



Universitat de Lleida

Novel Consistency-based Approaches for Dealing with Large-scale Multiple Sequence Alignments

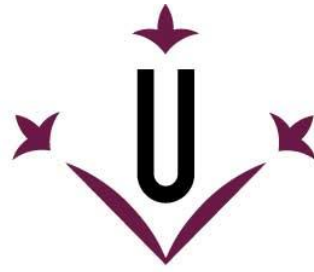
Jordi Lladós Segura

<http://hdl.handle.net/10803/663293>

ADVERTIMENT. L'accés als continguts d'aquesta tesi doctoral i la seva utilització ha de respectar els drets de la persona autora. Pot ser utilitzada per a consulta o estudi personal, així com en activitats o materials d'investigació i docència en els termes establerts a l'art. 32 del Text Refós de la Llei de Propietat Intel·lectual (RDL 1/1996). Per altres utilitzacions es requereix l'autorització prèvia i expressa de la persona autora. En qualsevol cas, en la utilització dels seus continguts caldrà indicar de forma clara el nom i cognoms de la persona autora i el títol de la tesi doctoral. No s'autoritza la seva reproducció o altres formes d'explotació efectuades amb finalitats de lucre ni la seva comunicació pública des d'un lloc aliè al servei TDX. Tampoc s'autoritza la presentació del seu contingut en una finestra o marc aliè a TDX (framing). Aquesta reserva de drets afecta tant als continguts de la tesi com als seus resums i índexs.

ADVERTENCIA. El acceso a los contenidos de esta tesis doctoral y su utilización debe respetar los derechos de la persona autora. Puede ser utilizada para consulta o estudio personal, así como en actividades o materiales de investigación y docencia en los términos establecidos en el art. 32 del Texto Refundido de la Ley de Propiedad Intelectual (RDL 1/1996). Para otros usos se requiere la autorización previa y expresa de la persona autora. En cualquier caso, en la utilización de sus contenidos se deberá indicar de forma clara el nombre y apellidos de la persona autora y el título de la tesis doctoral. No se autoriza su reproducción u otras formas de explotación efectuadas con fines lucrativos ni su comunicación pública desde un sitio ajeno al servicio TDR. Tampoco se autoriza la presentación de su contenido en una ventana o marco ajeno a TDR (framing). Esta reserva de derechos afecta tanto al contenido de la tesis como a sus resúmenes e índices.

WARNING. Access to the contents of this doctoral thesis and its use must respect the rights of the author. It can be used for reference or private study, as well as research and learning activities or materials in the terms established by the 32nd article of the Spanish Consolidated Copyright Act (RDL 1/1996). Express and previous authorization of the author is required for any other uses. In any case, when using its content, full name of the author and title of the thesis must be clearly indicated. Reproduction or other forms of for profit use or public communication from outside TDX service is not allowed. Presentation of its content in a window or frame external to TDX (framing) is not authorized either. These rights affect both the content of the thesis and its abstracts and indexes.



Universitat de Lleida

TESI DOCTORAL

**Novel Consistency-based Approaches for Dealing
with Large-scale Multiple Sequence Alignments**

Jordi Lladós Segura

Memòria presentada per optar al grau de Doctor per la Universitat de Lleida
Programa de Doctorat en Enginyeria i Tecnologies de la informació

Director/a
Fernando Guirado Fernández
Fernando Cores Prado

Tutor/a
Fernando Guirado Fernández

2018

Novel Consistency-based Approaches for Dealing with Large-scale Multiple Sequence Alignments

by

Jordi Lladós Segura

Submitted to the Department of Computer Science and Industrial Engineering
on June 1, 2018, in partial fulfillment of the
requirements for the degree of
PhD Thesis in Engineering and Information Technology

Abstract

Multiple Sequence Alignment (MSA) has become fundamental for performing sequence analysis in modern biology. With the advent of new high-throughput Next Generation Sequencing (NGS) technologies, the volume of data generated by sequencers has increased significantly. Thus, large-scale aligners are required. However, the heuristic nature of MSA methods together with their NP-hard computational complexity is slowing down the analysis of large-scale alignments with thousands of sequences or even whole genomes. Moreover, the accuracy of these methods is being drastically reduced when more sequences are aligned. Consistency methods proven to mitigate such errors add precomputed information (consistency library) for each pairwise to the alignment stage, and are capable of producing high-rated alignments. However, maintaining this whole collection of pairwise information in the memory limits the maximum number of sequences that can be dealt with at once.

The objective of this PhD is the study and proposal of new methods and tools to enable scalability for consistency-based MSA aligners, processing bigger datasets, improving their overall performance and the alignment accuracy.

The main obstacle to attain scalability on such methods is the library resource requirements (both memory and computing time) that grows quadratically with the number of sequences. Two methods are proposed to improve the scalability: 1) reducing the library information in order to fit it into the memory; and 2) storing the library data in secondary distributed storage, using the new Big Data paradigms (MapReduce, no-sql databases) and architectures (Hadoop) to calculate, store and access the library efficiently. In addition to the computational approaches, we propose an innovative secondary objective function to increase the accuracy of the final alignment.

The results demonstrate the effectiveness of the proposals, which improve the scalability, performance and accuracy of T-Coffee, the tool used to validate the different proposals.

Resum

L'alineament múltiple de seqüències (MSA) ha esdevingut fonamental per tal de realitzar anàlisis de seqüències a l'era de la biologia moderna. Amb l'arribada de les tecnologies de seqüenciació de nova generació d'alt rendiment (NGS), el volum de dades generades pels seqüenciadors s'ha incrementat significativament. Per tant, s'han de definir nous alineadors que puguin treballar a gran escala. No obstant això, la naturalesa heurística dels mètodes MSA juntament amb la complexitat computacional (NP-hard) està alentint l'anàlisi d'alineaments a gran escala que involucren milers de seqüències o fins i tot a genomes complets. A més, la qualitat d'aquests alineaments es veu dràsticament reduïda quan s'incrementa el nombre de seqüències. Els alineadors basats en consistència asseguren mitigar aquest problema mitjançant la inclusió d'informació precalculada (anomenada com a llibreria de consistència) per cada parell de seqüències a tractar a la fase d'alineament. Aquests mètodes són capaços de produir alineaments d'alta qualitat. No obstant això, mantenir aquest gran volum d'informació, que involucra a tots els parells de seqüències, a memòria limita la quantitat de seqüències que es poden tractar simultàniament.

L'objectiu d'aquest PhD és l'estudi i proposta de nous mètodes i eines per tal de permetre, als MSA basats en consistència, la capacitat d'escalar processant conjunts de dades molt més grans, millorant el rendiment i la qualitat de l'alineament.

El principal obstacle per tal d'aconseguir dita escalabilitat en aquests mètodes són els requisits de recursos de la llibreria (memòria i temps de còmput) els quals creixen quadràticament amb el nombre de seqüències. Al present treball de tesis es proposen dos mètodes per millorar l'escalabilitat: 1) reduir la informació de la llibreria per tal de mantenir-la a memòria; i 2) emmagatzemar les dades de la llibreria a un sistema d'emmagatzemament secundari distribuït, utilitzant els nous paradigmes de Big Data (MapReduce, bases de dades no-sql) i architectures (Hadoop) per calcular, mantenir i accedir a la llibreria eficientment. A més de l'enfocament computacional, s'ha desenvolupat una nova funció objectiu secundària que permet incrementar la qualitat de l'alineament final.

Els resultats demostren l'efectivitat de les propostes, les quals milloren l'escalabilitat, rendiment i qualitat de T-Coffee, l'eina emprada per validar les diferents propostes.

Resumen

El alineamiento múltiple de secuencias (MSA) se ha demostrado como fundamental para poder realizar análisis de secuencias en la era de la biología moderna. Con la llegada de las tecnologías de secuenciación de nueva generación y de altas prestaciones (NGS), el volumen de datos generados por los secuenciadores se ha incrementado significativamente. Por este motivo, es necesario desarrollar alineadores capaces de trabajar a gran escala. No obstante, la naturaleza heurística de los métodos de MSA, juntamente con su complejidad computacional (NP-hard) está retrasando el análisis de alineamientos a gran escala que involucran miles de secuencias o incluso a genomas completos. Además, la calidad de estos alineamientos se ve drásticamente reducida cuando se incrementa el número de secuencias a alinear. Los alineadores basados en consistencia permiten mitigar este problema añadiendo información precalculada (denominada librería de consistencia) para cada par de secuencias a tratar en la fase de alineamiento. Estos métodos son capaces de producir alineamientos de alta calidad. No obstante, almacenar este gran volumen de información, que involucra a todos los pares de secuencias, en memoria limita la cantidad de secuencias que se pueden tratar simultáneamente.

El objetivo de este PhD es el estudio y propuesta de nuevos métodos y herramientas que permitan a los MSA basados en consistencia, escalar (procesando un mayor número de secuencias), mejorando el rendimiento y la calidad del alineamiento.

El principal obstáculo para lograr dicha escalabilidad en estos métodos son los requisitos de recursos de la librería (memoria y tiempo de cómputo) los cuales crecen cuadráticamente con el número de secuencias. En el presente trabajo de tesis, se proponen dos métodos para mejorar la escalabilidad: 1) reducir la información de la librería para poder así mantenerla en memoria; y 2) almacenar los datos de la librería en un sistema de almacenamiento secundario distribuido, usando los nuevos paradigmas de Big Data (MapReduce, bases de datos no-sql) y arquitecturas (Hadoop) para calcular, almacenar y acceder a la librería eficientemente. Además del enfoque computacional, se ha desarrollado una nueva función objetivo secundaria para incrementar la calidad del alineamiento final.

Los resultados demuestran la efectividad de las propuestas, las cuales mejoran la escalabilidad, rendimiento y calidad de T-Coffee, la herramienta utilizada para validar las diferentes propuestas.

Acknowledgments

I would like to thank everyone who supported me during the course of this PhD. First of all, I want to acknowledge my supervisors, Dr. Fernando Cores Prado and Dr. Fernando Guirado Fernández, for their support and guidance over these years, and who encouraged and helped me to finish the research.

While working on the PhD, I met many members of the Group of Distributed Computing (GCD) from the Universitat de Lleida (UdL). I want to express my gratitude to the seniors who supported me behind the scenes: Concepció Roig, Francesc Giné, Francesc Solsona and Josep Ll. Lèrida. Then I wish to thank my co-workers, with whom I shared many breakfasts, lunches and deadlines at the university; Anabel Usié, Eloi Gabaldon, Ismael Arroyo, Ivan Teixidó, Jordi Mateo, Jordi Vilaplana, Josep Rius, Marc Gonzàlez, Marc Solé and Miquel Orobitg. Also thanks go to Montse Espunyes who helped with the administrative processes required for the PhD.

I also want to acknowledge all the members of the Edinburgh Data-Intensive Research (DIR) group at the University of Edinburgh. Mainly, I am grateful to Malcolm Atkinson and Rosa Filgueira for giving me the opportunity to visit the group and work with them.

To conclude, I would like to thank my closest friends and family for their support during these long years, especially my parents, Ramon and Rosa, who encouraged me to complete my studies and always showed me patience and understanding throughout the process.

Thanks to everyone.

Contents

1	Introduction	17
1.1	Computational Biology	19
1.1.1	Molecular Biology	19
1.1.2	Sequence Analysis	21
1.2	Sequence Alignment	23
1.2.1	Pairwise Alignment	25
1.2.2	Multiple Sequence Alignment	26
1.3	Distributed Computing Platforms	29
1.3.1	High-Performance Computing	30
1.3.2	Big Data Frameworks	35
1.4	Related Work	38
1.4.1	Multiple Sequence Alignment	38
1.4.2	Benchmarking	42
1.4.3	Next-Generation Sequencing in bioinformatics	43
1.5	Document structure	47
2	Methodology	49
2.1	Problem Statement	50
2.2	Main Objective	50
2.3	Milestones	52
2.4	Research Methodology	54

3	Papers	57
3.1	Recovering accuracy methods for scalable consistency library	57
3.1.1	Contributions to the state of the art	57
3.1.2	Paper 1: Recovering accuracy methods for scalable consistency library	58
3.2	PPCAS: implementation of a Probabilistic Pairwise model for Consistency-based multiple alignment in Apache Spark	60
3.2.1	Contributions to the state of the art	60
3.2.2	Paper 2: PPCAS: implementation of a Probabilistic Pairwise model for Consistency-based multiple alignment in Apache Spark	61
3.3	Scalable Consistency in T-Coffee through Apache Spark and Cassandra database	63
3.3.1	Contributions to the state of the art	63
3.3.2	Paper 3: Scalable Consistency in T-Coffee through Apache Spark and Cassandra database	64
3.4	Accurate consistency-based multiple sequence alignment reducing the memory footprint	66
3.4.1	Contributions to the state of the art	66
3.4.2	Paper 4: Accurate consistency-based multiple sequence alignment reducing the memory footprint	67
4	Global discussion of the results	69
4.1	Recovering accuracy methods for scalable consistency library	69
4.2	PPCAS: implementation of a Probabilistic Pairwise model for Consistency-based multiple alignment in Apache Spark	70
4.3	Scalable Consistency in T-Coffee through Apache Spark and Cassandra database	71
4.4	Accurate consistency-based multiple sequence alignment reducing the memory footprint	72
5	Global conclusions and future work	73

5.1	Conclusions	73
5.2	Future Work	77
A	Doctoral stay at the University of Edinburgh: A brief summary	79
B	Other contributions	83

List of Figures

1-1	DNA double helix.	20
1-2	The central dogma of molecular biology.	22
1-3	An example of bioinformatics workflow.	22
1-4	Cost per genome over recent years.	23
1-5	Multiple Sequence Alignment.	24
1-6	Example of global alignment	24
1-7	Example of local alignment	24
1-8	Needleman-Wunsch algorithm (dynamic programming algorithm). . .	25
1-9	Progressive alignment workflow.	28
1-10	Computer cluster.	32
1-11	Taxonomy of Flynn (1972).	33
1-12	Hadoop architecture and components.	36
1-13	Spark architecture.	37
1-14	T-Coffee algorithm stages.	40
1-15	Library structure.	40
1-16	T-Coffee execution time.	42
1-17	The Map process in SparkSW.	46
2-1	Memory hierarchy in computer architecture.	51
A-1	Overview of the 1000 genome-sequencing analysis workflow.	80
A-2	Processing the individuals of the 1000 genome using Apache Spark. .	81
A-3	Frequency overlap mutations of the 1000 genome using Apache Spark.	81
A-4	T-Coffee extension calculation using Apache Spark.	82

Chapter 1

Introduction

Understanding the structure of genes and proteins is a key to keeping people healthy and fighting off disease. In the 1950s, biology made a great breakthrough in this sense, when Frederick Sanger determined the structure of insulin [4], a hormone that regulates the sugar levels in the blood. This discovery established the foundations for sequencing proteins and enabled the practice of comparing sequences. However, manual comparisons turned out to be impractical. In order to speed up such processes, the possibilities of applying computational resources to biomedical problems started to take over. This led to the first approach between molecular biology and computer science in 1965, when Dayhoff published the *Atlas of Protein Sequence and Structure* [5] together with the first database of protein sequences, enabling new uses for computers in biology. Later on, in the late 1980s and the 1990s, computer tools able to align sequences appeared, thus solving the limitation of doing the comparisons manually. This combination of computer science and biology led to the appearance of the interdisciplinary field named bioinformatics, a field responsible for developing new methods and algorithms to solve formal and practical problems arising from the management and analysis of biological data.

With the advent of the high-throughput, high-volume, huge data generating capacity we have today, these tools started to become obsolete. Recent sequencing technologies, next-generation sequencing [20], allow deoxyribonucleic acid (DNA) and ribonucleic acid (RNA) to be sequenced faster and more cheaply than before, result-

ing in a greater number of larger sequences, even whole genomes, which could not previously be handled in a realistic time. This increase created the need for new data computation techniques to store and process higher volumes of information.

High-performance computing (HPC) enabled the use of parallel processing to run advanced applications efficiently, reliably and quickly, using the aggregation of multiple resources. Such infrastructures were able to deal with big amounts of data at first, but the continuous growth in this could not be handled by high-performance computing. Bigger datasets had to be processed and sent to the nodes through the network, producing bottlenecks and high delays. Due the need to analyze and process ever higher amounts of information, frameworks like Apache Hadoop [55] appeared and provided solutions for dealing with bigger datasets.

Multiple sequence alignment has been proven to be a powerful tool in many domains in bioinformatics and computational biology. It is used in many fields of study such as phylogenetic reconstruction, homology detection and 2D/3D structure prediction.

This PhD focuses on improving the performance and scalability of multiple sequence alignment tools which are not prepared for the next-generation sequencing. We deal with it from two approaches. The first one is about improving and optimizing the functions that characterize the multiple sequence alignment code, improving its execution times and accuracy, and the second one, regarding the scalability, enabling it to deal with bigger datasets taking advantage of the computational technologies that we have nowadays. To date, high-performance computing has proven to be able to speed up many important tools with computing power. Real time forecasting or sequencing a genome in a few days is a solid proof of this.

The rest of the introduction is structured as follows. First, there is a brief introduction to some key aspects of biology and the interdisciplinary field of computational biology followed by the most important step in the sequence analysis, the sequence alignment. Next, we explain the current parallel computing techniques and the related work in the literature, and finally, the document structure.

1.1 Computational Biology

The definition of computational biology and bioinformatics is often misinterpreted, in general terms. Bioinformatics refers to the use of computers in order to handle biological data. In practice, it deals with databases, file formats and pipeline designs, whereas the algorithms and tools to facilitate the biological analyses are included in computational biology. Both involve the analysis of biological data, particularly DNA, RNA and protein sequences to help the understanding of evolutionary aspects of molecular biology.

