The Malta Human Genome Project

Progress Report

Sara Ann Abdilla (188396M)

Supervisor(s): Dr Jean-Paul Ebejer

Faculty of ICT

University of Malta

December 2016

Submitted in partial fulfillment of the requirements for the degree of B.Sc. Computing Science (Hons.)

Contents

1 Introduction **1**

1.1 Motivation 2

1.2 Why is this a non-trivial problem? 2

2 Background Research and Literature Review **2**

2.1 Genome Compression Tools 2

2.2 Read Alignment Tools 2

2.3 MapReduce Framework 3

2.4 Genome Browsers 3

2.5 Gaps in Current Research 3

3 Aims and Objectives **3**

4 Methods Development **4**

4.1 Genome Compression 4

4.2 Sequence Alignment 4

4.3 Alignment Visualisation 4

4.4 Future Plans 4

5 Evaluation **4**

6 Deliverables **5**

6.1 Project Timeline 5

Abstract: Globally, human genomes are being assembled in order to aid in medical diagnosis, forensic research, genealogy and bioinformatics amongst other areas. While many countries have already had a reference genome constructed for their population, Malta is still in the process of accomplishing this - a National Maltese Human Reference Genome. In this dissertation, we propose to build computational algorithms for the visualisation and analysis of DNA by developing a compression tool and a sequence aligner as deliverables and possibly, a genome browser which would involve data visualisation capabilities.

1 Introduction

Over the years, there has been an increase in sequencing of human genomes in order to identify genetic diseases. This procedure is aided by a reference genome which is considered as a representative sample of a population's set of genes or DNA (to which individual's DNA sequences can be compared against) [1]. Each individual's DNA corresponds to an arrangement of approximately 3 billion bases; each base being one of four possible nucleotides: adenine, cytosine, guanine and thymine - oftenly referred to by the letters A, C, G and T respectively. DNA (deoxyribonucleic acid) could be described as a model of biological life which encompasses genetic information/genes [2, 3] - it is analogous to a storage device of genetic information.

Genome assembly/sequencing technologies and projects are rapidly advancing and their costs are decreasing. DNA variations and mutations may correlate to diseases so finding differences between a reference genome and the genome of a patient afflicted with a disease has important implications for medical diagnosis and treatment [4] as genes are hereditary from one generation to the next. Examples of these genome projects include the 100,000 Genomes Project in the UK and the International Cancer Genome Project [5, 6, 7].

The University of Malta is developing a national Maltese Human Reference Genome whereby whole genome sequencing of certain Maltese DNA samples (provided by the Malta BioBank) will be performed. The American human genome sequencing facility Complete Genomics, which was founded in 2006, is a partner in this project. This development is required in order to identify any DNA mutations which are predominant in the Maltese population. Comparing the provided Maltese DNA samples with any other human reference genome would not provide any accurate results as every distinct nationality has a diverse gene pool. This is because humans are descendant from diverse civilisations, so their evolutionary histories are different.

A read is a nucleotide sequence which is a fragment of the genome. It is gathered by sequencers from cloned fragments of the genome as depicted in Figure 1. This is done as DNA sequencing technologies are unable to read whole genomes, but sequencing reads pose no problems as they are shorter.

