compression, namely $C(x|y)$ and $C(y|x)$, and mode 9 for individual compression, namely $C(x)$ and $C(y)$.

---

it captures the notion of model-freezing discussed in Section 3.2. It differs from Section 3.2 only in the part that it does not use target models (class $\mathcal{T}$), and rather only reference models (class $\mathcal{R}$).

4.2.2.1 Is the NRC a distance?

For the NRC to be considered a distance it needs to respect all the properties formulated in Section 4.1.1.2, namely identity, symmetry and triangle inequality, as well as being always non-negative. Since $|x| \geq C(x\|y)$ the quotient between the terms in the NRC is always $0 < NRC \leq 1$. The identity property is also straightforward to see, since $C(x\|x)$ must be approximately zero, and therefore $C(x\|x)/|x| \approx 0$.

For the other two properties we give the respective proofs considering three random strings x, y, z where $y \in x$, $z \in x$ (x is given by the concatenation of y and z), $z \neq y$ (no information is shared between y and z) and $|y| > |z|$. These conditions can be better understood using Fig. 4.9.

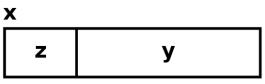


Figure 4.9: Illustration with three random objects x, y, z, where x is made by the concatenation of object y and z. There is no information shared between y and z.

Lemma 4.2.1. *The Normalized Relative Compression is not symmetric and, thus* $NRC(x, y) \neq NRC(y, x)$.

Proof. Considering that it is symmetric, $NRC(x,y) = NRC(y,x)$, on substituting we get $C(x\|y)/|x| = C(y\|x)/|y|$. Accordingly, $C(y\|x) \approx 0$, and hence, $NRC(y,x) \approx 0$. On the other hand, $C(x\|y) \approx |z|$, and hence, $NRC(x,y) \approx |z|/|x|$. Therefore, by *r.a.a.*, the non symmetric property is true. □

Lemma 4.2.2. *The Normalized Relative Compression does not respect triangle inequality,* $NRC(x,y) + NRC(y,z) \not\geq NRC(x,z)$.

Proof. Considering that it respects the triangle inequality and, hence $NRC(x,y)+NRC(y,z) \geq NRC(x,z)$, when substituting we get $C(x\|y)/|x| + C(y\|z)/|y| \geq C(x\|z)/|x|$. Accordingly, $C(x\|y) \approx |z|$, and hence, $NRC(x,y) \approx |z|/|x|$. On the other hand, $C(y\|z) \approx 0$, and hence, $NRC(y,z) \approx 0$. Finally, $C(x\|z) \approx |y|$, so, $NRC(y,z) \approx |y|/|x|$. Substituting, we get $|z|/|x| \geq |y|/|x|$. Since $|y| > |z|$, this is false. Therefore, by *r.a.a.*, the NRC does not respect the triangle inequality property. □

Therefore, given the negative proofs on the properties to be considered a distance, the NRC can not be considered a distance.

4.2.2.2 Advantages

The main advantage of the NRC is the absence of a self-similarity term, $C(x)$. Accordingly, the computation time and resources needed to compute the NRC are much lesser than to compute the NCCD. In fact, the NRC can be interpreted as the fraction of an object that cannot be represented by the other object (instead of a ratio of information quantities), which may be better correlated with the human notion of proximity. Contrarily to the NCCD, which depends on the ratio of compression terms, with implications on the convergence of the approximations, the NRC depends only on one term. Moreover, the setting up of the parameters for the relative compressor are much simpler, namely because it is only needed to set the parameters from the models according to the reference, $\mathcal{R}$. Furthermore, this is also related with less memory usage, namely because we do not need to create models to the target, $\mathcal{T}$. Finally, unlike the NCCD, the NRC behaves in a predictable way, where better compression gains cannot have a negative impact on quantification. This because, the NCCD uses a ratio between two approximation functions [150], while the NRC uses only one.

4.2.2.3 Method and tool

We have built a compressor based on a mixture of FCMs and XFCMs of only one model class, specifically belonging to the reference set $\mathcal{R}$, and respecting the conditions of a relative compressor enumerated above. The compressor is able to process n objects $(x, y, ..., z)$ using one as reference.

As indicated before, the computation of $C(x\|y)$ is faster than $C(x|y)$. Nevertheless, we also use multi-threading to compute each object (having the reference frozen in memory). This enables a much faster computation. We have created a tool with the described method to show some experimental results. The tool (smash-global), written in C language, is available at `https://github.com/pratas/smash-global/`, under GPL-2, and can be applied to any FASTA and multi-FASTA files. The tool, completely unsupervised, is being able to compute n sequences and automatically constructing a heatmap depicting the metrics in an image.

4.2.2.4 Experimental results

The NRC can not answer specific questions relatively to distances, since it is non symmetric and does not obey the triangular inequality, although it can infer insights into dissimilarity or completeness, for example those related to genomic proximity. We have computed the NRC using an avian and a primate dataset.

From the Avian Phylogenomics Project [1] we have downloaded several RNA sequences, namely representing the following species: Rifleman, Peking duck, Chuck will's widow, Bar tailed trogon, Emperor penguin, Grey crowned crane, Rhinoceros hornbill, Anna's hummingbird, Red legged seriema, Turkey vulture, Chimney swift, Killdeer, Macqueen's bustard, Speckled mousebird, Pigeon, American crow, Common cuckoo, Little egret, Sunbittern, Peregrine falcon, Northern fulmar, Red throated loon, Medium ground finch, White tailed eagle, Bald eagle, Cuckoo roller, Golden collared manakin, Budgerigar, Carmine bee eater, Brown mesite, Kea, Crested ibis, Hoatzin, Dalmatian pelican, White tailed tropicbird, Great cormorant, American flamingo, Downy woodpecker, Great crested grebe, Yellow thoated sandgrouse, Adelie penguin, Common ostrich, Zebra finch, Red crested turaco, White throated tinamou, Barn owl.

We have computed the NRC in the RNA sequences of the mentioned species, where the result is depicted in Fig. 4.10. As it can be seen, there are several species that have a low NRC value, namely the bald eagle / white tailed eagle, adelie penguin / emperor penguin. These are species that are known to be genetically close [109]. On the other hand, white throated thinamus, duck and ostrich seem to give lower compression capabilities when they are used as a reference, a characteristic that agrees with the possible age of the species [109]. There are several other patterns, however, we leave them to future analysis.

In Fig. 4.11 we present a heatmap corresponding to the NRC between the human and two primates (chimpanzee and gorilla). Moreover, we have also included the unlocalized, unplaced and mitochondrial sequences. Accordingly, we easily identify the same characteristics present in the NCCD heatmap. Besides, there are new evidences that have been hidden by the derived information distance, namely a high degree of homology from chromosome 19 relatively to others which conforms with a study showing that the largest number of small repeats is in chromosome 19 [151]. Moreover, we are able to see several correlations between extra sequence chromosomes (unplaced and unlocalized) and many chromosomes. The existence of extra sequences results from problems in assembling (that did not identify their chromosome origins). Nevertheless, the measure is able to bypass these problems. In spite of having more associated patterns we emphasize the high representability of mitochondrial DNA (mtDNA) given exclusively human chromosome 5. It is believed that sequences provided by mtDNA existe in chromosomes, although not concentrated in any specific chromosome [152], unlike the results reported by the measure. This point is extended to relative complexity profiles, addressed in Section 4.3.3 and in Fig. 4.19, showing where these low complexities occur in the mtDNA.

4.3 Local measures

This section is about looking at sequences or, more precisely, at graphical representations of sequences. In other words, it is about the summarization of data bearing in mind graphical

[1] `http://avian.genomics.cn/en/jsp/database.shtml`

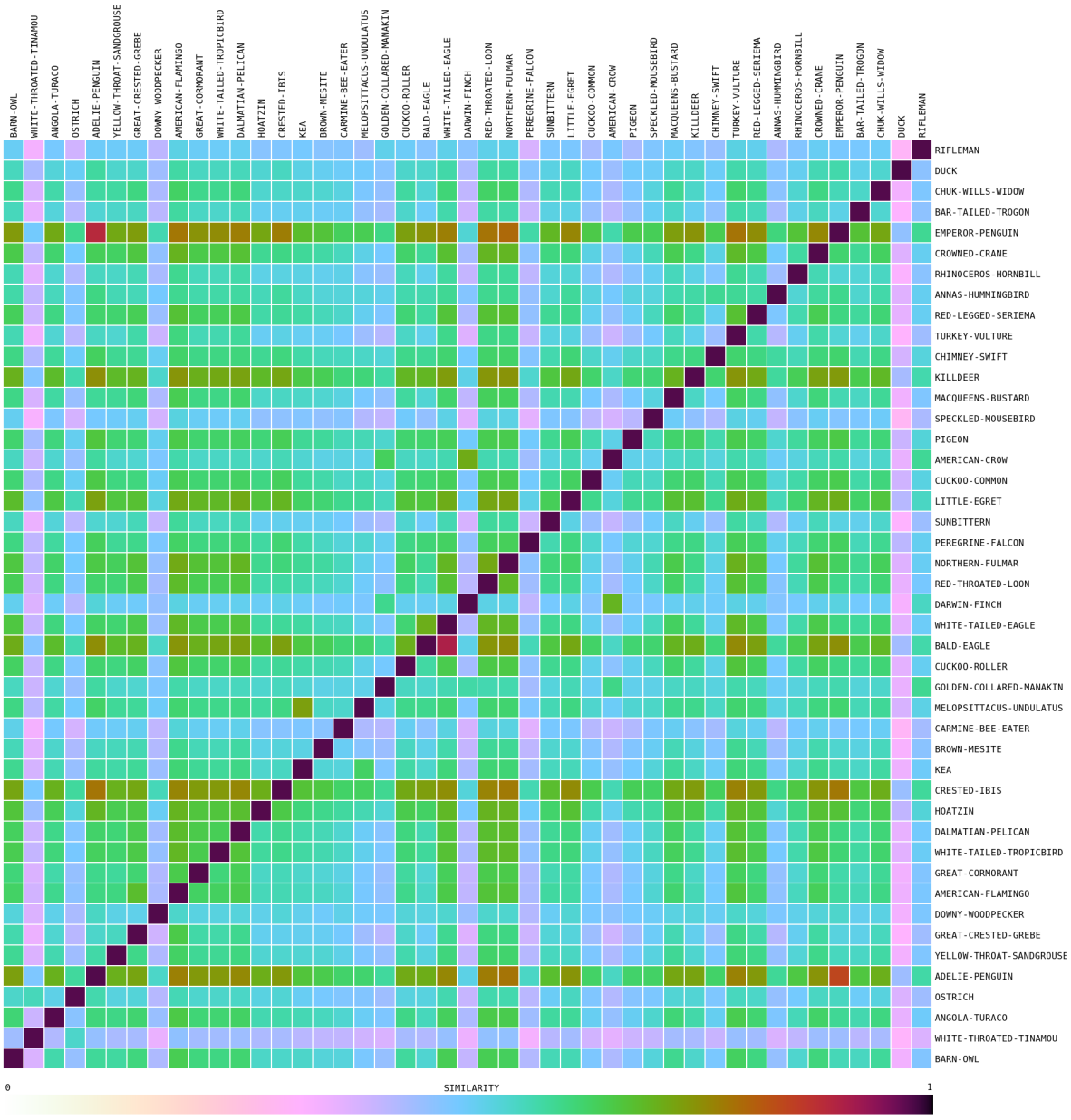


Figure 4.10: Inter-species RNA sequences heatmaps reporting Normalized Relative Compression values for 45 bird species according to [109].

representations, a problem related to some of the current challenges in large-scale computing [153].

The idea is old, as the sayings "a picture is worth a thousand words" and the century-old advertisement title "one look is worth a thousand words" show. In fact, the association of graphical information to sequences, namely DNA sequences, has been pursued for long. Sequence logos [154] and the chaos game display (CGR) [155] are two well-known examples. Most often, the underlying motivation is to look for and to display information related to the

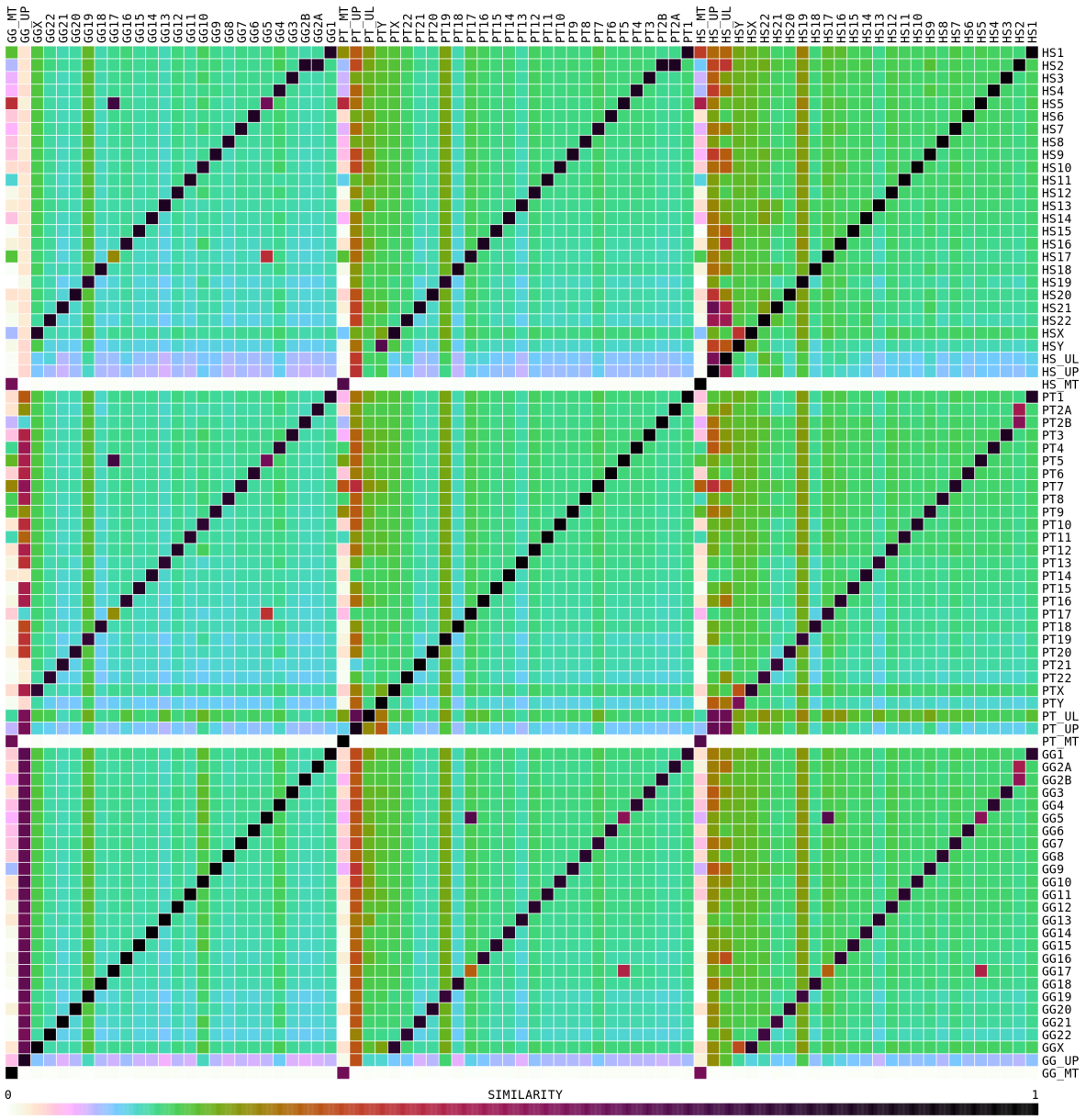


Figure 4.11: Inter-species chromosomal heatmaps reporting Normalized Relative Compression (NRC) values for *H. sapiens* (HS), *P. troglodytes* (PT) and *G. gorilla* (GG). The "UL", "UP" and "MT" represent respectively, unlocalized, unplaced and mitochondrial sequences. The higher the similarity, the lower the NRC. The tool (smash-global) ran with mode 25, using four reference models in a mixture.

---

degree of randomness of the sequences, hoping to find meaningful structure. The degree of randomness is intimately related with the complexity, predictability, compressibility, repeatability and, ultimately, with the information theoretic notion of entropy of a sequence. Other methods, use the graphical paradigm for presenting several parameters that can be obtained from a DNA sequence. For example, the Genome Atlas of Jensen *et al.* [156] allows the visual-

ization of information related to repeats, nucleotide composition, and structural parameters, in microbial genomes (the genome of *E. coli* is analyzed in [157] using this approach).

Some methods provide visual information of global properties of the DNA sequences. For example, CGR uses the distribution of points in an image to express the frequency of the oligonucleotides that compose the sequence [158]. From these CGR images, other global representations can be derived, such as genomic signatures [159, 160] or entropic profiles [161].

Originally [161], entropic profiles were estimated using global histograms of the oligonucleotide frequencies, calculated using CGR images. Later, they have been generalized by Vinga *et al.* [162] in order to calculate and visualize local entropic information. Other approaches for estimating the randomness along the sequence have also been proposed. For example, Crochemore *et al.* [163] used the number of different oligonucleotides that are found in a window of predefined size for estimating the entropy. Troyanskaya *et al.* [164] proposed the linguistic complexity, also calculated on a sliding window, as a measure of the local complexity of the DNA sequence.

Both the global and the local estimates of the randomness of a sequence provide useful information and both have shortcomings. The global estimates do not show how the characteristics change along the sequence and the local estimates fail to take into consideration the global properties of the sequence. This last drawback was addressed by Clift *et al.* [165] using the concept of sequence landscape. Using directed acyclic word graphs, they were able to construct plots displaying the number of times that oligonucleotides from the target sequence occur in a given source sequence. If the target and source sequences coincide, then the landscape provides information about self-similarities (repeats) of the target sequence.

The sequence landscapes of Clift *et al.* [165] seem to have been the first attempt of displaying local information while taking into account the global structure of the sequence. This idea was also pursed by Allison *et al.* [166], using a model that considers a sequence as a mixture of regions with little structure and regions that are approximate repeats. Based on this statistical model, they have produced information sequences, which quantify the amount of surprise of having a given base at a given position, knowing the remaining left (or right) part of the sequence. When plotted, these information sequences provide a quick overview of certain properties of the original symbolic sequence, allowing for example to easily identify zones of rich repetitive content [167, 82, 168].

The interest of complexity measures for DNA sequence analysis has been explored by several researchers, such as in [169, 170, 171]. The key measure is again the Kolmogorov complexity. As mentioned, the Kolmogorov complexity measure is non-computable and is usually approximated by other computable measures, such as, Lempel-Ziv complexity measures [172, 169], linguistic complexity measures [173], or compression-based complexity measures [73, 168, 85].

The information sequences of Allison *et al.* [166] are intimately related to data compression. The importance of data compression for pattern discovery in the context of DNA sequences was already recognized by Grumbach *et al.* [69] and, since then, it has been reinforced by others (e.g. [174, 167]). In fact, the existence of regularities in a sequence renders it algorithmically compressible. The algorithmic information content of a sequence is the size, in bits, of the shortest accurate description of the sequence.

In this section, we further explore the idea of information sequences. We take important steps forward, in what concerns the notion and foundations of complexity profiles, conditional complexity profiles and conditional exclusive profiles. Moreover, we give several applications for the different cases explored with specific developed computational

tools written in C/C++ language and available at `http://http://bioinformatics.ua.pt/software/dna-at-glance/` and `http://http://bioinformatics.ua.pt/software/geco/` (introduced in Chapter 3). The tools are used in different cases exploring the *S. pombe* genome (uid 127) and *H. sapiens* genome, both obtained from the NCBI (`ftp://ftp.ncbi.nlm.nih.gov/genomes/`).

4.3.1 Complexity profiles

A complexity profile can be seen as a numerical sequence $\overrightarrow{N}(x_i)$ containing values that express the predictability of each element from x given a compression function $C(x)$. As such, we define

$$\overrightarrow{N}(x_i) = C(x_i|x_1^{i-1}). \tag{4.15}$$

Moreover, $C(x)$ has a causal effect, which means that it is assumed that for $\overrightarrow{N}(x_i)$, we have to previously access the elements $\overrightarrow{N}(x_1), \overrightarrow{N}(x_{...}), \overrightarrow{N}(x_i)$ by order. Nevertheless, there might exist some classes of compression functions where the access might not need to respect causality or order, namely those more simplistic but with lower description capabilities.

The number of bits needed to describe the object x, can be computed as the sum of the number of bits of each x_i, namely

$$C(x) = \sum_{i=1}^{|x|} \overrightarrow{N}(x_i), \tag{4.16}$$

where as i increases, the compressor is asymptotically able to better predict the following outcomes, because it creates an internal model of the data. In other words, C is learning.

However, for a detection application, such as in motif searching, its computation might have some problems, namely because only after seeing the second time some repetitive region, C is able to describe it in $\overrightarrow{N}(x)$. Therefore, if we are searching for similar regions we might need to proceed in a more specific way, such as in computing the minimum of each element after computing $\overrightarrow{N}(x)$ and $\overleftarrow{N}(x)$, where

$$\overleftarrow{N}(x_{|x|-i+1}) = C(x_{|x|-i+1}|x_{|x|}^{|x|-i+2}), \tag{4.17}$$

assuming that it respects the order $i = 1, 2, ..., |x|$. Both (4.15) and (4.17) can be computed parallel. However, only after having their complete numerical sequences we are able to compute

$$\overleftrightarrow{N}(x_i) = \min\left\{\overrightarrow{N}(x_i), \overleftarrow{N}(x_i)\right\}, \tag{4.18}$$

where for a global measure we have

$$\overleftrightarrow{N}(x) = \sum_{i=1}^{|x|} \overleftrightarrow{N}(x_i). \tag{4.19}$$

Accordingly, in terms of complexity, the function $\overleftrightarrow{N}(x)$ describes a lower bound given by the model when assuming that there is an *oracle* giving information from future outcomes.

4.3.1.1 Applications

According to the defined compressor, we have extracted the complexity profiles, $\overrightarrow{\overleftarrow{N}}(x)$, for each of the three *S. pombe* chromosomes. These results are presented in Fig. 4.12. As it can be seen, there are locations of low information content which are clearly associated with DNA regions of biological interest, such as telomeric and centromere regions. Therefore, we have marked with letters A, C, D, F, G and I the telomeric regions and with letters B, E and H the centromere regions. Yet, these marked letters clearly identifies what is the long arm (q) and short arm (p) on each chromosome.

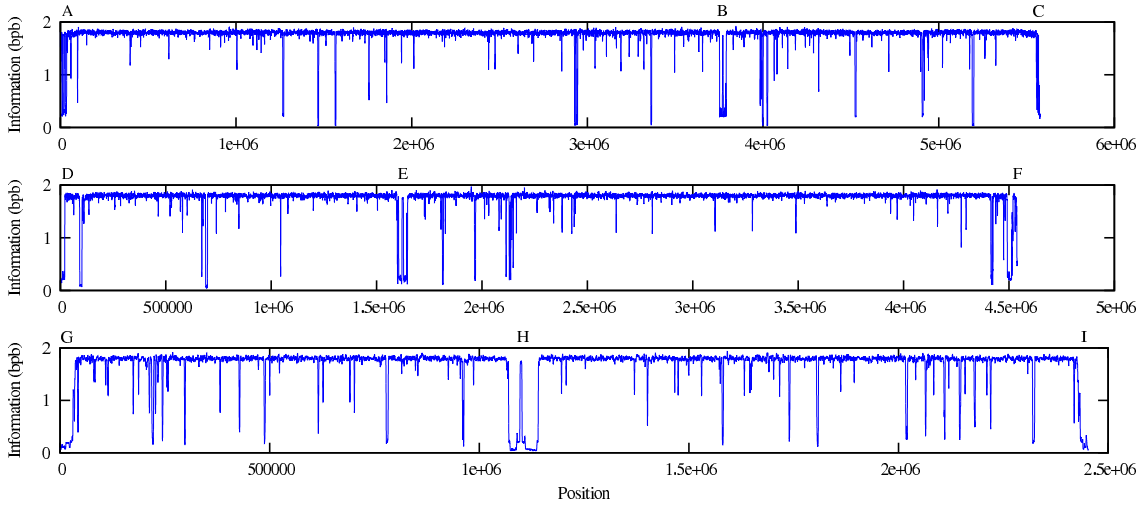


Figure 4.12: Complexity profiles for chromosome 1 (first row), chromosome 2 (second row) and chromosome 3 (third row) of *S. pombe*. The information content was processed in both directions, combined using the minimum value of each direction, and low-pass filtered using and a Blackman window of 1001 bases.

In some species, the centromeres are regions hard to find, due to the size of the sequence and nature. However, as Wood *et al.* [175] investigated, the *S. pombe* centromeres are large comparing to the budding yeast *S. cerevisiae*. In this way, we could easily identify them with complexity profiles (B, E and H). Moreover, as it can be seen in Fig. 4.13, the sizes of the centromeric regions vary inversely with the length of the chromosomes.

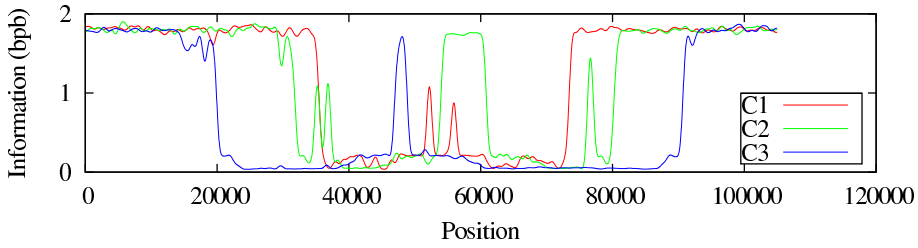


Figure 4.13: Complexity profiles of the centromeres from chromosome 1 (C1), 2 (C2) and 3 (C3). The information content was processed in both directions, combined using the minimum value of each direction, and low-pass filtered using a Blackman window of 1001 bases.