Computational biology is an interdisciplinary field that includes foundations in applied mathematics, statistics, molecular biology, genetics, genomics and computer science, but overall it is the melding of molecular biology with computer science, so some background to molecular biology and understanding its central dogma [6] is required.

1.1.1 Molecular Biology

The fundamental molecules responsible for the functioning of every living system are the cells. A multicellular organism has a cell and a cell nucleus. This nucleus contains the DNA, the hereditary material. All life depends on three critical molecules, namely DNA, RNA, and protein. The DNA molecules hold information on how the cell works and they must pass on the instructions for creating their constituent components to their descendants, the protein molecules that form enzymes that send signals to other cells and regulate gene activity and form the body's major components and finally, the RNA, which is an intermediary between the DNA and the proteins and provides templates to synthesize into protein.

Deoxyribonucleic Acid - DNA

DNA is made up from sugar molecule, phosphate (group 4) and different bases (nucleotides), adenine (A), thymine (T), guanine (G) and cytosine (C) as shown in Figure 1-1. The bases on one strand of DNA form base pairs with a second strand of DNA

to form the double helix, which represents the molecular appearance of DNA. The bases from one strand of a DNA helix are in essence a mirror image of the bases in the other strand – when there is an A in one strand there is a T in the other; when there is a C in one strand there is a G in the other. These “base pairing” rules are the key to understanding how DNA carries information and is copied into a new DNA strand (a cell must copy its DNA before it divides into two cells). The order of bases is referred to as the sequence that encodes the information in the DNA.

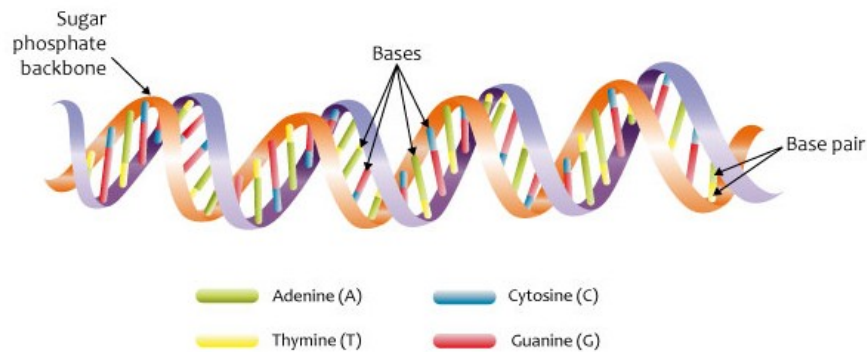


Figure 1-1: DNA double helix.

Ribonucleic Acid - RNA

RNA is often compared to a copy from a reference book, or a template, because it carries the same information as its DNA template but is not used for long-term storage. Like DNA, RNA is assembled as a chain of nucleotides, but unlike DNA it is more often found in nature as a single strand folded onto itself, rather than a paired double strand. When the cell needs to produce a certain protein, it activates the protein gene and produces multiple copies of that piece of DNA in the form of messenger RNA, or mRNA. The multiple copies of mRNA are then used to translate the genetic code into protein through the action of the cell’s protein manufacturing machinery, the ribosomes.

Protein

Like DNA, proteins are polymers: large chains assembled from prefabricated molecular units, which, in the case of proteins, are amino acids. There are twenty different types of amino acids that can be combined to make a protein. The sequence of amino acids determines each protein's unique 3-dimensional structure and its specific function. These proteins are essential in all living organisms, and are involved in DNA synthesis, RNA synthesis, the immune response, cell structure and much more. Proteins are more informative than DNA/RNA as they are characterized by 20 vs 4 characters. Also protein sequences offer a longer “look-back” time than the rest.

The Central Dogma

DNA carries the genetic information of a cell and consists of thousands of genes. Each gene serves as a recipe for how to build a protein molecule. Proteins perform important tasks for the cell functions or serve as building blocks. The flow of information from the genes determines the protein composition and thereby the functions of the cell. The process of DNA being processed into protein is shown in Figure 1-2 and is explained in the following paragraphs.

The DNA is situated in the nucleus, organized into chromosomes. Every cell must contain the genetic information so the DNA is duplicated before a cell divides (replication). When proteins are needed, the corresponding genes are transcribed into RNA (transcription). The RNA is first processed so that non-coding parts are removed (processing) and is then transported out of the nucleus (transport). Outside the nucleus, the proteins are built based upon the code in the RNA (translation).

1.1.2 Sequence Analysis

Sequence analysis is the process of subjecting a sequence to any of a wide range of analytical methods to understand its features, function, structure, or evolution. The analysis includes such methodologies as sequence alignment, searches of biological databases, and others. Biologists use many different types of workflows in order to

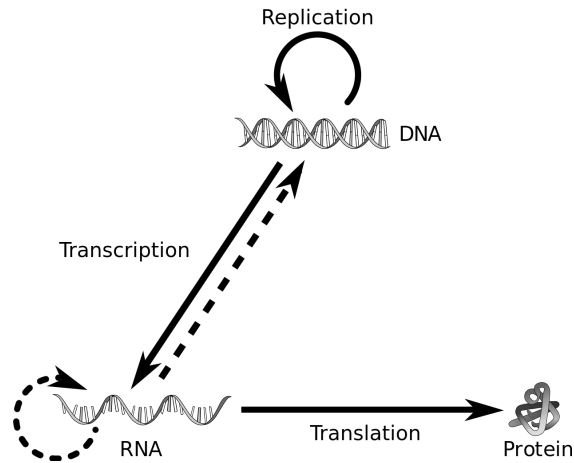


Figure 1-2: The central dogma of molecular biology.

carry out these analyses. An example of this kind is given in Figure 1-3 where (1) a file containing similar sequences, in *Fast All* (FASTA) format, is used as the input of the workflow. This format is a text-based way of representing DNA, RNA or protein sequences, where each sequence has its own name and description. Whatever the type of molecule, (2) the FASTA file is aligned by the Multiple Sequence Alignment tool of the researcher’s choice, and (3) this aligned file is used to compute an alignment for those sequences in PHYLogeny Inference Package (PHYLIP) format, a common format for generating phylogenetic trees [39]. Then, (4) an evolutionary model [19] should be selected which, in conjunction with the PHYLIP, will produce (5) the final phylogenetic tree.

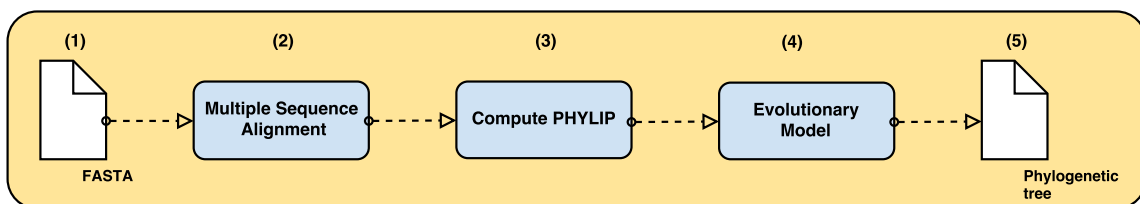


Figure 1-3: An example of bioinformatics workflow.

The final alignment lets the researcher identify regions of similarity that may be a consequence of functional, structural, or evolutionary relationships between the sequences. Meanwhile, the phylogenetic tree shows evolutionary relationships among various biological species or other entities.

Prior to this kind of analysis, it is important to note that the sequence to be analyzed should be obtained. Such process was done by Sanger sequencing method. This was the most widely-used sequencing method for approximately 40 years. It is quite simple, yet extremely laborious.

While some of the progress on sequencers relied on improvements to the Sanger method, the real game changer was the emergence of the NGS, a term for any machine not sequencing the Sanger way. Such technology is able to sequence an entire human genome within a single day. In contrast, the previous Sanger sequencing technology, used to decipher the human genome, required over a decade to deliver the final draft. Not only was the time reduced, but so were the costs. Figure 1-4 shows the cost per genome during the last decade. It is a fact that the NGS broke Moore's law and opened new challenges in the study of the human genome.

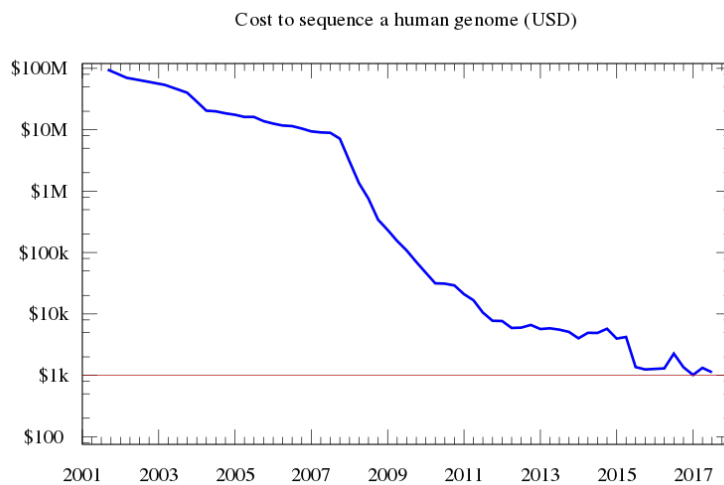


Figure 1-4: Cost per genome over recent years.

1.2 Sequence Alignment

As can be seen in Figure 1-5, sequence alignment consists of arranging sequences of DNA, RNA or protein. This facilitates the identification of similarities, mutations, divergences etc., between a group of sequences. Through sequence alignment, we are capable of exploring the mysteries of many aspects of biology and life.

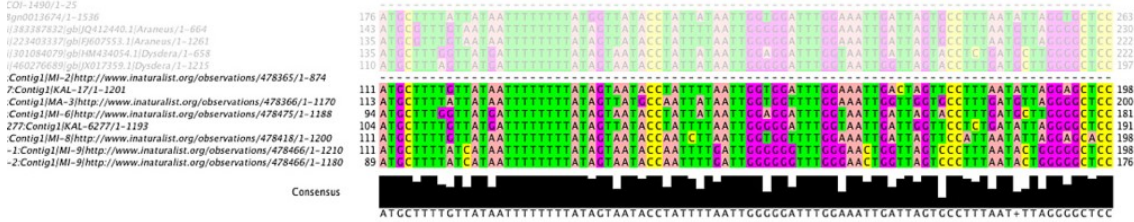


Figure 1-5: Multiple Sequence Alignment.

The alignments can be generated globally or locally. It basically depends on whether one sequence is a subsequence of the other or not. An example of each type is shown in Figures 1-6 and 1-7.

Global alignment is meant to be used for an end-to-end alignment, where the sequences are of similar length. It may end up with a low Identity and a lot of Gaps if the sizes of query and subject are dissimilar.

```

ACTACTAGATTACTTACGGATCAGGTACTTTAGAGGCTTGCAACCA
|||||        |||   |||||||              |||
ACTACTAGATT---ACGGATC--GTACTTTAGAGGCTAGCAACCA

```

Figure 1-6: Example of global alignment

Local alignment finds local regions with the highest level of Similarity, and is more suitable when one sequence is somewhat smaller than the other.

```

ACTACTAGATTACTTACGGATCAGGTACTTTAGAGGCTTGCAACCA
|||  |||   |||||||              |||
TACTCACGGATGAGGTACTTTAGAGG

```

Figure 1-7: Example of local alignment

The sequence alignment is divided in two different groups, depending on how many sequences are to be aligned. On the one hand, there is the pairwise alignment that can only be used between two sequences at a time. It results in an optimal alignment that can be solved in a feasible time.

On the other hand, multiple sequence alignment is an extension of pairwise alignment to incorporate three or more sequences at a time. MSA is essential for any

bioinformatic workflow and with the assumption being that as alignments improve, so do phylogenetic reconstructions, meaning that improving current methods can greatly affect the biology field. However, that is not easy, as most formulations of the problem lead to NP-hard (Non-deterministic Polynomial) [52] optimization problems.

1.2.1 Pairwise Alignment

Given two strings $S = (S_1, \dots, S_n)$ and $T = (T_1, \dots, T_n)$, a pairwise alignment of S and T is defined as an ordered set of pairings of (S_i, T_j) and gaps $(S_i, -)$ and $(-, T_j)$, $S_i \in S$ and $T_j \in T$. The alignment is reduced to the two original strings when all gaps in the alignment are deleted. Alignment by dynamic programming based on Gotoh [36] or Myers & Miller [28] guarantees that the resulting alignment is the optimal alignment or one of the equally optimal alignments.

The Needleman-Wunsch algorithm [42] is an example of a dynamic programming algorithm that finds the best possible global alignments between two strings. In Figure 1-8 shows that the problem is solved by filling a two dimensional matrix.

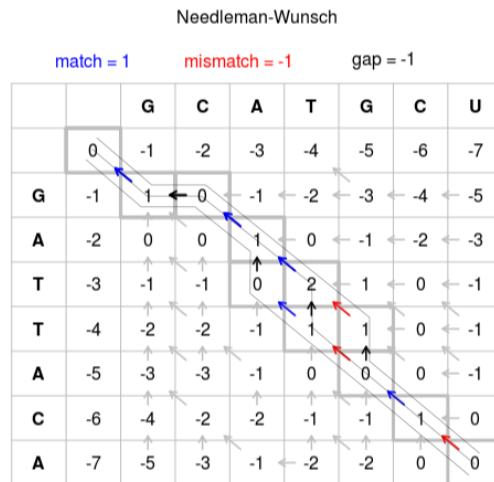


Figure 1-8: Needleman-Wunsch algorithm (dynamic programming algorithm).

All best possible alignments are found by filling the matrix according to the three score parameters: match, mismatch, and gap. It requires $O(nm)$ space and time where n and m are the lengths of the first and second strings, respectively. After

the matrix is filled with a scoring function, the algorithm generates the optimal path using a traceback technique. The traceback always begins with the last cell filled, the bottom-right, and it moves according to the value written in the cell (the higher the better). The path can take three different directions, (1) it can follow the diagonal, meaning that the letters from two sequences are aligned, (2) it can chose left, then a gap is introduced into the left sequence or (3) it can go up, so the gap is introduced into the top sequence. This process is repeated until the top-left cell is reached. In this particular case, it results in six possible best alignments as there is not necessarily a single correct alignment (GCATG-CU|G-ATTACA, GCA-TGCU|G-ATTACA and GCAT-GCU|G-ATTACA).

1.2.2 Multiple Sequence Alignment

As for the pairwise alignment, the goal is to find an alignment that maximizes a scoring function. In the case of pairwise alignment, the dynamic programming is fast and reliable, but if we extrapolate it to align more sequences, the time grows with the number of sequences. The time complexity can be defined as two sequences $O(n^2)$, three sequences $O(n^3)$ and k sequences $O(n^k)$. That is why the most appropriate way to solve this problem is to use heuristic methods. It may be impossible to identify Residues that align properly (structurally) throughout a multiple sequence alignment. Furthermore, a computational optimal alignment given a particular objective function does not guarantee that it will also be the biologically optimal one.

MSA require more complex computational algorithms than the pairwise alignment because of their increased difficulty. Below are the different heuristics used in the literature.

Exact Method

The exact methods of multiple alignment use dynamic programming and are guaranteed to find optimal solutions. However, they are not feasible for more than a few sequences.

The main difficulty in aligning multiple sequences by dynamic programming is the rapidly increasing need for memory and computational power. While aligning three sequences by dynamic programming has been implemented, it is not practical to align more than three sequences.

Progressive Method

Progressive alignment [30] is one of the most widely used heuristics. First, it aligns the most similar sequences and then adds the less related sequences successively to the alignment. This process can be defined by the following heuristic as shown in Figure 1-9:

1. Calculate the distance matrix between each pair of sequences: This matrix indicates the mutation/evolutionary events between all-against-all the sequences that are aligned. It can be built from a number of different sources, like substitution matrices, such as PAM [7] or BLOSUM [14], or distance-based methods, such as KTUP, UPGMA, etc.
2. Build the guide tree using the previously computed distance matrix. The guide tree is a visual representation of the distance matrix, but at the same time it also offers better integration in the programmer code. It allows the needed mobility between the nodes of the tree while preserving the similarity order in each step of the progressive alignment.
3. Do pairwise alignments following the guide tree: The pairwise alignments of sequences or groups of sequences is repeated successively until the root of the tree is reached. In each of these iterations, the dynamic programming proceeds to align the pair or groups of sequences.

The biggest problem of this method is that if an error is made in the initials steps of the alignment, it propagates until it reaches the root of the tree. For this reason, other approximations based on the progressive method appeared. These are the consistency-based method [35], which give more awareness of the other residue

position in the alignment in each step of the heuristic, and the iterative method, which refines the alignment by rebuilding the guide tree over and over until is done.