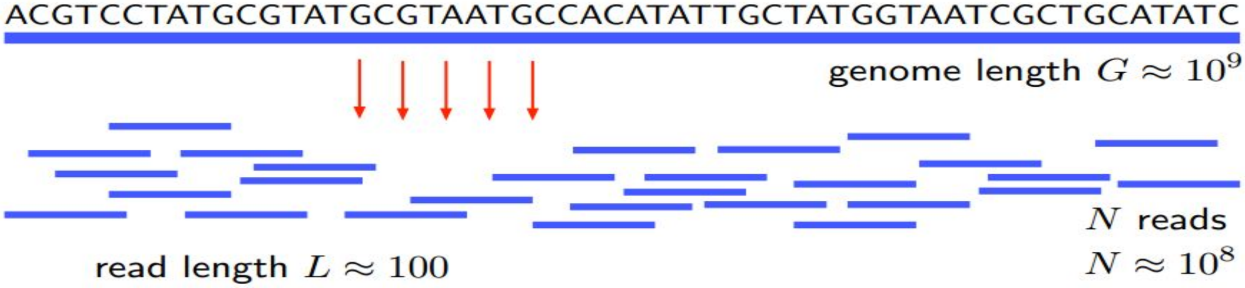

ACGTCCTATGCGTATGCGTAATGCCACATATTGCTATGGTAATCGCTGCATATC
genome length $G \approx 10^9$
N reads
$N \approx 10^8$
read length $L \approx 100$

Figure 1: DNA sequencing reads

The reading of an individual's DNA shows the likeliness of that person developing a disease, but the reading of a nations DNA shows why that population is more likely to develop a disease [4]. In order to accurately and efficiently match alignments between a human reference genome and the respective DNA sequencing reads, different algorithms may be applied. The most common aligners use Burrows-Wheeler Transform [5, 8] but there are multiple other algorithms; see section 2.2 for more details.

Thus, computationally, tools which deal with DNA analysis and visualisation need to be constructed in order to aid in this endeavour. These developments are only possible after the alignment of a set of sequencing reads against a reference genome is completed. For efficient alignment, parallelism and concurrency will be implemented in the system using the MapReduce framework; see section 2.3 for further details.

1.1 Motivation

While the national Maltese Human Reference Genome is gradually being developed, analysis tools for it are still in progress. This project will focus on constructing these DNA analysis and visualisation tools so it would be desirable to implement them in order to ensure the success of the entire Malta Human Genome Project (MHGP). Mainly, however, the project will be focused on introducing parallelisation and concurrency into these tools as existing tools could be used for some of the constructions.

1.2 Why is this a non-trivial problem?

A major problem with genome projects is that they have a large amount of data in the form of reads which is required to be aligned over the entire genome. Another problem is their efficiency with regards to this accurate alignment. When aligning a set of sequencing reads against the whole reference genome, different alignment methods will produce diverse performance especially when single and parallel process execution is considered. The more efficient the produced tools are, the easier it is to analyse and visualise the DNA.

2 Background Research and Literature Review

The development of a sequencing technology consists of multiple consecutive stages: genome compression, sequence alignment and genome browser construction. The following points detail some research which has already been conducted in these areas.

2.1 Genome Compression Tools

Genomes, particularly human ones, consist of a large amount of data - approximately 3,000 Mb (megabase pairs). In order to efficiently analyse them, said genomes are compressed using various methods.

For example, Chen et al devised the lossless *GenCompress* algorithm which implements certain established compression algorithms such as Lempel-Ziv [9], a variable-to-fixed-length code which parses the input sequence into non-overlapping DNA fragments of differing lengths while also constructing a dictionary of the fragments observed. By analysing approximate matches based on the evaluated edit distances, it was found that such a method not only achieves the best compression ratio but also finds common sections in DNA sequences [9]. Another example would be the lossless reference-based compression algorithm devised by Fritz et al [6]. It implements established components such as Golomb codes, optimal prefix codes (i.e. no codeword is a prefix of any other codeword in the relative system), and De Bruijn graphs, directed graphs symbolising overlaps between sequences, and was found to be quite efficient for read alignments similar to the reference genome. Other known algorithms are *Biocompress-2* and *Cfact* [9] along with *DNACompress* and *DNAZip* [6].

2.2 Read Alignment Tools

Next-Generation Sequencing (NGS) technologies are evolving rapidly and multiple read aligners are being produced; the most known being BWA, Bowtie, Soapv2, MAQ, BOAT, SHRiMP2 for the NGS platforms Illumina, Roche454 and ABI SOLiD [5, 7, 8]. These sequencers are given such importance due to the fact that genomic analysis would prove to be impossible without them. Reference genomes are produced using them so without this basic building block, no initial analysis can even commence.