According to Wood *et al.* [175], possibly more extended centromeric regions are required for proper mitotic and meitotic behavior when the chromosome arms are shorter.

4.3.2 Conditional complexity profiles

A conditional complexity profile is a complexity profile that explores the existence of extra information, namely the information from an object y, where we define

$$\overrightarrow{N}(x_i|y) = C(x_i|x_1^{i-1}, y). \tag{4.20}$$

For a global measure, we sum for all positions, i. e., we compute

$$C(x|y) = \sum_{i=1}^{|x|} \overrightarrow{N}(x_i|y), \tag{4.21}$$

According to the previous definitions, the NCCD can be computed symbol by symbol, in other words, we can decompose it to element complexity. As such, if we know that the complexity of x is higher or lower than of y, substituting on (4.13), we are able to write the NCCD as

$$\text{NCCD}(x,y) = \begin{cases} \dfrac{\sum_{i=1}^{|x|} \overrightarrow{N}(x_i|y)}{\sum_{i=1}^{|x|} \overrightarrow{N}(x_i)}, & C(x) \geq C(y) \\ \dfrac{\sum_{i=1}^{|y|} \overrightarrow{N}(y_i|x)}{\sum_{i=1}^{|y|} \overrightarrow{N}(y_i)}, & C(x) < C(y) \end{cases} \tag{4.22}$$

In the same way, as in the complexity profiles, for a detection application, such as in motif search, we compute the minimum of each element after computing $\overrightarrow{N}(x|y)$ and $\overleftarrow{N}(x|y)$, where

$$\overleftarrow{N}(x_{|x|-i+1}|y) = C(x_{|x|-i+1}|x_{|x|}^{|x|-i+2}, y), \tag{4.23}$$

assuming that it respects the order $i = 1, 2, ..., |x|$. Both $\overrightarrow{N}(x|y)$ and $\overleftarrow{N}(x|y)$ can be computed in parallel. However, only after having their complete numerical sequences we are able to compute

$$\overleftrightarrow{N}(x_i|y) = \min\{\overrightarrow{N}(x_i|y), \overleftarrow{N}(x_i|y)\} \tag{4.24}$$

where for a global measure we have

$$\overleftrightarrow{N}(x|y) = \sum_{i=1}^{|x|} \overleftrightarrow{N}(x_i|y). \tag{4.25}$$

In terms of complexity, the function $\overleftrightarrow{C}(x|y)$ describes a lower bound given by the model, furnishing extra auxiliary information from y, when assuming that there is an *oracle* giving information from future outcomes from x.

4.3.2.1 Applications

Based on the conditional complexity profiles, we made an inter-chromosomal study of the *S. pombe* genome. We have computed $\overrightarrow{\overleftarrow{N}}(x_i|y)$ using chromosome 3 as x and chromosome 1 as y. Likewise, we have computed the same process, although using chromosome 2 as y instead of chromosome 1. Both profiles are represented in Fig. 4.14, as well as the complexity profile $\overrightarrow{\overleftarrow{N}}(x)$ for chromosome 3, to unveil conditional information exclusively from y.

In Fig. 4.14, we have unveiled important regions marked with the letters A, B and C. Starting with the letter B, this region contains the 2529 bases of gene eft202 (from base 537326 to 539854, in chr. 3). According, the statistics that unveiled this gene were extracted also from 2529 bases of chromosome 1, which represent gene eft201 (from base 2907701 to 2910229, chr. 1, with ∼99% sequence similarity to gene eft202).

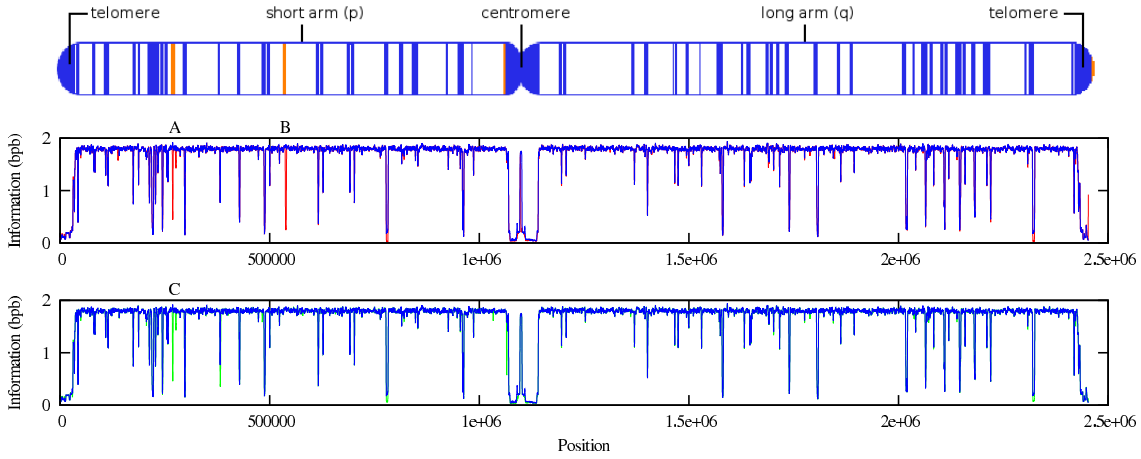


Figure 4.14: Conditional complexity profiles for chromosome 3 of *S. pombe*. The first row shows a representation for chromosome 3 and their long repetitive zones. The second row shows chromosome 3 (blue) with information added from chromosome 1 (green). The third row shows chromosome 3 (blue) with information added from chromosome 2 (red). The information content was processed in both directions, combined using the minimum value of each direction, and low-pass filtered using a Blackman window size of 1001.

In relation to the region marked with letter A (Fig. 4.14), we have verified that chromosome 1 unveiled a repetition in chromosome 3, that represents two highly similar genes (ef1a-a in chromosome 3 and ef1a-b in chromosome 1 with ∼98% sequence similarity). Moreover, chromosome 2 unveiled another very similar gene, ef1a-c, with ∼98% sequence similarity to both previous genes. In Fig. 4.15, there is an illustration that shows the relative position of these genes. In this illustration, letter A marks a region from base 4095202 to 4096584 (1383 bases, chr. 1). Letter B refers to base 626106 to 627488 (1383 bases, chr. 2), and letter C from base 268097 to 269479 (1383 bases, chr 3).

Interestingly, these very similar genes, present in all *S. pombe* chromosomes (a very rare property) and always in the short arm, have homologous in the following species: human, chimpanzee, dog, cow, rat, chicken, zebrafish, fruit fly, mosquito, *C. elegans*, *S. cerevisiae*, *K. lactis*, *E. gossypii*, *M. grisea* and *N. crassa*.

Figure 4.15: Illustration of the three chromosomes of *S. pombe* genome marked with genes ef1a-b (A), ef1a-c (B) and ef1a-a (C).

4.3.3 Relative complexity profiles

A relative (or conditional exclusive) complexity profile is a complexity profile that explores exclusively the existence of information from other object, namely the information from object y, where

$$\overrightarrow{N}(x_i\|y) = C(x_i\|y). \tag{4.26}$$

For a global measure, we sum the instants

$$C(x\|y) = \sum_{i=1}^{|x|} \overrightarrow{N}(x_i\|y). \tag{4.27}$$

According to the previous definitions, the NRC can be decomposed by elements, in other words, we can decompose it to element complexity. As such, substituting on (4.14), we are able to decompose the NRC up to

$$\mathrm{NRC}(x,y) = \frac{1}{|x|}\sum_{i=1}^{|x|} \overrightarrow{N}(x_i\|y). \tag{4.28}$$

If we consider a lossy relative compressor (L), based on multiple FCMs that compete to represent a certain block of b symbols, and ignore the side information (having $b = 1$) we are able to say that for each $N_L(x_i\|y)$ they have the same complexity, independently of the direction they are processed, and hence

$$\overrightarrow{N}_L(x_i\|y) = \overleftarrow{N}_L(x_i\|y). \tag{4.29}$$

If we consider a lossless relative compressor, with side information, where $b > 1$ and $|x| \mod b \neq 0$ and our intention is to hold (4.29), then we need to synchronize the blocks (given the processing direction). According to Fig. 4.16, the first block of $\overleftarrow{N}(x_i\|y)$ will have $|x| \mod b$ symbols, while the rest b symbols. Moreover, we assume that the side information is not being compressed according to a causal context. At this point, we have a competing lossless compressor that measures information regardless the order of the blocks. In fact, the computation can be strongly parallelized (having a maximum number of threads equal to $|x|/b$) without affecting the measure.

However, for compressors relying on competition, considering causal context side information (namely encoded by a FCM), or mixtures led to the following inequality

$$\overrightarrow{N}(x_i\|y) \neq \overleftarrow{N}(x_i\|y). \tag{4.30}$$

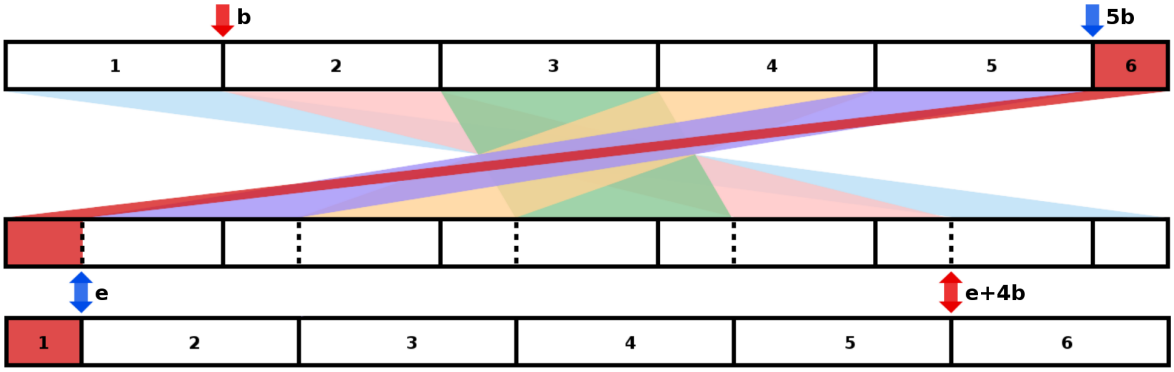


Figure 4.16: Synchronizing multiple blocks for processing $\overleftarrow{N}(x_i\|y)$ (bottom map) after reversing $\overrightarrow{N}(x_i\|y)$ (top map). The point b represents the block size, while e the extra given by $|x| \mod b$.

As an example we provide, in Fig. 4.17, a situation using a compressor based on a mixture, described previously, where the relative profiles differ according to the DIFF profile. We believe that, for large sequences and low b, these differences might turn to be insignificant.

Nevertheless, for overall quantities we get

$$C(x\|y) \approx \sum_{i=1}^{|x|} \overrightarrow{N}(x_i\|y) \approx \sum_{i=1}^{|x|} \overleftarrow{N}(x_i\|y), \tag{4.31}$$

which agree with the overall compressed values according to Fig. 4.17, where $\sum_{i=1}^{|x|} \overrightarrow{N}(x_i\|y) = 29,783,776$ and $\sum_{i=1}^{|x|} \overleftarrow{N}(x_i\|y) = 29,784,688$ bits, as well as in what it concerns overall resources (memory and computation time).

Therefore, both competing and cooperating approaches in learning processes create specific connections relatively to the direction, in spite of using in the overall approximately the same relative complexity.

4.3.3.1 Applications

The direct application of $C(x_i\|y)$ can be used to locate regions of similar information, such as identifying motifs. Fig. 4.18 depicts an example of a motif search analysis. In this case, the gene DCC (Deleted in Colorectal Carcinoma netrin 1 receptor) is used as a sample for a search in human chromosome 18 sequence. Accordingly, without loading a reference model, GeCo will only use the target models, in agreement with the running mode and, hence, will identify regions with high/low complexity in a blind mode search. These regions are mostly related with repetitive elements, such as transposons, telomeres and centromeres. Using a human reference DCC gene sequence, we easily identify the similar region containing the DCC gene. Moreover, when we use the chimpanzee DCC gene as a reference we are also able to identify the human DCC gene in the sequence of chromosome 18.

Another application is to further explore the results mentioned in Section 4.2.2.4. For the purpose we have computed a relative complexity profile of the mitochondrial sequence, using exclusively the human chromosome 5 as reference, according to Fig. 4.19. As it can be seen,

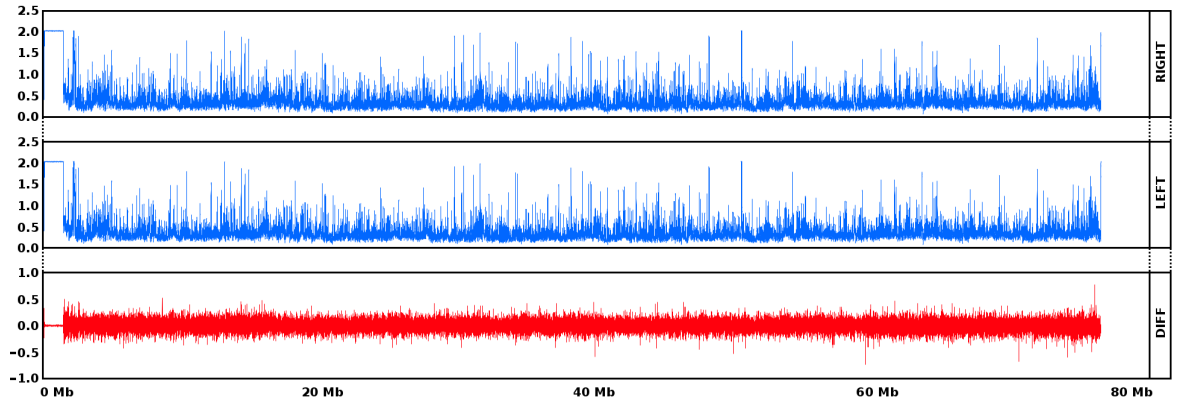


Figure 4.17: Computation of $\overrightarrow{N}(x_i\|y)$ (RIGHT) and $\overleftarrow{N}(x_i\|y)$ (LEFT) using the chromosomes 18 of human, as y, and chimpanzee, as x. The relative profiles, as well as the difference between them (DIFF), have been computed using GeCo tool and filtered with Goose framework. The plot depicts the complexity values in bits per base according to each i. GeCo ran with the following model parameters "-rm 4:1:0:0/0 -rm 10:1:0:0/0 -rm 13:10:0:0/0 -rm 20:500:1:3/50 -c 30". Goose filtering ran with a window of 1001, for LEFT and RIGHT, and 201, for DIFF.

there are many low complexity regions that are associated with important and annotated genes (given the NCBI map). Moreover, the associations are linked to chromosome 5 in a scattered way, where several are inversions.

There are many applications where these scenarios can be used, namely those who rely in comparative analysis [176], as it is further developed in Chapter 5.

4.3.4 Connection to relative compression

We studied the connection between compression types, namely between compression, conditional compression and relative compression, considering that the objects have been generated from a non-stationary source. As such, to relate them we need to follow a symbol by symbol approach.

If we consider a compressor using FCMs based on competition, with one $\mathcal{T}$ model and one $\mathcal{R}$ model, we are able to define a connection to relative compression, namely by

$$C(x_i|y) = \min\{C(x_i\|y), C(x_i)\} + S_i, \tag{4.32}$$

where S_i is the side information needed to describe the model that uses less bits to represent each symbol. When computing the sums we get

$$\begin{aligned} C(x|y) &= \sum_{i=1}^{|x|} \min\{C(x_i\|y), C(x_i)\} + \sum_{i=1}^{|x|} S_i \\ C(x|y) &= \sum_{i=1}^{|x|} \min\{C(x_i\|y), C(x_i)\} + S. \end{aligned} \tag{4.33}$$

As it can be seen, the $\min\{C(x_i\|y), C(x_i)\}$ acts as a lower bound and, therefore, the connection also holds for $\mathcal{R}_n$ and $\mathcal{T}_m$, in a competition scenario, without side information, and

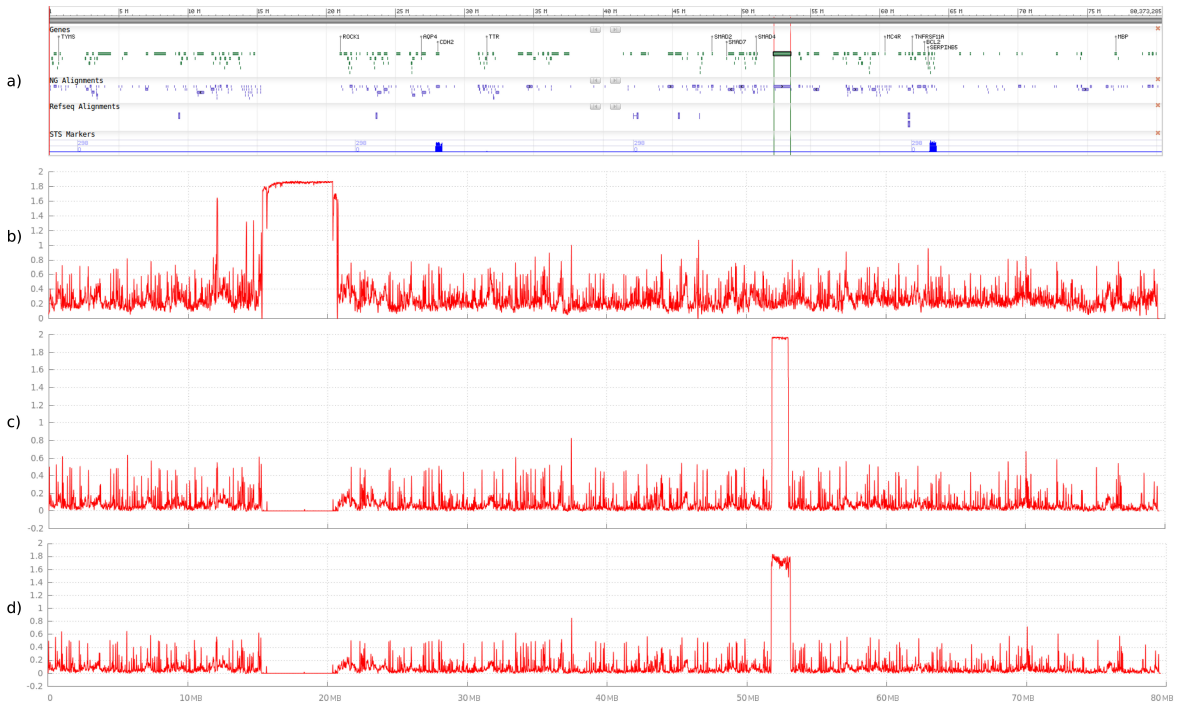


Figure 4.18: Human chromosome 18 redundancy profiles according to NCBI map; **a)** NCBI map with annotated regions with vertical bars delimiting DCC gene; **b)** redundancy profile without using conditional information, identifying mostly the centromere and several regions that might need zoom inspection; **c)** redundancy profile using DCC gene as conditional information, identifying efficiently DCC gene; **d)** redundancy profile using DCC chimpanzee gene as conditional information, identifying efficiently DCC gene in human. BLAST align tool ranks 98 % of identity between human and chimpanzee DCC gene. All the redundancy profiles have been computed using $R_i = 2 - C(x_i\|y)$. GeCo run with mode 1 and flag "-e". Filtering has been parameterized, using Goose filter tool, with 20,000 bases in a Hamming window.

---

hence,

$$C_L(x_i|y) = \min\{C_L(x_i\|y), C_L(x_i)\}, \tag{4.34}$$

where C_L is a lossy compressor that ignores side information (which is related to the decision of what is the best model) independently from the type of compression.

Although the minimum does not have an inverse, we can always approximate it as follows

$$\begin{aligned} C_L(x_i|y) &= \min\{C_L(x_i\|y), C_L(x_i)\} \\ C_L(x_i|y) &\approx \left(C_L(x_i\|y)^{-p} + C_L(x_i)^{-p}\right)^{-1/p} \\ C_L(x_i|y)^{-p} &\approx C_L(x_i\|y)^{-p} + C_L(x_i)^{-p} \\ C_L(x_i\|y)^{-p} &\approx C_L(x_i|y)^{-p} - C_L(x_i)^{-p}, \end{aligned} \tag{4.35}$$

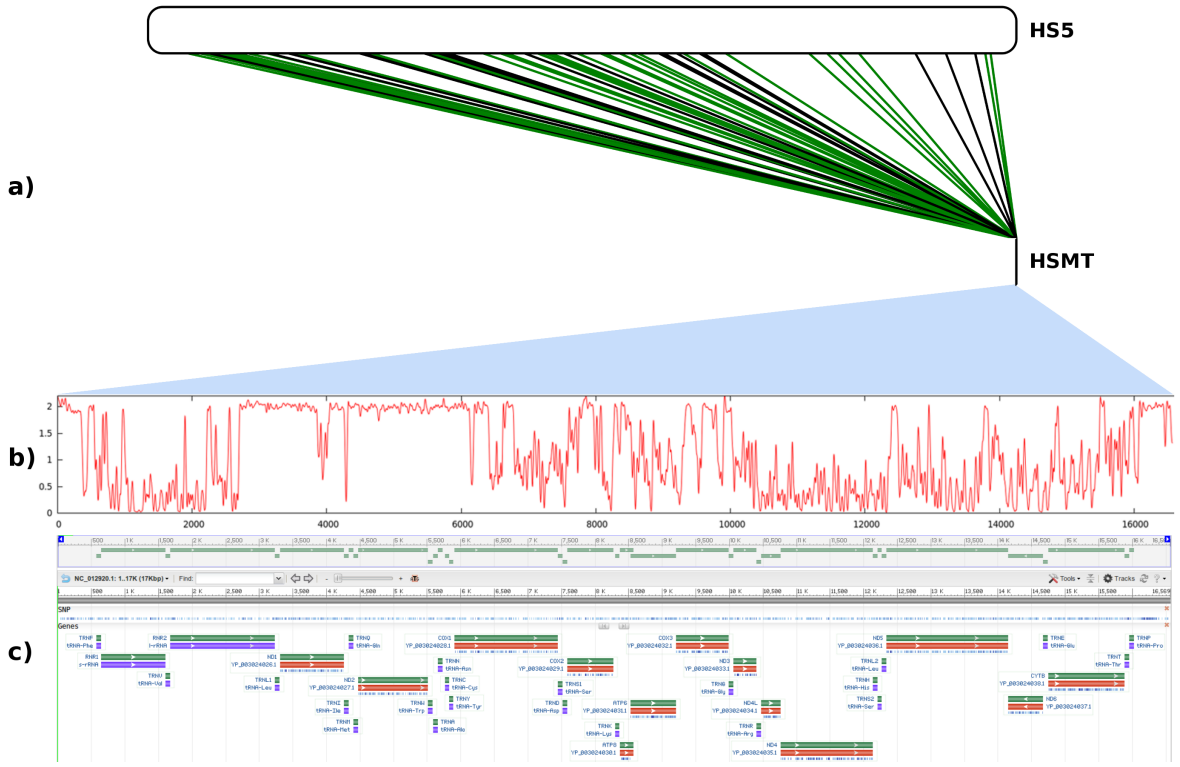


Figure 4.19: Relative profiles and maps for *H. sapiens* (HS) chromosome 5 and mitochondrial genome, sharing a high degree of relative complexity. **a)** The map has been computed using the smash tool (presented in Chapter 5), where green links represent inversions; **b)** the profile has been computed with with GeCo using chromosome 5 as reference; **c)** NCBI map with annotated regions.

---

and hence, for an isolated (lossy) conditional exclusive compression we get

$$C_L(x_i \| y) \approx \frac{1}{(C_L(x_i|y)^{-p} - C_L(x_i)^{-p})^{1/p}}, \tag{4.36}$$

considering $C_L(x_i|y) < C_L(x_i)$, since we are relying on an approximation, and where p is a large number.

The usage of model mixing requires a more elaborated connection, mainly because there is not direct side information, since the probabilities are mixed according to the performance of the models.

4.4 Conclusions

Compression-based global measures are a remarkable way to quantify the amount of information within objects or across them. In this chapter, we have proposed a way to compute the NID, without using the conjoint information but rather the conditional information. This simplistic computation view requires the definition of a conditional compressor, that we have described and explored. As an application, we have measured the distance between genomic

sequences, mostly chromosomes, within the same species and between different species, reporting several insights of evolution already known, but also several undocumented.

We have introduced a way to quantify relative information, namely through the normalized relative compression (NRC) that also requires a specific compressor. Its computation is deeply simpler, using much less resources (time and memory), and has the possibility to be computed using several parallel forms and, in some cases, it can be accessed without order (has we have shown in the local measures). We have measured the NRC within several RNA bird species and in chromosomes of several primates, being able to determinate most of the NCCD results (in the case of primates), but also unveiling new ones.

We have explored local measures that derive from the global, presenting several definitions and applications. The ability "to look at a DNA sequence" and immediately being able to identify specific regions has been shown to be a valuable tool, namely for identification of motif, centromere, telomere, homologous genes, among others. Finally, we have made a connection between the compressors that approximate the Kolmogorov complexity and the relative compression at a symbol scale. In the next chapter we will use the relative profiles in an unsupervised method to detect and visualise genomic rearrangements.

"Let's put all the pieces together. Push that string!"

Spider

5 Genomic rearrangements

In Section 4.3.3 we have formalized the conditional exclusive profiles, that measure the quantity of algorithmic information needed to describe a certain region given exclusively other. This notion of relative information provides a way for clustering similar regions. In this chapter, we develop a method to compute and visualise this type of regions that are directly associated to structural genomic rearrangements.