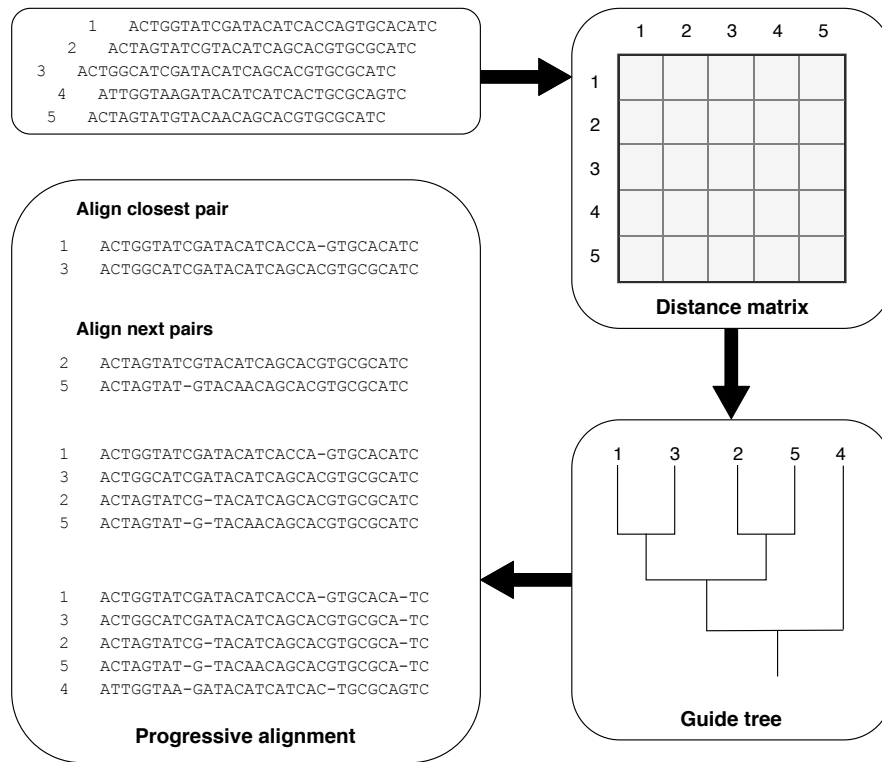


Figure 1-9: Progressive alignment workflow.

Consistency-based

The main idea behind tools using this method is to correct the errors caused in the first stages by the guide tree. The basis is the same as the progressive alignment strategy, with an ability to consider information from all the sequences during each alignment step, not just the one being aligned at that stage. Such information is enough to avoid aligning two residues that are not commonly together in a particular dataset. Despite the improvements over a standard progressive tool, the complexity of the execution time and memory requirements increase. Building the consistency represents an all-against-all pairwise alignment, which supposes $N * (N - 1)/2$ alignments and the order of $O(N^2L^2)$ in time and space, N being the number of sequences and L , the length of the sequence.

This PhD focuses on this improvement over the basic progressive method.

Iterative Approaches

Iterative MSA is similar to the progressive alignment. It is an optimization method that proceeds by realigning each sequence in the multiple alignment until the iterations consistently fail to improve the alignment. The most common optimizations are the use of the sum-of-pairs score or the guide tree. These optimizations are performed iteratively and continue until they reach some defined limit. As with many optimization problems, it can also get stuck in a local minimum, resulting in longer execution times.

Many tools have implemented hybrid methodologies between iterative- and consistency-based methods. Despite their notable alignments, these tools are not prepared for the Next-Generation Sequencing, meaning that when the number of sequences to be aligned increases, the execution times become unfeasible and the accuracy degrades. The use of parallel technologies and newer approaches should be able to achieve scalable Multiple Sequence Aligners.

1.3 Distributed Computing Platforms

Understanding the human brain, folding proteins, simulating earthquakes or even recreating the Big bang are some of many cases where scientists needed a lot of computing power to obtain results. In the 1960s, the first non-military supercomputers were introduced, monster machines from companies like Cray and IBM that were able to surpass by far the general-purpose computers. Over the next twenty years, computers gained greatly in popularity, information technology operations started to grow in complexity and the academic community and corporations became aware of their importance. In the 1990s, the first data centers as we know them appeared, facilities that centralized computer systems with thousands of processors to provide services including computing power, data storage and security. The convergence between computers and supercomputers also started, as the ideas that went into the design of supercomputers became part of personal computers. Nowadays, in the 21st century, massively parallel computers with thousands of processors are interconnected

by ultra-fast networks achieving the level of petaFLOPS. Currently, data centers have stepped up their energy efficiency thus making them more cost-effective.

The use of the data centers in order to deal with computational problems that are either too large for standard computers or would take too long is called High-Performance Computing (HPC). A desktop computer generally has a single processing chip, limited memory and small storage space, whereas an HPC infrastructure contains thousands of processors, vast memory and huge amounts of storage. HPC has already contributed enormously to scientific innovation, industrial and economic competitiveness, national and regional security, and the quality of human life. However, as the world steadily becomes more connected with an ever-increasing number of electronic devices, the growth in data and the way it can be used is also changing. Almost ninety percent of the data in the world today has been created in the last two years. Therefore, alternatives for dealing with Big Data have emerged, frameworks like Apache Hadoop and Apache Spark greatly improve the data locality of HPC. These frameworks bring the computers to the data. If we have a dataset with 1TB and the code to process it has 1MB, then it is more efficient to send the program to where the data is stored (residing on each node, which does both storage and computing) rather the other way round, as seen in HPC.

Next, the concepts of HPC and Big Data are presented in more detail.

1.3.1 High-Performance Computing

High-performance computing is the use of parallel processing to run advanced application programs efficiently, reliably and quickly. Thus, HPC uses aggregated resources, or Clusters (Figure 1-10); collections of computer nodes connected through a local network.

Cluster In computer science, a cluster is a set of nodes interconnected by a network fabric that act as a single system to enable high availability, load balancing and parallel processing. The idea of grouping several nodes together to increase the processing power is very practical in many applications, as it allows the work

that would not be reasonably done on a single node to be split. Furthermore, it has other benefits; better fault tolerance, higher availability and horizontal scaling. When more storage or processing power is needed, a new node can be added to the cluster, and it does not have to be the same. Heterogeneous systems are quite common in many organizations, where different kinds of resources are accumulated over the years, and these are grouped together to work as a cluster.

Grid computing This was meant to be the ideal successor to clusters. The main advantage of grid computing in the research field was to increase the available computing power to institutions without the need for new clusters, employing the use of multiple decentralized, heterogeneous and geographically-dispersed nodes, providing raw computing power on demand. It is similar to a power grid, where the user does not have to worry about the source of the computing power. As a drawback, each node or cluster is relatively independent of the others, given the geographic distance and low Internet bandwidth. Despite all the efforts put into the grid, its decentralized scheme and having to share the resources has not been widely adopted. Thus, an evolution of the grid and distributed computing paradigm called the cloud computing was proposed.

Cloud computing This is a new computing paradigm that provides a large pool of dynamically-scalable and virtual resources as a service on demand. So instead of requesting specific disk, CPU and memory resources, a service allocates the needed resources to physical resources. So, if an application requires one computing unit, then the service only allocates one computing unit, which may be shared with some other applications using the service. Its main advantage is that it provides large resources immediately without investing in new infrastructure.

In fact, parallel computing is better for modeling real world problems, as the world itself is full of events happening at the same time. Traditionally, programs were coded to run for serial computation, while parallel processing enables the use of multiple resources to solve a computational problem, breaking the problem into smaller tasks, so each processing element can execute a task simultaneously. In order

to fulfill its work, these tasks usually need to exchange data with each other. On doing so, a problem appears; synchronization. Coordinating parallel tasks in an execution workflow implies establishing synchronization points, where all the tasks have to arrive to move on. This may lead to some nodes or CPUs remaining idle, waiting for the slower node to reach the established point. In the literature, there is a qualitative measure to classify the amount of communications that a group of tasks produce. This is named granularity.

Coarse granularity This implies that there is a lot of computation work. Therefore, the communications are poor.

Fine granularity This is a symptom of small amounts of work and high levels of communications, as every time that the work is done, the tasks synchronize.

A good ratio between the work and communications is essential to produce highly efficient applications.

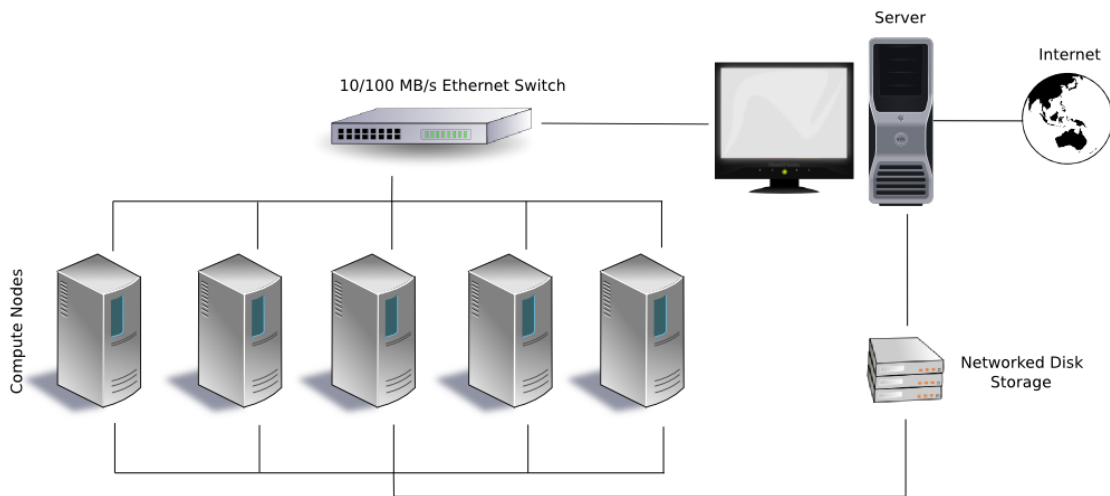


Figure 1-10: Computer cluster.

Many authors have proposed taxonomies to classify computer systems. There are two models proposed by Flynn that are worth mentioning for parallel processing

tasks. These are Single Instruction Multiple Data (SIMD) and Multiple Instruction Multiple Data (MIMD).

SIMD Figure 1-11a, where the same instructions are executed on multiple processing units (PU) simultaneously, this being usual in vectorial processing (GPUs). If a problem fits the SIMD taxonomy, it can reach a very high speedup since GPUs are much more powerful than CPUs.

MIMD Figure 1-11b, where different instructions are executed on different PUs, which is the standardized case for multi-core processing.

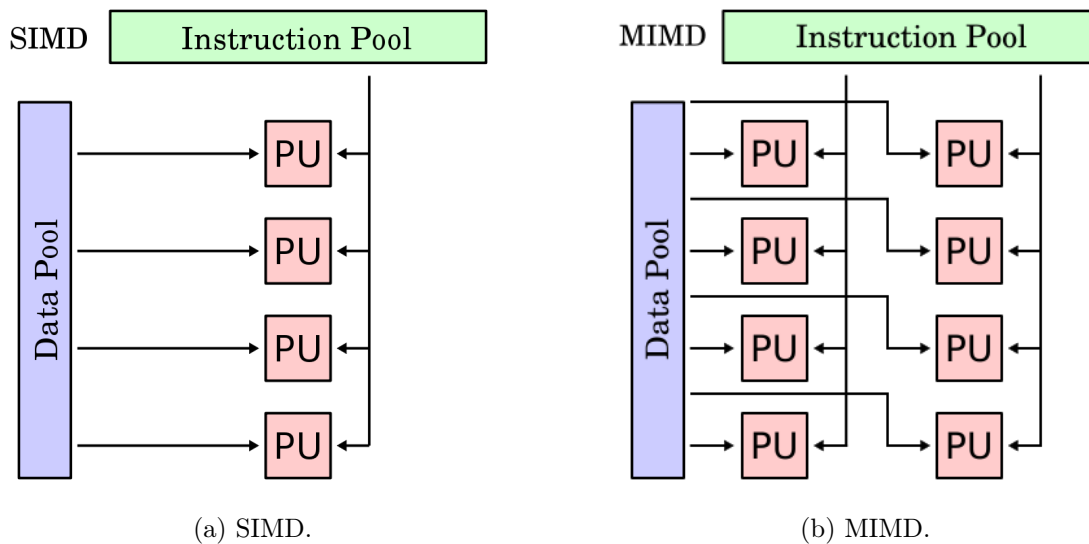


Figure 1-11: Taxonomy of Flynn (1972).

Whatever the taxonomy, we have the parallel memory architectures. Computer systems exploit the Shared-Memory and Distributed-Memory parallelism.

Shared-Memory This has multiple CPUs that share the same memory address space in one physical computer. Although the processors can operate independently, the modifications to memory are visible for every PU. This memory parallelism is divided into two classes depending on the distance from any memory location. These are Uniform Memory Access (UMA) and Non-Uniform Memory Access (NUMA). NUMA is opposite to UMA. While NUMA provides low latency

and high bandwidth to access local memory, accessing memory owned by other CPUs has higher latency and lower bandwidth performance. Instead, NUMA has the same uniform access time to any memory module in the system. Common programming techniques that apply this memory architecture are applications using OpenMP and Pthreads.

Distributed-Memory This refers to processors performing computations in the local memory and then using explicit messages to transfer data to remote processors, given that they do not have access to the same memory space. Its nature makes this parallelism NUMA and highly scalable in memory terms. Adding more CPUs increases the memory proportionately. The most common technique for Distributed-Memory is message passing. This technique manages data transfers between instances of a parallel program running on multiple processors in a parallel computing architecture. The message passing models consists of a number of processes, each working on some local data. Each process lacks the mechanism to access the memory of another directly. In order to share their data, they have to send and receive explicit messages between them. This approach is very flexible. It is left up to the programmer to explicitly divide data and work across each processor as well as manage the communications among them. This kind of parallelism is hugely penalized by the communications. The Gigabit Ethernet produces delays of around $100\mu\text{s}$, and the use of InfiniBand (a network specially designed for cluster communication) reduces the latency by 5-6.

Modern HPC systems are often a hybrid implementation of both memory concepts, as clusters have multiple computers with multiple CPUs.

In this PhD, many implementations are based on parallel infrastructures. The speedup, Equation 1.2, P being the parallel units and S the sequential code, helps us to determine the improvement in execution time of a task executed in two similar architectures with different resources, denoting how many times the parallel code is better than the sequential.

$$S_p = \frac{T_s}{T_n} \quad (1.1)$$

Otherwise the efficiency, Equation 1.2, is a meter of the size of the speedup improving per contributing unit.

$$E_p = \frac{S_p}{p} \quad (1.2)$$

Overall, the biggest drawback to HPC architectures is that the data is distributed through the network, and for Big Data applications, its CPU-bound nature. Big Data applications are characterized as I/O bound, where managing the data is the bottleneck.

1.3.2 Big Data Frameworks

A framework in Big Data is just a set of tools to facilitate the processing of datasets, the most widely-used frameworks being Apache Hadoop and Apache Spark.

Apache Hadoop

Hadoop is used for batch processing. It has two major components, as can be seen in Figure 1-12, the Hadoop distributed file system (HDFS) [3] which is used for storing data, and MapReduce [9]], used for processing data. HDFS is designed to read and store huge datasets across all the nodes, and due to its parallel nature, it provides high throughput operations. The MapReduce algorithm contains two important tasks, named Map and Reduce. Map takes a set of data and converts it into another set of data, where individual elements are broken down into tuples (key/value pairs). Secondly, reduce task takes the output from a map as an input and combines those data tuples into a smaller set of tuples with operators like aggregation/summation.

The Hadoop architecture is based on a master-slave model. The master manages, maintains and monitors the slaves, while the slaves provide the resources. To fulfill its duties, the master node runs a daemon for HDFS (Namenode) and Yarn (ResourceManager). The Namenode records the metadata of all the files stored in the cluster, the

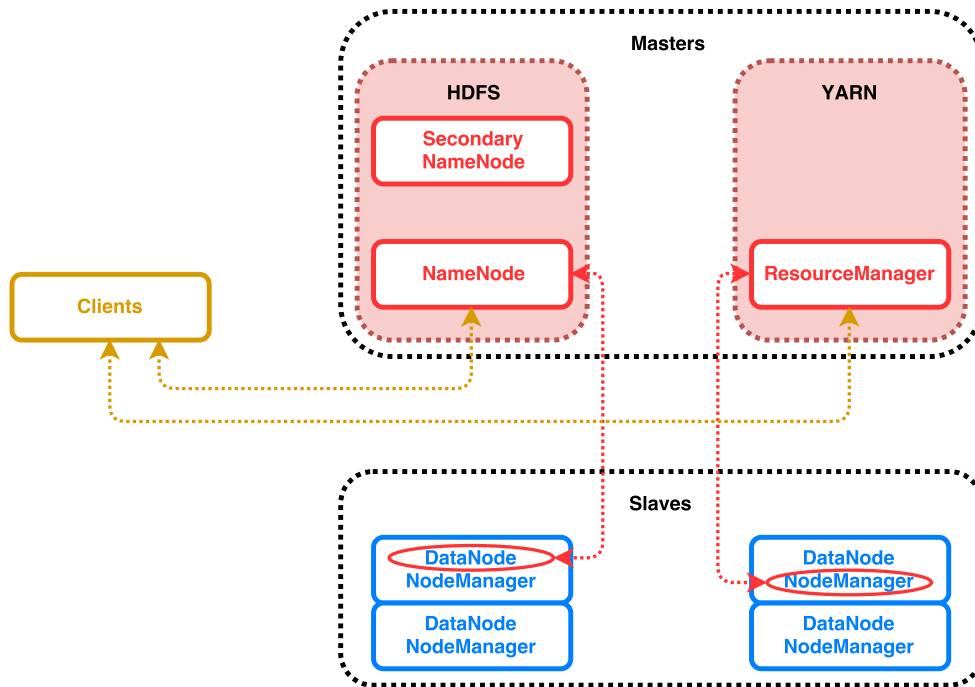


Figure 1-12: Hadoop architecture and components.

location of the blocks stored or even the size of the files. Without the NameNode, the filesystem would not work and all the files would be lost. The SecondaryNameNode is a backup node in case the main one fails. On the other side, the ResourceManager is in charge of scheduling the applications and managing the resources. The slaves run the DataNode and the NodeManager. The first is responsible for reading, writing, deleting and replicating the data, and the second is responsible for managing containers and monitoring resource utilization in each container.