The majority of read aligners implement Burrows-Wheeler Transform (BWT), which permutes the bases of a sequence into another sequence. Others use: FM-indexing which finds the number of occurrences of a read within a genome along with each occurrence's position, Needleman-Wunsch which globally finds similar regions between nucleotide sequences by comparing DNA fragments over the entirety of the sequences, and Smith-Waterman which is a variation of Needleman-Wunsch whereby similar regions are found locally by comparing the fragments over a substring of the sequences [5, 7, 8]. The efficiency of these aligners differs according to the task given; for example, while some may be more efficient with short reads, others may be more efficient with longer reads [10, 11]. Other known algorithms which assist in approximate read mapping are the Hamming distance as well as the Levenshtein/Edit distance. The former method evaluates the number of mismatches between two sequences with regards to substitutions while the latter method evaluates the number of mismatches between two sequences with regards to substitutions, insertions and deletions. These are not as commonly used due to their relative shortcomings. Both have an inefficient time complexity and the Hamming distance algorithm can even prove to be inaccurate as it only considers substitutions, unlike the Levenshtein distance which not only considers substitutions but also insertions and deletions.

2.3 MapReduce Framework

MapReduce is a parallel programming model which processes and generates large datasets whereby *Map* takes an input pair and returns a set of intermediate key-value pairs to *Reduce*, which then takes an intermediate key and a set of values for the respective key and merges them to return a smaller set of values. It was originally produced by Google to query trillions of web pages, but it has started to be included in other areas which also involve large datasets due to its parallelisation feature [12]; i.e. simultaneous process execution.

This model was used as a framework for read alignment by Menon et al. [13] due to its parallelisation, by incorporating suffix arrays (sorted arrays of all the suffixes of a sequence), BWT and cloud computing in their implementation. Unfortunately, the current implementation of this framework is not publicly available.

2.4 Genome Browsers

Genome browsers are graphical interfaces which are used for analysis in conjuction with genomic databases. They display the information found such that individuals are able to browse and visualise the stored genomes, leading to easier data seaching and analysis. Most browsers are web-based applications which allow certain customisations according to the user's requirements, but stand-alone browsers exist as well [14, 15]. An example of a web-based genome browser is *GBrowse* which was constructed by Stein et al. [15].

Genome browser construction has an intricate development process. Fortunately, many genome browser frameworks have already been constructed such as the most popular *GBrowse* (mentioned previously) as well as *Ensembl*, *JBrowse* and *LookSeq* among others. It should also be noted that there are two types of web-based genome browsers; multiple-species and species-specific browsers [14]. As this project deals with a human genome, a species-specific browser will be implemented. Following the Generic Model Organism Database (GMOD) project, there are multiple open-source tools for this type of browser; *GBrowse*, again, being one of the most used frameworks [14].

2.5 Gaps in Current Research

As mentioned in section 2.3, the read aligner which implements the MapReduce framework is not currently available. We will therefore be creating our own implementation with the use of this framework to parallelise the system; i.e. contribution to field.

3 Aims and Objectives

The aim of this project is to develop algorithms and build tools for the analysis of sequenced genomes from the Malta Human Genome Project. The main objectives of the system are as follows:

1. The compression of a reference genome such as that done by Chen et al. [9] and Fritz et al. [6];
2. The alignment of a reference genome against a number of sequencing reads such as that done by Lee et al. [8] and Li et al. [5];
3. The use of the MapReduce framework for read alignment to incorporate parallelisation and concurrency such as that done by Menon et al. [13].

Depending on project and time constraints, the construction of a genome browser could also be completed using novel and established components (such as that done by Stein et al. [15]) in order to reference and visualise DNA for comparitive genomics and to analyse DNA mutations.

4 Methods Development

This section details the components which have been implemented along with a description of future plans.