In fact, structural genomic rearrangements are a major source of intra- and inter-species variation. Chromosomal inversions, translocations, fissions and fusions, are part of the naturally occurring genetic diversity of individuals, are selectable and can confer environment-dependent advantages [177].

Chromosome rearrangements are also associated with disease, namely, developmental disorders and cancer. For example, many leukaemia patients present a reciprocal translocation between chromosomes 9 and 22, also known as the Philadelphia chromosome. This produces BCR-ABL fusion proteins that are constitutively active tyrosine kinases, contributing to tumour growth and proliferation [178].

Another striking example is the human inversion polymorphism in the 17q21 region, which contains the neurodegenerative disorder-associated gene *MAPT* (microtubule associated protein Tau). The direct oriented H1 haplotype is common and relates with increased Alzheimer's and Parkinson's disease risk, while the inverted H2 haplotype has higher frequencies in Southwest Asia and Southern Europe populations, particularly around the Mediterranean [179, 180]. Recurrent inversions are found in the primate lineage, where the H2 haplotype is the ancestral state, and recent work evidences that Neanderthals and Denisovans also carried the H1 allele [181].

How genome architecture changes contribute to speciation and which macroevolutionary events occurred through time are fundamental to understand the dynamics of chromosome evolution, and hence, the origins of species. In addition, chromosome alterations are hallmarks of cancer genomes with diagnosis and prognosis value [182], and are also used in prenatal and postnatal clinical settings.

Several insights into chromosome structure and evolution have been traditionally achieved by cytogenetic procedures such as G-banding, or molecular karyotyping approaches like fluo-

rescence in situ hybridisation (FISH) and, more recently, array-based methods [183]. However, in some groups, such as the great apes, access to samples is often difficult, e.g. due to ethical reasons. Also, these approaches can be time-consuming, expensive, or lack resolution, as opposed to computational solutions [184].

The advent of sequencing technology enabled the analysis of genomic sequences at nucleotide resolution. Nowadays, next-generation sequencing is bringing a substantial increase of speed, quality and reliability of the results for much less costs, although there is still promising space for improvements. The availability of sequenced genomes boosted computational methods into a new era, allowing some expensive and/or lengthy *wet lab* processes to be complemented by computational approaches [185].

Derived scientific insights from genomic sequences, including the conserved distribution of genes on the chromosomes of different species or synteny, have been mostly explored using sequence alignments [186, 187, 188, 189, 190, 191, 192, 193, 194, 195], while for visualization, a wide variety of strategies have been proposed [196, 197, 198, 199, 200]. Specifically, at a macro level the most popular are Mauve [189], Cinteny [201], Apollo [200], MEDEA (`http://www.broadinstitute.org/annotation/medea`), MizBee [202] and Circos [203], which are discussed in a review [204]). Although, the circle-based visualization is becoming very popular, for detecting block alignments and re-arrangements across very similar species, such as primates, an ideogram still seems to be the best approach.

We propose a computational method to detect signatures of chromosome evolution. The method is completely alignment-free and is based on the information content of the sequences being compared. The information content itself is estimated using data compression techniques. The resulting stand-alone algorithm depends only on two parameters.

We developed a tool by means of which the method can be tested in practice. The tool has been made publicly available and is described in detail. It is capable of producing an SVG image that shows the correspondence of regions between two sequences, together with a file reporting the respective positions and types of rearrangements.

Its performance is demonstrated with the help of several examples. Those involving synthetic sequences are intended to illustrate the underlying principles. More realistic case studies, involving prokaryotic and eukaryotic genomes, are also discussed. In particular, for intra-species and inter-species analysis, where for the later we obtain human/chimpanzee and human/orangutan complete chromosome maps.

For clarity, the potential and limitations of the tool and some of its design tradeoffs are discussed separately, following the description of the method. This separates limitations that are inherent to the method from those that are by-products of the current implementation, and that as such might be removed in future implementations.

5.1 The method

5.1.1 Creating models of the data

The method identifies small-scale or large-scale rearrangements between pairs of DNA sequences called the reference and the target. The method applies to arbitrary sequences, and therefore the reference and the target can be as large as an entire chromosome or genome. The goal of the method is to automatically detect regions in the target sequence that have information content similar to regions found in the reference. The method yields a set of

segments of the target sequence and, for each of these, the corresponding segment found in the reference sequence.

Both sequences are preprocessed such that their alphabet is $\Theta = \{\mathrm{A}, \mathrm{C}, \mathrm{G}, \mathrm{T}\}$. Symbols originally not belonging to Θ (for example, N's) are substituted by uniformly distributed symbols from Θ, in order to keep the original length of the sequence. These pseudo-random generated segments have high entropy relatively to the compressor (since the compressor is not able to determine the generation function and they have uniform distribution), therefore, with high probability will not share information with any other sequence, hence will not interfere with the matching process.

The core of the method involves the estimation of the amount of conditional information that is required to represent a certain region of the target, using exclusively information from the reference and, hence, relative complexity. Basically, if x and y are, respectively, the target and reference sequences, we compute a numerical sequence $\overrightarrow{N}(x_i \| y)$, where $1 \leq i \leq n$ and $n = |x|$ is the size of the target sequence. For a position i in the target sequence, $\overrightarrow{N}(x_i \| y)$ measures the number of bits required to represent the symbol located in that position, according to the aforementioned interpretation of conditional exclusive information (see Chapter 4).

To properly estimate $\overrightarrow{N}(x_i \| y)$, it is crucial to have a good model of the reference sequence y. We have chosen finite-context models (FCMs) for this purpose, namely because they seem to be the models that represent better genomic sequences given the compression capabilities using low computational resources, comparing with other models. An introduction to FCMs is made in Section 2.2.1. In this case we opted to set the parameter α to 0.001, forcing the estimator to behave approximately as a maximum likelihood estimator. In practice, this makes the segmentation process easier (see below). The number of bits that is required to represent symbol x_{i+1} using exclusively information from the reference sequence is given by

$$\overrightarrow{N}(x_{i+1} \| y), \tag{5.1}$$

according to the notion of relative complexity profiles in Section 4.3.3. We intend to clarify that to estimate the information content, any model can be used as long as it is able to perform conditional exclusive compression, and more specifically, be able to incorporate the complete memory of these sequences (see [120] for complete object representability notions).

5.1.2 Finding information-similar regions

As explained before, the core idea of the method is to compute, along the target sequence x, the amount of information required to represent x using exclusively information from the reference sequence y. Therefore, at a first stage, we end up with a numerical information sequence $\overrightarrow{N}(x_i \| y)$ of size $n = |x|$. Figure 5.1 illustrates how the method operates, using synthetic data generated with an appropriate tool (XS [108]). The target was created by manipulating some parts of the reference, as described in the figure.

Regions where $\overrightarrow{N}(x_i \| y)$ is small indicate a high level of information sharing with y. To mark them, we compare a smoothed version of the information sequence with a threshold (T). The result is the set of regions of interest of x, for the given reference y, which are denoted by $x^{(l)}, l = 1, 2, ..., L$.

It remains to find the regions of the reference y which are strongly associated with each $x^{(l)}$. To do this we invert the roles of the reference and the target. More precisely, each $x^{(l)}$ is now regarded as a reference, and y is taken as the target. We thus compute, for each

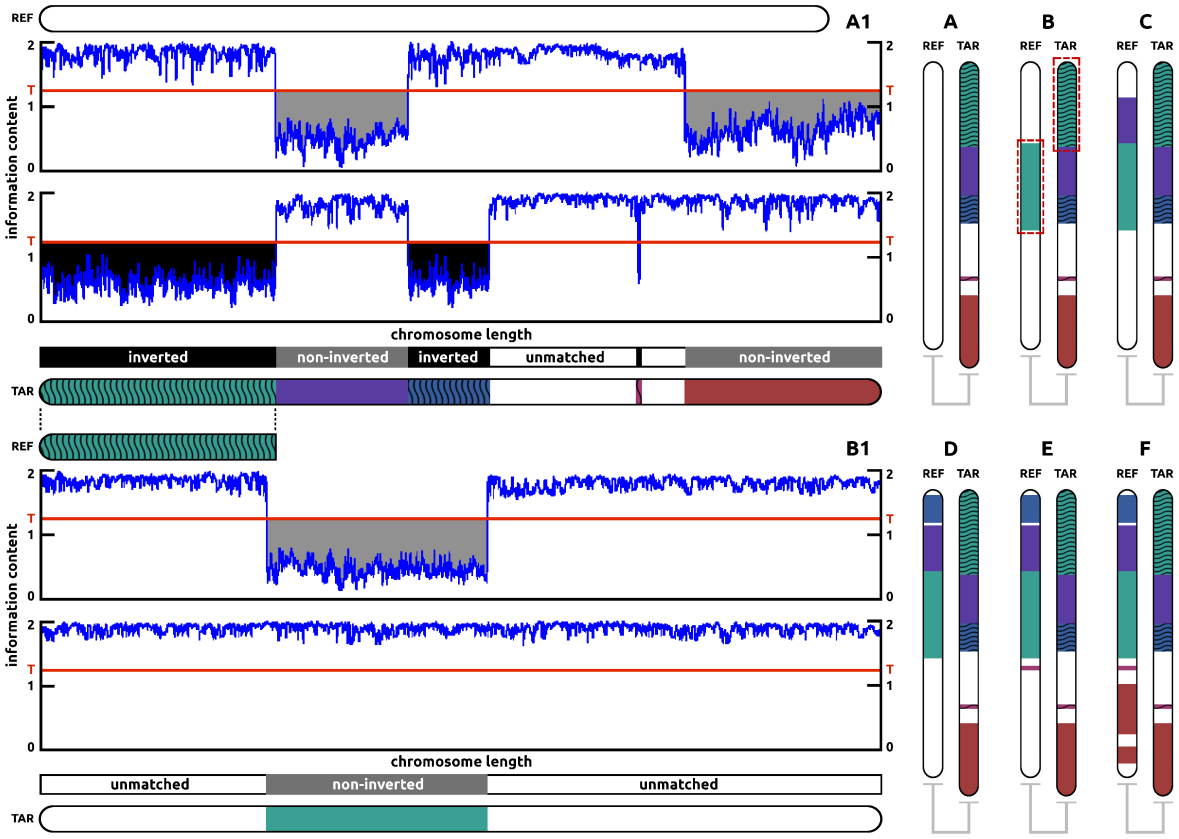


Figure 5.1: Similarity discovery, step by step. (**A**), scan the target to identify those of its regions that significantly share information content with the reference. (**B**) scan the reference to find those of its regions associated with each region identified at step **A**. Step (**C**), (**D**), (**E**), (**F**), repeat step **B** for each region identified at step **A**.

$l = 1, 2, ..., L$, the information sequences $\overrightarrow{N}(y_i \| x^{(l)})$, from which the regions of y associated with each $x^{(l)}$ can be found.

The described procedure can find pairs of regions that are similar in the sense of information-sharing, but does not take into account possible inversions. For this purpose, the reference sequence should be reverted, complemented and loaded into the FCM model. Then steps entirely similar to those described above need to be taken. Having done this, both inversions and direct homologies can be segmented in the target sequence.

If both the inverted and direct instances of a region are found to have high information content, then the region shares no information with the rest of the data and therefore it is left unmarked. This is the case with regions that are essentially unique and with unsequenced regions (those that originally contained N's, that have been replaced with random data).

5.2 The tool

5.2.1 Software availability

An implementation of the method (Smash) is freely available, under GPL-2 license, at `http://github.com/pratas/smash`. Smash is a tool that computes chromosome information maps, with an ideogram output architecture. The colors for each block are automatically calculated using the HSV (Hue, Saturation, Value) color space, where only the Hue varies. For more information about Smash, see the manual that follow the tool.

5.2.2 The threshold T

Smash has a command-line option by means of which the threshold T can be varied in the interval $[0, 2]$. The threshold can be regarded as a parameter. In general, the best T is data-dependent. The guiding principle is to choose T so that it selects regions of complexity sufficiently below the average. In practice, this is not difficult to achieve, but some experimentation may be required to obtain the best results.

As a rule, T should be smaller when working with similar species than when working with more distant species. For example, for the human/chimpanzee pair we used $T = 1.3$ but for the chicken/turkey pair we used $T = 1.95$. When working with entire chromosomes, the threshold can be adjusted to match the degree of divergence encountered.

5.2.3 Model depth

The model depth, described by the parameter k, must be an integer in the range $[1, 28]$. The default value ($k = 20$) works well for sequences, say, longer than 1 Mb (1,000,000 symbols). The default also works well for smaller sequences, although in this case the actual performance may depend on how repetitive they are. We have found out that there is often little practical need to tune k.

The relation between the model depth k and the estimated probabilities (which are directly related to the counters c_y), and the capabilities of Markov models in the context of DNA sequence modeling, have been treated in detail in [85, 205].

5.2.4 Compared with other methods

Accordingly, we have included two synthetic sequences of 4 kb each and made a comparative study with respect to other techniques: Mauve [189], and VISTA [191]. The synthetic sequences of these figures were simulated using XS [108] and randomly permuted using different block sizes and inversions with a program from the Goose framework (`https://github.com/pratas/goose/`). The first one, Fig. 5.2, contains a small number of block permutations and inversions to allow an easy comparison with the provided ground truth. The second one, Fig. 5.3, contains overlapping permutations and, therefore, is more complex.

In Figs. 5.4 and 5.5, the methods are compared in real prokaryotic and eukaryotic sequences, respectively. Fig. 5.4 addresses a large-scale comparative study between *Shigella flexneri* (NC_017328) and *Escherichia coli* (NC_017638), using Smash and Mauve. Regarding the eukaryotic example, we have used chromosome 3 of human and orangutan, depicted in Fig. 5.5. Smash spent 871 seconds (for the inside Smash map) and 1291 seconds (for the

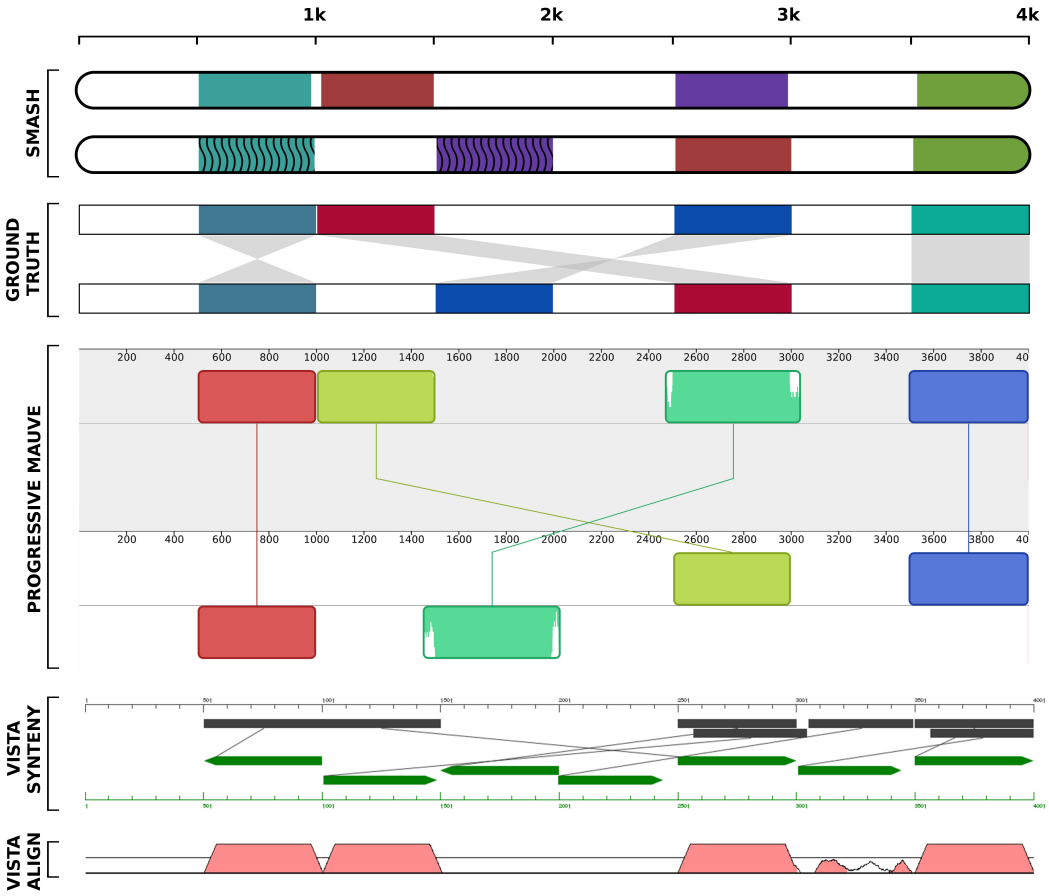


Figure 5.2: Comparison between Smash, Mauve and the VISTA methods on a synthetic sequence (4,000 bases) with a ground truth. Smash was ran with $T = 1.5$, $k = 10$ and discarding blocks smaller than 5 bases. VISTA was computed online, for both synteny and alignments.

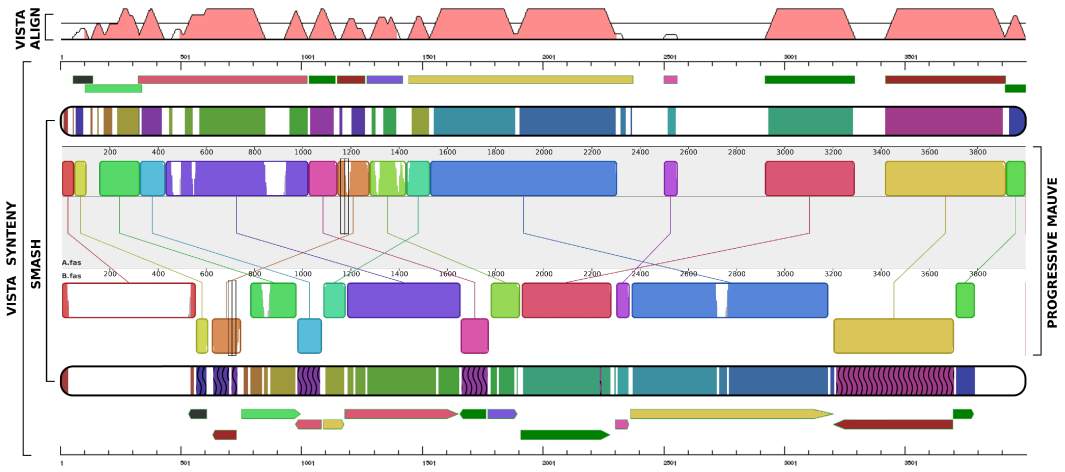


Figure 5.3: Comparison between Smash, Mauve and the VISTA methods on a more complex synthetic sequence (4,000 bases). Smash ran with $T = 1.5$, $k = 10$ and discarding blocks smaller than 5 bases. VISTA was computed online (syntheny) and the maps were adapted (colors instead of pointer lines) for better display. First sub-image depicts only the VISTA alignments.

outside Smash map) using a maximum memory peak of 2.4 GB, while Mauve spent 2633 seconds and used 6.1 GB of memory. The Mauve version used is already the improved (latest) version.

5.2.5 Commutativity

The method is commutative, that is, it has the potential to lead to the same results when the reference and the target are swapped. Smash can easily be made commutative as well. However, in most usage scenarios, there is a natural reference sequence. Furthermore, the assumption that one of the two sequences is the reference simplifies the algorithm and leads to time savings. For these two reasons, the current implementation of Smash is approximately commutative, but not exactly so.

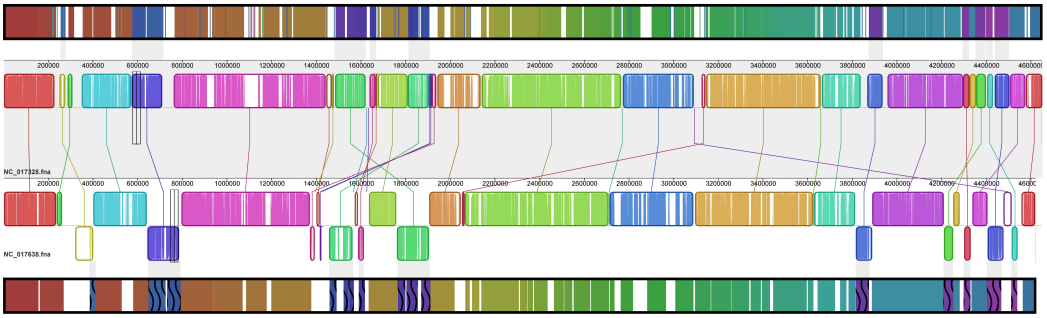


Figure 5.4: Comparison between Smash and Mauve methods on *S. flexneri* and *E. coli* bacterial genomes. Smash was ran using $T = 1.8$, $k = 20$ and discarding blocks smaller than 20 kb.

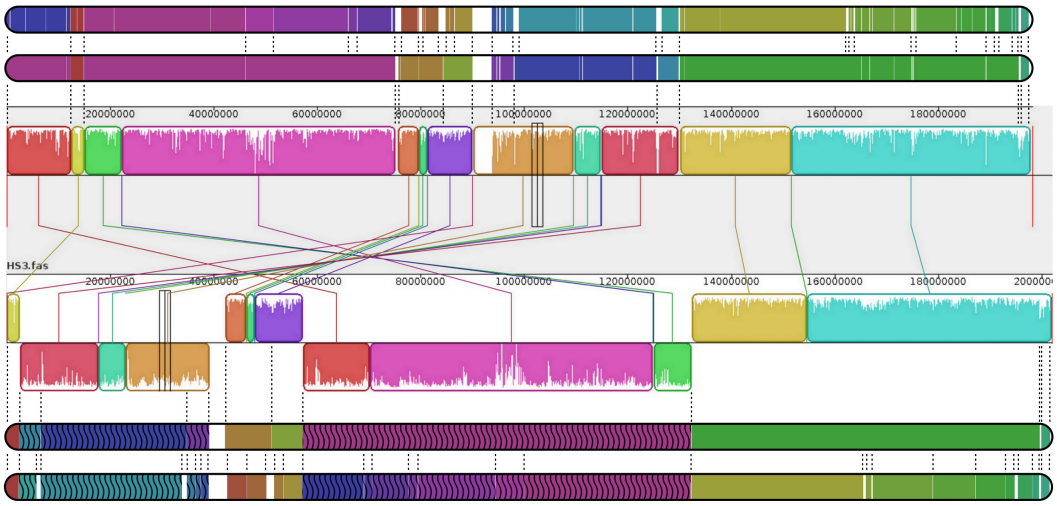


Figure 5.5: Comparison between Smash and Mauve methods on chromosome 3 of *H. sapiens* and *P. abelii*. Smash was ran using $T = 1.5$ (inside Smash map) and $T = 1.6$ (outside Smash map), $k = 20$ and discarding blocks smaller than 1 Mb.

To illustrate this, we performed additional experiments using both prokaryotic and eukaryotic genomes. For the prokaryotes, we have used *Shigella flexneri* (NC_017328) and *Escherichia coli* (NC_017638). As can be seen in Fig. 5.6, the maps are similar (apart from some differences in color and reversed pattern assignment, due to the automatic coloring method used). Nevertheless, it is possible to spot small differences, mainly because we have discarded matched regions smaller than 20 kb. Fig. 5.7, which illustrates the human/chimp pair, shows that at a larger scale these small differences tend to disappear.

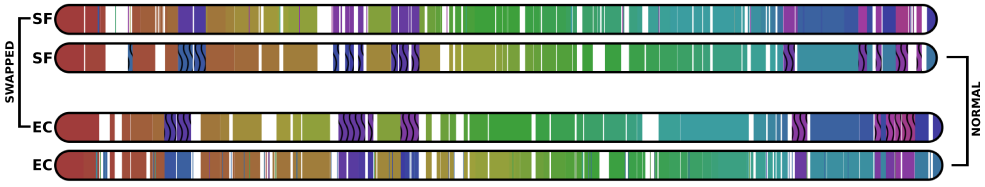


Figure 5.6: Smash result when the reference and the target are swapped. "SF" stands for *S. flexneri* and "EC" for the *E. coli* bacterial genomes. Smash was ran using $T = 1.8$, $k = 20$ and discarding blocks smaller than 20 kb.

5.2.6 Working with distant genomes

Smash does work for more distant genomes than, say, the human/chimpanzee pair studied in detail next. This is shown e.g. by the chicken/turkey map of chromosome 1 depicted in Figure 5.8. According to TimeTree [206], *Gallus gallus* and *Meleagris gallopavo* have an estimated divergence time of 44.6 million years (MY), while between *Homo sapiens* and *Pan troglodytes* or *Pongo abelii* the divergence times are estimated as 6.3 MY and 15.7 MY, respectively.

We emphasize, however, that Smash can be applied to pairs of sequences that are even

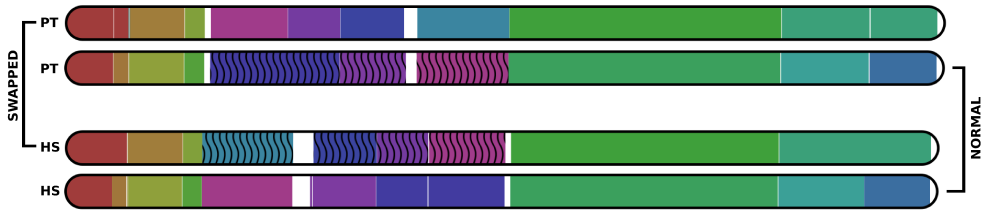


Figure 5.7: Smash result when the reference and the target are swapped. "HS" denotes the *H. sapiens* whereas "PT" indicates the *P. troglodytes* eukaryotic genomes. Smash was ran using $T = 1.3$, $k = 20$ and discarding blocks smaller than 1 Mb.