The issue with Hadoop is its poor performance in real time data analytics. While Hadoop processes data on disk, Spark supports in-memory computing, and this enables it to query data much faster than disk-based engines (100x faster, as it uses the main memory).

Apache Spark

The most important thing in Apache Spark is the Resilient Distributed Datasets (RDDs). RDDs are the fundamental units of data in spark, which is a distributed collection of elements across cluster nodes and that can perform parallel operations

on distributed data. Then there is laziness (an RDD is only computed when its data is needed), and persistence (while an RDD may be accessed multiple times, it is only computed once).

The way the Spark application operates is shown in Figure 1-13. The process that runs the user code that creates RDDs, performs transformation and action, and creates SparkContext, is called the driver. When the Driver process needs resources to run jobs/tasks, it asks the ClusterManager for resources. The ClusterManager allocates the resources and uses the Workers to create Executors for the Driver. Each worker can run as many executors as the CPU cores it has. Given the flexibility of Spark, the user should specify these parameters.

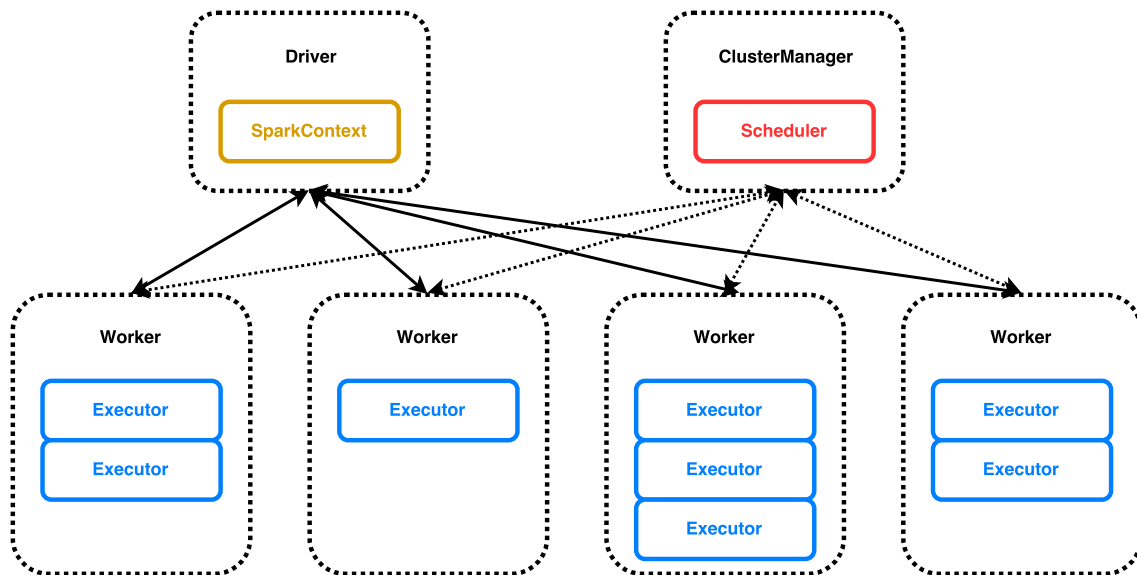


Figure 1-13: Spark architecture.

Spark is easier to develop as it knows how to operate on data, it requires no abstractions. It supports SQL queries, streaming data, machine learning, packages and graph data processing. It may be coded in a variety of languages, such as Scala, Java, Python and R.

Both approaches benefit good performance by having the processor close to the data it needs to process, so the computation is assigned to occur where the data resides.

1.4 Related Work

Sequence aligners in the literature are based on Gotoh or Myers & Miller's dynamic programming techniques, but each aligner has its basis in different scoring functions, such as a consistency library or substitution matrices like PAM and BLOSUM. Whatever the method, it allows an optimal final alignment for two sequences to be performed.

1.4.1 Multiple Sequence Alignment

The most popular progressive alignment implementation is the Clustal family, [15], especially the ClustalW weighted variant proposed by [49], which is used by a large number of biologists. However, the main drawback of these methods is that errors made in the early stages not only propagate to the final alignment but may also increase the likelihood of misalignment due to incorrect Conservation signals. To lessen the early-error propagation, iterative and consistency-based methods were proposed.

Iterative algorithms, like MUSCLE [12], MAFFT [16] and Opal [53], overcome the greediness of progressive alignment methods through a process of alignment refinement in order to optimize the final result. These approaches start with an initial solution, which is improved using iterative steps. Evolutionary and genetic algorithms are an enhancement of iterative algorithms that use a stochastic process to improve the solution. Evolutionary methods start with an initial population of individual solutions and make them evolve using crossover and mutation operations in order to select the best individual based on its fitness (alignment accuracy). A good example of such genetic algorithms are Rubber Band Technique Genetic Algorithm (RBT-GA) [48], Multiple Sequence Alignment Genetic Algorithm (MSA-GA) [13] and Vertical Decomposition Genetic Algorithm (VDGA) [29]. Recently, a few works have been implemented on multi-objective genetic based methods to cope with the different fitness functions used to optimize the alignment: Multiobjective Optimizer for Sequence Alignments based on Structural Evaluations (MO-SAStrE) [38], Multiple Sequence Alignment with Affine Gap by using Multi-Objective Genetic Algorithm

(MSAGMOGA) [18] and Hybrid Multiobjective Memetic Metaheuristic for Multiple Sequence Alignment (H4MA) [40].

Consistency-based methods use consistency information from all-against-all pairwise alignments to improve the final alignment accuracy. Gotoh introduced consistency to identify anchor points for reducing the search space of an MSA. Since then, some MSA tools based on consistency have appeared in the literature.

T-Coffee

T-Coffee [34] is the most representative method in this category. Although it can produce high alignment accuracy, it is the one with highest complexity in memory requirements and execution time. It is an MSA tool that combines the consistency-based scoring function COFFEE [33] with the progressive alignment algorithm. Furthermore, it increases the alignment accuracy by seeking consistency among a set of global and local pairwise alignments. The algorithm for aligning two sequences or two pre-aligned groups is divided into three main stages, as shown in Figure 1-14, (1) calculate the Primary Library, (2) generate Extended Library, and (3) build the final alignment:

1. *Primary Library.* The primary library is a collection of data obtained from computing all-against-all possible pairwise alignments. It can be represented by an $N \times N$ matrix (see Figure 1-15), where each cell $S_i - S_j$ when $i \neq j$ contains a list of residue matches between those sequences. Each residue match is represented by a constraint/entry $\{x, y, W_{(x,y)}\}$, x being a residue of S_i matched with y a residue of S_j and a weight $W_{(x,y)}$ representing its correctness. Each constraint list is used in the progressive alignment stage to fill the dynamic programming matrix.

The size of the CL is in the order of $O(N^2L^2)$, where N^2 is given by all the possible combinations of sequences without repetition and L^2 by the worst scenario in one pairwise (there are no matches between both sequences).

2. *Extended Library.* The extension of the library is a re-weighting process that

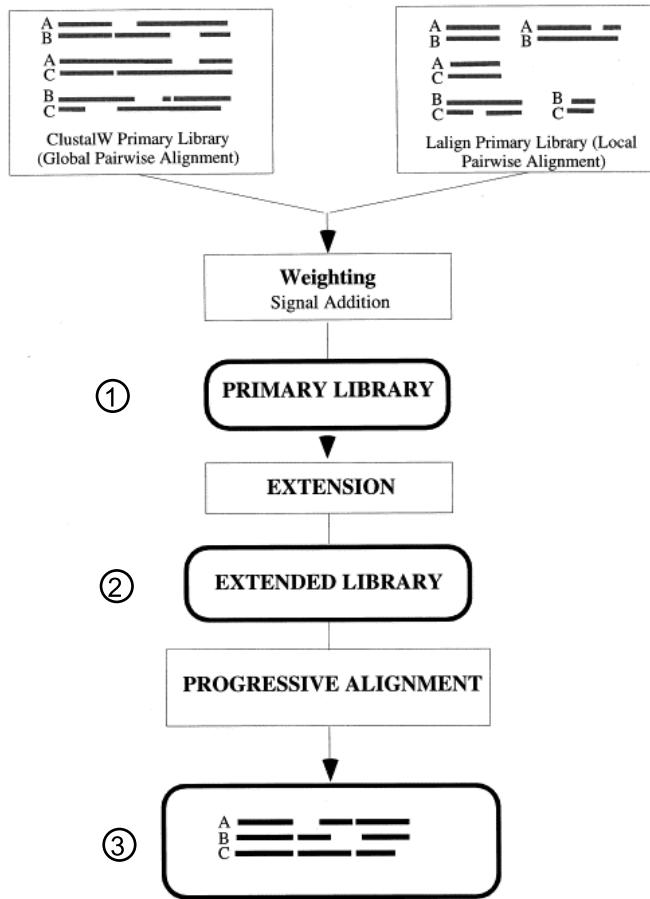


Figure 1-14: T-Coffee algorithm stages.

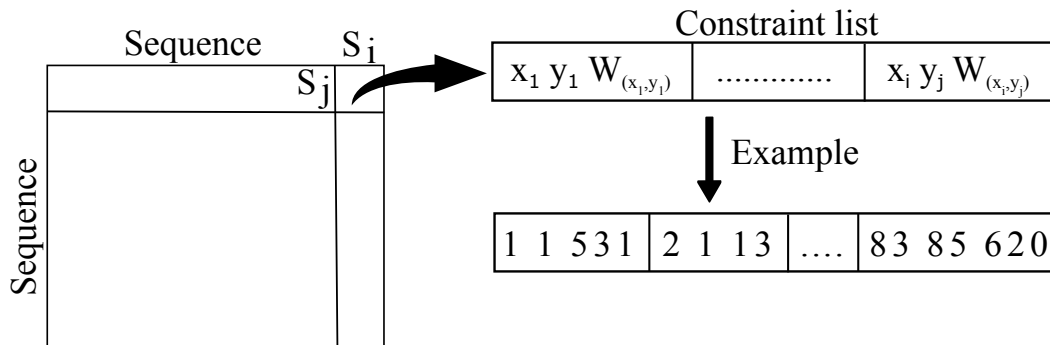


Figure 1-15: Library structure.

uses the transitivity property to include indirect information about constraints. For instance, given an MSA containing three sequences x, y, and z, if position x_i aligns with position z_k and position z_k aligns with y_j in the projected x-z and z-y alignments, then to be consistent, the x_i must align with y_j in the projected x-y alignment. Such extra information is able to provide valuable data about whether a residue is aligned in a column with other sequences, helping to mitigate the errors caused by the guide tree. The library extension is performed on-the-fly during the progressive alignment stage, as a prior extending of all the primary library would lead to many unused entries, higher memory requirements and longer execution times.

3. *Progressive Alignment strategy.* The MSA is based on the successive construction of pair-wise alignments. It starts by aligning the two most closely related sequences, and then adds sequences in the order defined by a guide tree. The guide tree is generated using a distance matrix obtained by all-against-all pair-wise alignments. In T-Coffee, the alignments are performed maximizing the COFFEE objective function that uses the weights in the extended library instead of the traditional Substitution matrix weights and gap penalties that other MSA tools use.

The main drawback of consistency-based aligners is the high computational resources (CPU and memory) required to calculate and store the consistency information. For example, the consistency library in T-Coffee has a complexity of $O(N^2L^2)$, N being the number of sequences and L their average length. Figure 1-16 represents the increase in execution time depending on the number of sequences, showing the quadratic growth of the execution time. Such requirements mean that the method is not scalable, but is limited to aligning a few hundred sequences on a typical desktop computer. On average, aligning 500 sequences requires around 8GB of memory. Therefore, these aligners are not feasible for large-scale alignments with thousands of sequences.

Newer MSA tools adopted both proposals, resulting in hybrids between iterative

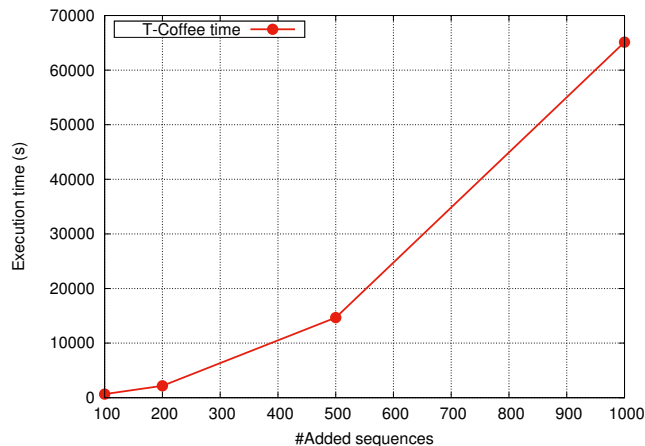


Figure 1-16: T-Coffee execution time.

and consistency-based methods. Do et al. presented ProbCons in [10]. It was a modification of the traditional sum-of-pairs scoring system that incorporates Hidden Markov Models to specify the probability distribution over all alignments between a pair of sequences. Furthermore, Subramanian et al. developed a new tool, DIALIGN-T, in [47], which formulated consistency based on finding ungapped local alignments via segment-to-segment comparisons that determine new weights using consistency. Another method based on consistency, the LINSI variant of MAFFT [16], was presented by Katoh et al. in [17]. This uses a new objective function combining the WSP score from Gotoh and the COFFEE-like score ([32]), which evaluates the consistency between a multiple alignment and pairwise alignments.

1.4.2 Benchmarking

In order to validate the correctness and evaluate the performance of an MSA tool, a reliable MSA benchmark is crucial. It should be able to represent real problems encountered when aligning sets of complex sequences. Since the emergence of the sequence alignment, many databases, such as BALiBASE [50], HomFam [44], HOMSTRAD [26], Prefab [11] and SABmark [51], have been built. These databases have usually been constructed using 3D protein structural alignments, and are thus independent of sequence alignment methods. The quality of the MSA programs has typically been assessed by an accuracy score that measures the proportion of cor-

rectly aligned residue pairs in the alignment. Each benchmark uses a different tool to compare the user alignment against a reference alignment. There is also a set of scoring schemes. The Total Column Score (TC) is a binary score that tests the ability of the programs to align all the sequences correctly, the sum-of-pairs (SP) is used to measure the degree to which the program succeeds in aligning some of the sequences in an alignment, the Q score is the number of correctly-aligned residue pairs divided by the number of residue pairs in the reference alignment, and the Column Score (CS) is the proportion of columns of residues correctly aligned between the test and reference alignments.

One of the most standardized and widely-used benchmarks for protein analyses is BALiBASE, which is specifically designed to serve as an evaluation resource to address all the problems encountered when aligning complete sequences. However, its largest dataset is rather short (150 sequences). An example of BALiBASE benchmark results is shown in Table 1.1. The first column indicates the aligner tool used in the run, the results for BALiBASE subgroupings are in columns 2–7, and finally the average score over all families is given in the eighth column.

In order to validate the use for larger sequences, HomFam provides datasets with thousands of entries using Pfam families. So as to score the results of aligning a Pfam family, the HOMSTRAD site contains some reference alignments and the corresponding Pfam family. These references are previously de-aligned and shuffled into the dataset. After the alignment process, the reference sequences are extracted and compared with the originals in HOMSTRAD.

1.4.3 Next-Generation Sequencing in bioinformatics

Comparative genomics, comparative and human medicine, as well as Multiple Sequence Alignment are challenged by the high-throughput sequencing era. The need to manage more information is forcing all the tools that solve such problems to migrate to the big-data paradigm. The problem of scalability is common to all these tools and algorithms. Therefore, recent bioinformatics tools have already taken advantage of new technologies to improve their performance and scalability. These have

Table 1.1: Accuracy comparison of MSA tools, using the SP score from BALiBASE.

<i>Aligner</i>	<i>RV11</i>	<i>RV12</i>	<i>RV20</i>	<i>RV30</i>	<i>RV40</i>	<i>RV50</i>	<i>Average</i>
MSAProbs	0.562	0.885	0.852	0.765	0.827	0.782	0.785
ProbCons	0.557	0.883	0.846	0.759	0.807	0.776	0.778
Probalign	0.537	0.883	0.844	0.750	0.826	0.767	0.774
MAFFT	0.521	0.874	0.845	0.762	0.829	0.776	0.771
ClustalΩ	0.481	0.847	0.823	0.760	0.799	0.736	0.748
T-Coffee	0.502	0.845	0.820	0.730	0.800	0.733	0.743
Muscle	0.465	0.846	0.809	0.713	0.760	0.706	0.724
Dialign-tx	0.423	0.814	0.789	0.648	0.710	0.662	0.682
ClustalW	0.415	0.798	0.773	0.636	0.696	0.649	0.669

been developed using HPC and later, the MapReduce framework and its distributed file system. The latter has demonstrated the ability to store and process petabytes of information in a timely and cost-effective manner.

There are MapReduce solutions for traditional algorithms like blast [25], CloudBlast [24], which encourage the use of the MapReduce approach for the execution of large-scale bioinformatics applications, delivering speedups of 57x against 52.4x for the MPI version with the same number of CPUs, and SparkBLAST [8], which outperforms the equivalent system implemented on Hadoop in terms of speedup and execution times given the in-memory operations.

