PhD Proposal

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

Introduction

Computational pangenomics is becoming increasingly essential in the field of biomedical and personalized medicine, mainly thanks to the **genome-wide association studies (*GWAS*)**. Unfortunately, in order to complete these types of studies a large amount of data is needed, data that 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. From a biological point of view, it is also interesting to note that, as it has been pointed out by the study of *1000 Genome Project*, 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 samples, in the next few years. It is therefore clear that all these data are now a limit to the state of art algorithms and data structures.
The problem of *pattern matching* is one of the most studied topics in the field of algorithmics and bioinformatics. 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*)**, thanks to the studies of Ferragina and Manzini, who proposed its use together with indexing via *FM-index*. The use of such algorithms, such as *BTW*, is essential in alignment algorithms such as **BLAST**, based of the *seed-and-extend paradigm*, where short strings, the so-called *seed*, are choosen to be the starting point of the alignment. Then, from the seed, the algorithm proceed to extend the match in order to compute the alignment, increasing the efficiency of the algorithm.
Furthermore, in recent years, as introduced, there has been a change of interest in the field of bioinformatics. Until a few years ago the research was focused on the study of a **linear sequence of a genome** while now the researchers are beginning to deepen the topic of the **pangenome**, which term was introduced by Tettelin in 2005. briefly, the *pangenome* is a compact rappresentation of multiple genomes, encoding the variations in a multitude of samples from the same specie. 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].
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 regarding the problem of *pattern matching*. 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]. Together with this data structure, the authors proposed the concept of **matching statistics (*MS*)** in order 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** in order to further optimize the pattern search.
Against this background, various algorithms and data structures have been implemented in order to study *haplotypes* and *genotypes*, such as 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 the complete set of genes contained in the DNA. So, from 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 in order to study outocomes 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. Thus 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 structure, developed in order to handle the study of haplotypes sequences,

is the **positional Burrows-Wheeler transform (*PBWT*)**, proposed by Durbin in 2014 [4]. Using this particular data structure (which will be described below), it is possible to study efficiently a collection of haplotypes but only in the bi-allelic case. Furthermore variants of the *PBWT* have been studied for handling the multiallelic case. The use of the *PBWT* is found in many software for the study of haplotypes and in various genotype imputation methods, that are studies that infers unobserved genotypes in a sample of individuals [5].
During the development of my master thesis, I worked to create a run length encoded variant of the *PBWT*, the **RLPBWT**, using and adapting the various theories developed for the *RLBWT*, in collaboration with the authors of *MONI* and *PHONI*. In this context, my PhD is going to be focused on the development of new algorithms in various topics related to open problems in the study of the *haplotyping/genotyping*, of the *genome variants* and of the imputation issues related. My intention is also to deepen experimental themes of *pattern matching* and of the *pangenome graph*, in detail the new developments on *BWT* and indexing structures, as well as *succinct data structures*, with particular attention to the use of *bitvectors*.

State of the art

Now I present a brief overview of the main algorithms, data structures, methods etc ... that will be the core of my studies during PhD.

Bitvectors **Bitvectors** are one of the most important data structure when mentioning *succinct data structures*. A *bitvector* is an array on n bits which 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 to calculate how many occurrences of one are up to a certain index. Instead, the *select function* allows to obtain 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 $0 < i \leq rank_B(n)$, where $n = |B|$, 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 $\mathcal{O}(n)$ bits in memory. In more practical terms, there are several implementations of the same within **SDSL (*Succint Data Structures Library*)**, one of the most important C++ library used in bioinformatics. As the implementation changes (for example *plain bitvector, interleaved bitvector, sparse bivector* etc...) the computational time of the two operations varies (usually only one of the two is in constant time) as well as the amount of additional bits needed. An example of the use of bitvectors is tracking the runs in the run-length encoded implementations of *BWT* and *PBWT*, where we put one at each head of run, allowing fast operations along the runs themselves.

RLBWT The **Burrows-Wheeler Transform (*BWT*)** was introduced in 1994 in order 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 Burrows–Wheeler Transform (*RLBWT*)**. Given a text T, $RLBWT_T$ is a rappresentation 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 being 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 at the end of every run. The algorithm for querying through the *RLBWT* takes advantage of other methods, such as the use of **thresholds** (minimum *LCP* value between two consecutive runs of the same character) in *MONI*, and the use of **longest common extension (*LCE*) query** (to compute the right equal common extension between two position in the text) in *PHONI*. Both solutions use **straight-line programs (*SLP*)** briefly a *grammar-compression* algorithm based on a *context-free grammar*, for *random access* in *MONI* and for *Longest Common extensions (LCE) queries* in *PHONI*. The purpose of the two projects is 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 right or left. Furthermore, using a particular function called φ (and 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 position in the suffix array.
Thanks to these and other methods, it was possible to perform pattern matching efficiently even on long sequences of nucleotides, such as those studied in a pangenomic context. In fact, most studies in the field of bioinformatics start with the resolution of pattern matching problems and, as a direct consequence, of alignment problems.