Figure 5.8: Smash computation of the *M. gallopavo* (top map) and *G. gallus* (bottom map) chromosome 1. Smash has been computed using a threshold $T = 1.95$, a context size of $k = 20$ and discarding blocks smaller than 1 Mb.

more distant. Regardless of the exact nature of the reference and target, Smash will find the rearrangements present, even if one or both sequences are synthetic (computer generated). This can be useful to develop a better understanding of how Smash works, or for testing purposes.

5.2.7 Working with unassembled sequences or assembling errors

One of the advantages of Smash is that it works even when the reference is not assembled. Therefore, it can be used with references composed of non-assembled reads obtained directly from the NGS sequencers. In fact, although next-generation sequencing made low cost high speed sequencing possible, it also decreased the size of sequencing reads [50]. On the other hand, most of the primate assembled sequences use the human genome as a reference. This might be problematic, because of the assumption that humans and the other primates exhibit a high degree of homology, which might not always be true [137]. Hence, it might be important to measure similarity against non-aligned references.

Fig. 5.9 depict the results of Smash over chromosome 18 of human and chimp using random permutations of blocks with different size, showing its robustness when fragmented references are used. Smash spent less than 8 minutes for each computation.

Smash is able to identify regions containing shared information even when one of the sequences is block-permuted, a capability that may be of interest to measure sequence similarity, e.g. when one of the sequences is not assembled, or when there are assembly errors. Obviously, the identification of the precise genomic rearrangements that took place will have to be deferred until final assembly takes place.

5.3 Results and Discussion

To illustrate the potential of the method, we show the chromosomal information maps in a two line approach: (human) intra-species and (primates) inter-species. For the first one,

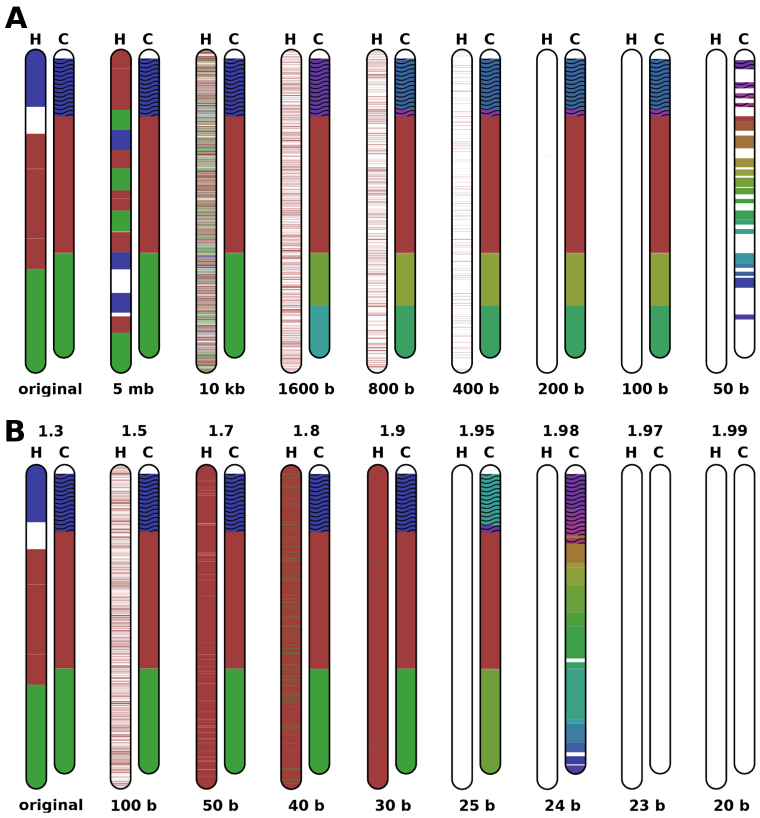
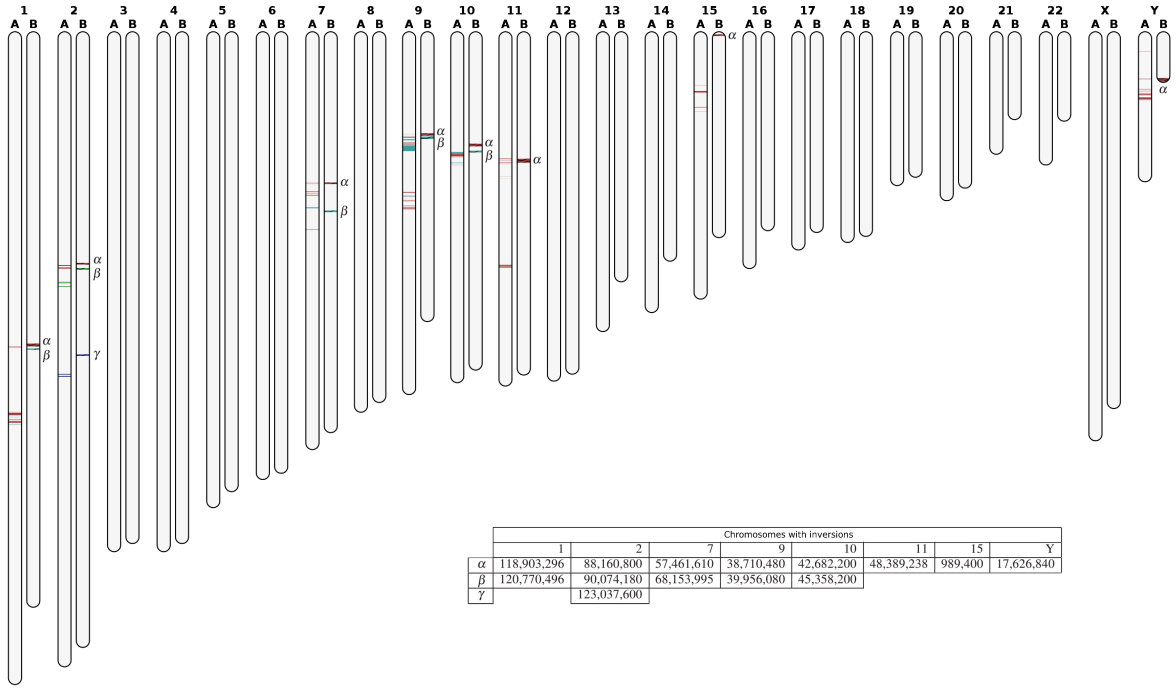


Figure 5.9: Smash computation over *P. troglodytes* chromosome 18, using as reference permuted blocks of different sizes from *H. sapiens* chromosome 18. Colors are only consistent for each run of the tool and, therefore, may not be consistent from one run to another run, where the sequences or the parameters are changed. (**A**) Smash was executed using $T = 1.3$ and $k = 20$. (**B**) Smash was executed using a variable threshold T (upper value) and $k = 20$.



	Chromosomes with inversions							
	1	2	7	9	10	11	15	Y
α	118,903,296	88,160,800	57,461,610	38,710,480	42,682,200	48,389,238	989,400	17,626,840
β	120,770,496	90,074,180	68,153,995	39,956,080	45,358,200			
γ		123,037,600						

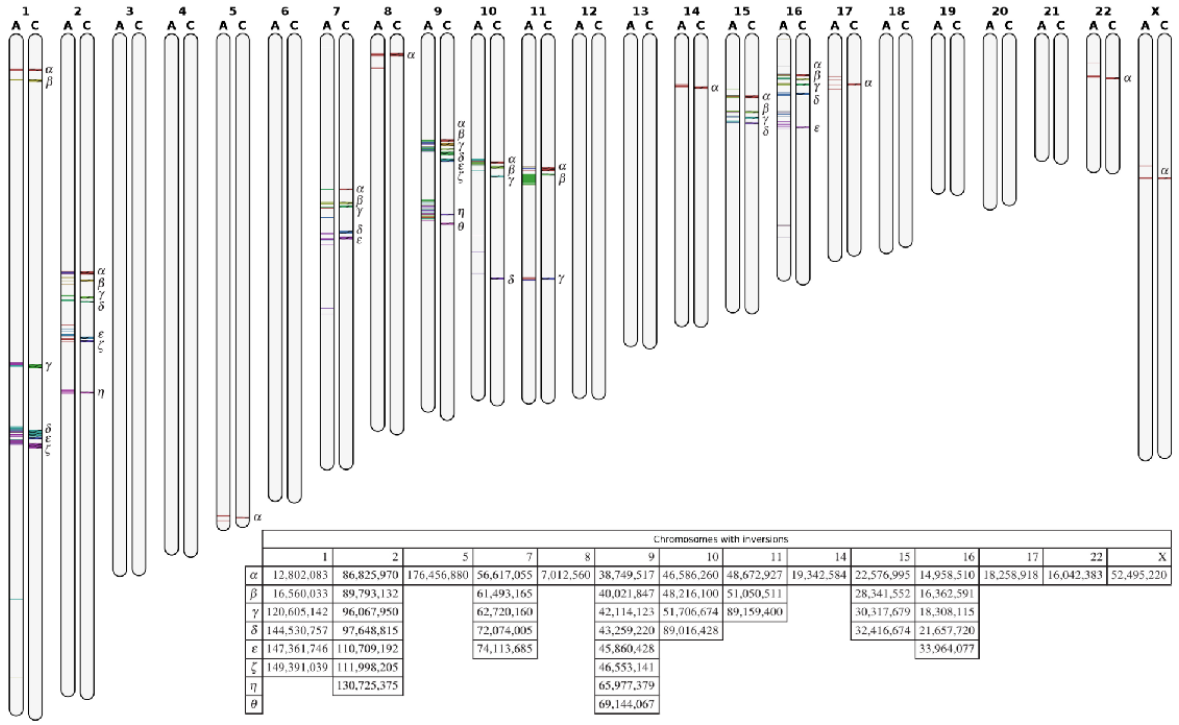
Figure 5.10: Large-scale inversions between GRC (A) and HuRef (B) assemblies for each chromosome. The information maps show exact or approximate inversions with length higher than 500 kb. Each position associated with inversions, in the HuRef chromosomes, is reported in the table and marked with a Greek letter according to the map.

since there is a high degree of similarity, we only assess the large-scale inversions larger than 500 kb, while for the second one we make a full large-scale analysis and also include several small-scale analysis.

5.3.1 Human intra-species maps

Advances in sequencing technology have increased the number of digital human genomes, raising conditions towards intra-species characteristics and diversity research. However, the *de novo* assembly of the next generation sequencing (NGS) reads is still problematic, mainly because the alignment of the reads from these new genomes to a high quality reference genome remains a critical aspect of data interpretation. Nevertheless, the human reference assembly is the highest quality mammalian assembly available. The main reference genome assemblies are those from the Genome Reference Consortium (GRC 38) [129], the J. Craig Venter Institute (HuRef) [207] and the Washington U. School of Medicine (CHM 1.1).

We use Smash on these different human assemblies, with respect to only inversions. According Fig. 5.10 shows the maps from the large-scale inversions between A (GRC) and B (HuRef), while Fig. 5.11 shows the inversions between A (GRC) and C (CHM) assemblies. In respect to A/B, there are inversions in chromosomes 1, 2, 7, 9, 10, 11, 15 and Y. Specifically to chromosome 1, the inversions are contained in the pericentric regions (around 119 Mb). This region is also inverted between human and chimpanzee species, although in a much larger density, as it can be seen in further results on the next sub-section.



	Chromosomes with inversions													
	1	2	5	7	8	9	10	11	14	15	16	17	22	X
α	12,802,083	86,825,970	176,456,880	56,617,055	7,012,560	38,749,517	46,586,260	48,672,927	19,342,584	22,576,995	14,958,510	18,258,918	16,042,383	52,495,220
β	16,560,033	89,793,132		61,493,165		40,021,847	48,216,100	51,050,511		28,341,552	16,362,591			
γ	120,605,142	96,067,950		62,720,160		42,114,123	51,706,674	89,159,400		30,317,679	18,308,115			
δ	144,530,757	97,648,815		72,074,005		43,259,220	89,016,428			32,416,674	21,657,720			
ε	147,361,746	110,709,192		74,113,685		45,860,428					33,964,077			
ζ	149,391,039	111,998,205				46,553,141								
η		130,725,375				65,977,379								
θ						69,144,067								

Figure 5.11: Large-scale inversions between GRC (A) and CHM (C) assemblies for each chromosome. The information maps show exact or approximate inversions with length higher than 500 kb. Each position associated with inversions, in the CHM chromosomes, is reported in the table and marked with a Greek letter according to the map.

On the other hand, for A/C the inversions are present between chromosomes 1, 2, 5, 7, 8, 9, 10, 11, 14, 15, 16, 17, 22, X. From these, most of the inversions are contained in pericentric regions, a major factor of dynamism across individuals of the same species.

5.3.2 Primate inter-species maps

We used Smash to compute the chromosomal information maps for inter-species, namely in the pairs human-chimpanzee, human-orangutan and also a translocation example in human-gorilla. The *Homo sapiens* (GRC), *Pan troglodytes*, *Gorilla Gorilla* and *Pongo abelii* reference assembled chromosomes were downloaded from the NCBI. In order to create the human-chimpanzee map, we have concatenated chromosomes 2A and 2B of the chimpanzee, ran Smash once per chromosome (totaling 23 runs), then manually corrected the associated picture regarding the hypothetical centromere between 2A and 2B, and finally grouped all the maps in one global picture (the one shown in Fig. 5.12). A similar process was done for the human/orangutan map, shown in Fig. 5.14. The results obtained confirm and extend previous work based on orthologous gene distribution, array comparative genomic hybridisation (array CGH) and FISH approaches [208, 209, 148].

Figure 5.12 shows the complete information maps between human and chimpanzee genomes, using chromosome pairwise comparisons, which are characterized by several inversions, in chromosomes 1, 4, 5, 7, 12, 15, 17, 18, and Y. All known pericentric inversions were detected

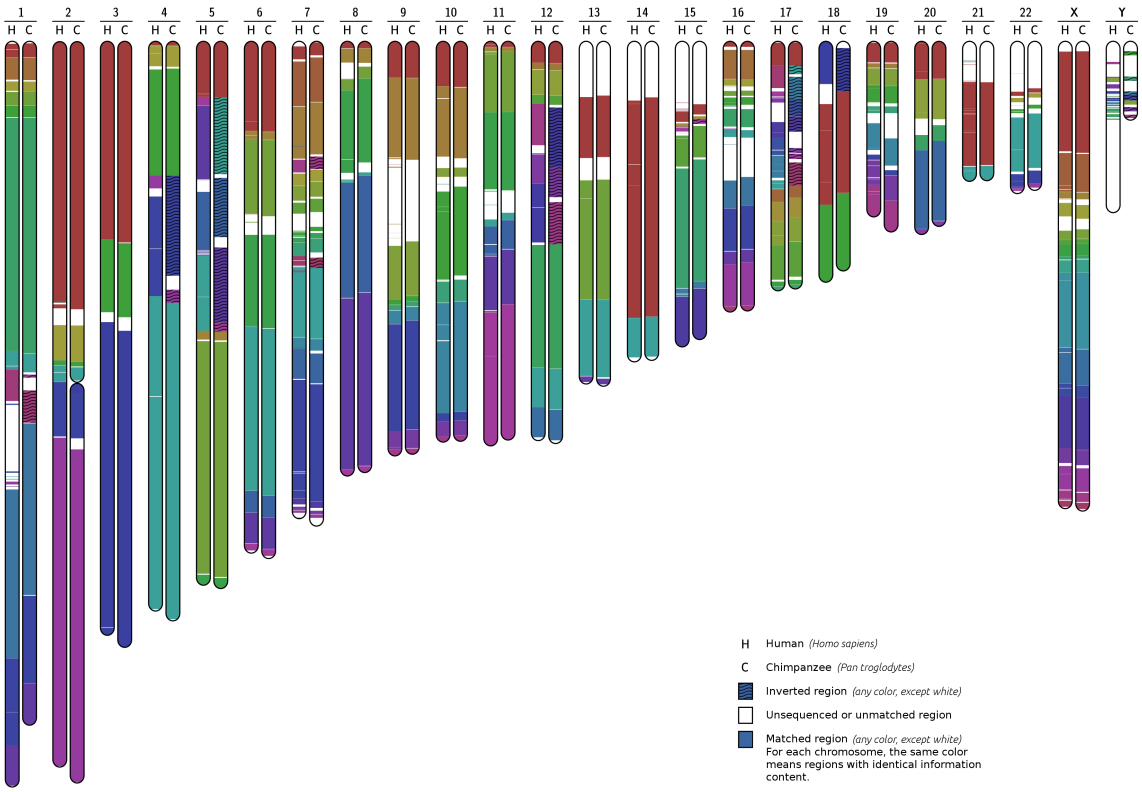


Figure 5.12: Human chimpanzee chromosomal map, obtained from chromosome pairwise comparison. Inversions can be observed in chromosomes 1, 4, 5, 7, 12, 15, 17, 18, and Y. Chromosomes 2A and 2B of chimpanzee have been fused for a more concise representation.

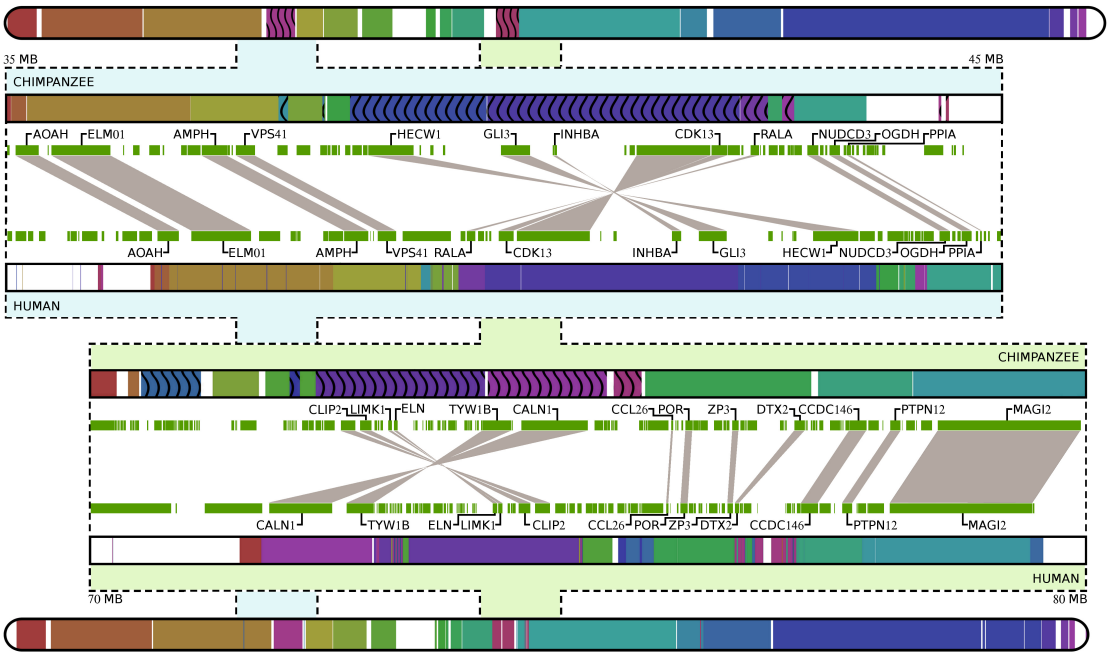


Figure 5.13: Progressive human and chimpanzee chromosome 7 information maps. For each chromosomes two regions have been extracted (35 MB to 45 MB and 70 MB to 80 MB). The progressive maps for these sub-regions show the genes involved in the paracentric inversions detected.

by our method with the exception of inversions in chromosomes 9 and 16 that are located in regions with limited available sequence information [129]. The structural rearrangements observed in the chimpanzee Y chromosome agree with previous reports [137], where variable copy number and position of Y-specific genes was found among chimpanzees (*Pan troglodytes*) but not among bonobo (*P. paniscus*), gorilla (*Gorilla gorilla gorilla* and *G. beringei graueri*) or orangutan (*Pongo pygmaeus* and *P. abelii*) lineages [210].

In addition, we identify inversions in chromosome 7 (Fig. 5.13) that were only partially described before [209]. Despite their importance, inversions are traditionally difficult to detect and new experimental approaches have been recently developed to improve the available tools [211]. These two inversions are located in 7p14.1 and 7q11.23 around the *GLI3* and *ELN* genes, respectively, and both are associated with human disorders. Namely, the Greig cephalopolysyndactyly syndrome is caused by mutations, deletion or rearrangements in the region containing the *GLI3* transcription factor that affect the development of the limbs, head and face, and is characterized by the presence of extra fingers or toes [212]. The Williams-Beuren syndrome (WBS) is a neurodevelopmental disease with distinctive facial and behavioral features, as well as several degrees of intellectual disability, caused by deletions of genes including *ELN* [213]. Curiously, inversion polymorphisms are present in a significant proportion of parents from WBS patients [214, 213], which is also observed in the 17q21.31 region [215], suggesting that structural variants enhance some microdeletion syndromes. Given the structural differences observed in these chromosomal regions, one might speculate that they have contributed to evolutionary innovation and the emergence of lineage-specific phenotypes.

Figure 5.14 depicts the complete information maps between human and orangutan. It

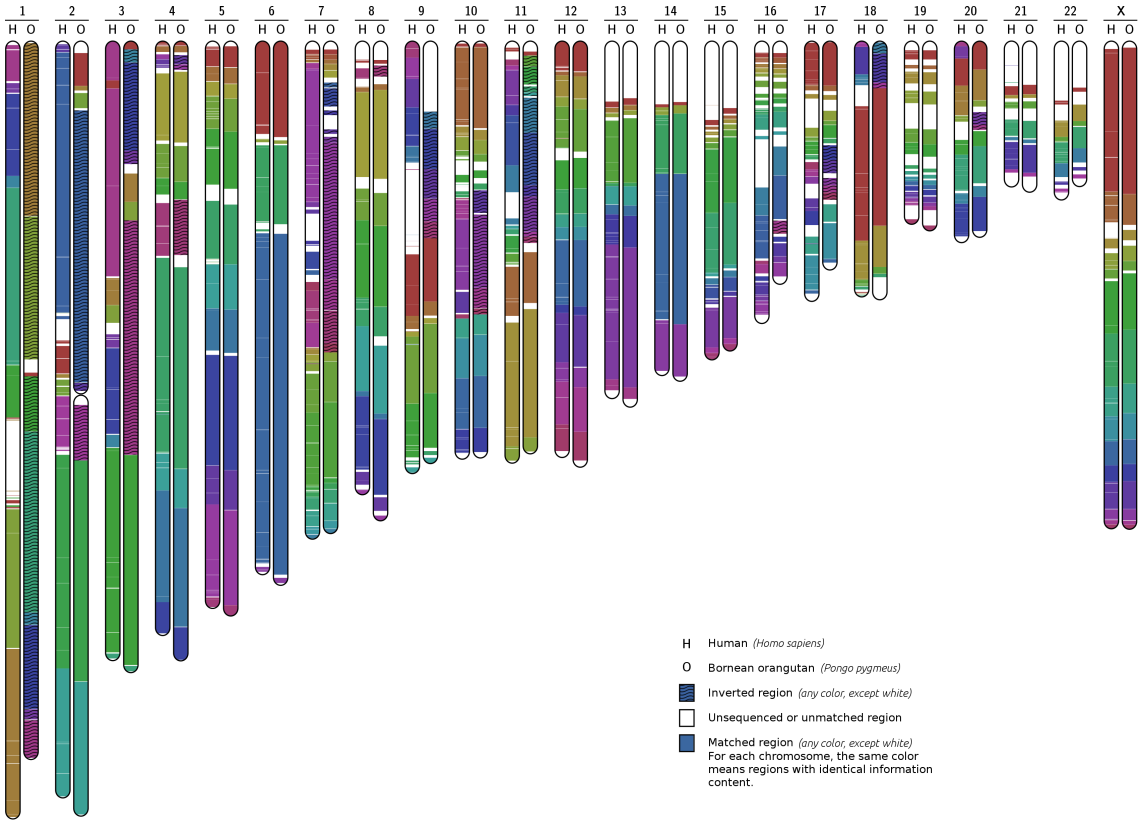


Figure 5.14: Human orangutan chromosomal map, obtained from chromosome pairwise comparison. Inversions are present in chromosomes 2, 3, 4, 7, 8, 9, 10, 11, 16, 17, 18 and 20. Chromosomes 2A and 2B of orangutan have been fused for a more concise representation.

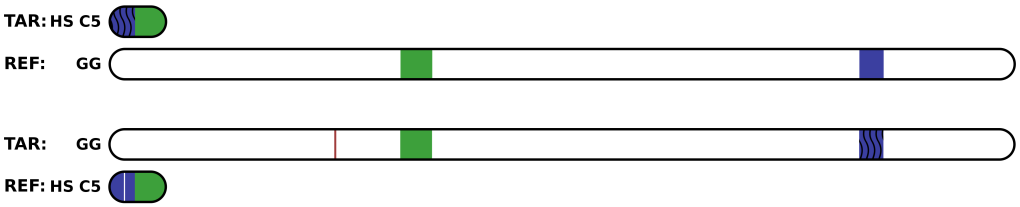


Figure 5.15: Detection of a translocation between the whole genome of gorilla, GG (all chromosomes concatenated), and human chromosome 5, HS C5. Smash was ran twice using the default parameters. In the first case, GG was used as the reference and HS C5 as the target (it required ≈ 22 GB of memory and ≈ 224 minutes), while in the second case HS C5 was used as reference and GG as the target (requiring ≈ 2 GB of memory and ≈ 115 minutes to run).

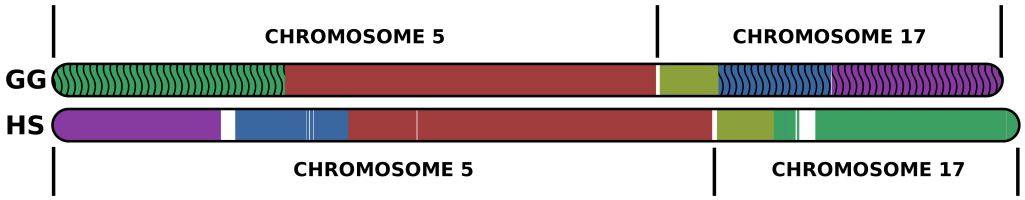


Figure 5.16: Detection of a translocation between gorilla and human chromosomes 5 and 17. The sequences have been concatenated (chromosome 5 and 17) in each species. In the middle of each concatenation we introduced one million of N symbols to facilitate the location of the concatenation breakpoint. Smash was ran using the default parameters.