In the area of search and mapping short reads against a reference genome, applications such as CloudBurst [43] and CloudAligner [31], implement traditional algorithms like RMAP [46] using the MapReduce paradigm. The former proposes a new parallel read-mapping algorithm optimized for mapping next-generation sequence data to the human genome and other reference genomes, achieving good speedups vs RMAP executing on a single core. Although CloudAligner achieves higher performance, the current existing MapReduce-based applications were not designed to process the long reads produced by the NGS. Besides, it is important to understand when to use MapReduce, as a wrong use of this paradigm may be counterproductive.

In this PhD we focus on MSA tools, and it is known that when the number of sequences to be aligned increases, there is an increase in the execution time and a degradation of accuracy [45]. The utilization of HPC and Big Data infrastructures

[58] has recently given computational biologist researchers an opportunity to achieve scalable, efficient and reliable computing performance on Linux clusters and cloud computing services.

Parallel implementations based on the main heuristics, such as ClustalW-MPI [21], Parallel-TCoffee [57], were implemented using the MPI standard, rebuilding its stages in order to be executed on distributed memory machines. Another approaches have used Compute Unified Device Architecture (CUDA) to reduce the execution time of MSA applications, a parallel computing platform to build general-purpose applications for graphic processing units. MSA-CUDA [22] or GPU-ClustalW [23] are examples of such applications. Although all of these approaches improve their original algorithm, they exhibit scalability problems when the number of sequences increases. These are due to data dependencies and memory requirements.

Newer solutions use Hadoop to surpass such limitations. Sadasivam [41] proposes a novel approach that combines the dynamic programming algorithm with the computational parallelism of Hadoop data grids to improve accuracy and accelerate Multiple Sequence Alignment. They get profit of the principle of block splitting in Hadoop which, coupled with its scalability, facilitates the aligning of large sequences.

Spark has also emerged as an enabling technology for large-scale MSAs. Wiewiórka [54] developed SparkSeq, a general-purpose tool used to build genomic analysis pipelines in Scala and run them in an interactive way, tuned for processing big alignment data in the cloud with nucleotide precision. It combines the Picard Java Development Kit for Sequence Alignment/Map format (SAM) (Picard SAM JDK) via Hadoop-BAM library and Apache Spark to introduce versatile sequencing analyses into the MapReduce environment.

Zhao [56] developed SparkSW, which can carry out the Smith-Waterman algorithm, a dynamic programming algorithm for local sequence alignment. As shown in Figure 1-17, it takes advantage of the HDFS to store the input FASTA database in multiple blocks, and uses Apache Spark RDDs to do the processing. The Map task splits the protein database in order to obtain multiple pairwise alignments simultaneously, and score them against a substitution matrix. Finally, it filters the higher

scored pairwise using transformation functions provided by Spark. This study reveals that the Apache Spark framework provides an efficient solution to facilitate dealing with the ever-increasing size of biological sequence databases.

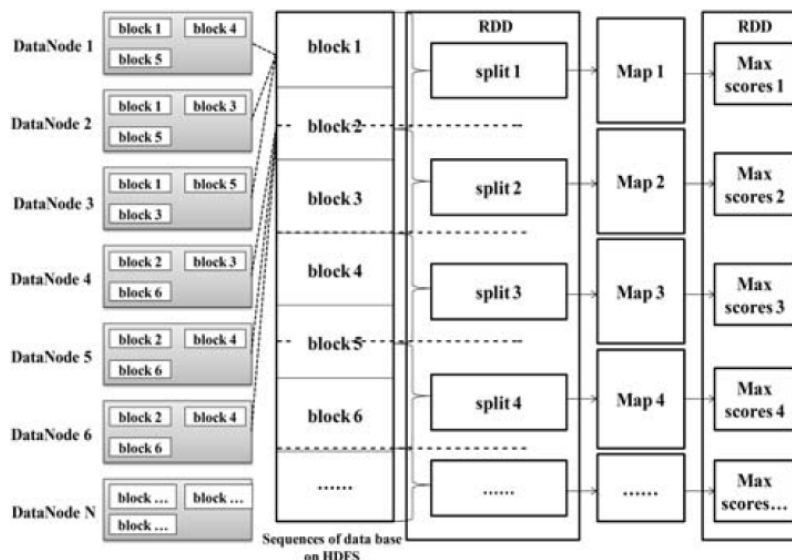


Figure 1-17: The Map process in SparkSW.

PASTASpark [1] adapts the PASTA (Practical Alignments using SATé and TrAnsitivty) MSA tool to use Big Data frameworks. The PASTA iterative workflow consists of four steps or phases. In the second phase, which is the most CPU-intensive and time consuming, the subsets are aligned by default using MAFFT. Due to this, PASTASpark focuses on parallelizing this step. In particular, it uses Spark RDDs as a task scheduler to invoke MAFFT, where each map task encompasses the aligner parameters and the subset to be aligned. The main advantage over the original PASTA is that it reduces the I/O cost significantly.

Overall, one of the most significant MSA tools in the literature is T-Coffee. Its accuracy is better than the average and the consistency approach is a good alternative over the standard progressive alignment, although it has flaws in its scalability and execution times. Focusing on the NGS, the more numerous and larger the sequences are, the more errors may be produced by the guide tree. Consistency methods are proven to be less susceptible to such errors, and scaling them may offer this strength to the alignment. From the research point of view, T-Coffee is a good tool to validate

our algorithms and methods, given its use of consistency as a whole library, which leads to the generation and processing of huge amounts of data. With the use of novel methods and algorithms, and taking advantage of newer Big Data frameworks, we propose multiple enchantments to consistency-based methods, in order to improve their scalability and performance from the computational and biological point of view.

1.5 Document structure

The rest of this PhD dissertation is organized as follows. The chapter 2 presents the starting point of the work and states the objectives and milestones that have to be achieved. Also it presents an introduction to the research methodology used. In chapter 3, I present the most relevant publications submitted during this PhD. Each publication is related to one of the milestones presented in Chapter 2. Next, chapter 4 presents a global discussion of the results obtained for each publication, and finally, chapter 5 presents the principal conclusions extracted from the PhD and advances some of the possible future research lines.

Chapter 2

Methodology

The problem of Multiple Sequence Alignment with a consistency-based library has been studied for many years in the literature. Due to this, there are many different tools that try to solve such problems. Among them, I decided to use T-Coffee as the test application, where I will implement and verify the contributions. Nevertheless, the new improvements and methods proposed in this PhD may be used and implemented on other aligners.

MSA is a heuristic problem. Therefore, there are many parameters and functions to consider, such as the residue substitution matrices, the gap penalties, the generation of the guide trees, etc. Recent methods for those functions should be studied, and if they can improve the performance, implemented. Furthermore, in order to deal with the complexity of this kind of problem, I focus on (1) reducing its complexity, optimizing the objective function and other time/memory consuming functions, like the guide tree calculation, or (2) maintaining the complexity, parallelizing the problem, and adding more resources to the computing infrastructure.

This PhD looks at those approaches, trying to optimize the current functions to increase their performance, and using new technologies in the field of distributed computing to take advantage of parallelism when generating an alignment.

This chapter presents the problem statement along with the starting point for the PhD. Then, the general objective of the PhD is framed, followed by the milestones that have been considered necessary and finally the research methodology used.

2.1 Problem Statement

Few years ago, the Distributed Computing Research Group at the University of Lleida (GCD) became interested in sequence alignment, since it is a complex computational problem and a challenge in the field. The research in this area has produced two PhDs in the university:

- Montañola [27] contributed to reducing the computation time and memory usage of the MSA T-Coffee, adapting it to work with threads rather than processes. He also designed a parallel pairwise method, with an efficient mapping of sequences to nodes, and finally a method to determine the minimal amount of system resources required to solve a problem of a determined size.
- Orobitg [37] proposed three approaches to reduce some limitations of the MSA methods. The first aimed to improve the parallelism degree of the progressive alignment stage, generating more balanced guide trees, resulting in faster execution times and small accuracy variances. The second focused on reducing the size of the consistency library by a percentage in order to reduce the memory consumption, although the library maintained its quadratic growth. Finally, the last proposal was based on generating multiple guide trees with slight variations for a sequence to be aligned and thus generate several alignments in parallel, finally to output the one that achieved a greater scoring metric.

This PhD tackles some challenges that matches the future work proposed by Orobitg, like large-scale aligners, which are indispensable taking into consideration the huge volume of data generated nowadays, the use of consistency to guarantee a minimal accuracy over distributed memory environments, and the overall adaptation to new parallel paradigms like MapReduce and CUDA.

2.2 Main Objective

The main objective of this PhD is:

The study and proposal of new methods and tools to enable scalability into consistency-based MSA tools, aligning bigger datasets, and improving their overall performance (requirements and accuracy).

The proposed methods aim to provide better performance for the application. On the one hand, I mainly focused on building and accessing a scalable consistency library, one that grows quadratically in memory requirements needs quite a tune up to fit in memory or a change in the paradigm. The latter implies lowering one level in the memory hierarchy (Figure 2-1) from the main memory to mass storage. Thus, the capacity can be enhanced by the order of terabytes. Although the cost of increasing the capacity has repercussions on the access times, DRAM access time is around 50ns, while accessing a hard disk requires around 10ms (a difference of an order of magnitude of 5), increasing the total execution time exponentially. Therefore, the way the aligner accesses the consistency needs to be redesigned, providing new mechanisms to reduce the high latencies of the secondary memory. On the other hand, the biological aim of the tool is taken into account. Having results in a reasonable amount of time is important, but accuracy is as important. The internal functions and algorithms used by MSAs analyzed to improve its accuracy with the most representative benchmarks.

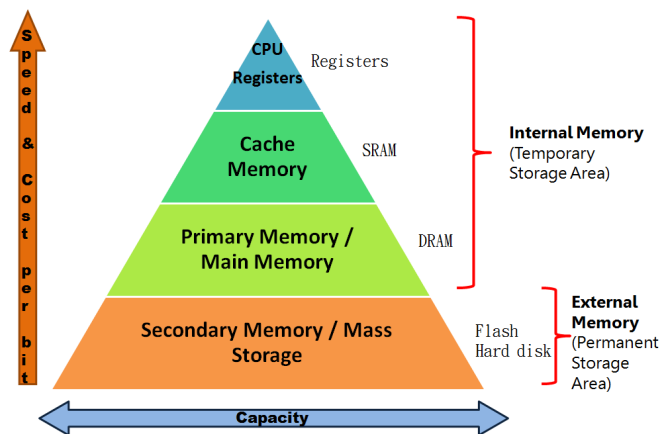


Figure 2-1: Memory hierarchy in computer architecture.

2.3 Milestones

In order to achieve the proposed main scalability objectives, there are two possible approaches: 1) reduce its size in order to fit the computer memory or 2) retain all the data on distributed storage, where we can attain huge capacities. Scaling this problem directly affects the accuracy of the final alignment. Therefore, the method should be refined to maintain its accuracy when aligning bigger datasets.

The main objective has been divided into specific milestones. These milestones are organized in a way that allows us to understand and analyze the problem. It can be summarized as; a previous study of the art and related works, the generation of a consistency library with fixed sizes, and also the adaption of its generation and allocation to a MapReduce paradigm, to be able to execute, store and access large libraries in a distributed Hadoop infrastructure and finally, I aim to improve the accuracy of the alignment by incorporating current techniques into different stages of the alignment.

M0. Exhaustive study of the state of the art in the field of consistency-based aligners.

Prior to designing and building our methods, it is essential to understand the current approaches in the literature that solve the MSA problem with consistency. There are many techniques and each of these has its downsides and benefits. We also have to be aware of the most innovative solutions in data processing tools that have appeared lately.

M1. Study and proposal of a new consistency-based library with a linear complexity in space.

One of the most important stages of a consistency-based MSA tool is to build the primary library. All the method relies on it to generate a high quality alignment. One way to enable the scalability is to delimit the library size to the computer's main memory. To this end, we must select the most important constraints in the library and discard the rest. In this case, the method which

selects the information is critical, as the accuracy of the final alignment is heavily affected by it. As the basis of the method, it is really important to understand its behavior, so an in-depth study was carried out to be able to select the most important parts of the library and to study how reducing its size could affect the final alignment accuracy.

M2. Migrating the consistency library to massive data-processing frameworks.

It is known that generating consistency is quite expensive in time. The first idea that we had, given its embarrassing parallel nature, was to migrate all these calculations to a cluster-like architecture to boost the execution time. However later, when newer technologies appeared for processing high amounts of data, we focused on building a scalable primary library using Big Data frameworks with MapReduce techniques and storing them on a Hadoop distributed file system. In order to use such data for real time alignment, it was decided to move the consistency from HDFS to a distributed database. This allows the aligner to access the library using an SQL query like system. Finally, a two layer in-memory cache was designed to speed up the library accesses, and the database queries were also optimized to minimize the latency and improve the efficiency.

M3. Validate algorithms that improve consistency-based aligners accuracy.

It is well known that, from the point of view of computer science, the scalability and the execution time values are really important. However, from the point of view of biologists, the main aim of the algorithms is to provide the highest possible accuracy. However, these different goals can be contradictory as computer resources are finite and the problem to solve is extremely hard. In this final milestone, a new objective function is proposed, and evaluated over T-Coffee, in order to guarantee a minimum alignment quality even when high library reductions are applied.

2.4 Research Methodology

In computer science, there is no standard defined for the research methodology, as this field is composed of a merger of different scientific and engineering fields, each with its own scientific method. That is why many scientific methods have been proposed to deal with its complex nature. This PhD work follows the directives of the hypothetico-deductive method, adapted to the computer science field as proposed by Adrion [2]. In this method, research is divided into 4 different parts that can be repeated depending on the results.

1. **Observe the existing solutions.**

This part consists of an in-depth analysis of the related work done in the same or similar fields. It is necessary in order to avoid working on a problem that has already been solved and to produce high-quality, up-to-date research. Moreover, the study of other scientific fields can provide new ideas to be applied in the field. I analyzed the MSA tools presented in the field and the novel techniques used in computer science to improve the scalability and reduce the execution time of computational problems, and obtain a great deal of information on how to treat the problem with different infrastructures and application requirements.

2. **Propose better solutions.**

With the wide knowledge gained during the study of the related work, it was time to analyze the existing solutions for the purpose of improving them. In this part, some techniques in the literature were also adapted to consistency-based MSA tools, and innovative improvements to increase the performance of our proposals were proposed.

3. **Build or develop new solutions.**

In this part, new consistency-based alignment techniques were proposed and implemented. When problems appear, step 2 was repeated to find a new solution. This step can also lead us to repeat step 1 to analyze how a specific

problem is solved in the literature. This part is where the proposed techniques are implemented.

4. Measure and analyze the new solution.

Finally, the solutions are tested and evaluated with the literature proposals in order to compare the results. If the results are not good enough, the previous steps are repeated until the solutions are improved.

Chapter 3

Papers

In this chapter, the most representative papers for achieving the PhD objective are presented. Each of the papers is related with the milestones presented above.

3.1 Recovering accuracy methods for scalable consistency library

In this paper, a first attempt at reducing the memory requirements of the consistency library is presented. The proposed method is able to determine which constraints should be discarded from the library, and thus fix its size to a desired amount. A drastic reduction of the library sharply decreases its accuracy. Therefore, I propose three techniques to reduce the accuracy loss.

This proposal is able to attain a scalable library which is related to milestone M1 in the PhD.

3.1.1 Contributions to the state of the art

The following paper proposes the Bound Library Method (BLM) which enables scalability in a consistency-based MSA tool. The method has the total amount of available memory to store the consistency library as an input parameter and determines whether each new constraint must be stored in function of its score. It is able to

achieve scalability, although there is a negative impact on accuracy. To reduce this degradation of accuracy and attain a better alignment, three innovative approaches were implemented in T-Coffee. The first approach can only be used when there are still some residues not directly related to the pair of constraints that are being computed (Related consistency). Otherwise, the other two methods are only used when there is no information at all (Dynamic and static substitution matrix). Those three methods can be used alone or combined with each other.

3.1.2 Paper 1: Recovering accuracy methods for scalable consistency library

The paper presented can be found in the following publication:

Authors: Jordi Lladós, Fernando Guirado, Fernando Cores, Josep Lluís Lèrida and Cedric Notredame

Title: Recovering accuracy methods for scalable consistency library

Journal: Journal of Supercomputing

Volume: 71 **Issue:** 5 **Pages:** 1833–1845

Year: 2015

Impact Factor(SCI/SSHI/AHCI): 1.088 **SJR:** 0.456

Quartile and Subject(SCI/SSHI/AHCI): Computer Science, Theory & Methods, Q2 (47 of 105) **Quartile and Subject(SJR):** Software, Q2 (146 of 333)

ISSN: 1573-0484

DOI: 10.1007/s11227-014-1362-z

Recovering accuracy methods for scalable consistency library

Jordi Lladós · Fernando Guirado ·
Fernando Cores · Josep Lluís Lériða ·
Cedric Notredame

Published online: 31 December 2014

© The Author(s) 2014. This article is published with open access at Springerlink.com

Abstract Multiple sequence alignment (MSA) is crucial for high-throughput next generation sequencing applications. Large-scale alignments with thousands of sequences are necessary for these applications. However, the quality of the alignment of current MSA tools decreases sharply when the number of sequences grows to several thousand. This accuracy degradation can be mitigated using global consistency information as in the T-Coffee MSA-Tool, which implements a consistency library. However, consistency-based methods do not scale well because of the computational resources required to calculate and store the consistency information, which grows quadratically. In this paper, we propose an alternative method for building the consistency-library. To allow unlimited scalability, consistency information must be discarded to avoid exceeding the environment memory. Our first approach deals with the memory limitation by identifying the most important entries, which provide better consistency. This method is able to achieve scalability, although there is a negative impact on accuracy. The second proposal, aims to reduce this degradation of accuracy, with three different methods presented to attain a better alignment.

