Ph.D. Proposal

Davide Cozzi, 829827, d.cozzi@campus.unimib.it

Introduction

Computational pangenomics is a new emerging research field in computational biology. This concept is based on the idea we need to move from the traditional view of a reference genome as a *linear sequence* to the one where we consider the *genetic variations* or *variants*, in a large collection of sequences of species. This new reference is called a *pangenome*. Pangenomics is becoming increasingly essential in the field of biomedical and personalized medicine, mainly thanks to the **genome-wide association studies (*GWAS*)** since GWAS studies need to analyze a large collection of individual genomes. Unfortunately, to complete these types of studies on a large amount of data, such data must be indexed, queried and analyzed. Just to give some examples the *homo sapiens reference genome (GRCh38.p14)*, has a size of $\sim 3.1Gb$ and contains $\sim 59,265$ genes, as reported by NCBI. Biological studies done with the *1000 Genome Project* have pointed out that there are over 88 million variants between those human genomes. Among these variants, 84.7 million are **Single Nucleotide Polymorphisms (*SNPs*)**, 3.6 million are **short insertions/deletions (*indel*)** and 60000 are **structural variants**, involving more than 50 nucleotides. Moreover, it is necessary to consider that the objective is the sequencing of at least 100 thousand individuals, in the next few years. Therefore, it is clear that dealing with all these data is challenging for the current state of art algorithms and data structures for sequence analysis. An example is represented by a classical research topic in sequence comparison: *pattern matching*, where the input is a long text (a genome) and a short pattern (a read) and the output is the list of all occurrences of the pattern as substrings of the text. The problem of *pattern matching* is one of the most fundamental topics in the field of algorithmics and bioinformatics. We can notice that such problem has linear solution in time, if there is proper indexing of the text. The interest in such problems is due to the need to align sequences or to search for specific patterns within the *DNA*. In this context, a large number of data structures and algorithms have been modeled. Among these, one of the most used is the **Burrows-Wheeler transform (*BWT*)** together with indexing via *FM-index*. Thanks to these algorithms and to the fact that it is returned an interval over the *BWT*, the problem of pattern matching, over an indexed text, has become linear on the length of the pattern. The use of such algorithms is essential in speeding alignment algorithms such as **BLAST**, based on the *seed-and-extend paradigm*, where short strings, the so-called *seed*, are chosen to be the starting point of the alignment. Then, from the seed, the algorithm proceeds to extend the match to compute the alignment, increasing the efficiency of the algorithm. The key point is the ability to compress large texts and query the compressed texts themselves. Furthermore, in recent years, as introduced, there has been a change of interest in the field of bioinformatics. Now the researchers are deepening the topic of the **pangenome**, which term was introduced in 2005 for dealing with the comparison of genes in bacterial species. Briefly, the *pangenome* is a compact representation of multiple genomes, encoding the variations in a multitude of samples from the same species. In fact, the need to take into account the high variability in population genomes as well as the specificity of an individual genome in a personalized approach to medicine is rapidly pushing the abandonment of the traditional paradigm of using a single reference genome [1].

The main focus of this proposal is improving the current state of the art of pattern matching algorithms and data structures to manage biological sequences, with a focus on spatial complexity. In fact, the design of data structures to handle large collections of data, filling the gap between the genomic data production and the current computational state of the art, will have an impact on the ability of molecular biologists to analyze the current amount of data, represented by pangenome graphs.