---

shows that orangutan chromosome 1 is in the opposite direction as compared with human. Moreover, there are large inversions in chromosomes 2, 3, 4, 7, 8, 9, 10, 11, 16, 17, 18 and 20. Although there are fewer data available, the results are consistent with previous cytogenetic approaches that identified new rearrangements on the orangutan genome, specifically, a pericentric inversion on chromosome 1, complex rearrangements on chromosome 2 and a subtelomeric deletion on chromosome 19 [216]. Also, recent evidence suggests that the orangutan genome maintains the ancestral chromosomal state with observable differences in most chromosomes when compared with humans, including chromosomes 1, 2, 3, 7, 10, 11 and 18 [208].

The method and the implementation here described allows the detection of large-scale and small-scale genomic rearrangements, including balanced translocations and inversions that are not detected by array-CGH or chromosome alterations that are below the limits of microscopy, thus, extending the possibilities of genome-wide structure characterization with a single tool. In Figs. 5.15 and 5.16 we provide an example of a translocation between chromosomes 5 and 17 of human and gorilla. As it can be seen, after concatenating the sequences, Smash was able to detect a well known translocation that is one of the bases of gorilla speciation foundations [141].

Specifically, in Fig. 5.15 we present the detection results provided by Smash both for the case where the whole concatenated genome of the gorilla is used as reference (top) as well as when used as target (bottom). In the first case (top map), the ≈ 3 GB of reference required ≈ 22 GB of memory to run. We can clearly identify the region with homology to the human

chromosome splited in two major blocks, placed in the regions corresponding to the positions of chromosomes 5 and 17 of gorilla (the precise locations can be obtained from the files produced by Smash). The bottom map shows similar results, but at the cost of considerable less resources (just 2 GB of memory and 115 minutes of computing time), because in this case the shortest sequence played the role of reference sequence. Fig. 5.16 provides more detail regarding this translocation, because in this case only chromosomes 5 and 17 of both species were concatenated and hence considered.

Smash compares pairs of sequences. These pairs can be built using single chromosomes, as shown in Figs. 5.12 and 5.14, or sets of chromosomes concatenated in a single sequence, as in the example of the translocation shown in Figs. 5.15 and 5.16. In either case, Smash looks for and reports the position of regions that are similar, from the point of view of information content. Hence, in the examples provided in Figs. 5.12 and 5.14, only the regions that are similar in each pair of chromosomes are reported. To have a full view, it would be required either to run Smash in each possible pair of chromosomes (i.e., all possible pairs formed between the set of human chromosomes and the set of chimpanzee chromosomes, or by concatenating in a single sequence the whole genome of each species). Naturally, when very large sequences are involved (for example, entire genomes concatenated), the visualization granularity is reduced and the computational resources increase.

5.4 Conclusions

Chromosome rearrangements can drive adaptation and evolution of novel traits, but they can be deleterious as well. Here, we show that compression-based models are remarkably capable of detecting signatures of genomic chromosomal evolution, namely to determine how information flows between sequences of the same species and across species. The method is alignment-free and universal, in the sense that it can accept any input pair of genomic sequences, and depends only on two parameters.

A tool that implements the method has been made available for download. General guidelines have been given on how to select the values of its two parameters, which do not affect its performance in an overly sensitive way. Its advantages and limitations have been discussed.

The tool and the ideas that underlie its design may lead to new insights about important genomic questions, since it allows blind unsupervised detection of rearrangements and similarities between genomic sequences. An obvious example is the detection of evolutionary patterns across species, as demonstrated in the examples, but the tool has similar potential for diagnosis and genetic counselling. The detection of rearrangements in cancer genomes at high resolution levels is also considered important, in connection with risk stratification and personalized therapeutics.

"Sometimes say it best, specifically about something, when you say exactly nothing at all."

P. J. Green

6 Relative uniqueness

In the previous Chapter we have emphasized the regions that share relative information. In this Chapter, we address their complement, in other words, we focus on regions that occur in a given sequence but are absent from other or others. Therefore, our intention is to detect relative uniqueness. For the purpose we introduce the notion of relative absent words (RAWs).

RAWs are sub-sequences that do not occur in a given sequence (reference) and occur in another specific sequence (target). Consider a target sequence, x, and a reference sequence, y, both drawn from the finite alphabet Θ. We say that β is a factor of x if x can be expressed as $x = u\beta v$, with uv denoting the concatenation between sequences u and v. We denote by $\mathcal{W}_k(x)$ the set of all k-size words (or factors) of x. Also, we represent the set of all k-size words *not in* x as $\overline{\mathcal{W}_k(x)}$. For each word size k, we define the set of all words that exist in x but do not exist in y by

$$\mathcal{R}_k(x, y) = \mathcal{W}_k(x) \cap \overline{\mathcal{W}_k(y)}. \tag{6.1}$$

We define the subset of words that are minimal as

$$\mathcal{M}_k(x, y) = \{\beta \in \mathcal{R}_k(x, y) : \mathcal{W}_{k-1}(\beta) \cap \mathcal{M}_{k-1}(x, y) = \emptyset\}, \tag{6.2}$$

i.e., a minimal absent word of size k cannot contain any minimal absent word of size less than k. In particular, $l\beta r$ is a minimal absent word of sequence x, where l and r are single letters from Θ, if $l\beta r$ is not a word of x but both $l\beta$ and βr are (see [217] for more).

In order to unveil the presence of RAWs in a sequence x, we compute a binary sequence reporting their presence or absence along the sequence, using exclusively the model of y. As such, a relative uniqueness profile is given by

$$U(x_i \| y), \tag{6.3}$$

where U does not need to respect causality. Therefore, after loading the model from y and freezing we can access to any i.

Fig. 6.1 depicts the behavior, on changing the order of the model (k), of relative uniqueness and relative complexity (on synthetic data). Accordingly, we see that as the order of the model increases, its describing capabilities increase. Therefore, they start to behave as

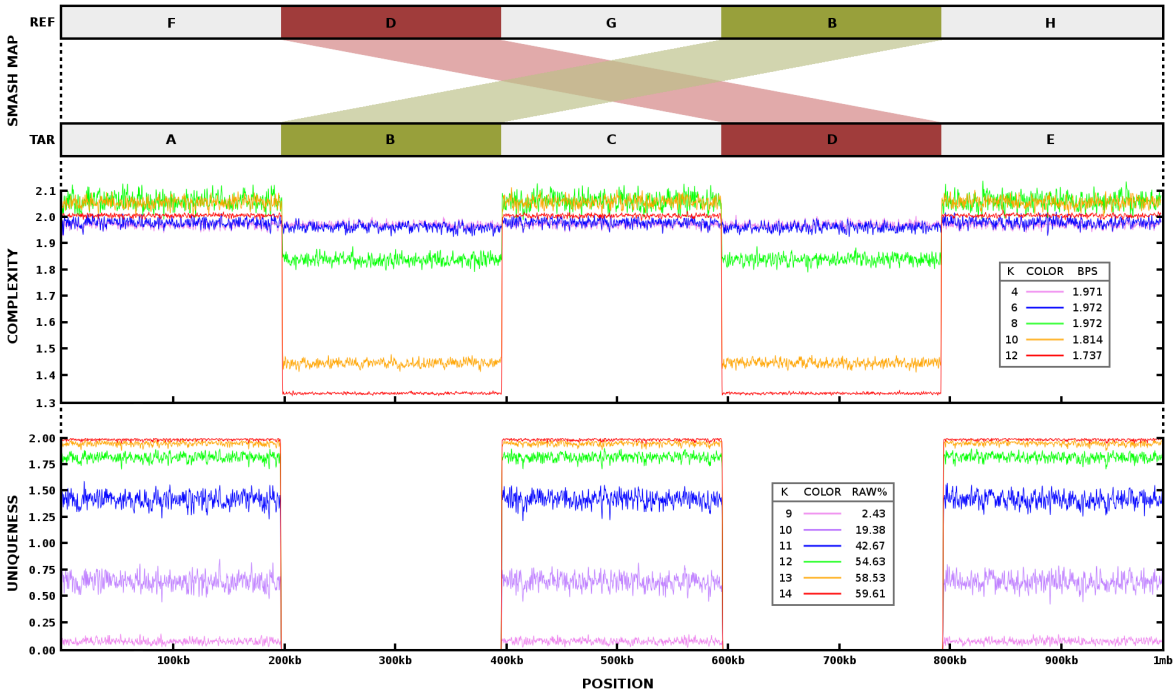


Figure 6.1: Visual perception of relative complexity and relative uniqueness $U(x_i\|y)$, given a reference (REF) and target (TAR) sequences. The relative complexity has been computed using a single FCM model where only the context-order (k) vary. The relative uniqueness has been computed using a binary model of k-order. The first panel depicts a smash map with the relations between reference and target sequence according to the procedure described in the previous chapter. The sequences have been pseudo-random simulated using Goose framework (`http://github.com/pratas/goose`) where the distribution of the symbols is according to A = T = 30% and C = G = 20%. The "BPS" stands for bits per symbol, while "RAW%" for the percentage of unique relative k-words.

complementary models. Nevertheless, the relative complexity model is based on counts, while the relative uniqueness profile is binary. Thus, if in a probabilistic model the counts of the relative complexity are uniform, both profiles will not behave exactly as complementary. In fact, the binary model can be seen as a simplification of a model based on counts.

Although minimal absent words have been studied before to describe properties of prokaryotic and eukaryotic genomes and to develop methods for phylogeny construction or PCR primer design [218, 219, 220, 217, 221], their practical usage as an entire set, for personalized medicine, is limited. According, one has to rely on relative comparisons, mainly because there is the need for differential identification of sequences that are derived from a pathogen genome but absent from its host. Moreover, they need to be minimal to maximize the probability, when concatenating the word to another word, of still being absent words in a presence of some alteration. Therefore, we are particularly interested in the non-empty set $\mathcal{M}_k(x, \bar{y})$ corresponding to the smallest k, referred as Minimal Relative Absent Words (MRAWs), for a personalized medicine application, while the interest in RAWs (not minimal) stands for relative uniqueness detection.

6.1 Personalized medicine application

For a personalized medicine application we use a large reference sequence, namely one corresponding to the human genome, while the target (or targets) are usually very small, namely those corresponding to a virus or bacteria. Since $y \gg x$, for detecting MRAWs the algorithm will spent most of its time loading each k-mer of y, for $k \in \{k_1, k_2, ..., k_n\}$.

Every k_j is computed according to (6.3), using a k-mer model that uses a numerical index between 0 and $|\Theta|^{k_j} - 1$. Each index is updated, with the information of presence or absence of each respective word, in a simple table array for $k_j \leq 16$, while for larger k_j in a hash table. Although there is the possibility to search for a $k_j > 16$, in practice the MRAWs are used for $k_j \leq 16$. The memory, Ω, required for k is given by

$$\Omega_k = \chi \sum_{j=k_1}^{k_n} |\Theta|^j, \tag{6.4}$$

where χ is the precision of the memory. Current implementation sets $\chi = 8$ bits, although it can be easily decreased to 1 bit. For a common search, having $k =\in \{11, 12, 13, 14\}$, it would be needed 340 MBytes. As it can be seen, $\Omega_k = 340$ MBytes regardless the size of y, in nowadays computers, are very low memory numbers.

On the other hand, the time resource is a demanding task in this situation, namely because there is the need to load each k_j from y (large sequence). Aware of this, we have created a method that uses parallel computing for loading each k-mer model. Therefore, for each k_j we compute it with a thread, T_j, having a speedup near 1.

After loading the reference, the models are kept frozen. Here it starts the detection of RAWs for each target sequence according to equation 6.1. In this phase there is no need for parallel computing since, for practical applications, the size of the target(s) are very small.

6.1.1 Software availability

The tool (EAGLE), written in C language, with the implementation of the method, supporting multi-threading, is available at `http://bioinformatics.ua.pt/software/eagle`, under GPL-2, and can be applied to any emerging pathogens or to show evidence of evolutionary patterns and signatures across species.

6.1.2 Ebola virus in human

Ebola virus (EBOV) is a negative strand-RNA virus from the *Filoviridae* family that causes high mortality hemorrhagic fevers, for which no vaccine or treatment currently exist [222]. There are five *Ebolavirus* species, namely,

- *Zaire*,
- *Sudan*,
- *Bundibugyo*,
- *Tai Forest*,
- *Reston*,

with the first (1976) and major (2014) outbreaks caused by the type species *Zaire ebolavirus* [223]. The numbers of the largest ever EBOV outbreak are worrying and continue escalating, with over 25500 cases and 10500 deaths (April 8, 2015) from the virus mainly in Guinea, Liberia, and Sierra Leone, according to the World Health Organization. The current outbreak is also the first where transmission has occurred outside Africa, with reported cases in Europe (Spain) and America (USA) [224]. Promising vaccine candidate tests are being rushed to face the epidemics and could be available within a few months [225]. These yet experimental therapies include, for example, recombinant viral vectors [226] or antibodies that target the viral glycoprotein (GP) [227, 222], but innovative approaches are still needed for the development of diagnosis tools and identification of druggable targets.

We used the current EBOV outbreak sequences, which were recently published [228], to discover and characterize the minimal relative absent words that are present in EBOV genomes but absent from the human genome. Moreover, we show that these words are also absent from the other *Ebolavirus* species and even from the genomes obtained from previous outbreaks. Thus, the sequences that we identify are species-specific and important for future development of diagnosis or therapeutic strategies for EBOV.

We have used the full GRC-38 human reference genome [129] downloaded from the NCBI, including the mitochondrial, unplaced and unlocalized sequences. The sequences of 99 EBOV genomes from the current outbreak in Sierra Leone [228] and additional 66 *Ebolavirus* genomes have been also downloaded from NCBI (see in `https://github.com/pratas/eagle/` scripts to download and process all the results. See [229] for additional sequence references).

Fig. 6.2 shows the computation for word sizes 12, 13, and 14. As expected, the number of absent words decreases as the k-mer size decreases. Specifically, for $k = 11$ (not represented), there are no EBOV RAWs. On the other hand, for $k = 12$, three groups of points emerge (RAW1, RAW2 and RAW3) representing the position of a relative absent word in each of the 99 unaligned viral genomes (Fig. 6.2-a). Alignments of 124 *Ebolavirus* sequences including additional EBOV genomes from the current outbreak in Guinea [223] and from previous outbreaks, show that the identified MRAWs fall into conserved protein regions (Fig. 6.2-b). However, several mutations can be found in the genomes that discriminate between the different species of *Ebolavirus* and even between EBOV sequences from the current and previous outbreaks (Fig. 6.2-c).

The identification of these viral genome signatures is important for quick diagnosis in outbreak scenarios. Additional analysis with all 165 Ebolavirus genomes confirmed these results (see Fig. 6.1.2). In particular, RAW1 is conserved within EBOV and can distinguish EBOV from other *Ebolavirus* species. RAW2 is conserved in all sequences from the West African 2014 outbreak in Guinea, Sierra Leone and Liberia, and only one nucleotide difference exists between these sequences and unrelated outbreak genomes. RAW3 is also conserved at the species level, excluding the four EBOV 1976/77 genomes, and can distinguish between all *Ebolavirus* species, as it can be seen in Fig. 6.1.2.

From the three EBOV sequence motifs absent in the human genome, the first (RAW1) is included in the virus nucleoprotein (NP), while the other two (RAW2 and RAW3) fall within the sequence of the viral RNA-polymerase (L-protein) (Fig. 6.2-c). Previous studies show that the N-terminal region of EBOV NP participates in both the formation of nucleocapsid-like structures through NP-NP interactions and in the replication of the viral genome [230], and RAW1 sequence (TTTCGCCCGACT) is part of this N-terminal region. The L-protein (LP) produces the viral transcripts to be translated by host ribosomes and is involved in the replication of the viral genome as well. The LP contains the two additional MRAWs, RAW2

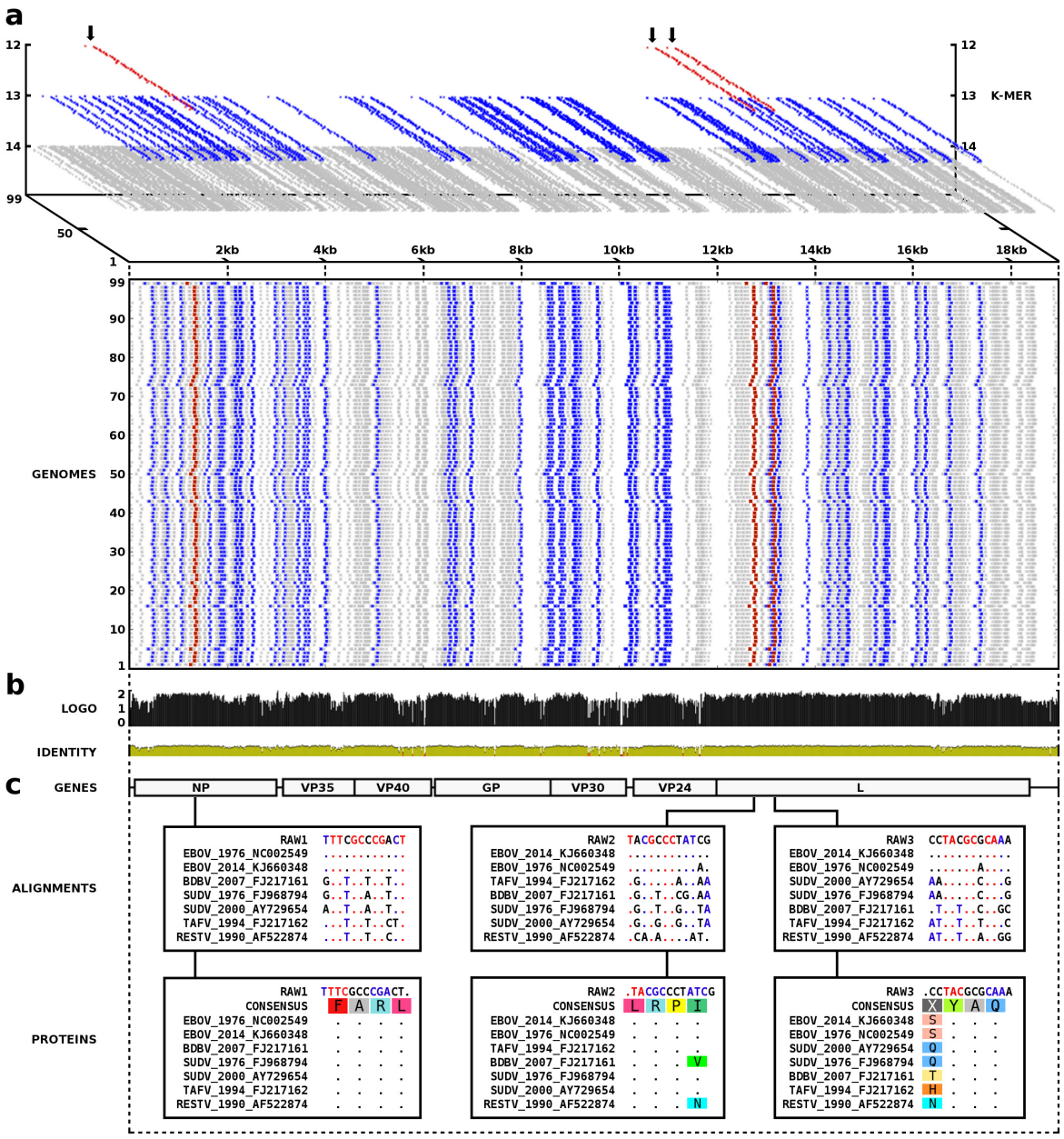


Figure 6.2: Ebola virus minimal absent words relatively to the human complete genome. (**a**) Relative absent words (RAWs) were identified in 99 unaligned genomes from the current outbreak in Sierra Leone (2014) and are highlighted in red ($k = 12$), blue ($k = 13$) and gray ($k = 14$). (**b**) Whole genome alignments from 124 published *Ebolavirus* genomes were obtained from [228] and visualized in Geneious (created by Biomatters, available from `http://www.geneious.com`). Sequence logos and identity define conserved regions. (**c**) Regions corresponding to the identified RAWs are shown in genome location and both as nucleic acid and protein alignments. The *Ebolavirus* reference genomes are displayed, as well as selected representative sequences where nucleotide differences are observed.

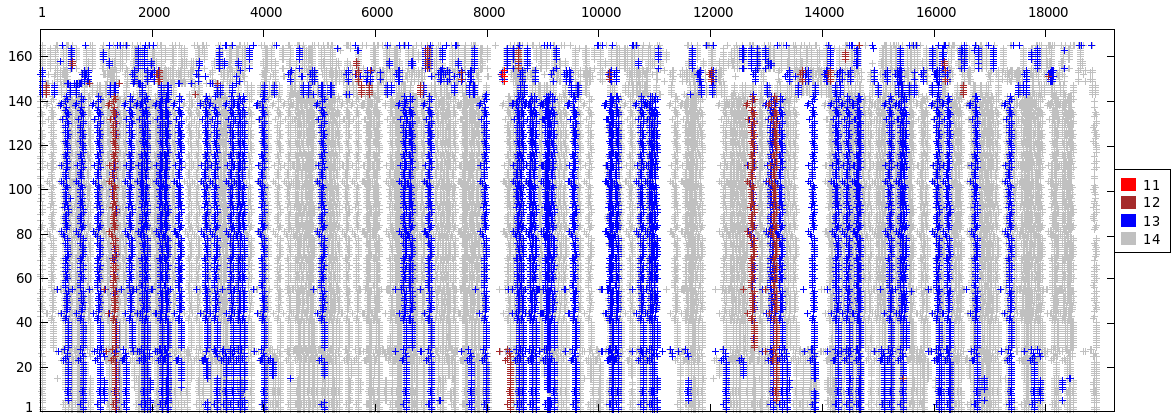


Figure 6.3: Identification of relative absent words in 165 *Ebolavirus* genomes with the human genome as reference. RAW sequences are shown in red (k=11), dark red (k=12), blue (k=13) and gray (k=14). 11-mer RAWs are exclusive of *Reston ebolavirus*. 1-24, *Zaire ebolavirus* (EBOV) genomes from previous outbreaks; 25-28, EBOV genomes from the 2014 DRC (Democratic Republic of the Congo) unrelated outbreak; 29-142, EBOV genomes from the West African 2014 (current) outbreak; 143-147, *Bundibugyo ebolavirus* (BDBV) genomes; 148-154, *Reston ebolavirus* (RESTV) genomes; 155-164, *Sudan ebolavirus* (SUDV) genomes; and 165, *Tai Forest ebolavirus* (TAFV) genome.

(TACGCCCTATCG) and RAW3 (CCTACGCGCAAA). Both NP and LP are critical for the virus life cycle and constitute good targets for therapeutic intervention.

Screening for new antiviral compounds could benefit from knowledge of their protein structures. For EBOV, most protein structures are unknown except for the C-terminal domain of NP, GP, VP24 and VP35 [231], thus, we have predicted the structure of the N-terminal regions of the EBOV NP and LP by homology modeling (Fig. 6.4, 6.5 and 6.6). These structural models show that the amino acids corresponding to the RAW1 motif are enclosed within the structure, while RAW2 and RAW3 are exposed at the protein surface, which can justify its higher degree of conservation.

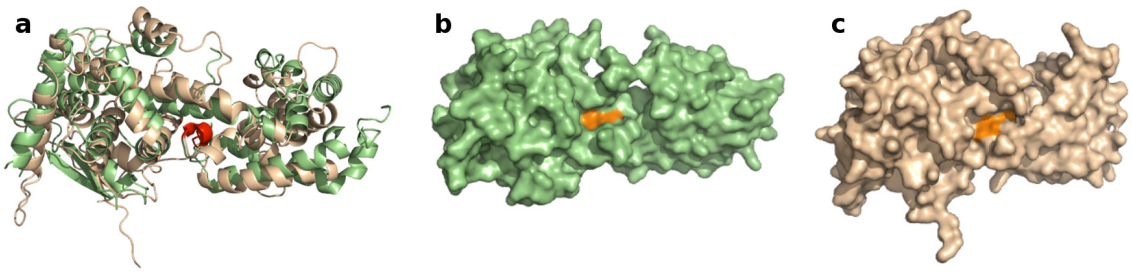


Figure 6.4: Structure of the N-terminal region from the Ebola virus Nucleoprotein (residues 1-380). **a**) Superposition of the model (wheat color) and corresponding template structure (green). **b** and **c**) Surface view of template and model respectively. Amino acid residues corresponding to the RAW sequence are highlighted.

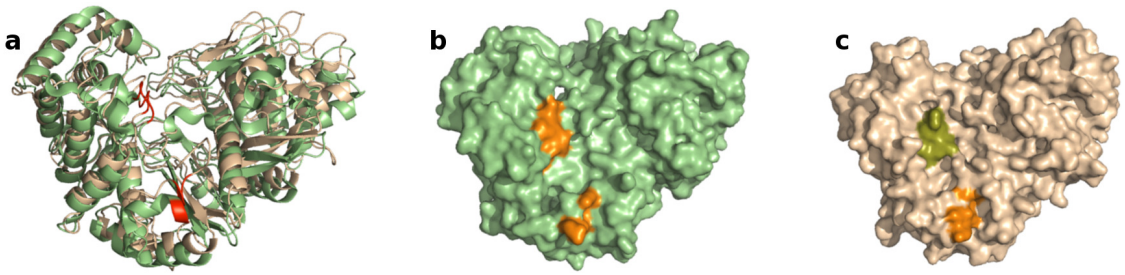


Figure 6.5: Structure of the N-terminal region from the Ebola virus RNA-polymerase (residues 177-805). **a**) Superposition of the model (wheat color) and corresponding template structure (green). **b** and **c**) Surface view of template and model respectively. Amino acid residues corresponding to the RAW sequence are highlighted.