J. Lladós (✉) · F. Guirado · F. Cores · J. L. Lériða
Department of Computer Science, Universitat de Lleida, Lleida, Spain
e-mail: jordi.llados@diei.udl.cat

F. Guirado
e-mail: f.guirado@diei.udl.cat

F. Cores
e-mail: fcores@diei.udl.cat

J. L. Lériða
e-mail: jlerida@diei.udl.cat

C. Notredame
Comparative Bioinformatics Group, Center for Genomic Regulation, Barcelona, Spain
e-mail: cedric.notredame@crg.es

3.2 PPCAS: implementation of a Probabilistic Pairwise model for Consistency-based multiple alignment in Apache Spark

In the following paper, a MapReduce approach to dealing with the probabilistic pairwise model for consistency-based methods is proposed. This is aimed at parallelization over Big Data infrastructures. Many highly-rated MSA tools, such as MAFFT, ProbCons and T-Coffee (TC), use the probabilistic consistency as a prior step to the progressive alignment stage in order to improve the final accuracy.

Contrary to the previous work, where we focused on reducing the library size to improve the computational performance, we implemented a scalable stand-alone tool to build the whole library relying on Apache Spark. The consistency library may be cataloged as embarrassingly parallel, as it evaluates $N * (N - 1)/2$ combinations, N being the number of sequences, where each of these combinations has no dependencies on the others. With the advent of the Next-Gen Sequencing, the number and length of the sequences to be aligned have grown exponentially, with the corresponding negative impact on execution time and memory requirements. The use of massive data-processing techniques can provide a solution to these limitations.

This proposal is able to generate massive consistency libraries in feasible times, which is related to milestone M2 in the PhD.

3.2.1 Contributions to the state of the art

The paper proposes PPCAS, an implementation of a probabilistic pairwise model for consistency-based multiple alignment in Apache Spark, which enables the scalability of the primary library computation.

Adapting algorithms to the MapReduce paradigm used in the Big Data frameworks is not trivial, and the programming languages being interpreted are much slower than compiled ones. Thus, the use of Python with the Ctypes extension was selected. This provides C language compatibility data types and also the ability to

call on external shared libraries. PPCAS is able to define all the task combinations and distribute them in a balanced way among the distributed workers (executors). Each executor has to perform a subset of the pairwise combinations and generate its library, and finally save it in the HDFS file system.

3.2.2 Paper 2: PPCAS: implementation of a Probabilistic Pairwise model for Consistency-based multiple alignment in Apache Spark

The paper presented can be found in the following publication:

Authors: Jordi Lladós, Fernando Guirado and Fernando Cores

Title: PPCAS: implementation of a Probabilistic Pairwise model for Consistency-based multiple alignment in Apache Spark

Book series: Lecture Notes in Computer Science

Volume: 10393 **Pages:** 601-610

Year: 2017




SJR: 0.295

Quartile and Subject(SJR): Computer Science, Q2 (2 of 16)

ISBN: 978-3-319-65482-9

DOI: 10.1007/978-3-319-65482-9_45

PPCAS: Implementation of a Probabilistic Pairwise Model for Consistency-Based Multiple Alignment in Apache Spark

Jordi Lladós^() , Fernando Guirado , and Fernando Cores 

INSPIRES Research Center, Universitat de Lleida, Jaume II, 69, 25001 Lleida, Spain
{jordi.llados,f.guirado,fcores}@diei.udl.cat

Abstract. Large-scale data processing techniques, currently known as Big-Data, are used to manage the huge amount of data that are generated by sequencers. Although these techniques have significant advantages, few biological applications have adopted them. In the Bioinformatic scientific area, Multiple Sequence Alignment (MSA) tools are widely applied for evolution and phylogenetic analysis, homology and domain structure prediction. Highly-rated MSA tools, such as MAFFT, ProbCons and T-Coffee (TC), use the probabilistic consistency as a prior step to the progressive alignment stage in order to improve the final accuracy. In this paper, a novel approach named PPCAS (Probabilistic Pairwise model for Consistency-based multiple alignment in Apache Spark) is presented. PPCAS is based on the MapReduce processing paradigm in order to enable large datasets to be processed with the aim of improving the performance and scalability of the original algorithm.

Keywords: Multiple Sequence Alignment · Consistency · Spark · MapReduce

1 Introduction

The probabilistic pairwise model [10] is an important step in all consistency-based MSA tools. A probabilistic model can simulate a whole class of objects, assigning an associated probability to each one. In the multiple alignment field, the objects are defined as a pair of residues from the input set of sequences, and the associated weight is the probability of being aligned [14]. For any two sequences, there are many possibilities of residue matches, $Length(sequence_1) * Length(sequence_2)$. The probabilistic model assigns each residue match a score. The higher this is, the better. For a complete dataset of sequences, the collection of the all the residue matches, which implies all the pairs of sequence evaluations, is known as the Consistency Library. This library is used to guide the progressive alignment and thus improve the final pairwise accuracy. A well-known MSA tool that uses consistency is T-Coffee [3].

The computation of the consistency library evaluates $N * (N - 1)/2$ combinations, N being the number of sequences, and that may be cataloged as

3.3 Scalable Consistency in T-Coffee through Apache Spark and Cassandra database

In this paper, a Big Data version of T-Coffee for performing large-scale alignments is presented. The application integrates consistency information through the Cassandra database, previously generated by the MapReduce processing paradigm, in order to enable large datasets to be processed with the aim of improving the performance and scalability of the original algorithm.

Using the secondary storage to save the consistency provides sufficient capacity to hold constraints for thousands of sequences, although accessing raw data from a disk may be fatal for access times. We plan to use a distributed NoSQL database to integrate consistency into T-Coffee efficiently.

This proposal is able to use consistency libraries in secondary storage generated by Big Data frameworks, which is related to milestone M2 in the PhD.

3.3.1 Contributions to the state of the art

The following paper proposes Big Data T-Coffee (BDT-Coffee), a tool able to produce a quality alignment relying on Spark to build the consistency library and integrating it into the aligner through Apache Cassandra.

The use of an intermediary to access the library, in this case a database, fits the problem perfectly. The progressive alignment stage requests all the constraints that are aligned with the same sequence-residue, which can be stored with the same key and retrieved together when needed. Furthermore, Cassandra is able to map together the entries with the same keys, which provides good data locality. The problem resides in the number of accesses, a small dataset of 4 sequences with an average length of 90 residues has its execution time multiplied by 100, from 0.02 seconds to 2 seconds when querying from Cassandra. This was due a high number of repeated queries in the progressive alignment stage. In order to avoid repeated queries to Cassandra, it was decided to implement a memory cache to hold the consistency. Thus, the second

and subsequent times that a primary-library data related to a sequence-residue is needed by the program, it may already be in the memory. However, if the library is bigger than the main memory of the node running T-Coffee, a replacement policy is used. finally, we also re-implemented the Cassandra keyspace to be query friendly, adding together multiple residues from the same sequence to reduce the number of queries, but enlarging its size.

3.3.2 Paper 3: Scalable Consistency in T-Coffee through Apache Spark and Cassandra database

The paper presented can be found in the following publication:

Authors: Jordi Lladós, Fernando Cores and Fernando Guirado

Title: Scalable Consistency in T-Coffee through Apache Spark and Cassandra database

Journal: Journal of Computational Biology

Volume: 25 **Issue:** 8 **Pages:** 894-906

Year: 2018

Impact Factor(SCI/SSHI/AHCI): 1.032 **SJR:** 1.257

Quartile and Subject(SCI/SSHI/AHCI): Statistics & Probability, Q2 (60 of 124) **Quartile and Subject(SJR):** Computational Theory and Mathematics, Q1 (21 of 123)

ISSN: 1066-5277

DOI: 10.1089/cmb.2018.0084

Scalable Consistency in T-Coffee through Apache Spark and Cassandra database

Jordi Lladós^{1*}, Fernando Cores¹, Fernando Guirado¹

INSPIRES Research Center, Universitat de Lleida.

Jaume II. 69, 25001 Lleida, Spain

¹{jordi.llados, fcores, f.guirado}@diei.udl.cat

Abstract. Next-generation sequencing (NGS), also known as high-throughput sequencing, has increased the volume of genetic data processed by sequencers. In the bioinformatic scientific area, highly-rated Multiple Sequence Alignment (MSA) tools, such as MAFFT, ProbCons and T-Coffee (TC), use the probabilistic consistency as a prior step to the progressive alignment stage in order to improve the final accuracy. However, such methods are severely limited by the memory required to store the consistency information. Big-data processing and persistence techniques, are used to manage and store the huge amount of information that are generated. Although these techniques have significant advantages, few biological applications have adopted them. In this paper, a novel approach named BDT-COFFEE (Big Data Tree-based Consistency Objective Function For alignment Evaluation) is presented. BDT-COFFEE is based on the integration of consistency information through Cassandra database in TC, previously generated by the MapReduce processing paradigm, in order to enable large datasets to be processed with the aim of improving the performance and scalability of the original algorithm.

Key words: MSA, T-Coffee, Hadoop, Spark, Cassandra, large-scale alignments

1 Introduction

The construction of MSA from individual sequences is essential for a wide range of applications in bioinformatics (Chatzou *et al.*, 2015). MSAs are fundamental for nearly all aspects of post-genomic biological research. In addition to the role that the MSAs play in advancing our understanding of the evolution and diversity of life, they also provide a platform on which algorithms that predict protein structure and function can be based. Due to this, automatic high-quality MSAs are crucial to guaranteeing the reliability and success of such studies. However, given that the best computational match cannot correspond to the best biological meaning, it is well known that the problem leads to NP-hard (Wang L, 1994).

The NGS revolution has drastically reduced the time and cost requirements for sequencing large genomes, producing massive amounts of data. So, for large-scale analysis, such technical issues as speed and scalability also become important parameters (Muller *et al.*, 2009). High-throughput comparative analyses require automated and fast pipelines that include numerous MSAs as the starting point for structural and functional studies (Thompson and Poch, 2006) and phylogenomic approaches (Dunn *et al.*, 2008). However, the larger and longer the sequence datasets to align are, the higher is the error introduced and the lower the alignment accuracy. Some methods allow computation of larger data sets while sacrificing quality, and others produce high-quality alignments, but scale badly with the number of sequences and require huge amounts of time to provide the solution (Sievers *et al.*, 2013).

3.4 Accurate consistency-based multiple sequence alignment reducing the memory footprint

Unlike the previous papers, where the optimizations were mainly focused on increasing the performance and scalability of consistency-based methods, with the use of policies that reduce the primary library or migrating the algorithm to Big Data frameworks. In this paper, we redefine the dynamic programming behavior by adding a secondary objective function, which, in conjunction with reduced libraries, aims to produce highly-rated alignments in less time. This proposal was implemented and evaluated on the T-Coffee MSA tool.

This is related to milestone M3 in the PhD.

3.4.1 Contributions to the state of the art

The following paper proposes a Matrix-Based secondary objective function, evaluated on the T-Coffee MSA tool, and named MBT-Coffee. The proposal incorporates the benefits of two different kinds of objective functions, matrix and consistency based.

The addition of a secondary objective function based on a matrix is able to boost the overall accuracy and guarantees a minimal alignment accuracy when a drastic library reduction is required. This procedure has been integrated into the progressive alignment stage, where the closest sequences are aligned following the order of the guide tree. Such alignment is achieved by the use of a dynamic programming algorithm, which builds a matrix to generate the alignment path using a backtracking technique. This matrix was originally filled by the COFFEE function, which resulted in many empty cells when a reduction was triggered. Our proposal is able to manage this by returning an average with both objective functions, which ensures that each cell will have a value, and the backtrack will be able to decide correctly whether it has to insert a gap or align the residues.

3.4.2 Paper 4: Accurate consistency-based multiple sequence alignment reducing the memory footprint

The paper presented can be found in the following publication:

Authors: Jordi Lladós, Fernando Cores, Fernando Guirado and Josep L. Lèrida

Title: Accurate consistency-based multiple sequence alignment reducing the memory footprint

Journal: Computer Methods and Programs in Biomedicine

Volume: 208 **Issue:** C **Pages:** 106237

Year: 2021

Impact Factor(SCI/SSHI/AHCI): 5.428 **SJR:** 0.924

Quartile and Subject(SCI/SSHI/AHCI): Computer Science, Theory & Methods, Q1 (13 of 110) **Quartile and Subject(SJR):** Computer Science Applications, Q1 (128 of 638)

ISSN: 0169-2607

DOI: 10.1016/j.cmpb.2021.106237



Accurate consistency-based MSA reducing the memory footprint

Jordi Lladós*, Fernando Cores, Fernando Guirado, Josep L. Llérida

INSPIRES Research Center, Universitat de Lleida, Jaume II, 69, 25001 Lleida, Spain

ARTICLE INFO

Article history:

Received 13 December 2020

Accepted 8 June 2021

Keywords:

Multiple sequence alignment

Consistency

T-coffee

Dynamic programming

ABSTRACT

Background and Objective: The emergence of Next-Generation sequencing has created a push for faster and more accurate multiple sequence alignment tools. The growing number of sequences and their longer sizes, which require the use of increased system resources and produce less accurate results, are heavily challenging to these applications. Consistency-based methods have the most intensive CPU and memory usage requirements. We hypothesize that library reductions can enhance the scalability and performance of consistency-based multiple sequence alignment tools; however, we have previously shown a noticeable impact on the accuracy when extreme reductions were performed. **Methods:** In this study, we propose Matrix-Based T-Coffee, a consistency-based method that uses library reductions in conjunction with a complementary objective function. The proposed method, implemented in T-Coffee, can mitigate the accuracy loss caused by low memory resources. **Results:** The use of a complementary objective function with a library reduction of $\geq 30\%$ improved the accuracy of T-Coffee. Interestingly, $\geq 50\%$ library reduction achieved lower execution times and better overall scalability. **Conclusions:** Matrix-Based T-Coffee benefits from accurate alignments while achieving better scalability. This leads to a reduction in memory footprint and execution time. In addition, these enhancements could be applied to other aligners based on consistency.

© 2021 The Author(s). Published by Elsevier B.V.

This is an open access article under the CC BY-NC-ND license (<http://creativecommons.org/licenses/by-nc-nd/4.0/>)

1. Introduction

Multiple sequence alignment (MSA) is important in several research domains in molecular biology and bioinformatics, such as epidemiology, phylogenetic tree reconstruction, 3D structure prediction, and hidden Markov modelling (HMM). These fields use MSA to infer residue-level homology, structural, or functional identity [1,2].

The optimal alignment of two sequences can be performed using the Needleman-Wunsch (NW) algorithm and dynamic programming [3,4]. All the best possible alignments are found by completing the Dynamic Programming (DP) matrix according to three score parameters: match, mismatch, and gap scores, which are obtained using substitution matrices such as PAM and BLOSUM. Next, the algorithm generates the optimal path using a traceback technique. The time complexity for this algorithm grows exponentially with the number of sequences and their lengths; therefore, heuristic algorithms are required. In this case, the goal of MSA tools is to seek an alignment that maximizes its accuracy, as approximated

by the sum of the similarities for all pairs of sequences (SP score) or the Total Column (TC) score.

Progressive alignment is a widely used heuristic. This process builds a final MSA by combining the pairwise alignments. It starts by using the most similar pair of sequences and ends with the more distantly related, following the order of a guide tree. The most popular progressive alignment implementation is the Clustal family; ClustalW [5] and Clustal Ω [6] are the most representative approaches. The biggest drawback of this method is that if an error is made in the initial steps of the alignment, this is propagated until it reaches the root of the tree. Therefore, other approximations based on the progressive method have appeared, such as iterative algorithms or consistency-based methods. In iterative algorithms, such as MUSCLE [7], MAFFT [8], and ProbCons [9], the greediness of the progressive alignment method is overcome through a process of alignment refinement that optimizes the obtained result. Evolutionary and genetic algorithms [10–12] are an enhancement of the iterative algorithms that use a stochastic process to improve the final solution. Alternatively, consistency-based methods use consistency information about different pairwise alignments to improve the result. T-Coffee (TC) [13] is the most representative method in this category. It combines the COFFEE consistency-based scoring function [14] with the progressive alignment algorithm.

* Corresponding author.

E-mail addresses: jordi.llados@udl.cat (J. Lladós), fernando.cores@udl.cat (F. Cores), fernando.guirado@udl.cat (F. Guirado), josepluis.lleirida@udl.cat (J.L. Llérida).

Chapter 4

Global discussion of the results

In this chapter, the results and conclusions of each paper presented in Chapter 3 are discussed.

4.1 Recovering accuracy methods for scalable consistency library

In order to build a scalable consistency library, a reduced version of the library with a heuristic criteria named the Bound Library Method (BLM) is proposed. Its goal is to maintain the best consistency information while drastically limiting the size of memory that the library may use. The main drawback of hugely delimiting the library resides in lower accuracy results, as important constraints to finding the correct path in the progressive alignment may be discarded. To overcome the accuracy loss, it was decided to propose methods to recover it by using information about the same residue positions in other sequences, a global consistency matrix and substitution matrices.

The validation was done using the BAliBASE and HomFam benchmarks, the former to test accuracy and the latter for scalability. We show that the BLM is able to reduce the number of entries used by TC, maintaining them at a certain level. Regarding accuracy, BAliBASE shows more decay the smaller the library is, but the recovery methods are able to restore its degradation significantly.