Preliminaries

Thanks to the last developments in sequencing technologies, with **Next Generation Sequencing (*NGS*)** and **third-generation sequencing**, which had led both to reduce the costs of single sequencing and to produce sequences of ever-higher quality in less and less time, the researchers were able to theorize the **pangenome graph**. Furthermore, the new amount of sequences has led to new algorithms for *pattern matching* for collections of sequences. In 2021, Rossi et al. proposed *MONI* as a data structure to handle a **run-length encoded version of BWT (*RLBWT*)**, with the ultimate intention of indexing and using multiple genomes as a reference [2]. With this type of encoding, it is possible to further improve the compression of the text, having that consecutive identical characters, the so-called *runs*, are compactly stored, making it possible to study pangenomic sequences. According to Rossi et al., *BWA-MEM*, one of the most used read aligners, uses between 1.1 and 3.8 times more memory than *MONI*. Together with this data structure, the authors used the concept of **matching statistics (*MS*)** to efficiently compute the matches between a pattern and a text. A recent improvement has been made through the implementation of *PHONI* [3], where the **longest-common-extension (*LCE*) queries**, achieving the same results as *MONI* with only a single sweep over *MS* array instead of two as in *MONI* (details in the next section). On the other side, several computational problems arising from computational biology have led to the development and implementation of algorithms and data structures for analyzing specific sequence data such as *haplotypes* and *genotypes* in the **genotyping variants problem**. Briefly, we could define *haplotypes* as a combination of allelic variants, each one inherited from a parent. Instead, the *genotype* is combined information of the haplotypes. For example, humans have two haplotypes, being diploid, and the genotype is their combined information. So, since 2005, publication of the **GWAS** has begun. The goal of this type of studies is to screen the *pangenome* looking for associations between genetic variants, for example to study the outcomes of diseases. In this particular historical period, it is impossible not to mention viruses. In fact, by nature, viruses replicate a lot but often in a not perfect way, during infections. This produces many inexact clones, referred as **viral haplotypes**, which, taken together, form the **viral pangenome**. The identification of all these haplotypes is crucial both to the study of the spread of viruses and to the production of efficient drugs, in a context of high pharmacological resistance. One of the most important data structures, developed to handle the study of haplotypes sequences, is the **positional Burrows-Wheeler transform (*PBWT*)**, proposed by Durbin in 2014 [4]. *PBWT* aims to represent efficiently an haplotypes panel, storing them in a queryable compressed way (details will be described later). According to Durbin, the used memory by *PBWT* is nearly six times smaller than the raw data. These spatial results are necessary to process real information. Using this particular data structure, it is possible to study only biallelic panel, over an alphabet $\Sigma \in \{0, 1\}$, for example, to extract common patterns in a set of haplotypes. Thanks to the compressed representation of the data, the use of the *PBWT* is found in many software for the study of haplotypes and in various genotype imputation methods that infer unobserved genotypes in a sample of individuals [5].

State of the art

Now I present a brief overview of the main algorithms, data structures and tools that are the state of the art of computational pangenomics.

Efficient and compact boolean data structures. If $\mathcal{X}$ is the optimal number of bits needed to store some data, a representation of this information is defined *succint* if it takes $\mathcal{X} + o(\mathcal{X})$ bits of space. **Bitvectors** are one of the most important *succint data structures.* A *bitvector* is an array on n bits that allows two particular operations, called **rank** and **select**, in addition to the classic operations on boolean arrays, such as *random access* in *constant time.* More in detail, the *rank function* allows calculating how many occurrences of one are up to a certain index. Instead, the *select function* allows obtaining the index of every one present in the *bitvector.* Formally, given a bitvector B, such that $|B| = n$, and given an index i, such that $0 \leq i < n$, we can define $rank_B(i) = \sum_{k=0}^{k<i} B[k]$. Instead, about the select function, given an integer i, such that $0 < i \leq rank_B(n)$, we can define $select(i) = \min\{j \mid rank_B(j+1) = i\}$. From a theoretical point of view, these two operations can be supported in *constant time*, with the additional cost of $o(n)$ bits in memory. There are several implementations of the same (for example *plain bitvector, interleaved bitvector, sparse bivector*, etc...) within **SDSL (*Succint Data Structures Library*)**, one of the most important C++ library used in bioinformatics proposed by Gog et al. in 2014. Thanks to the various implementations, both the computational time of the two main operations and the additional bits needed vary, allowing a better choice of the best variant possible depending on the use case. Due to their compactness in memory, *bitvectors* are widely used in algorithms for the analysis of biological sequences. An example of the use is tracking the runs in the run-length encoded implementations of *BWT* and *PBWT*, where we put one at each head of every run, allowing fast operations for indexing and mapping.