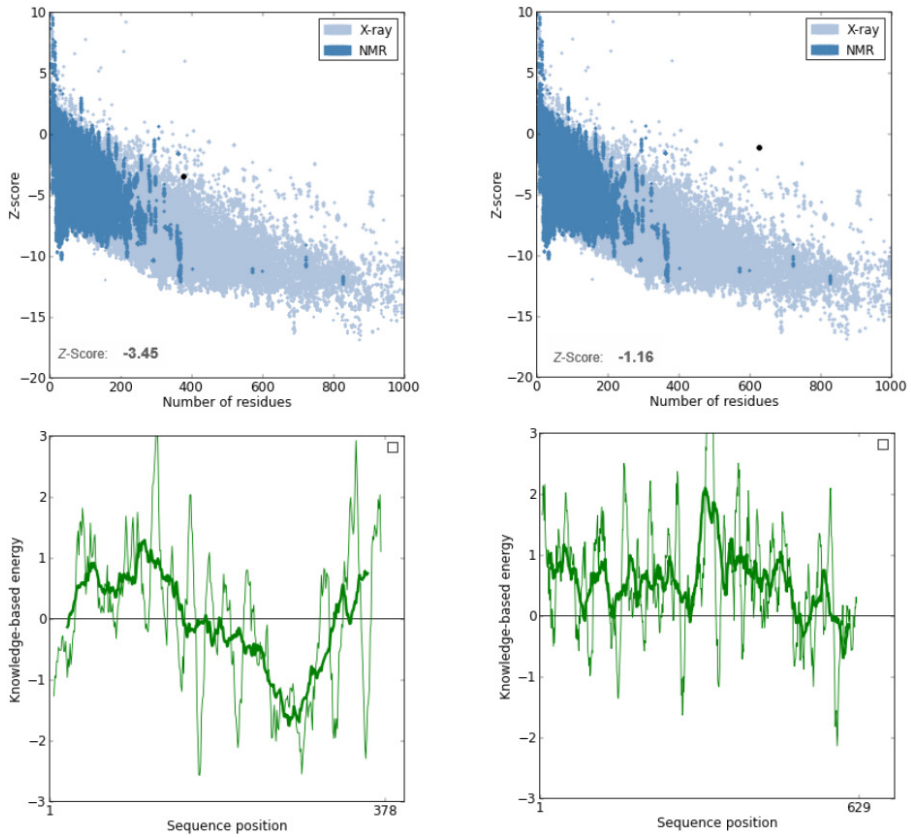


Figure 6.6: Evaluation of the model for Ebola virus Nucleoprotein (NP) and RNA-polymerase (LP). The quality of the predicted 3D structures for Ebola virus NP (left) and LP (right), overall (top) and locally (bottom) was estimated in ProSA-web. The models were obtained with MODELLER using the Nipah virus nucleoprotein (PDB ID:4CO6) and the BVDV (bovine viral diarrhea virus) RNA polymerase (PDB ID:1S48) as templates, respectively.

The personalized medicine field is now closer to clinical practice with the advances of next-generation sequencing technologies. Personalized therapeutics are a possibility and their development is essential with the emergence of resistance to current available drugs. Additionally, quick diagnosis is required for emerging pathogens and in epidemics such as the current Ebola outbreak. Here, we have detected MRAWs in the human genome that are present in EBOV genomes, and identified nucleotide differences in some of these sequences that can distinguish between *Ebolavirus* species and outbreaks. Also, we show that the corresponding amino acid sequences are conserved within EBOV. These results can now be further explored for diagnosis and therapeutics, sometimes mentioned as theranostics [232]. Namely, RAW nucleotide sequences can be used in diagnosis to design primers that identify *Ebolavirus* infections or distinguish between *Ebolavirus* species. For PCR-based methods, longer sequences and multiplex reactions can be developed to avoid primer binding bias. Additional nucleotide or protein-based strategies for therapeutics can be envisaged, as discussed below.

One problem in developing efficient EBOV treatments is the virus ability to evade the immune system. The viral glycoprotein (GP) is a major target because it mediates attachment and entry into the host cells. However, in addition to the surface envelope protein, the GP gene also produces fragment, soluble glycoproteins that are secreted and direct the immune system to produce antibodies for variable and non-essential regions of the virus [233, 234].

As current efforts based on the viral GP might prove ineffective, additional targets should be sought. Our results show that the viral nucleoprotein (NP) and polymerase (LP) can be attractive targets. As the amino acid sequences of all three 12-mer RAWs are conserved within EBOV, these regions can be used to screen for small molecule inhibitors.

In particular, RAW1 is conserved in all *Ebolavirus* NP proteins, which can indicate a functional or structural role. And, considering that the protein model predicts that RAW2 and RAW3 are relatively close in the 3D structure and in exposed domains, these regions can be used to develop novel antibodies. Also, a recently described mechanism shows that the polymerase (LP) from Ebola and Marburg viruses is capable of editing transcripts, resulting in increased variability in the produced proteins, and that the most edited mRNAs are the Ebola GP and Marburg nucleoprotein (NP) and LP itself [235]. Thus, the use of combined therapies towards multiple proteins can be more effective, as suggested by studies to develop vaccines for Lassa virus that target both NP and GP [236, 237].

RNA-based strategies such as RNA interference (RNAi) or antissense therapies are also promising approaches to silence target specific gene expression. The RAW sequences that we have identified can be used to develop RNAi or antissense probes that bind viral transcripts and prevent their translation, thus, inhibiting viral replication without blocking the host mRNAs. Translation of these technologies into clinical applications have been slowed by challenges in the delivery of small RNAs into cells, but recent developments in delivery systems are bridging the bench to bedside gap [238, 239]. Among these, gold or lipid nanoparticles [240, 241] were shown to be effective against cancer and viral infections, including EBOV [242]. Gold-nanobeacons can be applied as a combined diagnosis and therapy tool for effective testing, including in low-cost settings [243] and, with this purpose, advances in peptide nucleic acid (PNA) probes for viral detection are also taking place [244, 44]. Whichever the technology, the identification of genome signatures for rapid evolving species such as Ebola viruses will be useful for the development of both diagnosis and therapeutics.

6.2 Unique regions detection

The identification of regions that are present in a species but absent in other can be used also to detect novelty and signatures across intra- and inter-species. Moreover, in a presence of aligned sequences from one side of the sequences (targets) we are able to visualise where they occur since we rely on uniqueness profiles (6.3).

According to the previous section, we are interested in creating a model of a reference to detect sub-sequences that are present in a target, in order to detect novel regions relatively to a reference, or better, we are interested to find RAWs. Unlike the previous section, we only use a k-mer model. Besides, this model has higher depth (typically $k = 30$).

If we used a binary vector to store in memory (RAM) all the entries, and only using (in computation) 1 bit to say if a k-mer exists or not, we would need 4^k bits. Using a regular $k = 30$, we would need 131072 Terabytes of memory. This is impracticable on current computers.

Considering to use a hash table for such a model would become a feasible task. However, the memory would become dependent on the number of inserted elements, because a hash table increases as the number of new elements inserted increases. Moreover, for the volume of data in this case would become a hard task, although feasible.

A third option is a probabilistic data structure, namely a Bloom filter [245], trading space resources by precision. Notwithstanding, the usage of a very large Bloom filter (with the number of hash functions optimized), can give very high probabilities of becoming very similar to deterministic. Nevertheless, for this case, we do not need very large lengths and precise results, since we want to find regions (RAWs) and not mRAWs. This seems the most efficient choice.

For using a Bloom filter, we set a vector of dimension m and the number of hash functions h, obtaining a balance that is also related with the number of elements that are filtered n. Asymptotically, for a given m and n, the value of the number of hash functions that minimizes the false positive probability is

$$h = \frac{m}{n} \ln 2, \tag{6.5}$$

that can be seen as

$$2^{-h} \approx 0.6185^{m/n}. \tag{6.6}$$

The more elements that are added to the set, the larger the probability of false positives. The required number of bits m, given n and a desired false positive probability p, assuming that the optimal value of h is used, can be computed by substituting the optimal value of h in

$$p = \left(1 - e^{-(m/n \ln 2)n/m}\right)^{(m/n \ln 2)} \tag{6.7}$$

that can be seen as

$$\ln p = -\frac{m}{n} \left(\ln 2\right)^2, \tag{6.8}$$

and finally

$$m = -\frac{n \ln p}{(\ln 2)^2}. \tag{6.9}$$

This means that asymptotically, for a given false positive probability p, the length of a Bloom filter m is proportional to the number of elements being filtered n.

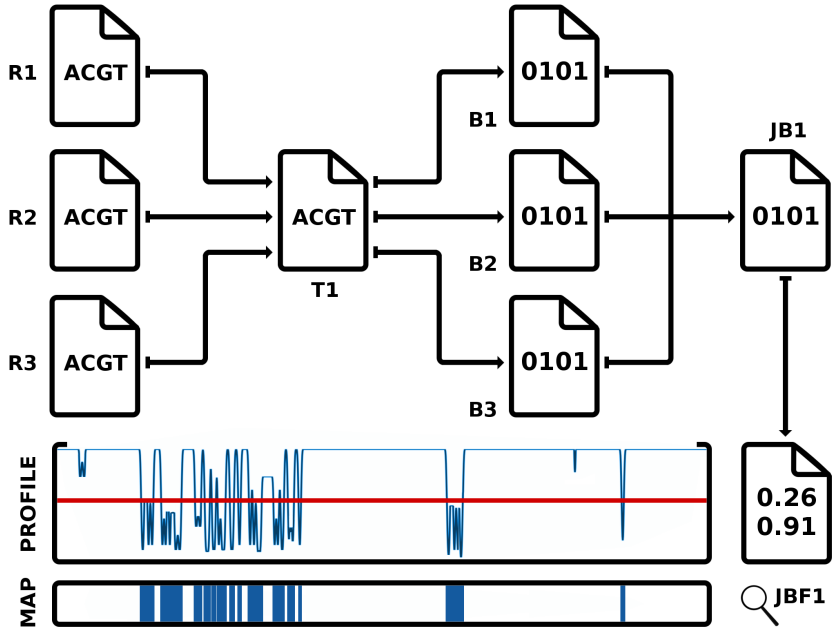


Figure 6.7: Visual description of the method. The genomic sequences contained in the files $R1, R2$ and $R3$ are independently processed against a target, $T1$. From each computation is generated a binary sequence, $B1, B2$ and $B3$, describing the presence/absence of a RAW according to the order of $T1$. After, the binary files are computed using a logic Or ($\vee$), $B1 \vee B2 \vee B3$ and the result is $JB1$. The $JB1$ sequence is then (low-pass) filtered resulting in the real sequence described as $JBF1$. Finally, a threshold (line in red) is used to segment the information contained in the $JBF1$, where each segmented region is represented in the RAWs map.

For finite values, the false positive probability for a finite Bloom filter with m bits, n elements, and h hash functions is at most

$$\left(1 - e^{-h(n+0.5)/(m-1)}\right)^h, \tag{6.10}$$

having a penalty for at most half an extra element and at most one fewer bit. For more information and details see [246].

This method allows whole genome analysis using T_n targets and R_n references. To solve this we write to disk each RAW detected from R_i according to each T_i. After, for each T_i the RAWs are only considered if they exist in all R_i. A file containing the whole genome RAWs according to each T_i is stored (these are the unique regions).

An example of the method, from sequences to maps, using 3 reference sequences and 1 target is depicted in Fig. 6.7. For T_n targets, the process is repeated recursively n times. Moreover, when using inverted repeats, the reverse complement sequence is also loaded into memory (for the same reference model).

For visualizing the unique regions, after a low-pass filtering of a binary sequence containing the presence/absence of RAWs, a threshold is used to segment these regions and then they are computed in a visual map (see Fig. 6.7 for an example).

In Fig. 6.8 we have ran the tool (CHESTER) against several synthetic sequences, that

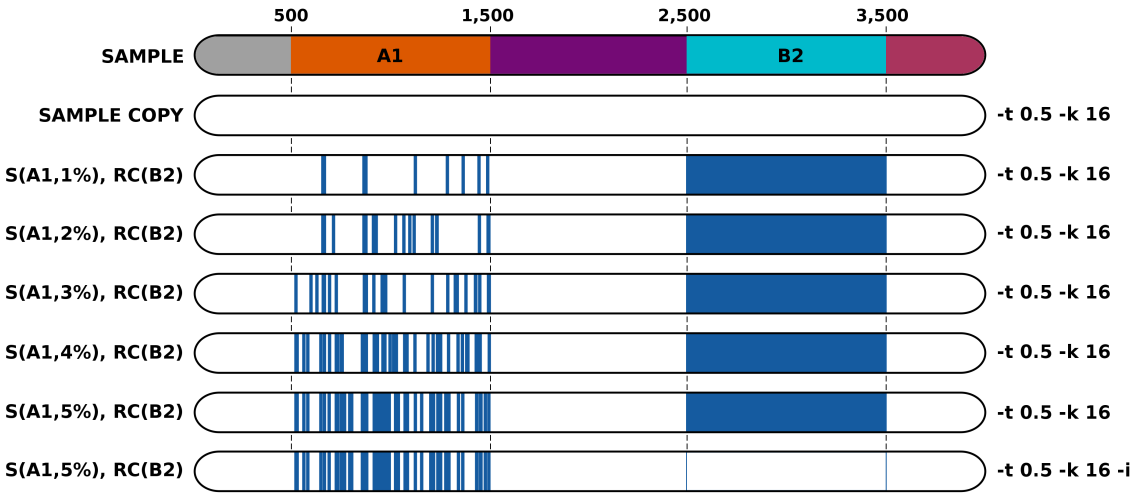


Figure 6.8: Running CHESTER using several ground truth sequences. Blocks $A1$ and $B2$ have been edited according to the functions referred on the left. Function S stands for a substitution mutation of the input block with the defined percentage. Function RC applies the reverse complement of the input. CHESTER running parameters are defined on the right, using a threshold of 0.5 and a k-mer size of 16, while only the bottom map has been run using inversions. The blue color on the computed maps represents the unique regions according to the RAWs.

have been suited manipulated to better understand the tool. According, the method identifies as novel the regions that are mutated (in $A1$) and also the inversions ($B2$). When the tool ran with the "-i" parameter, and therefore prepared to handle inversions, these where successfully not reported. When dealing with sequenced data, the sequences might have several inverted regions due to errors of assemblage or sequencing. As we have shown in Fig. 6.8, this method is prepared to overcome those limitations.

6.2.1 Software availability

The tool (CHESTER), written in C language, with the implementation of the method is available at `http://github.com/pratas/chester`, under GPL-2, and can be applied to any genomic sequences, supporting FASTA, FASTQ and SEQ (ACGTN) format, which can be used to find and visualise unique regions and signatures.

6.2.2 Unique human regions relatively to other primates

For the experiments we have used: the reference human genome (GRC-38) [129], the reference chimpanzee genome (2.1.4), the reference gorilla genome (3.1) and the reference orangutan genome (2.0.2). The sequences have been downloaded from the NCBI. The sequences of gorilla and orangutan representing the chromosome Y have not been yet sequenced and therefore are not present in the study. On the other hand, we have also included the unlocalized, unplaced and mitochondrial sequences in order to bypass most assembly challenges.

We ran CHESTER on those sequences obtaining the map in Fig. 6.9. According, the larger blue areas identify the centromeres, namely corresponding to very repetitive DNA.

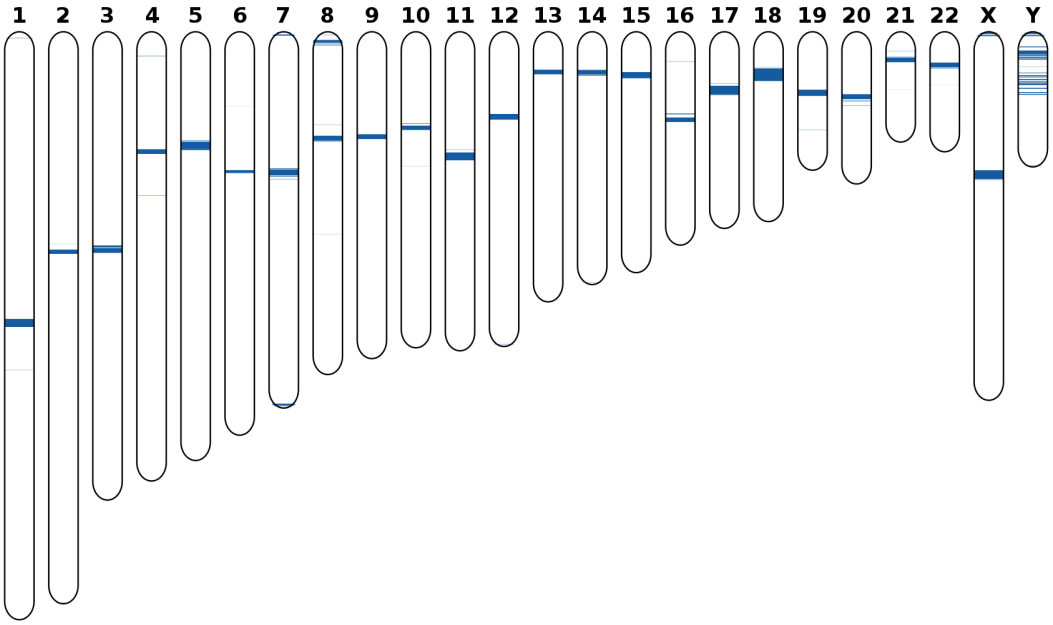


Figure 6.9: Human unique region maps according to chimpanzee, gorilla and orangutan using CHESTER with $t = 0.6$ and $k = 30$. Chromosome Y of gorilla and orangutan were not present, by lack of sequencing data, and therefore Y map is only according to chimpanzee.

On the other hand, the smaller areas contain several genes and pseudogenes (genes that are not expressed [247]) associated for instance to immunology, blood, smell and brain. Besides there are several identified motifs in these regions (on the NCBI and Ensemble), although considered of importance their nature has not yet been understood.

From those sub-sequences which are reasonably understood, we highlight HCP5 HLA complex P5, MAFK, GALNT9, OR11H12, OR11H11, SHOX short stature homebox.

For example, on human chromosome 14 the OR11H12 olfactory receptor is a gene associated with olfactory receptors that interact with odorant molecules in the nose, to initiate a neuronal response that triggers the perception of a smell. These findings are confirmed with other recent studies that show the loss of olfactory function only in the hominid evolution and therefore the consequent genomic sequence alteration [248].

Genetic and/or genomic human relative uniqueness can, perhaps, be seen as a product of genome interactions with environment, behavior and culture. These systems seem ultimately linked with the irreversible process of learning [249]. As any thermodynamic semi-isolated system can only, asymptotically, increase their complexity, and therefore, showing remarkably distinct unique features along time, specially when are characterized by higher learning accelerations.

6.3 Conclusions

Relative uniqueness identifies regions present in a sequence that are absent from a sequence or several sequences. The definition of relative uniqueness is specifically related with the definition of relative absent words. Their fundamentals have been introduced along with two applications: one for personalized medicine, namely using the *Ebolavirus*, and other for whole

genome analysis. The first one, identified three minimal sequences found in *Ebolavirus* that are absent from human DNA. Moreover, these three sub-sequences proved to be sufficient to classify different *Ebolavirus* sub-species. The second, was used to visualize where each human chromosome is unique relatively to chimpanzee, gorilla and orangutan. The spoted regions seem to be related with important genes, where several are already documented while others seem to be new identifications.

7

Conclusions and future work

Genomic sequences are large codified messages describing most of the structure of all known living organisms. Since the presentation of the first genomic sequence, a huge amount of genomics data have been generated, with diversified characteristics, rendering the data deluge phenomenon a serious problem in most genomics centers. As such, most of the data are discarded (when possible), while other are compressed using general purpose algorithms, often attaining modest data reduction results.

To face this problem we have introduced a new compressor based on an algorithmic entropy filter, applying a preprocessing analysis technique, leading to substantial improvements on the savings in memory resources, particularly in the decompression process. Besides, it yields tremendous improvements in the compression ratio, specially in highly repetitive datasets, such as in genomic collections. In future work, we might develop different scalable levels of algorithmic entropy.

However, it is mostly limited (by performance) to collections of similar sequences. Therefore, we have proposed a universal (multiple purpose) genomic sequence compressor. We have used a mixture of two classes, reference and target models, that we explore with finite-context models (FCMs) or eXtended FCMs (XFCMs). The XFCMs have been introduced as new error tolerant high-order FCMs. Together with memory representability using cache-hashes they ensure flexibility given hardware specifications. The results show very good adaptability of the compressor to multiple types and characteristics of genomic sequences.

Ultimately, the compressor is a program that tries to learn and approximate the objects nature. Therefore, it is directly related with the complexity of an object or between objects and, hence, we use the compressors learning and describing capabilities and measure complexity on genomic sequences. For this purpose, we have proposed a way to compute the Normalized Information Distance (NID), without using the conjoint information, but rather the conditional information. This simplistic computation requires the definition of a conditional compressor, according to the universal genomic compressor that we have developed. As an application, we have measured the distance between genomic sequences, mostly chromosomes, within the same species and between different species, reporting several insights of evolution already known, but also several undocumented.

Moreover, we have introduced a way to quantify relative information, namely through the normalized relative compression (NRC), also requiring a specific compressor. The NRC computation is much simpler, using less resources (time and memory), and has the possibility to be computed using several parallel forms and, in some cases, it can be accessed without order (as we have shown in the local measures). We have measured the NRC within 45 RNA bird species and in chromosomes of several primates, being able to confirm most of the NCCD results, but also spotting a high correlation between mitochondrial DNA and chromosome 5. The NRC gave proves to be a very important matter to explore in future work, namely a suitable manipulation to turn it into a distance (for example, by maxims or sums). Moreover, we believe that it can be related with a temporal notion, such as a relative logical depth.

We have explored local measures that cumulatively make the respective global measures, presenting several definitions and applications. Mostly, the applications give the ability "to look at a DNA sequence" and immediately identify specific regions, namely motif, centromere, telomere, homologous genes, among others. As a specific automatic unsupervised tool we have explored it in chromosome rearrangements. The tool and the ideas that underlie its design may lead to new insights about important genomic questions, since it allows blind unsupervised detection of rearrangements and similarities between genomic sequences. An example is the detection of evolutionary patterns across species, as demonstrated in the examples. Mostly, the tool led to the study and unveiling of important characteristics that may have happened in the past. Actually, we are able to look and understand events from million years ago. But the tool has similar potential for diagnosis and genetic counseling. The detection of rearrangements in cancer genomes at high resolution levels is also considered important, in connection with risk stratification and personalized therapeutics.

Finally, as a complement, we have introduced and explored relative uniqueness. Relative uniqueness identifies regions present in a sequence that are absent from a sequence or several sequences. The definition of relative uniqueness is specifically related to the definition of relative absent words (RAWs). We introduced their fundamentals along with two applications: one for personalized medicine, namely using the *Ebolavirus*, and other for whole genome analysis. In the first one, we identified three minimal sequences found in *Ebolavirus* that are absent from human DNA. These three sub-sequences proved to be enough to classify different *Ebolavirus* sub-species. However, the method is not limited to the *Ebolavirus*, it can be used in any hostage/pathogen. The second, was used to visualize where each human chromosome is unique relatively to several primates. The spotted regions seem to be related with important genes, specifying the roots of human evolution.

During this thesis we have presented several blind unsupervised methods to compute relative complexity or uniqueness. These methods are fully automatic. Besides, they are parallelizable, generally faster than the existing ones (when they exist), universal (in the sense of genomic input) and with better describing capabilities. Therefore, after the development of these methods we will assemble a fully automatic ANI ("Artificial Narrow Intelligence") being able to search for most characteristics that might exist within or between data. This will require the continuous development of faster and better approximation tools, but mostly, biologists to analyze the existent facts. We believe that the findings will increase faster than the human capability to understand them.

Bibliography

[1] H. Nyquist. Certain factors affecting telegraph speed. *Trans. of the American Institute of Electrical Engineers*, XLIII:412–422, January 1924.

[2] H. Nyquist. Certain topics in telegraph transmission theory. *Trans. of the American Institute of Electrical Engineers*, 47(2):617–644, April 1928.

[3] R. V. L. Hartley. Transmission of information. *Bell System Technical Journal*, 7(3):535–563, 1928.

[4] P. J. S. G. Ferreira and R. Higgins. The establishment of sampling as a scientific principle—a striking case of multiple discovery. *Notices of the AMS*, 58(10):1446–1450, 2011.

[5] A. Turing. On computable numbers, with an application to the Entscheidungsproblem. *Proceedings of the London Mathematical Society*, 42(2):230–265, 1936.

[6] C. E. Shannon. A mathematical theory of communication. *Bell System Technical Journal*, 27:379–423,623–656, 1948.

[7] A. B. Pippard. *Elements of classical thermodynamics: for advanced students of physics*. Cambridge University Press, 1964.

[8] J. C. Maxwell. *Theory of heat*. Courier Corporation, 2012.

[9] C. H. Bennett. Demons, engines and the second law. *Scientific American*, 257(5):108–116, 1987.

[10] A. N. Kolmogorov. Three approaches to the quantitative definition of information. *Problems of Information Transmission*, 1(1):1–7, 1965.

[11] K. Gödel. Über formal unentscheidbare Sätze der Principia Mathematica und verwandter Systeme I. *Monatshefte für mathematik und physik*, 38(1):173–198, 1931.