PBWT Based on the theories of the *BWT*, in 2014, Durbin devised the **positional Burrows–Wheeler transform (*PBWT*)**, in order 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 for 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 already mentioned multiallelic version or the **dynamic PBWT (*d-PBWT*)**, 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 Wlliams and Mumey, published in 2020, who also studied *maximal perfect haplotype blocks*, but with the addition of *missing data*, 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 to make genotype imputation in order to predict unobserved genotypes from a panel with milions of haplotypes. Due to the size of the panel, the use of *PBWT* has been proved to be inevitable, further demonstrating the importance of this data structure.
Durbin himself, the author of the *PBWT*, with Shchur et al. in 2019, proposed a use of his data structure in *GWAS* context. In fact, they studied the associations between genetic variants in order to identify signals of natural selection and to build the so-called **ancestral recombination graph**, that contains complete information about history of samples.
For the sake of completeness, an example where *PBWT* is not used is **Ranbow**, proposed by Moeinzadeh et al. in 2020. The aim of this project was the haplotype reconstruction of polyploid genome from short read sequencing data, studying also the multi-allelic case.

Research goals

Therefore, for my master's thesis, I had to rethink the concept of *Matching Statistics* for *PBWT* (tracking a *row* of the panel instead of the *pos*), how to compute the *SLP* for the panel, how to use *thresholds*/*LCE queries* and how to get the same behaviour of the φ function, in order to obtain the **RLPBWT**, combining the ideas related to the *RLBWT* with those of the *PBWT*, eventually, as for example for the φ function, developing a simple new data structure. 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 PhD in order to be able to deepen the computer science potential in the biological context. Summarizing, the most important research goals of my PhD are:

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. A first step in this direction was made by Naseri with the already cited *m-PBWT*. Talking about spatial complexity, 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 the *LF-mapping*, with more than two alleles, too.
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 really 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* so, I should probably deal with parametric or approximate algorithms, based on researches already developed in *BIAS*.
3. **Genotype imputation & other goals**. As introduced, I have developed a wide-range interest in the application of computer science to biology. So, I do not intend to forget the actual use of the data structures that I will study and develop during my PhD. The management of missing and multiallelic data can further improve the state of the art of *GWAS* and of genotype imputation, allowing even more precise inference of unobserved genotypes. All this must be read in the perspective of a continuous rise of available data, not only regarding the human sequencing.
   I would also point out my interest in deepening the issues related to pattern matching, indexing structures and pangenome graph. During my PhD, it's my intention to focus on all the BIAS laboratory's topics of interest, from the point of view of bioinformatics and experimental algorithmics.

From a more technical point of view I mainly focused on the use of the **C++ programming language**, during both my bachelor and master thesis projects. I managed to do this because of the availability of efficient libraries, as the already cited *SDSL*, that has become a standard in bioinformatics. 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 project, including Christina Boucher (University of Florida) and Travis Gagie (Dalhousie University), in order to improve the quality of my PhD program.

References

[1] Jasmijn A. Baaijens, Paola Bonizzoni, Christina Boucher, Gianluca Della Vedova, Yuri Pirola, Raffaella Rizzi, and Jouni Sirén. Computational graph pangenomics: a tutorial on data structures and their applications. *Natural Computing*, 21(1):81–108, Mar 2022.

[2] Massimiliano Rossi, Marco Oliva, Ben Langmead, Travis Gagie, and Christina Boucher. Moni: a pangenomics index for finding mems. *bioRxiv*, 2021.

[3] Christina Boucher, Travis Gagie, I Tomohiro, Dominik Köppl, Ben Langmead, Giovanni Manzini, Gonzalo Navarro, Alejandro Pacheco, and Massimiliano Rossi. Phoni: Streamed matching statistics with multi-genome references. In *2021 Data Compression Conference (DCC)*, pages 193–202. IEEE, 2021.

[4] Richard Durbin. Efficient haplotype matching and storage using the positional Burrows–Wheeler transform (PBWT). *Bioinformatics*, 30(9):1266–1272, 01 2014.

[5] Simone Rubinacci, Olivier Delaneau, and Jonathan Marchini. Genotype imputation using the positional burrows wheeler transform. *PLOS Genetics*, 16(11):e1009049, Nov 2020.