Run-length encoding and succinct data structures. The **Burrows-Wheeler Transform (*BWT*)** was introduced in 1994 to compress texts but it has been used widely in bioinformatics, above all thanks to the already cited *FM-index.* Speaking of *pangenome*, linear indexing via *FM-index* is no longer the best solution as it does not handle the large repetitions there are in this new type of sequences. In 2005 Mäniken and Navarro defined the **Run-Length encoded Burrows–Wheeler Transform (*RLBWT*)**. Given a text T, $RLBWT_T$ is a representation of BWT_T with a compact storage of consecutive equal characters, the so-called *runs*. With this new perspective, the algorithms have changed from being linear over the length of the text, n, to be linear over the number of runs, r, so sub-linear over the length of the text. The new indexing method, introduced by Gagie et al., was called **r-index** and it corresponds to the *RLBWT* plus the *suffix array sampling* at the beginning and the end of every run. The algorithm for querying through the *RLBWT* takes advantage of other methods, such as the use of **thresholds**, defined as the minimum *LCP* value between two consecutive runs of the same character, in *MONI*. Instead, in *PHONI*, the authors use the **longest common extension (*LCE*) query**, to compute the *MS*. Formally, an *LCE query*, given two positions i and j in a text T, such that $|T| = n$, compute the length of the longest common prefix between $T[i : n-1]$ and $T[j : n-1]$, so an *LCE* is the right equal common extension between two positions in the text. Both solutions operate on a compressed representation of the text, via a **straight-line program (*SLP*)**. Briefly, an *SLP* is based on a *grammar-compression* algorithm, by a *context-free grammar*, and here it is used for *random access* and *LCE queries*. The purpose of the two projects is the computation of the **matching statistics (*MS*)**. Given a pattern P and a text T, the *MS* of P in respect to T is an array M of pairs position/length (*pos*/*len*), $|M| = |P|$, such that $T[M[i].pos : M[i].pos+M[i].len-1] = P[i : i+M[i].len]-1]$ and $P[i : i + M[i].len]$ does not occur in T. Given *MS*, we can compute every **Maximal Exact Match (*MEM*)** of a pattern in a text. Given a text T and a pattern P, a substring of the pattern $P[i : i+l-1]$, of length l, is a *MEM* of P in T if $P[i : i+l-1]$ is a substring of T but neither $P[i-1 : i+l-1]$ nor $P[i : i+l]$ are, so if the substring cannot be extended either to the right or to the left. Furthermore, using a particular function called φ (and its inverse φ^{-1}), based on the use of the **inverse suffix array (*ISA*)**, it has been possible to find all starting positions of all copies of P in T from starting position of the match extracted by *MS*, quickly calculating, given a position p in *SA*, the previous and next index stored in the suffix array. So, it was possible to perform pattern matching efficiently even on long sequences of nucleotides, such as those studied in a pangenomic context.

PBWT. Based on the theories of the *BWT*, in 2014, Durbin devised the **positional Burrows–Wheeler transform (*PBWT*)**, to solve the problem of pattern matching on panels (matrices) of haplotypes. In detail, he analyzed a panel X with M haplotypes and N biallelic sites. This data structure is based on a *reversed-prefix ordering* at each column k that produces two different multidimensional arrays. The first one is the set of the **prefix arrays**, denoted by a, which contains the index of the haplotype m in the original panel, for each column k and each position i of a_k. More formally, we can say that $a_k[i] = m$ iff X_m is the i-th haplotype in the reversed-prefix ordering at column k. We can note x_m, such that $a_k[i] = m$, could be denoted by y_i^k. The second bidimensional array is the set of the **divergence arrays**, denoted by d, which indicates the index of the starting column of the longest common suffix, ending in column k, between a row and its previous one, at reversed-prefix ordering at column k. More formally, we can define $d_k[i] = h$ iff h is the smallest column index such that $y_i^k[h, k) = y_{i-1}^k[h, k)$. Thanks to these two bidimensional arrays, it is possible to compute all matches within X longer than a minimum length L, all set-maximal matches within X in linear time and all set-maximal matches, maximal in the width of the match, from a new sequence z to X in $\mathcal{O}(M^2N)$. Since its development, there has been a growth in research based on it, both in terms of variant design, such as the **multiallelic PBWT** or the **dynamic PBWT**, where the static PBWT is generalized to a dynamic data structure via linked lists, and in terms of its use to study haplotype panels. A first example could be found in the paper of Alanko et al., published in 2021, where the authors used the *PBWT* to look for *maximal perfect haplotype block*, within a haplotypes panel. These types of algorithms are essential for the identification of genomic regions that show signatures of natural selection. Another interesting paper is the one by Williams and Mumey, published in 2020, who also studied *maximal perfect haplotype blocks*, but with the addition of a *missing data* management attempt, handled with the help of wildcards, using the *PBWT*. A very intriguing tool, talking about *GWAS*, is **IMPUTE5**, proposed by Rubinacci et al. in 2020. The main purpose of this software is making *genotype imputation* to predict unobserved genotypes from a panel with millions of haplotypes, so it was necessary to use *PBWT*. Improving this data structure, it will be possible to help the *GWAS* on tumors and other disorders.