4.1 Genome Compression

The genome compression method implemented compresses the acquired human genome, having a size of approximately 3GB, into a binary file by means of integers (each integer being of length 4 bytes). Each nucleotide base is assigned two bits to represent it as there are only 4 possible bases. Chen et al. used the same deduction when developing *GenCompress* [9]. The implementation used acheived a compression rate of 70%, from roughly 3GB to 1GB, and it was observed to be accurate after certain decoding tests took place.

4.2 Sequence Alignment

Four possible alignment methods were then taken into consideration; the number of human sequencing reads being 28,094,847 with each read having a length of 60 bases; i.e. 30 base-pairs/bp. The Hamming distance and the Levenshtein distance methods both return a measure of the similarity between two sequences, k-mer indexing evaluates all the possible subsequences of a sequence of length k into an index and the most frequently used FM-indexing evaluates all the possible subsequences of a sequence, using BWT, into an index.

These aligners were first implemented and tested on string inputs; i.e. on an uncompressed genome of marginally less size than the human genome to confirm their applicability. Some of these functions were then converted in order to instead support integer inputs due to the previously mentioned genome compression. This was done by using bitwise operations and by finding any patterns in the integer sequence; i.e. any repetitive integers or pairs of integers or triples and so on in the compressed reference genome.

4.3 Alignment Visualisation

The data visualisation tool's initial construction is based on *Tkinter*, Python's standard Graphical User Interface (GUI) package. The result is a depiction of the reads aligned with the genome, with lines representing the position and length of each alignment/match. As of now, its interactivity is in the form of clicking a line at a certain position to output the offset of the match at the point of clicking.

4.4 Future Plans

As for future developments, firstly, the read alignment methods need to all be converted to support integer inputs (due to the genome compression applied) and then compared in order to deduce the most efficient one. An improvement on the BWT implementation will be done by incorporating the MapReduce parallel programming model, as described by Menon et al. [13], in order to accelerate the transformation.

Secondly, the data visualisation tool needs to be further refined; i.e. become more interactive in terms of genome analysis. This could be accomplished by outputting the DNA mutation found at the point of clicking instead of just the offset, or perhaps even both.

Optionally, depending on the project's time constraints, the genome browser tool could also be constructed using the established technologies (described in section 3) for efficient comparitive genomics; the visualisation tool being incorporated in this. Each method developed will be analysed and compared to deduce the most feasible implementation.

5 Evaluation

The main evaluation techniques proposed are as follows:

1. Comparitive review of the developed read aligner with the other developed aligners and with existing methods by comparing match rates with the corresponding time taken;
2. Comparitive review of the parallel MapReduce read aligner against a single process execution;
3. Review of the developed human genome visualisation tool by testing that its interface is easily understood by most users by, for example, checking that the data is output in a clear manner;

4. Possible comparitive review of the constructed genome browser by referencing DNA and analysing DNA mutations followed by comparison of results with existing methods.