In scalability terms, maintaining a fixed amount of constraints greatly enhances the number of sequences that are aligned. With 2GB of main memory, TC was able to align 400 sequences, and TC-BLM, 1,400. Furthermore, execution times are reduced. Fewer constraints in the library implies fewer computations in the on-the-fly extension.

4.2 PPCAS: implementation of a Probabilistic Pairwise model for Consistency-based multiple alignment in Apache Spark

Generating the probabilistic pairwise model for consistency-based tools with large datasets is a CPU- and memory-intensive task. Its performance on a desktop computer is very limited and time consuming. In this paper, a novel approach named PPCAS (probabilistic pairwise model for consistency-based multiple alignment in Apache Spark) is presented. Its goal is to produce a quality library relying on a Hadoop infrastructure with Spark, using the MapReduce paradigm, able to generate a library of thousands of sequences with the same memory requirements, and improving performance as more nodes are added to the infrastructure.

To assess the good functioning of PPCAS, various experimental studies were done. First, given that the T-Coffee aligner is able to use external libraries, that capability and BALiBASE were used to prove that the libraries built with PPCAS are valid, resulting in almost equivalent accuracy alignments. Next, regardless of the execution time required to calculate the consistency library, a single node comparison was done. In this case, T-Coffee was outperformed, and by a greater more margin the larger the number of sequences that were aligned. After, to demonstrate the benefits of using a Big Data infrastructure, scalability was measured adding more computing nodes with a fixed size of 1,000 sequences. With 20 computing nodes, the results showed linear speedups, by 18.18x over the single node execution time and 29.45x over the TC version. Finally, scalability was evaluated by increasing the number of sequences

with 20 nodes. PPCAS was able to deal with 20,000 sequences in an average time of 64,013 seconds and an output size of 1,15TB.

4.3 Scalable Consistency in T-Coffee through Apache Spark and Cassandra database

Storing the consistency in a distributed file system allows us to enhance the scalability. Using Apache Cassandra database, we are able to obtain the benefits of a distributed file system, and the SQL-like queries to retrieve the needed data from the library. The use of big consistency libraries to obtain a final alignment is dealt with in this paper. A novel aligner named BDT-Coffee is presented. It is able to produce a quality alignment using the consistency data generated by PPCAS and integrate it into the aligner by optimally querying Apache Cassandra.

In order to reduce the impact of using the hard drives as the main memory, which produces high numbers of queries and random accesses, two levels of memory dynamic cache were implemented, the Consistency Cache (CC) and the Score Cache (SC). Such addition is able to solve the huge number of disk accesses, as well as improving the original T-Coffee execution time. Besides, the accuracy remains equivalent, as the BALiBASE benchmark demonstrates. Finally, the scalability is evaluated by a Homfam dataset. While BDT-Coffee is able to align up to 5,000 sequences, T-Coffee crashes at 1,000 as it fills the main memory (8GB). Furthermore, it is really important to highlight the execution time speedup. Both the library generation and aligning process are faster, and they respectively achieve speedups of 17.3x and 3.4x over T-Coffee when aligning 1,000 sequences.

4.4 Accurate consistency-based multiple sequence alignment reducing the memory footprint

The computation of the progressive alignment is the core of many MSA tools. However, the final accuracy of the method depends highly on the dynamic programming and its objective function. In this paper, a novel method, evaluated on the T-Coffee MSA tool and named MBT-Coffee, is proposed. Its main goal is to improve the original dynamic programming phase by providing better information to the back-tracking matrix and being able to replace the information lost when using reduced consistency libraries.

A first validation of the tool is carried out by using the BALiBASE benchmark, demonstrating that the proposal is able to produce better alignments than the original T-Coffee with barely 30% of its library, and having its execution time reduced by 55%. However, the best scenario seems to be produced at 50% of its library, where the accuracy hardly improves with more consistency, and the execution time is reduced by 37%. Such an increase also enables MBT-Coffee with 50% of the library to overcome Clustal Ω in accuracy terms. The final validation deals with larger datasets using some sets from the HomFam benchmark. In this case, the improvement in accuracy over the original method is far greater than before, reaching values above 0.1 on the SP score, and halving the execution times. This improvement also implies that the former is able to exceed the other MSA tools analyzed, obtaining a better accuracy than MAFFT-LINSI and MSAProbs.

Chapter 5

Global conclusions and future work

In this chapter, we present overall conclusions for all the research work done during the PhD. We also discuss the open research lines that may be followed by future works within this scope.

5.1 Conclusions

In this work, different algorithms and methods to improve consistency-based MSA tools were designed and implemented, and successfully validated using T-Coffee. Some of the proposals are based on improving the complexity of the library. Given a better heuristic to reduce the library and together with Apache Spark, we were able to generate the consistency quickly and without memory restrictions. Other improvements were based on improving the accesses to the main library, using databases such as Apache Cassandra and improving the locality of the data in internal functions of the application. Finally, the quality of the alignments was also improved, modifying the behavior of the dynamic programming. This is essential for a tool to be used by biologists.

Taking as a reference the milestones presented in chapter 2, the published papers, together with the goals achieved are listed below. The contributions included in Chapter 3 are highlighted in bold:

M1. Study and proposal of a consistency-based library with a linear com-

plexity in space.

This milestone mainly focused on reducing the complexity of the main library. The first proposal was the bound library method as the primary library. The method provided a drastic reduction in memory, chosen by the user, enabling scalability, but with an implicit reduction of the method accuracy. Next, we proposed the use of multiple solutions to partially mitigate the accuracy loss produced by the library reduction.

Although the results were good, they could be improved, so we decided to do a deeper analysis of the consistency library and investigate its behavior further. With the use of multiple scripts and a modified genetic algorithm, we were able to extract the necessary premises to maintain or discard constraints of the library correctly.

- The constraints with a higher weight are more representative than the rest. In the optimally generated libraries, 64% of them are in this percentile.
- The most closely related leaves of the guide trees must have more constraints. It is reasonable, as they are the first to be aligned, their accuracy must be ensured to avoid propagating an error to the rest of the alignment.
- The constraints have to cover all the domain of the alignment. There has to be a balance between the higher and lower weighted constraints, so the aligner has information about all the residue positions and not just blocks of data with high-rated areas.

These premises were used to define the pattern that was followed in the design of the method, and the memory-efficient consistency Library (MEL) was presented. MEL was able to obtain accuracy similar to the original T-Coffee method using half the memory, and reducing it further produced better accuracy than the previously proposed BLM.

The contributions published in this milestone are presented below:

- Lladós, J., Guirado, F., Cores, F. Scalable Consistency for Large-Scale Multiple Sequence Alignments. *Proceedings of the 14th International Conference on Computational and Mathematical Methods in Science and Engineering, CMMSE 2014*. ISBN: 978-84-616-9216-3
- Lladós, J., Guirado, F., Cores, F., Lérída, J. L. and Notredame, C. Recovering accuracy methods for scalable consistency library. *The Journal of Supercomputing, SUPE 2015*, 71(5):1833-1845. ISSN: 1573-0484
- Lladós, J., Cores, F., Guirado, F. Efficient Consistency Library for Multiple Sequence Alignment Tools. *Proceedings of the 17th International Conference on Computational and Mathematical Methods in Science and Engineering, CMMSE 2017*, 4:1269-1280. ISBN: 978-84-617-8694-7

M2. Migrating the consistency to massive data-processing frameworks.

Given the degradation in accuracy from reducing the library, this milestone focused on scaling the probabilistic pairwise model for consistency-based aligners with the help of Apache Spark and its subsequent use in an aligner. First, PPCAS was proposed. This tool was able to generate large libraries using a Hadoop infrastructure to improve its calculation efficiency. As a counterpart, a lot of data had to be managed to actually use it. Using T-Coffee as a validation software, we managed to successfully integrate those massive libraries into it through a distributed database, but the latency generated by accessing the data resulted in huge execution times. Nevertheless, the implementation of multiple cache levels and various optimizations resulted in excellent results which outperformed the original T-Coffee in two aspects, (1) scalability, the use of secondary storage as a back-end enhances the amount of consistency by far, while the memory remains fixed and (2) execution time, both the generation of the library plus the alignment stage are faster than with T-Coffee, and these increase even more with more computing nodes being added. Finally, we also implemented a distributed memory-efficient consistency library (DMEL) onto

PPCAS, which shares the proven benefit of reducing the memory by half with no accuracy losses and also further reducing the execution time of the alignment stage.

The contributions published in this milestone are presented below:

- Lladós, J., Guirado, F., Cores, F. **PPCAS: Implementation of a Probabilistic Pairwise Model for Consistency-Based Multiple Alignment in Apache Spark.** *5th International Workshop on Parallelism in Bioinformatics, PBIO 2017. Springer LNCS proceedings, vol. 10393 pp.601-610. ISBN 978-3-319-65482-9*
- Lladós, J., Cores, F., Guirado, F. Optimization of Consistency-Based Multiple Sequence Alignment using Big Data technologies. *The Journal of Supercomputing, SUPE 2018. ISSN: 1573-0484*
- Lladós, J., Cores, F., Guirado, F. Scalable Consistency in T-Coffee through Apache Spark and Cassandra database. *The Journal of Computational Biology, JCB 2018, 25(8):894-906. ISSN: 1066-5277*

M3. Validate algorithms that improve consistency-based aligners accuracy.

In this final milestone, I mainly focused on how to improve the final accuracy of the alignment. To do this, I had to look for a way of improving the objective function to deliver more accurate scores. The proposal, implemented in T-Coffee, is able to generate better alignments than the original method. Furthermore, it is perfectly integrated with MEL, acquiring its benefits and solving its downsides.

The contributions published in this milestone are presented below:

- Lladós, J., Cores, F., Guirado, F. and Lériida, J. L. **Accurate consistency-based multiple sequence alignment reducing the memory footprint.** *Computer methods and programs in biomedicine, CMPB 2021, 208(C):106237. ISSN:0169-2607*

5.2 Future Work

While the PhD was being carried out, different ideas arose that could not be considered for lack of time and that could be done to continue improving the MSA with consistency.

Generating the extended library beforehand

The on-the-fly library extension is a CPU intensive stage of the progressive alignment and it is very time consuming. It also has the downside that it repeats many computations because a pair sequence-residue will be aligned many times, and thus the extension will be repeated. Parallelizing such computations and having all this data beforehand would save a great deal of time during the execution, but would extend all the primary library results in out-sized amounts of data, as many combinations are not really used. Finding a way to deal with this issue will improve the long wait times of T-Coffee.

Validate the scalable consistency library in other MSA tools

T-Coffee is not the only method that uses consistency. There are many other MSAs that use it in their workflows. Using PPCAS and the library reducing policies should increase the performance of other tools on the market.

Design and implement a progressive method focused on Big Data

Given that we have already designed and implemented a stand-alone map-reduce model to build consistency libraries, we can produce a new MSA aligner based 100% on the map-reduce paradigm.

Appendix A

Doctoral stay at the University of Edinburgh: A brief summary

During the stay abroad at the Data-Intensive Research Group we did the first insights into Apache Spark. To start with, we created an Apache spark cluster in the cloud, using virtual machines generated with Openstack. After learning the basics of Spark, we engaged a test case used in a department of the University of Edinburgh. This test case identifies mutational overlaps using data from the 1000 genomes project in order to provide a null distribution for rigorous statistical evaluation of potential disease-related mutations. The 1000 genomes project provides a reference for human variation, having reconstructed the genomes of 2,504 individuals across 26 different populations to energize these approaches. Figure A-1 shows the scientific workflow used to solve such a test case.

The purpose of this test case is to find overlaps in mutations among individuals in the 1000 genome. After choosing the chromosome to analyze and downloading the respective set of individuals (which person has which mutations) and sift scores (how severe it is), it cross-matches the data and filters it by the desired population (locations around the globe). It then analyses the pair overlap mutation and the frequency overlap mutations. In this work, we focus only on the latter, which measures the frequency of overlapping in mutations by selecting a number of random individuals (in this work we selected 25 random individuals), and selecting all single-nucleotide

polymorphisms (SNPs: the most common type of genetic variation between people) without taking into account their SIFT scores. For example, if variant 1, variant 20, variant 42, and variant 80 are the only variants which presents 3 overlappings among individuals, we could say that the frequency of 3 overlappings among that group of individuals is 4 mutations.

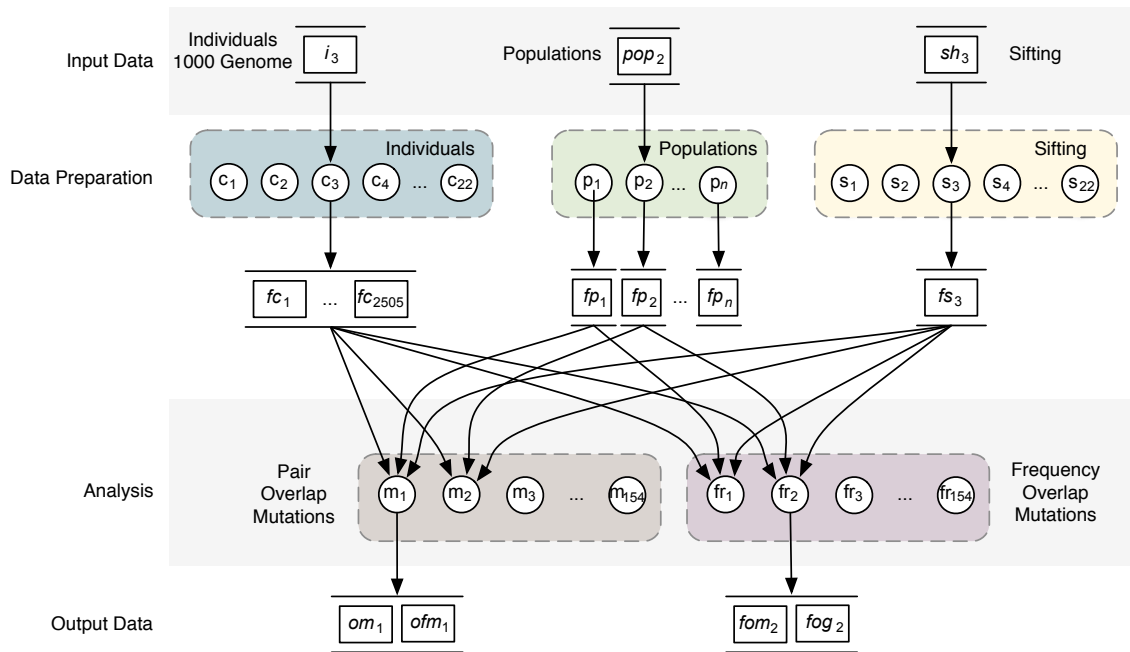


Figure A-1: Overview of the 1000 genome-sequencing analysis workflow.

The original version of the test case is coded in bash and is a sequential code. This produces long execution times and high memory usage. As an example, obtaining the frequency overlap mutations for chromosome 22 in the individuals in Europe produces memory peaks of 73,972Mb and takes 77,218s to be fully executed.

Instead, migrating this program to Apache Spark with the MapReduce paradigm, and only a node with 4 cores and 8GB of RAM, the execution time is reduced to 1,560s, almost a 50x speedup with lower requirements.

Figure A-2 shows a small example where we have 3 individuals (HG000XX) and only two rows of data about them (there are thousands in the real file). First, we filter the individuals who have mutations on both alleles (1|1) with a map operation,

using the individual as a key and the data that we are interested in as a value. Then, we GroupByKey() the result. Now the individuals are prepared to be used. We can intersect them with the desired population with the cogroup() function, and then proceed with the analysis.

```
u'#CHROM POS ID REF ALT QUAL FILTER INFO FORMAT HG00096 HG00097 HG00099'
u'22 16050075 rs587697622 A G other_suff... 1|1 0|0 0|0'
u'22 16054740 rs55926024 A G other_suff... 1|1 0|0 0|0'

[u'HG00096', [u'16050075', u'rs587697622', u'A', u'G']]
[u'HG00096', [u'16054740', u'rs55926024', u'A', u'G']]

[u'HG00096', [u'16050075', u'rs587697622', u'A', u'G'], [u'16054740', u'rs55926024', u'A', u'G']]
```

Figure A-2: Processing the individuals of the 1000 genome using Apache Spark.

The frequency overlap mutations are calculated in Figure A-3. First, a map operation is used to change the RDD key to the variant, assign them a value of 1 and reduceByKey() with an ADD operation. Next, a second key change is done, using the previous value as a key with a new value of 1 and reduceByKey() with an ADD operation. The result is that the frequency of 1 overlapping among that group of individuals is 2 mutations.

```
[u'HG00096', [u'16050075', u'rs587697622', u'A', u'G'], [u'16054740', u'rs55926024', u'A', u'G']]
[(u'rs587697622', 1), (u'rs55926024', 1)]
[(u'1', 1), (u'1', 1)]
[(u'1', 2)]
```

Figure A-3: Frequency overlap mutations of the 1000 genome using Apache Spark.

Given the good results obtained in this case, we implemented the T-Coffee library extension in Apache Spark. Note that the extension process is done on-the-fly in T-Coffee to avoid generating all the possible reweighing combinations. The problem is that if we want to calculate it in a Big Data environment beforehand, we have to generate all the possibilities.