Research goals

For my master's thesis, I had to adapt the concept described for *RLBWT* in *MONI* and *PHONI*. Obviously, there are some limitations, as in Durbin's *PBWT*, such as studying only biallelic panels and ignoring the management of missing data, which are very frequent in real cases.

During my master's degree, I have deepened some theoretical issues related to algorithms, especially in bioinformatics, and some modeling and inference topics, related to computational systems biology. So, it is my interest to continue my studies with the Ph.D. to be able to deepen the computer science potential in the biological context. Summarizing, the most important research goal of my Ph.D. is designing and validating algorithms and data structures to deal with open problems in pangenomics, such as:

1. **Multiallelic data**. Thanks to the growth in the production of genotypic data, the number of multiallelic sites is expected to grow, as well as the number of sample(even if there is only a 2% presence of triallelic sites, actually known in the human genome). Moreover, not only such sites could be more than expected but they are usually not considered by the majority of tools. The first step in this direction was made by Naseri with the already cited *m-PBWT*. Talking about spatial complexity, the run-length encoded version of the stored arrays for the *FM-index* could allow the management of increasingly large data for the imputation phases. In this context, the aim is to implement a new version of the *RLPBWT* that can manage multiallelic data, adapting the current use of bitvectors to handle efficiently also the *LF-mapping* with more than two alleles.
2. **Missing data**. A second extension, that would be more complicated to design, is a *RLPBWT* version that admits the presence of missing data. Most algorithms and data structures mentioned above assume that they work on exact data. In reality, real data can contain errors or even gaps, both mainly due to the imperfections of sequencing technologies, although now they have high correctness rates. Unfortunately, this is still an open problem, even if most of the errors are corrected in a preliminary step (mostly by heuristic methods). Handling missing data is known to be *hard* because every gap could assume any possible value. Having said this, I should probably deal with *parametric* or *approximate algorithms*, based on research already developed in *BIAS*.
3. **Algorithms for pangenome graphs & other goals**. Another research goal is to deepen the study of *pangenome graphs*. In fact, it is possible to interpret haplotype sequences as a pangenome graphs, in a compact way via the **Graph BWT (*GBWT*)**, even if it is more a *multi-string BWT* than a classical graph. A first improvement would be spatial optimization. There are a lot of open problems regarding the use of *GBWT*, such as handling missing/erroneous data and representing complex and nested variants. Once I have tried to solve the problem of missing data for the *PBWT/RLPBWT*, I could think of an adaptation for the *GBWT*, because it would be possible to analyze more precisely real data. I would also point out my interest in deepening all the other issues related to pattern matching, indexing structures and pangenome graphs, according to the *BIAS* laboratory's topics of interest.

The theoretical research requires an experimental verification, for this purpose some technologies that will be used during my Ph.D. must be mentioned. During both my bachelor's and master's degree, I mainly focused on the use of the **C++ programming language**, thanks to the availability of efficient libraries, such as the already cited *SDSL*. However, in addition to *C++*, I had the opportunity to deepen **Python**, with libraries such as *biopython*, and **Rust**, with recently developed libraries such as *bio-rust* (even if still not complete from an algorithmic point of view). Another point of interest is the analysis and the development of efficient algorithms based on parallel computing (also on GPU), that are increasingly in use in both *bioinformatics* and *computational systems biology*. To conclude this proposal, I also would point out the intention to remain in contact with various researchers, with whom I have already collaborated during my master thesis, including Christina Boucher (University of Florida) and Travis Gagie (Dalhousie University).

References

[1] The Computational Pan-Genomics Consortium. "Computational pan-genomics: status, promises and challenges". In: *Briefings in Bioinformatics* 19.1 (Oct. 2016), pp. 118–135.

[2] M. Rossi, M. Oliva, et al. "MONI: A Pangenomic Index for Finding Maximal Exact Matches". In: *Journal of Computational Biology* (Feb. 2022).

[3] C. Boucher, T. Gagie, et al. "PHONI: Streamed matching statistics with multi-genome references". In: *2021 Data Compression Conference (DCC)*. IEEE. 2021, pp. 193–202.

[4] R. Durbin. "Efficient haplotype matching and storage using the positional Burrows–Wheeler transform (PBWT)". In: *Bioinformatics* 30.9 (Jan. 2014), pp. 1266–1272.

[5] S. Rubinacci, O. Delaneau, et al. "Genotype imputation using the Positional Burrows Wheeler Transform". In: *PLOS Genetics* 16.11 (Nov. 2020), e1009049.