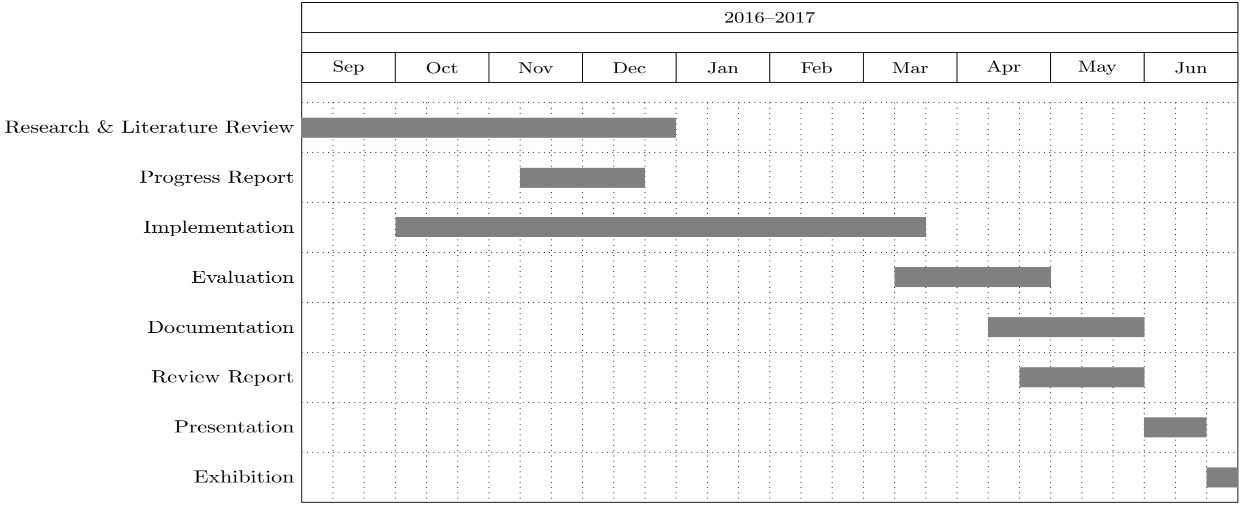
6 Deliverables

This project's deliverables are as follows:

1. The Final Year Report which will include all the relevant background information required to understand said project, a detailed explanation and the evaluation results;
2. The implementation (code) of the designed and developed system;
3. The documentation to explain to the users how the system should be employed; i.e. a user manual.

6.1 Project Timeline

The milestone schedule is represented in the following Gantt chart:



References

[1] B. Berger, J. Peng, and M. Singh, "Computational solutions for omics data," tech. rep., USA, Mar. 2014.

[2] L. Hunter, *Artificial Intelligence and Molecular Biology.* AAAI Press, 445 Burgess Drive, Menlo Park, California 94025, USA: MIT Press, 1993.

[3] A. M. Lesk, *Introduction to Genomics.* Great Claredon Street, Oxford, UK: Oxford University Press, 2 ed., 2012.

[4] "The hidden history of the maltese genome," *Think Magazine*, vol. 16, pp. 19–25, Apr. 2016.

[5] H. Li and R. Durbin, "Fast and accurate short read alignment with burrowswheeler transform," tech. rep., Wellcome Trust Sanger Institute, Wellcome Trust Genome Campus, Cambridge, CB10 1SA, UK, May 2009.

[6] M. H.-Y. Fritz, R. Leinonen, G. Cochrane, and E. Birney, "Efficient storage of high throughput dna sequencing data using reference-based compression," tech. rep., United Kingdom, Jan. 2011.

[7] L. Huang, V. Popic, and S. Batzoglou, "Short read alignment with populations of genomes," tech. rep., 2013.

[8] D. Lee, F. Hormozdiari, H. Xin, F. Hach, O. Mutlu, and C. Alkan, "Fast and accurate mapping of complete genomics reads," tech. rep., Oct. 2014.

[9] X. Chen, S. Kwong, and M. Li, "A compression algorithm for dna sequences and its applications in genome comparison," tech. rep., New York, NY, USA, Apr. 2000.

[10] M. Ruffalo, T. LaFramboise, and M. Koyutrk, "Comparative analysis of algorithms for next-generation sequencing read alignment," tech. rep., July 2011.

[11] J. Shang, F. Zhu, W. Vongsangnak, Y. Tang, W. Zhang, , and B. Shen, "Evaluation and comparison of multiple aligners for next-generation sequencing data analysis," tech. rep., Mar. 2014.

[12] J. Dean and S. Ghemawat, "Mapreduce: Simplified data processing on large clusters," tech. rep., Jan. 2008.

[13] R. K. Menon, G. P. Bhat, and M. C. Schatz, "Rapid parallel genome indexing with mapreduce," tech. rep., June 2011.

[14] JunWang, L. Kong, G. Gao, and J. Luo, "A brief introduction to web-based genome browsers," tech. rep., July 2012.

[15] L. D. Stein, C. Mungall, S. Shu, M. Caudy, M. Mangone, A. Day, E. Nickerson, J. E. Stajich, T. W. Harris, A. Arva, and S. Lewis, "The generic genome browser: A building block for a model organism system database," tech. rep., Dec. 2002.