When the progressive alignment asks for a combination of sequences: $(S_1 - S_2)$ and residues: $(T - G)$, there are two important scores to be calculated. The score that encompasses the direct and transitive consistency between $S_1 - T \longleftrightarrow S_2 - G$, and the max_score, which contains a sum of all the constraints aligned with $S_1 - T$ and a

sum of all the constraints aligned with $S_2 - G$. Figure A-4 shows a migration example of generating the extension over Spark with a reduced library of 4 constraints (the library generated for 4 sequences is up to 2,000 constraints). The problematic process is the first row, where we generate the transitive constraints of the input library. The GroupByKey() shuffles the data between the infrastructure nodes in order to group the keys and do further operations with them. This generates the same problem as in HPC, where the data is sent through the network and causes long delays. The other problem is noted in step (4) Map. We can see that 6 constraints are generated by 1T [2A 3F 4A], all the possible pair combinations from the braked values. So, each executor on the node is generating data, which is quite the opposite of the aim of Spark. This implementation worked flawlessly with small datasets, but generating the extended library for thousands of sequences causes long delays in GroupByKey() and crashes in the executors memory, as the amount of data generated by step (4) grows exponentially with the number of constraints it has.

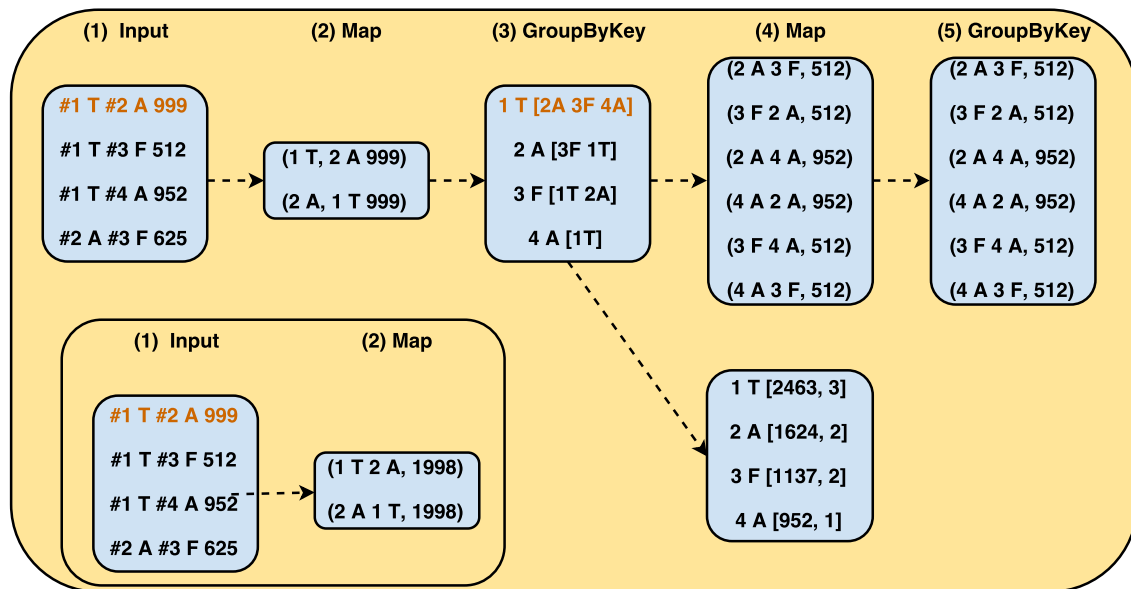


Figure A-4: T-Coffee extension calculation using Apache Spark.

Appendix B

Other contributions

- **Related to the bioinformatics field**

- Orobitg, M., Guirado, F., Cores, F., Lladós, J. and Notredame, C. High Performance computing improvements on bioinformatics consistency-based multiple sequence alignment tools. *Parallel Computing, PARCO 2014*, 42:18-34. ISBN: 978-84-933682-3-4
- Orobitg, M., Lladós, J., Guirado, F., Cores, F. and Notredame, C. Scalability and accuracy improvements of consistency-based multiple sequence alignment tools. *Proceedings of the 20th European MPI Users' Group Meeting, EuroMPI' 2013*, pp259-264. ISBN: 978-1-4503-1903-4

- **Unrelated to the bioinformatics field**

- Lladós, J., Mateo, J., Cores, F., Lérída, J. L. and Giné, F. Incentivación del aprendizaje de programación en las Ingenierías. Un caso práctico. *Actas de las XXIV Jornadas de paralelismo, JP 2013*, pp384-389. ISBN: 978-84-695-8330-2
- Mateo, J., Lladós, J., Lérída, J. L., Plà, L. M. and Solsona F. Paralelización del Algoritmo de Descomposición Cluster Benders. *Actas de las XXIV Jornadas de paralelismo, JP 2013*, pp390-395. ISBN: 978-84-695-8330-2

- Lladós, J., Arroyo, I., Cores, F., L rida, J. L., and Gin , F. RoboDist: Una Plataforma de juego P2P para incentivar la programaci n en los grados de Ingenier a. *Actas de las XXIII Jornadas de paralelismo, JP 2012*, pp501-506. ISBN: 978-84-695-4471-6
- Blanco, H., Llados, J., Guirado, F. and L rida, J. L. Ordering and Allocating Parallel Jobs on Multi-Cluster Systems. *Proceedings of the 14th International Conference on Computational and Mathematical Methods in Science and Engineering, CMMSE 2012*, pp196-206. ISBN: 978-84-612-5510-5
- Llad s, J., Pallej , T., Tresanchez, M., Teixid , M., Font, D. and Palac n, J. Experiencia de auto localizaci n en un recinto universitario a partir de la red WiFi existente. *Proceedings of the 11th Annual Seminar on Automation, Industrial Electronics and Instrumentation, SAAEI 2011*. ISBN: 978-84-933682-3-4

Glossary

Conservation Changes at a specific position of an amino acid (or less commonly, DNA) sequence that preserve the physico-chemical properties of the original residue. 38

Gaps Positions at which a letter is paired with a null are called gaps. 24

Identity The extent to which two (nucleotide or amino acid) sequences are invariant. 24

Residues The term residue refers to either a single base constituent from a nucleotide sequence, or a single amino acid constituent from a protein. This is a useful term when one wants to speak collectively about these two types of biological sequences. 26

Similarity The extent to which nucleotide or protein sequences are related. It is based upon identity plus conservation. 24

Substitution matrix A substitution matrix contains values proportional to the probability that amino acid *i* mutates into amino acid *j* for all pairs of amino acids. 41

Bibliography

- [1] Pena T. F. Pichel-J. C. Abuín, J. M. Pastaspark: multiple sequence alignment meets big data. *Bioinformatics*, 33(18):2948–2950, 2017.
- [2] W. R. Adrion. Research methodology in software engineering. In *Summary of the Dagstuhl Workshop on Future Directions in Software Engineering*, volume 18, pages 36–37. ACM SIGSOFT Software Engineering Notes, 1993.
- [3] Karun A.K. and Chitharanjan K. A review on hadoop - hdfs infrastructure extensions. In *2013 IEEE Conference on Information Communication Technologies*, pages 132–137, 2013.
- [4] Stretton AOW. The first sequence. fred sanger and insulin. *Genetics*, 162(2):527–532, 2002.
- [5] Alfred. Burger. Atlas of protein sequence and structure 1969. *Journal of Medicinal Chemistry*, 13(2):337–337, 1970.
- [6] Francis Crick. Central dogma of molecular biology. *Nature*, 227:561—563, 1970.
- [7] MO Dayhoff, RM Schwartz, and BC Orcutt. A model of evolutionary change in proteins. In *Atlas of protein sequence and structure*, volume 5, pages 345–352. National Biomedical Research Foundation Silver Spring, MD, 1978.
- [8] dos Santos Tostes-C. Dávila A. M. Senger H. da Silva F. A. de Castro, M. R. Sparkblast: scalable blast processing using in-memory operations. *BMC bioinformatics*, 18(1):318, 2017.
- [9] Ghemawat S. Dean, J. Mapreduce: A flexible data processing tool. *Communications of the ACM*, 53(1):72–77, 2010.
- [10] Chuong B Do, Michael Brudno, and Serafim Batzoglou. Prob cons: probabilistic consistency-based multiple alignment of amino acid sequences. In *AAAI*, pages 703–708, 2004.
- [11] R. C. Edgar. Muscle: a multiple sequence alignment method with reduced time and space complexity. *BMC bioinformatics*, 5(1):113, 2004.
- [12] Robert C Edgar. Muscle: multiple sequence alignment with high accuracy and high throughput. *Nucleic acids research*, 32(5):1792–1797, 2004.

- [13] Cedric Gondro and Brian P Kinghorn. A simple genetic algorithm for multiple sequence alignment. *Genetics and Molecular Research*, 6(4):964–982, 2007.
- [14] Steven Henikoff and Jorja G Henikoff. Amino acid substitution matrices from protein blocks. *Proceedings of the National Academy of Sciences*, 89(22):10915–10919, 1992.
- [15] Desmond G Higgins and Paul M Sharp. Clustal: a package for performing multiple sequence alignment on a microcomputer. *Gene*, 73(1):237–244, 1988.
- [16] Kazutaka Katoh and Daron M Standley. Mafft multiple sequence alignment software version 7: improvements in performance and usability. *Molecular biology and evolution*, 30(4):772–780, 2013.
- [17] Misawa K Katoh K and Miyata T. Kuma K. Mafft: a novel method for rapid multiple sequence alignment based on fast fourier transform. *Nucleic Acids Research*, 30(14):3059–3066, 2002.
- [18] Mehmet Kaya, Abdullah Sarhan, and Reda Alhajj. Multiple sequence alignment with affine gap by using multi-objective genetic algorithm. *Computer methods and programs in biomedicine*, 114(1):38–49, 2014.
- [19] Nasir A Kim KM and Caetano-Anollés G. The importance of using realistic evolutionary models for retrodicting proteomes. *Biochimie*, 99:129–137, 2014.
- [20] DanielC. Koboldt. The next-generation sequencing revolution and its impact on genomics. *Cell*, 155(1):27–38, 2013.
- [21] Kuo-Bin Li. Clustalw-mpi: Clustalw analysis using distributed and parallel computing. *Bioinformatics*, 19(12):1585–1586, 2003.
- [22] Schmidt B. Maskell D. L. Liu, Y. Msa-cuda: multiple sequence alignment on graphics processing units with cuda. In *Application-specific Systems, Architectures and Processors, 2009. ASAP 2009. 20th IEEE International Conference on*, pages 121–128. IEEE, 2009.
- [23] Schmidt B. Voss G. Müller-Wittig W. Liu, W. Gpu-clustalw: Using graphics hardware to accelerate multiple sequence alignment. In *International Conference on High-Performance Computing*, pages 363–374. Springer, 2006.
- [24] A. Matsunaga, M. Tsugawa, and J. Fortes. Cloudblast: Combining mapreduce and virtualization on distributed resources for bioinformatics applications. In *2008 IEEE Fourth International Conference on eScience*, pages 222–229, 2008.
- [25] Scott McGinnis and Thomas L. Madden. Blast: at the core of a powerful and diverse set of sequence analysis tools. *Nucleic Acids Research*, 32(2):20–25, 2004.
- [26] Deane C. M. Blundell T. L. Overington J. P. Mizuguchi, K. Homstrad: a database of protein structure alignments for homologous families. *Protein science*, 7(11):2469–2471, 1998.

- [27] A. Montañaola. *The pairwise problem with High Performance Computing Systems, contextualized as a key part to solve the Multiple Sequence Alignment problem*. PhD thesis, Universitat de Lleida, 2016.
- [28] Eugene W. Myers and Webb Miller. Optimal alignments in linear space. *Bioinformatics*, 4(1):11–17, 1988.
- [29] Farhana Naznin, Ruhul Sarker, and Daryl Essam. Vertical decomposition with genetic algorithm for multiple sequence alignment. *BMC bioinformatics*, 12(1):353, 2011.
- [30] Guo X. Pan Y. Nguyen, K. Multiple sequences alignment algorithms. *Multiple Biological Sequence Alignment*, 5:69–101, 2016.
- [31] Tung Nguyen, Weisong Shi, and Douglas Ruden. Cloudaligner: A fast and full-featured mapreduce based tool for sequence mapping. *BMC research notes*, 4(1):171, 2011.
- [32] C Notredame, L Holm, and D G Higgins. Coffee: an objective function for multiple sequence alignments. *Bioinformatics*, 14(5):407–422, 1998.
- [33] Cédric Notredame, Liisa Holm, and Desmond G. Higgins. Coffee: an objective function for multiple sequence alignments. *Bioinformatics (Oxford, England)*, 14(5):407–422, 1998.
- [34] Higgins DG Notredame C and Heringa J. T-coffee: A novel method for fast and accurate multiple sequence alignment. *Journal of molecular biology*, 302(1):205–217, 2000.
- [35] Gotoh O. Consistency of optimal sequence alignments. *Bulletin of Mathematical Biology*, 52(4):359–373, 1990.
- [36] Gotoh O. Heuristic alignment methods. *Methods in Molecular Biology*, 1079:29–43, 2014.
- [37] M. Orobitg. *High performance computing on biological sequence alignment*. PhD thesis, Universitat de Lleida, 2013.
- [38] Francisco M Ortuno, Olga Valenzuela, Fernando Rojas, Hector Pomares, Javier P Florido, Jose M Urquiza, and Ignacio Rojas. Optimizing multiple sequence alignments using a genetic algorithm based on three objectives: structural information, non-gaps percentage and totally conserved columns. *Bioinformatics*, 29(17):2112–2121, 2013.
- [39] Chen Y. Pan Y. Qin, L. and L. Chen. A novel approach to phylogenetic tree construction using stochastic optimization and clustering. *BMC Bioinformatics*, 7(4):S24, 2006.

- [40] Álvaro Rubio-Largo, Miguel A Vega-Rodríguez, and David L González-Álvarez. A hybrid multiobjective memetic metaheuristic for multiple sequence alignment. *IEEE Transactions on Evolutionary Computation*, 20(4):499–514, 2016.
- [41] G Sudha Sadasivam and G Baktavatchalam. A novel approach to multiple sequence alignment using hadoop data grids. In *Proceedings of the 2010 Workshop on Massive Data Analytics on the Cloud*, page 2. ACM, 2010.
- [42] Needleman SB and Wunsch CD. A general method applicable to the search for similarities in the amino acid sequence of two proteins. *Journal of Molecular Biology*, 48(3):443–453, 1970.
- [43] Michael C. Schatz. Cloudburst: highly sensitive read mapping with mapreduce. *Bioinformatics*, 25(11):1363–1369, 2009.
- [44] Dineen D. Wilm A. Higgins D. G. Sievers, F. Making automated multiple alignments of very large numbers of protein sequences. *Bioinformatics*, 29(8):989–995, 2013.
- [45] Fabian Sievers, David Dineen, Andreas Wilm, and Desmond G Higgins. Making automated multiple alignments of very large numbers of protein sequences. *Bioinformatics*, 29(8):989–995, 2013.
- [46] Andrew D. Smith, Wen-Yu Chung, Emily Hodges, Jude Kendall, Greg Hannon, James Hicks, Zhenyu Xuan, and Michael Q. Zhang. Updates to the rmap short-read mapping software. *Bioinformatics*, 25(21):2841–2842, 2009.
- [47] Weyer-Menkhoff J Subramanian A and Morgenstern B. Kaufmann M. Dialign-t: an improved algorithm for segment-based multiple sequence alignment. *BMC Bioinformatics.*, 6(1):66, 2005.
- [48] Javid Taheri and Albert Y Zomaya. Rbt-ga: a novel metaheuristic for solving the multiple sequence alignment problem. *Bmc Genomics*, 10(1):S10, 2009.
- [49] Julie D Thompson, Desmond G Higgins, and Toby J Gibson. Clustal w: improving the sensitivity of progressive multiple sequence alignment through sequence weighting, position-specific gap penalties and weight matrix choice. *Nucleic acids research*, 22(22):4673–4680, 1994.
- [50] Koehl P. Ripp R. Poch O. Thompson, J. D. Balibase 3.0: latest developments of the multiple sequence alignment benchmark. *Proteins: Structure, Function, and Bioinformatics*, 61(1):127–136, 2005.
- [51] Lasters I. Wyns L. Van Walle, I. Sabmark—a benchmark for sequence alignment that covers the entire known fold space. *Bioinformatics*, 21(7):1267–1268, 2004.
- [52] Jiang T. Wang L. On the complexity of multiple sequence alignment. *Journal of Computational Biology*, 1(4):337–348, 1994.

- [53] Travis J. Wheeler and John D. Kececioglu. Multiple alignment by aligning alignments. *Bioinformatics*, 23(13):559–568, 2007.
- [54] M. S. Wiewiórka, A. Messina, A. Pacholewska, S. Maffioletti, P. Gawrysiak, and M. J. Okoniewski. Sparkseq: fast, scalable, cloud-ready tool for the interactive genomic data analysis with nucleotide precision. *Bioinformatics*, 30, 2014.
- [55] Yunquan Zhang, Ting Cao, Shigang Li, Xinhui Tian, Liang Yuan, Haipeng Jia, and Athanasios V Vasilakos. Parallel processing systems for big data: a survey. *Proceedings of the IEEE*, 104(11):2114–2136, 2016.
- [56] G. Zhao, C. Ling, and D. Sun. Sparksw: Scalable distributed computing system for large-scale biological sequence alignment. In *2015 15th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing*, pages 845–852, 2015.
- [57] Yang X. Rospondek A. Aluru S. Zola, J. Parallel-tcoffee: A parallel multiple sequence aligner. *ISCA PDCS*, 7:248–253, 2007.
- [58] Quan Zou, Xu-Bin Li, Wen-Rui Jiang, Zi-Yu Lin, Gui-Lin Li, and Ke Chen. Survey of mapreduce frame operation in bioinformatics. *Briefings in Bioinformatics*, 15(4):637–647, 2014.