[12] G. J. Chaitin. On the length of programs for computing finite binary sequences. *Journal of the ACM*, 13:547–569, 1966.

[13] R. J. Solomonoff. A formal theory of inductive inference. Part I. *Information and Control*, 7(1):1–22, March 1964.

[14] R. J. Solomonoff. A formal theory of inductive inference. Part II. *Information and Control*, 7(2):224–254, June 1964.

[15] C. H. Bennett, P. Gács, M. Li P. M. B. Vitányi, and W. H. Zurek. Information distance. *IEEE Trans. on Information Theory*, 44(4):1407–1423, July 1998.

[16] C. H. Bennett and R. Landauer. The fundamental physical limits of computation. *Scientific American*, 253(1):48–56, 1985.

[17] W. Heisenberg. Über den anschaulichen Inhalt der quantentheoretischen Kinematik und Mechanik. *Zeitschrift für Physik*, 43(3-4):172–198, 1927.

[18] A. Einstein, B. Podolsky, and N. Rosen. Can quantum-mechanical description of physical reality be considered complete? *Physical Review*, 47(10):777, 1935.

[19] R. Cilibrasi and P. M. B. Vitányi. Clustering by compression. *IEEE Trans. on Information Theory*, 51(4):1523–1545, April 2005.

[20] Coulson A.R. Sanger F., Nicklen S. DNA sequencing with chain-terminating inhibitors. *Natl Acad Sci*, 74(12):5463–5467, 1977.

[21] B. Berger, J. Peng, and M. Singh. Computational solutions for omics data. *Nature Reviews Genetics*, 14:333–346, May 2013.

[22] C. Darwin. On the origin of species by means of natural selection, or the preservation of favoured races in the struggle for life, 1859. Available at `http://darwin-online.org.uk/`.

[23] Carl R Woese, Otto Kandler, and Mark L Wheelis. Towards a natural system of organisms: proposal for the domains archaea, bacteria, and eucarya. *Proceedings of the National Academy of Sciences*, 87(12):4576–4579, 1990.

[24] Sam Griffiths-Jones, Russell J Grocock, Stijn Van Dongen, Alex Bateman, and Anton J Enright. mirbase: microrna sequences, targets and gene nomenclature. *Nucleic acids research*, 34(suppl 1):D140–D144, 2006.

[25] Calvin B Harley, A Bruce Futcher, and Carol W Greider. Telomeres shorten during ageing of human fibroblasts. *Nature*, 345(6274):458–460, 1990.

[26] Michael L. Pace. Bacterial mortality and the fate of bacterial production. *Hydrobiologia*, 159(1):41–49, 1988.

[27] David M Raup and J John Sepkoski Jr. Mass extinctions in the marine fossil record. *Science*, 215(4539):1501–1503, 1982.

[28] S Blair Hedges. The origin and evolution of model organisms. *Nature Reviews Genetics*, 3(11):838–849, 2002.

[29] Laura Wegener Parfrey, Jessica Grant, Yonas I. Tekle, Erica Lasek-Nesselquist, Hilary G. Morrison, Mitchell L. Sogin, David J. Patterson, and Laura A. Katz. Broadly sampled multigene analyses yield a well-resolved eukaryotic tree of life. *Systematic Biology*, 59(5):518–533, 2010.

[30] János Podani, Zoltán N Oltvai, Hawoong Jeong, Bálint Tombor, A-L Barabási, and E Szathmary. Comparable system-level organization of archaea and eukaryotes. *Nature genetics*, 29(1):54–56, 2001.

[31] Dongying Wu, Philip Hugenholtz, Konstantinos Mavromatis, Rüdiger Pukall, Eileen Dalin, Natalia N Ivanova, Victor Kunin, Lynne Goodwin, Martin Wu, Brian J Tindall, et al. A phylogeny-driven genomic encyclopaedia of bacteria and archaea. *Nature*, 462(7276):1056–1060, 2009.

[32] Eugene V Koonin, Tatiana G Senkevich, and Valerian V Dolja. The ancient virus world and evolution of cells. *Biol Direct*, 1(1):29, 2006.

[33] Nicolas Delaroque and Wilhelm Boland. The genome of the brown alga ectocarpus siliculosus contains a series of viral dna pieces, suggesting an ancient association with large dsdna viruses. *BMC evolutionary biology*, 8(1):110, 2008.

[34] Florian Maumus, Aline Epert, Fabien Nogué, and Guillaume Blanc. Plant genomes enclose footprints of past infections by giant virus relatives. *Nature communications*, 5, 2014.

[35] Jonathan Filée. Multiple occurrences of giant virus core genes acquired by eukaryotic genomes: The visible part of the iceberg? *Virology*, 466:53–59, 2014.

[36] Philippe Colson, Xavier De Lamballerie, Natalya Yutin, Sassan Asgari, Yves Bigot, Dennis K Bideshi, Xiao-Wen Cheng, Brian A Federici, James L Van Etten, Eugene V Koonin, et al. megavirales, a proposed new order for eukaryotic nucleocytoplasmic large dna viruses. *Archives of virology*, 158(12):2517–2521, 2013.

[37] Patrick Forterre, Mart Krupovic, and David Prangishvili. Cellular domains and viral lineages. *Trends in Microbiology*, 22(10):554–558, 2014.

[38] Elizabeth Pennisi. Ever-bigger viruses shake tree of life. *Science*, 341(6143):226–227, 2013.

[39] Carlos Canchaya, Ghislain Fournous, Sandra Chibani-Chennoufi, Marie-Lise Dillmann, and Harald Brüssow. Phage as agents of lateral gene transfer. *Current opinion in microbiology*, 6(4):417–424, 2003.

[40] Felisa Wolfe-Simon, Jodi Switzer Blum, Thomas R. Kulp, Gwyneth W. Gordon, Shelley E. Hoeft, Jennifer Pett-Ridge, John F. Stolz, Samuel M. Webb, Peter K. Weber, Paul C. W. Davies, Ariel D. Anbar, and Ronald S. Oremland. A bacterium that can grow by using arsenic instead of phosphorus. *Science*, 332(6034):1163–1166, 2011.

[41] Tobias J. Erb, Patrick Kiefer, Bodo Hattendorf, Detlef Gnther, and Julia A. Vorholt. Gfaj-1 is an arsenate-resistant, phosphate-dependent organism. *Science*, 337(6093):467–470, 2012.

[42] J. D. Watson and F. H. C. Crick. A structure for deoxyribose nucleic acid. *Nature*, 171:737–738, 1953.

[43] E. Chargaff. Chemical specificity of nucleic acids and mechanism of their enzymatic degradation. *Experientia*, 6(6):201–209, 1950.

[44] Shang-Hong Zhang and Ya-Zhi Huang. Limited contribution of stem-loop potential to symmetry of single-stranded genomic DNA. *Bioinformatics*, 26(4):478–485, 2010.

[45] David Mitchell and Robert Bridge. A test of Chargaff's second rule. *Biochemical and Biophysical Research Communications*, 340:90–94, 2006.

[46] G. E. Zentner and S. Henikoff. Regulation of nucleosome dynamics by histone modifications. *Nature structural & molecular biology*, 20(3):259–266, 2013.

[47] N. D. Hastie, M. Dempster, M. G. Dunlop, A. M. Thompson, D. K. Green, and R. C. Allshire. Telomere reduction in human colorectal carcinoma and with ageing. *Nature*, 346(6287):866–868, 1990.

[48] M. Ridley. *Genome: The Autobiography of a Species in 23 Chapters*. HarperCollins, 2013.

[49] L. Rowen, G. Mahairas, and L. Hood. Sequencing the human genome. *Science*, 278:605–607, October 1997.

[50] J. Tomkins. How genomes are sequenced and why it matters: Implications for studies in comparative genomics of humans and chimpanzees. *Answers Research Journal*, 4:81–88, 2011.

[51] K. Sayood. *Introduction to data compression*. Morgan Kaufmann, 4th edition, 2012.

[52] D. Salomon. *Data compression - The complete reference*. Springer, 3nd edition, 2004.

[53] David Huffman. A method for the construction of minimum-redundancy codes. *Proceedings of the IRE*, 40(9):1098–1101, September 1952.

[54] S. Golomb. Run-length encodings. *IEEE Trans. on Information Theory*, 12(3):399–401, July 1966.

[55] M. Burrows and D. J. Wheeler. *A block-sorting lossless data compression algorithm*. Digital Systems Research Center, May 1994.

[56] J. Ziv and A. Lempel. A universal algorithm for sequential data compression. *IEEE Trans. on Information Theory*, 23:337–343, 1977.

[57] J. Rissanen and G. G. Langdon, Jr. Arithmetic coding. *IBM J. Res. Develop.*, 23(2):149–162, March 1979.

[58] J. Rissanen and G. G. Langdon, Jr. Universal modeling and coding. *IEEE Trans. on Information Theory*, 27(1):12–23, January 1981.

[59] T. C. Bell, J. G. Cleary, and I. H. Witten. *Text compression*. Prentice Hall, 1990.

[60] I. H. Witten, R. M. Neal, and J. G. Cleary. Arithmetic coding for data compression. *Communications of the ACM*, 30(6):520–540, June 1987.

[61] I. H. Witten and T. C. Bell. The zero-frequency problem: estimating the probabilities of novel events in adaptive text compression. *IEEE Trans. on Information Theory*, 37(4):1085–1094, July 1991.

[62] A. Moffat, R. M. Neal, and I. H. Witten. Arithmetic coding revisited. *ACM Trans. Inf. Syst.*, 16(3):256–294, 1998.

[63] J. G. Cleary and I. H. Witten. Data compression using adaptive coding and partial string matching. *IEEE Trans. on Communications*, 32(4):396–402, April 1984.

[64] M. V. Mahoney. Adaptive weighing of context models for lossless data compression. Technical Report CS-2005-16, Florida Institute of Technology CS Dept., Melbourne, FL, 2005.

[65] D. Salomon. *Data compression - The complete reference*. Springer, 4th edition, 2007.

[66] K. Sayood. *Introduction to data compression*. Morgan Kaufmann, 3rd edition, 2006.

[67] G. Street. *Introduction to bioinformatics*. Athens Auckland Bangkok Bogot Buenos Aires, and Karachi Kolkata Kuala Lumpur Madrid Melbourne, 2002.

[68] F. Lillo, S. Basile, and R. Mantegna. Comparative genomics study of inverted repeats in bacteria. *Bioinformatics*, 18(7):971–979, 2002.

[69] S. Grumbach and F. Tahi. Compression of DNA sequences. In *Proc. of the Data Compression Conf., DCC-93*, pages 340–350, Snowbird, Utah, 1993.

[70] S. Grumbach and F. Tahi. A new challenge for compression algorithms: genetic sequences. *Information Processing & Management*, 30(6):875–886, 1994.

[71] E. Rivals, J.-P. Delahaye, M. Dauchet, and O. Delgrange. A guaranteed compression scheme for repetitive DNA sequences. In *Proc. of the Data Compression Conf., DCC-96*, page 453, Snowbird, Utah, 1996.

[72] D. Loewenstern and P. N. Yianilos. Significantly lower entropy estimates for natural DNA sequences. In *Proc. of the Data Compression Conf., DCC-97*, pages 151–160, Snowbird, Utah, March 1997.

[73] X. Chen, S. Kwong, and M. Li. A compression algorithm for DNA sequences and its applications in genome comparison. In K. Asai, S. Miyano, and T. Takagi, editors, *Genome Informatics 1999: Proc. of the 10th Workshop*, pages 51–61, Tokyo, Japan, 1999.

[74] T. Matsumoto, K. Sadakane, and H. Imai. Biological sequence compression algorithms. In A. K. Dunker, A. Konagaya, S. Miyano, and T. Takagi, editors, *Genome Informatics 2000: Proc. of the 11th Workshop*, pages 43–52, Tokyo, Japan, 2000.

[75] X. Chen, S. Kwong, and M. Li. A compression algorithm for DNA sequences. *IEEE Engineering in Medicine and Biology Magazine*, 20:61–66, 2001.

[76] X. Chen, M. Li, B. Ma, and J. Tromp. DNACompress: fast and effective DNA sequence compression. *Bioinformatics*, 18(12):1696–1698, 2002.

[77] I. Tabus, G. Korodi, and J. Rissanen. DNA sequence compression using the normalized maximum likelihood model for discrete regression. In *Proc. of the Data Compression Conf., DCC-2003*, pages 253–262, Snowbird, Utah, 2003.

[78] G. Manzini and M. Rastero. A simple and fast DNA compressor. *Software—Practice and Experience*, 34:1397–1411, 2004.

[79] G. Korodi and I. Tabus. An efficient normalized maximum likelihood algorithm for DNA sequence compression. *ACM Trans. on Information Systems*, 23(1):3–34, January 2005.

[80] B. Behzadi and F. Le Fessant. DNA compression challenge revisited. In *Combinatorial Pattern Matching: Proc. of CPM-2005*, volume 3537 of *LNCS*, pages 190–200, Jeju Island, Korea, June 2005. Springer-Verlag.

[81] G. Korodi and I. Tabus. Normalized maximum likelihood model of order-1 for the compression of DNA sequences. In *Proc. of the Data Compression Conf., DCC-2007*, pages 33–42, Snowbird, Utah, March 2007.

[82] M. D. Cao, T. I. Dix, L. Allison, and C. Mears. A simple statistical algorithm for biological sequence compression. In *Proc. of the Data Compression Conf., DCC-2007*, pages 43–52, Snowbird, Utah, March 2007.

[83] Z. Zhu, J. Zhou, Z. Ji, and Y. Shi. DNA sequence compression using adaptive particle swarm optimization-based memetic algorithm. *IEEE Trans. on Evolutionary Computation*, 15(5):643–658, 2011.

[84] A. J. Pinho, D. Pratas, and P. J. S. G. Ferreira. Bacteria DNA sequence compression using a mixture of finite-context models. In *Proc. of the IEEE Workshop on Statistical Signal Processing*, Nice, France, June 2011.

[85] A. J. Pinho, P. J. S. G. Ferreira, A. J. R. Neves, and C. A. C. Bastos. On the representability of complete genomes by multiple competing finite-context (Markov) models. *PLoS ONE*, 6(6):e21588, 2011.

[86] T. Bose, M. H. Mohammed, A. Dutta, and S. S. Mande. BIND–an algorithm for lossless compression of nucleotide sequence data. *Journal of Biosciences*, 37(4):785–789, 2012.

[87] W. Dai, H. Xiong, X. Jiang, and L. Ohno-Machado. An adaptive difference distribution-based coding with hierarchical tree structure for DNA sequence compression. In *Proc. of the Data Compression Conf., DCC-2013*, pages 371–380. IEEE, 2013.

[88] Pinghao Li, Shuang Wang, Jihoon Kim, Hongkai Xiong, Lucila Ohno-Machado, and Xiaoqian Jiang. DNA-COMPACT: DNA compression based on a pattern-aware contextual modeling technique. *PLoS ONE*, 8(11):e80377, 2013.

[89] S. Wandelt, M. Bux, and U. Leser. Trends in genome compression. *Current Bioinformatics*, 2013.

[90] S. Deorowicz and S. Grabowski. Data compression for sequencing data. *Algorithms for Molecular Biology*, 8(1):25, 2013.

[91] D. Pratas and A. J. Pinho. Compressing the human genome using exclusively Markov models. In *Advances in Intelligent and Soft Computing, Proc. of the 5th Int. Conf. on Practical Applications of Computational Biology & Bioinformatics, PACBB 2011*, volume 93, pages 213–220, April 2011.

[92] Diogo Pratas and Armando J. Pinho. M6: a method for compressing complete genomes using markov models. In *Doctoral Symposium in Informatics Engineering*, page 6, 2012.

[93] E. S. Lander. Initial impact of the sequencing of the human genome. *Nature*, 470:187–197, 2011.

[94] S. Christley, Y. Lu, C. Li, and X. Xie. Human genomes as email attachments. *Bioinformatics*, 25(2):274–275, 2009.

[95] M. C. Brandon, D. C. Wallace, and P. Baldi. Data structures and compression algorithms for genomic sequence data. *Bioinformatics*, 25(14):1731–1738, 2009.

[96] C. Wang and D. Zhang. A novel compression tool for efficient storage of genome resequencing data. *Nucleic Acids Research*, 39(7):e45, 2011.

[97] S. Kuruppu, S. J. Puglisi, and J. Zobel. Optimized relative Lempel-Ziv compression of genomes. In *Proc. of the 34th Australian Computer Science Conference, ACSC-2011*, volume 11, pages 91–98, 2011.

[98] W. Tembe, J. Lowey, and E. Suh. G-SQZ: compact encoding of genomic sequence and quality data. *Bioinformatics*, 26(17):2192–2194, 2010.

[99] S. Deorowicz and S. Grabowski. Compression of DNA sequence reads in FASTQ format. *Bioinformatics*, 27(6):860–862, 2011.

[100] M. H.-Y. Fritz, R. Leinonen, G. Cochrane, and E. Birney. Efficient storage of high throughput DNA sequencing data using reference-based compression. *Genome Research*, 21:734–740, 2011.

[101] C. Kozanitis, C. Saunders, S. Kruglyak, V. Bafna, and G. Varghese. Compressing genomic sequence fragments using SlimGene. *Journal of Computational Biology*, 18(3):401–413, 2011.

[102] A. J. Pinho, D. Pratas, and S. P. Garcia. GReEn: a tool for efficient compression of genome resequencing data. *Nucleic Acids Research*, 40(4):e27, 2012.

[103] S. Wandelt and U. Leser. FRESCO: referential compression of highly similar sequences. *IEEE/ACM Trans. on Computational Biology and Bioinformatics*, 10(5):1275–1288, 2013.

[104] S. Deorowicz, A. Danek, and M. Niemiec. GDC 2: Compression of large collections of genomes. *Scientific Reports*, 5(11565):1–12, 2015.

[105] I. Ochoa, M. Hernaez, and T. Weissman. iDoComp: a compression scheme for assembled genomes. *Bioinformatics*, page btu698, 2014.

[106] A. J. Pinho, A. J. R. Neves, and P. J. S. G. Ferreira. Inverted-repeats-aware finite-context models for DNA coding. In *Proc. of the 16th European Signal Processing Conf., EUSIPCO-2008*, Lausanne, Switzerland, August 2008.

[107] J. L. Carter and M. N. Wegman. Universal classes of hash functions. *Journal of Computer and System Sciences*, 18(2):143–154, 1979.

[108] D. Pratas, A. J. Pinho, and J. M. O. S. Rodrigues. XS: a FASTQ read simulator. *BMC Research Notes*, 7(1):40, 2014.

[109] Erich D Jarvis, Siavash Mirarab, Andre J Aberer, Bo Li, Peter Houde, Cai Li, Simon YW Ho, Brant C Faircloth, Benoit Nabholz, Jason T Howard, et al. Whole-genome analyses resolve early branches in the tree of life of modern birds. *Science*, 346(6215):1320–1331, 2014.

[110] J. K. Bonfield and M. V. Mahoney. Compression of FASTQ and SAM format sequencing data. *PLoS ONE*, 8(3):e59190, March 2013.

[111] M. H. Mohammed, A. Dutta, T. Bose, S. Chadaram, and Sharmila S. Mande. DELIMINATE - a fast and efficient method for loss-less compression of genomic sequences. *Bioinformatics*, 28(19):2527–2529, 2012.

[112] A. J. Pinho and D. Pratas. MFCompress: a compression tool for fasta and multi-fasta data. *Bioinformatics*, October 2013.

[113] S. Grabowski, S. Deorowicz, and Ł. Roguski. Disk-based compression of data from genome sequencing. *Bioinformatics*, 31(9):1389–1395, 2015.

[114] C. S. Wallace and D. M. Boulton. An information measure for classification. *The Computer Journal*, 11(2):185–194, August 1968.

[115] J. Rissanen. Modeling by shortest data description. *Automatica*, 14:465–471, 1978.

[116] M. Li and P. Vitányi. *An introduction to Kolmogorov complexity and its applications*. Springer Science & Business Media, 2013.

[117] D. Hammer, A. Romashchenko, A. Shen, and N. Vereshchagin. Inequalities for Shannon entropy and Kolmogorov complexity. *Journal of Computer and System Sciences*, 60(2):442–464, 2000.

[118] Andrei N Kolmogorov. Logical basis for information theory and probability theory. *Information Theory, IEEE Transactions on*, 14(5):662–664, 1968.

[119] M. Li, X. Chen, X. Li, B. Ma, and P. M. B. Vitányi. The similarity metric. *IEEE Trans. on Information Theory*, 50(12):3250–3264, December 2004.

[120] M. Cebrián, M. Alfonseca, and A. Ortega. Common pitfalls using the normalized compression distance: what to watch out for in a compressor. *Communications in Information and Systems*, 5(4):367–384, 2005.

[121] S. Kullback and R. Leibler. On information and sufficiency. *Annals of Mathematical Statistics*, 22, 1951.

[122] R. Hamming. Error detecting and error correcting codes. *Bell System Technical Journal*, 29(2):147–160, 1950.

[123] V. Levenshtein. Binary codes capable of correcting deletions, insertions and reversals. In *Soviet physics doklady*, volume 10, page 707, 1966.

[124] J. Ziv and N. Merhav. A measure of relative entropy between individual sequences with application to universal classification. *IEEE Trans. on Information Theory*, 39(4):1270–1279, July 1993.

[125] Sven Helmer, Nikolaus Augsten, and Michael Böhlen. Measuring structural similarity of semistructured data based on information-theoretic approaches. *The VLDB JournalThe International Journal on Very Large Data Bases*, 21(5):677–702, 2012.

[126] D. Cerra, M. Datcu, and P. Reinartz. Authorship analysis based on data compression. *Pattern Recognition Letters*, 42:79–84, 2014.

[127] D. P. Coutinho and M. Figueiredo. Text classification using compression-based dissimilarity measures. *Int. Journal of Pattern Recognition and Artificial Intelligence*, 29(5), 2015.

[128] N. Nikvand and Z. Wang. Generic image similarity based on Kolmogorov complexity. In *Proc. of the IEEE Int. Conf. on Image Processing, ICIP-2010*, pages 309–312, Hong Kong, September 2010.

[129] D. Church, M. Deanna, V. Schneider, et al. Modernizing reference genome assemblies. *PLoS Biology*, 9(7):e1001091, 2011.

[130] M. Cebrián, M. Alfonseca, and A. Ortega. The normalized compression distance is resistant to noise. *IEEE Trans. on Information Theory*, 53(5):1895–1900, 2007.

[131] D. Pratas, A. J. Pinho, and S. Garcia. Exon: a web-based software toolkit for DNA sequence analysis. In *6th Int. Conf. on Practical Applications of Computational Biology & Bioinformatics*, pages 217–224. Springer, 2012.

[132] Alix Boc, Vladimir Makarenkov, et al. T-rex: a web server for inferring, validating and visualizing phylogenetic trees and networks. *Nucleic Acids Research*, 40(W1):W573–W579, 2012.

[133] François Chevenet, Christine Brun, Anne-Laure Bañuls, Bernard Jacq, and Richard Christen. Treedyn: towards dynamic graphics and annotations for analyses of trees. *BMC bioinformatics*, 7(1):439, 2006.

[134] Asif T Chinwalla, Lisa L Cook, Kimberly D Delehaunty, Ginger A Fewell, Lucinda A Fulton, Robert S Fulton, Tina A Graves, LaDeana W Hillier, Elaine R Mardis, John D McPherson, et al. Initial sequencing and comparative analysis of the mouse genome. *Nature*, 420(6915):520–562, 2002.

[135] Sharon Anderson, Alan T Bankier, Bart G Barrell, MHL De Bruijn, Alan R Coulson, Jacques Drouin, IC Eperon, DP Nierlich, Bruce A Roe, Frederick Sanger, et al. Sequence and organization of the human mitochondrial genome. *Nature*, 290:457–465, 1981.

[136] C. Ramsdell et al. Comparative genome mapping of the deer mouse (Peromyscus maniculatus) reveals greater similarity to rat (Rattus norvegicus) than to the lab mouse (Mus musculus). *BMC Evolutionary Biology*, 8(1):65, 2008.

[137] J. Hughes et al. Chimpanzee and human Y chromosomes are remarkably divergent in structure and gene content. *Nature*, 463(7280):536–539, 2010.

[138] Thomas Henry Huxley. *Evidence as to mans place in nature by Thomas Henry Huxley*. Williams and Norgate, 1863.

[139] Charles Darwin. *The descent ofman, andselection in relation to sex*. London: Murray, 1871.

[140] J. Ijdo, A. Baldini, D. Ward, S. Reeders, and R. Wells. Origin of human chromosome 2: an ancestral telomere-telomere fusion. *Proceedings of the National Academy of Sciences USA*, 88(20):9051–9055, 1991.

[141] R. Samonte and E. Eichler. Segmental duplications and the evolution of the primate genome. *Nature Reviews Genetics*, 3(1):65–72, 2002.

[142] Anna Jauch, Johannes Wienberg, Roscoe Stanyon, N Arnold, S Tofanelli, T Ishida, and Thomas Cremer. Reconstruction of genomic rearrangements in great apes and gibbons by chromosome painting. *Proceedings of the National Academy of Sciences*, 89(18):8611–8615, 1992.

[143] M Shibusawa, M Nishibori, Chizuko Nishida-Umehara, Masaoki Tsudzuki, Julio Masabanda, Darren K Griffin, and Yoichi Matsuda. Karyotypic evolution in the galliformes: an examination of the process of karyotypic evolution by comparison of the molecular cytogenetic findings with the molecular phylogeny. *Cytogenetic and genome research*, 106(1):111–119, 2004.

[144] Yang Zhang, Xiaojun Zhang, Thomas H O'Hare, William S Payne, Jennifer J Dong, Chantel F Scheuring, Meiping Zhang, James J Huang, Mi-Kyung Lee, Mary E Delany, et al. A comparative physical map reveals the pattern of chromosomal evolution between the turkey (meleagris gallopavo) and chicken (gallus gallus) genomes. *BMC genomics*, 12(1):447, 2011.

[145] The Chimpanzee Sequencing, Analysis Consortium, et al. Initial sequence of the chimpanzee genome and comparison with the human genome. *Nature*, 437(7055):69–87, 2005.

[146] Aylwyn Scally, Julien Y Dutheil, LaDeana W Hillier, Gregory E Jordan, Ian Goodhead, Javier Herrero, Asger Hobolth, Tuuli Lappalainen, Thomas Mailund, Tomas Marques-Bonet, et al. Insights into hominid evolution from the gorilla genome sequence. *Nature*, 483(7388):169–175, 2012.

[147] Devin P Locke, LaDeana W Hillier, Wesley C Warren, Kim C Worley, Lynne V Nazareth, Donna M Muzny, Shiaw-Pyng Yang, Zhengyuan Wang, Asif T Chinwalla, Pat Minx, et al. Comparative and demographic analysis of orang-utan genomes. *Nature*, 469(7331):529–533, 2011.

[148] D. Locke, R. Segraves, L. Carbone, N. Archidiacono, D. Albertson, D. Pinkel, and E. Eichler. Large-scale variation among human and great ape genomes determined by array comparative genomic hybridization. *Genome Research*, 13(3):347–357, 2003.

[149] Marta Farré and Aurora Ruiz-Herrera. Role of chromosomal reorganisations in the human-chimpanzee speciation. *eLS*, 2014.

[150] S. A. Terwijn, L. Torenvliet, and P. M. B. Vitányi. Nonapproximability of the normalized information distance. *Journal of Computer and System Sciences*, 77:738–742, 2011.

[151] I. Berg, D. Bosnacki, and P. Hilbers. Large scale analysis of small repeats via mining of the human genome. In *20th Int. Workshop on Database and Expert Systems Application, DEXA'09*, pages 198–202, September 2009.

[152] Junko Tsuji, Martin C Frith, Kentaro Tomii, and Paul Horton. Mammalian numt insertion is non-random. *Nucleic acids research*, 40(18):9073–9088, 2012.

[153] Pak Chung Wong, Han-Wei Shen, Christopher R. Jonhson, Chaomei Chen, and Robert B. Ross. The top 10 challenges in extreme-scale visual analytics. *IEEE Computer Graphics and Applications*, 32(4):63–67, 2012.

[154] T. D. Schneider and R. M. Stephens. Sequence logos: a new way to display consensus sequences. *Nucleic Acids Research*, 18(20):6097–6100, 1990.

[155] H. J. Jeffrey. Chaos game representation of gene structure. *Nucleic Acids Research*, 18(8):2163–2170, 1990.

[156] L. J. Jensen, C. Friis, and D. W. Ussery. Three views of microbial genomes. *Research in Microbiology*, 150:773–777, 1999.

[157] A. G. Pedersen, L. J. Jensen, S. Brunak, H.-H. Staerfeldt, and D. W. Ussery. A DNA structural atlas for *Escherichia coli*. *Journal of Molecular Biology*, 299:907–930, 2000.

[158] N. Goldman. Nucleotide, dinucleotide and trinucleotide frequencies explain patterns observed in chaos game representations of DNA sequences. *Nucleic Acids Research*, 21(10):2487–2491, 1993.

[159] P. J. Deschavanne, A. Giron, J. Vilain, G. Fagot, and B. Fertil. Genomic signature: characterization and classification of species assessed by chaos game representation of sequences. *Molecular Biology and Evolution*, 16(10):1391–1399, 1999.

[160] B. Fertil, M. Massin, S. Lespinats, C. Devic, P. Dumee, and A. Giron. GENSTYLE: exploration and analysis of DNA sequences with genomic signature. *Nucleic Acids Research*, 33:W512–W515, 2005.

[161] J. L. Oliver, P. Bernaola-Galván, J. Guerrero-García, and R. Román-Roldán. Entropic profiles of DNA sequences through chaos-game-derived images. *Journal of Theoretical Biology*, 160:457–470, 1993.

[162] S. Vinga and J. S. Almeida. Local Renyi entropic profiles of DNA sequences. *BMC Bioinformatics*, 8(393), 2007.

[163] M. Crochemore and R. Vérin. Zones of low entropy in genomic sequences. *Computers & Chemistry*, pages 275–282, 1999.

[164] O. G. Troyanskaya, O. Arbell, Y. Koren, G. M. Landau, and A. Bolshoy. Sequence complexity profiles of prokaryotic genomic sequences: a fast algorithm for calculating linguistic complexity. *Bioinformatics*, 18(5):679–688, 2002.

[165] B. Clift, D. Haussler, R. McConnell, T. D. Schneider, and G. D Stormo. Sequence landscapes. *Nucleic Acids Research*, 14(1):141–158, 1986.

[166] L. Allison, L. Stern, T. Edgoose, and T. I. Dix. Sequence complexity for biological sequence analysis. *Computers & Chemistry*, 24:43–55, 2000.

[167] L. Stern, L. Allison, R. L. Coppel, and T. I. Dix. Discovering patterns in *Plasmodium falciparum* genomic DNA. *Molecular & Biochemical Parasitology*, 118:174–186, 2001.

[168] T. I. Dix, D. R. Powell, L. Allison, J. Bernal, S. Jaeger, and L. Stern. Comparative analysis of long DNA sequences by per element information content using different contexts. *BMC Bioinformatics*, 8(Suppl. 2):S10, 2007.

[169] V. D. Gusev, L. A. Nemytikova, and N. A. Chuzhanova. On the complexity measures of genetic sequences. *Bioinformatics*, 15(12):994–999, 1999.

[170] F. Nan and D. Adjeroh. On the complexity measures for biological sequences. In *Proc. of the IEEE Computational Systems Bioinformatics Conference, CSB-2004*, Stanford, CA, August 2004.

[171] L. Pirhaji, M. Kargar, A. Sheari, H. Poormohammadi, M. Sadeghi, H. Pezeshk, and C. Eslahchi. The performances of the chi-square test and complexity measures for signal recognition in biological sequences. *Journal of Theoretical Biology*, 251(2):380–387, 2008.

[172] A. Lempel and J. Ziv. On the complexity of finite sequences. *IEEE Trans. on Information Theory*, 22(1):75–81, January 1976.

[173] G. Gordon. Multi-dimensional linguistic complexity. *Journal of Biomolecular Structure & Dynamics*, 20(6):747–750, 2003.

[174] E. Rivals, O. Delgrange, J.-P. Delahaye, M. Dauchet, M.-O. Delorme, A. Hénaut, and E. Ollivier. Detection of significant patterns by compression algorithms: the case of approximate tandem repeats in DNA sequences. *Computer Applications in the Biosciences*, 13:131–136, 1997.

[175] V. Wood et al. The genome sequence of *Schizosaccharomyces pombe*. *Nature*, 415(6874):871–80, February 2002.

[176] D. Pratas, R. M. Silva, A. J. Pinho, and P. J. S. G. Ferreira. An alignment-free method to find and visualise rearrangements between pairs of dna sequences. *Scientific Reports*, 5:10203, May 2015.

[177] A. Avelar, L. Perfeito, I. Gordo, and M. Ferreira. Genome architecture is a selectable trait that can be maintained by antagonistic pleiotropy. *Nature Communications*, 4, 2013.

[178] H. Lee, J. Thompson, E. Wang, and M. Wetzler. Philadelphia chromosome-positive acute lymphoblastic leukemia. *Cancer*, 117(8):1583–1594, 2011.

[179] M. Zody, Z. Jiang, H. Fung, F. Antonacci, L. Hillier, M. Cardone, T. Graves, J. Kidd, Z. Cheng, Abouelleil A, et al. Evolutionary toggling of the MAPT 17q21. 31 inversion region. *Nature Genetics*, 40(9):1076–1083, 2008.

[180] M. Donnelly, P. Paschou, E. Grigorenko, D. Gurwitz, S. Mehdi, S. Kajuna, C. Barta, S. Kungulilo, N. Karoma, R. Lu, et al. The distribution and most recent common ancestor of the 17q21 inversion in humans. *The American Journal of Human Genetics*, 86(2):161–171, 2010.

[181] N. Setó-Salvia, F. Sánchez-Quinto, E. Carbonell, C. Lorenzo, D. Comas, and J. Clarimón. Using the neanderthal and denisova genetic data to understand the common MAPT 17q21 inversion in modern humans. *Human Biology*, 84(6):1, 2013.

[182] M. Meyerso, S. Gabriel, and G. Getz. Advances in understanding cancer genomes through second-generation sequencing. *Nature Reviews Genetics*, 11(10):685–696, 2010.

[183] K. Das and P. Tan. Molecular cytogenetics: recent developments and applications in cancer. *Clinical Genetics*, 84(4):315–325, 2013.

[184] T. Wang, C. Maierhofer, M. Speicher, C. Lengauer, B. Vogelstein, K. Kinzler, and V. Velculescu. Digital karyotyping. *Proceedings of the National Academy of Sciences USA*, 99(25):16156–16161, 2002.

[185] Martin Kircher. Analysis of high-throughput ancient DNA sequencing data. In *Ancient DNA*, pages 197–228. Springer, 2012.

[186] Michael Brudno, Sanket Malde, Alexander Poliakov, Chuong B Do, Olivier Couronne, Inna Dubchak, and Serafim Batzoglou. Glocal alignment: finding rearrangements during alignment. *Bioinformatics*, 19(suppl 1):i54–i62, 2003.

[187] Scott Schwartz, W James Kent, Arian Smit, Zheng Zhang, Robert Baertsch, Ross C Hardison, David Haussler, and Webb Miller. Human–mouse alignments with blastz. *Genome research*, 13(1):103–107, 2003.

[188] Colin N Dewey. Aligning multiple whole genomes with mercator and mavid. In *Comparative genomics*, pages 221–235. Springer, 2008.

[189] Aaron E Darling, Bob Mau, and Nicole T Perna. progressiveMauve: multiple genome alignment with gene gain, loss and rearrangement. *PLoS ONE*, 5(6):e11147, 2010.

[190] Inna Dubchak, Alexander Poliakov, Andrey Kislyuk, and Michael Brudno. Multiple whole-genome alignments without a reference organism. *Genome research*, 19(4):682–689, 2009.

[191] Kelly A Frazer, Lior Pachter, Alexander Poliakov, Edward M Rubin, and Inna Dubchak. Vista: computational tools for comparative genomics. *Nucleic acids research*, 32(suppl 2):W273–W279, 2004.

[192] Adam Siepel, Gill Bejerano, Jakob S Pedersen, Angie S Hinrichs, Minmei Hou, Kate Rosenbloom, Hiram Clawson, John Spieth, LaDeana W Hillier, Stephen Richards, et al. Evolutionarily conserved elements in vertebrate, insect, worm, and yeast genomes. *Genome research*, 15(8):1034–1050, 2005.

[193] Donna Karolchik, Gill Bejerano, Angie S Hinrichs, Robert M Kuhn, Webb Miller, Kate R Rosenbloom, Ann S Zweig, David Haussler, and W James Kent. Comparative genomic analysis using the ucsc genome browser. In *Comparative Genomics*, pages 17–33. Springer, 2008.

[194] Shyam Prabhakar, Francis Poulin, Malak Shoukry, Veena Afzal, Edward M Rubin, Olivier Couronne, and Len A Pennacchio. Close sequence comparisons are sufficient to identify human cis-regulatory elements. *Genome research*, 16(7):855–863, 2006.

[195] Simon G Gregory, Mandeep Sekhon, Jacqueline Schein, Shaying Zhao, Kazutoyo Osoegawa, Carol E Scott, Richard S Evans, Paul W Burridge, Tony V Cox, Christopher A Fox, et al. A physical map of the mouse genome. *Nature*, 418(6899):743–750, 2002.

[196] Brian J Haas, Arthur L Delcher, Jennifer R Wortman, and Steven L Salzberg. Dagchainer: a tool for mining segmental genome duplications and synteny. *Bioinformatics*, 20(18):3643–3646, 2004.

[197] Stefan Kurtz, Adam Phillippy, Arthur L Delcher, Michael Smoot, Martin Shumway, Corina Antonescu, and Steven L Salzberg. Versatile and open software for comparing large genomes. *Genome biology*, 5(2):R12, 2004.

[198] Yoshiyuki Ohtsubo, Wakako Ikeda-Ohtsubo, Yuji Nagata, and Masataka Tsuda. Genomematcher: a graphical user interface for dna sequence comparison. *BMC bioinformatics*, 9(1):376, 2008.

[199] Nicholas H Putnam, Mansi Srivastava, Uffe Hellsten, Bill Dirks, Jarrod Chapman, Asaf Salamov, Astrid Terry, Harris Shapiro, Erika Lindquist, Vladimir V Kapitonov, et al. Sea anemone genome reveals ancestral eumetazoan gene repertoire and genomic organization. *science*, 317(5834):86–94, 2007.

[200] Suzanna E Lewis, SM Searle, N Harris, M Gibson, V Lyer, J Richter, C Wiel, L Bayraktaroglir, E Birney, MA Crosby, et al. Apollo: a sequence annotation editor. *Genome Biol*, 3(12):1–14, 2002.

[201] A. Sinha and J. Meller. Cinteny: flexible analysis and visualization of synteny and genome rearrangements in multiple organisms. *BMC bioinformatics*, 8(1):82, 2007.

[202] Miriah Meyer, Tamara Munzner, and Hanspeter Pfister. Mizbee: a multiscale synteny browser. *Visualization and Computer Graphics, IEEE Transactions on*, 15(6):897–904, 2009.

[203] Martin Krzywinski, Jacqueline Schein, İnanç Birol, Joseph Connors, Randy Gascoyne, Doug Horsman, Steven J Jones, and Marco A Marra. Circos: an information aesthetic for comparative genomics. *Genome research*, 19(9):1639–1645, 2009.

[204] C. Nielsen, M. Cantor, I. Dubchak, D. Gordon, and T. Wang. Visualizing genomes: techniques and challenges. *Nature methods*, 7:S5–S15, 2010.

[205] D. Pratas and A. J. Pinho. Exploring deep Markov models in genomic data compression using sequence pre-analysis. In *Proc. of the 22th European Signal Processing Conf., EUSIPCO-2014*, pages 2395–2399, Lisbon, Portugal, September 2014.

[206] S. Blair Hedges, Joel Dudley, and Sudhir Kumar. TimeTree: a public knowledge-base of divergence times among organisms. *Bioinformatics*, 22(23):2971–2972, 2006.

[207] S. Levy, G. Sutton, P. C. Ng, L. Feuk, A. L. Halpern, et al. The diploid genome sequence of an individual human. *PLoS Biology*, 5:2113–2144, 2007.

[208] M. Farré, D. Micheletti, and A. Ruiz-Herrera. Recombination rates and genomic shuffling in human and chimpanzee — a new twist in the chromosomal speciation theory. *Molecular Biology and Evolution*, 30(4):853–864, 2013.

[209] L. Feuk, J. MacDonald, T. Tang, A. Carson, M. Li, G. Rao, R. Khaja, and S. Scherer. Discovery of human inversion polymorphisms by comparative analysis of human and chimpanzee DNA sequence assemblies. *PLoS Genetics*, 1(4):e56, 2005.

[210] G. Greve, E. Alechine, J. Pasantes, C. Hodler, W. Rietschel, T. Robinson, and W. Schempp. Y-chromosome variation in hominids: intraspecific variation is limited to the polygamous chimpanzee. *PLoS ONE*, 6(12):e29311, 2011.

[211] F. Ray, E. Zimmerman, B. Robinson, M. Cornforth, J. Bedford, E. Goodwin, and S. Bailey. Directional genomic hybridization for chromosomal inversion discovery and detection. *Chromosome Research*, 21(2):165–174, 2013.

[212] L. Biesecker. The Greig cephalopolysyndactyly syndrome. *Orphanet J Rare Dis*, 3(10):238, 2008.

[213] I. Cuscó, R. Corominas, M. Bayés, R. Flores, N. Rivera-Brugués, V. Campuzano, and L. Pérez-Jurado. Copy number variation at the 7q11. 23 segmental duplications is a susceptibility factor for the Williams-Beuren syndrome deletion. *Genome Research*, 18(5):683–694, 2008.

[214] L. Osborne, M. Li, B. Pober, D. Chitayat, J. Bodurtha, A. Mandel, T. Costa, T. Grebe, S. Cox, L. Tsuie, et al. A 1.5 million–base pair inversion polymorphism in families with Williams-Beuren syndrome. *Nature Genetics*, 29(3):321–325, 2001.

[215] A. Sharp, S. Hansen, R. Selzer, Z. Cheng, R. Regan, J. Hurst, H. Stewart, S. Price, E. Blair, R. Hennekam, et al. Discovery of previously unidentified genomic disorders from the duplication architecture of the human genome. *Nature Genetics*, 38(9):1038–1042, 2006.

[216] A. Weise, M. Gross, S. Schmidt, F. Reichelt, U. Claussen, and T. Liehr. New aspects of chromosomal evolution in the gorilla and the orangutan. *International Journal of Molecular Medicine*, 19(3):437–443, 2007.

[217] A. J. Pinho, P. J. S. G. Ferreira, S. P. Garcia, and J. M. O. S. Rodrigues. On finding minimal absent words. *BMC Bioinformatics*, 10(137), May 2009.

[218] Supaporn Chairungsee and Maxime Crochemore. Using minimal absent words to build phylogeny. *Theoretical Computer Science*, 450:109–116, 2012.

[219] S. P. Garcia, A. J. Pinho, J. M. O. S. Rodrigues, C. A. C. Bastos, and P. J. S. G. Ferreira. Minimal absent words in prokaryotic and eukaryotic genomes. *PLoS ONE*, 6(1):e16065, January 2011.

[220] Zong-Da Wu, Tao Jiang, and Wu-Jie Su. Efficient computation of shortest absent words in a genomic sequence. *Information Processing Letters*, 110(14):596–601, 2010.

[221] J. Herold, S. Kurtz, and R. Giegerich. Efficient computation of absent words in genomic sequences. *BMC Bioinformatics*, 9(1):167, 2008.

[222] Uzma N Sarwar, Pamela Costner, Mary E Enama, Nina Berkowitz, Zonghui Hu, Cynthia S Hendel, Sandra Sitar, Sarah Plummer, Sabue Mulangu, Robert T Bailer, et al. Safety and immunogenicity of DNA vaccines encoding Ebolavirus and Marburgvirus wild-type glycoproteins in a phase I clinical trial. *J. Infect. Dis.*, page jiu511, 2014.

[223] Sylvain Baize, Delphine Pannetier, Lisa Oestereich, Toni Rieger, Lamine Koivogui, N'Faly Magassouba, Barr Soropogui, Mamadou Saliou Sow, Sakoba Keta, Hilde De Clerck, Amanda Tiffany, Gemma Dominguez, Mathieu Loua, Alexis Traor, Moussa Koli, Emmanuel Roland Malano, Emmanuel Heleze, Anne Bocquin, Stephane Mly, Herv Raoul, Valrie Caro, Dniel Cadar, Martin Gabriel, Meike Pahlmann, Dennis Tappe, Jonas Schmidt-Chanasit, Benido Impouma, Abdoul Karim Diallo, Pierre Formenty, Michel Van Herp, and Stephan Gnther. Emergence of Zaire Ebola virus disease in Guinea. *N. Engl. J. Med.*, 371(15):1418–1425, 2014.

[224] D Butler and L Morello. Ebola by the numbers: The size, spread and cost of an outbreak. *Nature*, 514(7522):284–285, 2014.

[225] Anne Gulland. Clinical trials of Ebola therapies to begin in December. *BMJ*, 349, 2014.

[226] Steven M Jones, Heinz Feldmann, Ute Ströher, Joan B Geisbert, Lisa Fernando, Allen Grolla, Hans-Dieter Klenk, Nancy J Sullivan, Viktor E Volchkov, Elizabeth A Fritz, et al. Live attenuated recombinant vaccine protects nonhuman primates against Ebola and Marburg viruses. *Nat. Med.*, 11(7):786–790, 2005.

[227] Brian M Friedrich, John C Trefry, Julia E Biggins, Lisa E Hensley, Anna N Honko, Darci R Smith, and Gene G Olinger. Potential vaccines and post-exposure treatments for filovirus infections. *Viruses*, 4(9):1619–1650, 2012.

[228] Stephen K Gire, Augustine Goba, Kristian G Andersen, Rachel SG Sealfon, Daniel J Park, Lansana Kanneh, Simbirie Jalloh, Mambu Momoh, Mohamed Fullah, Gytis Dudas, et al. Genomic surveillance elucidates Ebola virus origin and transmission during the 2014 outbreak. *Science*, page 1259657, 2014.

[229] R. M. Silva, D. Pratas, L. Castro, A. J. Pinho, and P. J. S. G. Ferreira. Three minimal sequences found in Ebola virus genomes and absent from human DNA. *Bioinformatics*, 31(15):2421–2425, April 2015.

[230] Shinji Watanabe, Takeshi Noda, and Yoshihiro Kawaoka. Functional mapping of the nucleoprotein of Ebola virus. *Journal of Virology*, 80(8):3743–3751, 2006.

[231] Amy C Shurtleff, Tam L Nguyen, David A Kingery, and Sina Bavari. Therapeutics for filovirus infection: traditional approaches and progress towards in silico drug design. *Expert Opin. Drug Discov.*, 7(10):935–954, 2012.

[232] Francois J Picard and Michel G Bergeron. Rapid molecular theranostics in infectious diseases. *Drug discovery today*, 7(21):1092–1101, 2002.

[233] Jonathan D Cook and Jeffrey E Lee. The secret life of viral entry glycoproteins: moonlighting in immune evasion. *PLOS pathog.*, 9(5):e1003258, 2013.

[234] Gopi S Mohan, Wenfang Li, Ling Ye, Richard W Compans, and Chinglai Yang. Antigenic subversion: a novel mechanism of host immune evasion by Ebola virus. *PLOS Pathog.*, 8(12):e1003065, 2012.

[235] Reed S Shabman, Omar J Jabado, Chad E Mire, Timothy B Stockwell, Megan Edwards, Milind Mahajan, Thomas W Geisbert, and Christopher F Basler. Deep sequencing identifies noncanonical editing of Ebola and Marburg virus RNAs in infected cells. *mBio*, 5(6):e02011–14, 2014.

[236] SP Fisher-Hoch, L Hutwagner, B Brown, and JB McCormick. Effective vaccine for Lassa fever. *J. Virol*, 74(15):6777–6783, 2000.

[237] Igor S Lukashevich. Advanced vaccine candidates for Lassa fever. *Viruses*, 4(11):2514–2557, 2012.

[238] Erika Check Hayden. RNA interference rebooted. *Nature*, 508:443, 2014.

[239] Hao Yin, Rosemary L Kanasty, Ahmed A Eltoukhy, Arturo J Vegas, J Robert Dorkin, and Daniel G Anderson. Non-viral vectors for gene-based therapy. *Nat. Rev. Genet.*, 15(8):541–555, 2014.

[240] João Conde, Miguel Larguinho, Ana Cordeiro, Luís R Raposo, Pedro M Costa, Susana Santos, Mário S Diniz, Alexandra R Fernandes, and Pedro V Baptista. Gold-nanobeacons for gene therapy: evaluation of genotoxicity, cell toxicity and proteome profiling analysis. *Nanotoxicology*, 8(5):521–532, 2014.

[241] Mohamed Shehata Draz, Binbin Amanda Fang, Pengfei Zhang, Zhi Hu, Shenda Gu, Kevin C Weng, Joe W Gray, and Fanqing Frank Chen. Nanoparticle-mediated systemic delivery of siRNA for treatment of cancers and viral infections. *Theranostics*, 4(9):872, 2014.

[242] Thomas W Geisbert, Amy CH Lee, Marjorie Robbins, Joan B Geisbert, Anna N Honko, Vandana Sood, Joshua C Johnson, Susan de Jong, Iran Tavakoli, Adam Judge, et al. Postexposure protection of non-human primates against a lethal Ebola virus challenge with RNA interference: a proof-of-concept study. *The Lancet*, 375(9729):1896–1905, 2010.

[243] MN Costa, B Veigas, JM Jacob, DS Santos, J Gomes, PV Baptista, R Martins, J Inácio, and E Fortunato. A low cost, safe, disposable, rapid and self-sustainable paper-based platform for diagnostic testing: lab-on-paper. *Nanotechnology*, 25(9):094006, 2014.

[244] Vinay G Joshi, Kantaraja Chindera, Arvind Kumar Singh, Aditya P Sahoo, Vikas D Dighe, Dimpal Thakuria, Ashok K Tiwari, and Satish Kumar. Rapid label-free visual assay for the detection and quantification of viral RNA using peptide nucleic acid (PNA) and gold nanoparticles (AuNPs). *Anal. Chim. Acta*, 795:1–7, 2013.

[245] B. H. Bloom. Space/time trade-offs in hash coding with allowable errors. *Communications of the ACM*, 13(7):422–426, July 1970.

[246] Andrei A. Broder and Michael Mitzenmacher. Network applications of Bloom filters: a survey. *Internet Mathematics*, 1(4):485–509, 2004.

[247] Elio F Vanin. Processed pseudogenes: characteristics and evolution. *Annual review of genetics*, 19(1):253–272, 1985.

[248] Graham M Hughes, Emma C Teeling, and Desmond G Higgins. Loss of olfactory receptor function in hominin evolution. *PloS one*, 9(1):e84714, 2014.

[249] Ajit Varki, Daniel H Geschwind, and Evan E Eichler. Human uniqueness: genome interactions with environment, behaviour and culture. *Nature Reviews Genetics*, 9(10):749–763, 2008.