PhD Proposal

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

Introduction

The problem of *pattern matching* is one of the most studied topics in the field of algorithmics and bioinformatics. For example the interest in such problems is due to the need to align sequences or search for specific patterns within the *DNA*.
In this context, a large number of data structure 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*.
Furthermore, in recent years, in the field of bioinformatics, there has been a change of interest. While until a few years ago the research was focused, as anticipated, on the study of a **linear sequence of a genome**, the researchers have begun to deepen the topic of the **pangenome**, which term was introduced by Tettelin in 2005. Infact 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, which have 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**.
From a biological point of view it is indeed interesting to note that, as pointed out by the study of *1000 Genome Project*, over 88 million variants, 84.7 million are **Single Nucleotide Polymorphisms (*SNPs*)**, 3.6 million are **short insertions/deletions (*indel*)** and 60000 **structural variants**, involving more than 50 nucleotides. All these variants are now a limit to the traditional use of a *linear sequence of a genome*.
Against this background, various algorithms and various data structures have been implemented in order to study *haplotypes* and *genotypes*, for example to study the **genotyping variants problem**. Briefly we could define *haplotypes* as a combination of allelic variants, inherited from a parent. Instead we can define the *genotype* as the complete set of genes contained in the DNA.
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 [2]. With 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 cases, in the original implementation. Furthermore, for example, variants of the *PBWT* have also been studied for handling the multiallelic case.
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 [3]. 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, regarding the *RLBWT*, has been made through the implementation of *PHONI* [4], where the **longest-common-extension (*LCE*) queries** are used to compute the *len* of *MS* in a single sweep over the pattern.
During the development of my master's thesis I worked in collaboration with the authors of *MONI* and *PHONI* in order to create a run length encoded variant of the *PBWT*, the **RLPBWT**, using and adapting the various theories developed for the *RLBWT*. In this context my PhD will be focused on the development of new algorithms in various topics related to open problems in the study of the *pangenome graph*, of the *haplotyping/genotyping* topics and *genome variants* problems. It is also my intention to deepen the more experimental themes relating to *pattern matching*, in detail the new developments on *BWT* and new indexing structures, as well as the theme of *succinct data structures*,

with particular attention to the use of *bitvectors*.

State of the art

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

BWT

The **Burrows-Wheeler Transform (*BWT*)** was introduced in 1994 in order to compress texts but it has had then wide use in bioinformatics, above all thanks to the already cited *FM-index*. Given a text T, \$-terminated, such that $|T| = n$, we can define, denoting with SA_T the *suffix array* of T, the BWT_T as: $BWT_T[i] = T[SA_T[i] - 1]$, if $SA_T[i] > 0$, and $BWT_T[i] = \$$ otherwise. Less formally we can say that $BWT_T[i]$ is the character that precedes the i-th suffix in the lexicographically order. It is important to note that this transform is reversible so we can reconstruct the text T from its transform BWT_T using the so-called **LF-mapping**. Given BWT_T and an array, called F_T, with all the characters of T in the lexicographically order, we can say that, thanks to the *LF-mapping*, the j-th occurrence of a certain character in BWT_T corresponds to the j-th occurrence of the same character in F_T, so we can reconstruct T starting from its last character \$. With the use of the *LF-mapping* we can perform the *backward-search* in order to use the BWT_T to look for a pattern P within T. This can be done efficiently thank to the *FM-index* which consists of two functions. The first one is C function, such that, given an alphabet Σ (that includes the ending character \$), $C : \Sigma \to [1, n]$. This function, given a character $\sigma \in \Sigma$ returns the number of occurrences of characters lexicographically minor than the one given as argument in T. The second one is the Occ function, $Occ : \Sigma \times [1, n] \to [1, n]$, which has as arguments a character $\sigma \in \Sigma$ and an index i of BWT_T and returns the count of occurrences of σ in $BWT_T[1, i]$.
The use of BWT has allowed the construction of efficient algorithms both in the field of pattern matching and in that of sequence alignment.

Bitvectors

Bitvectors are ones 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 possible on boolean arrays, such as *random access* in *constant time*. More in detail the *rank function* allows to calculate how many values of one are up to a certain index. Instead the *select function* allows to obtain the index of any 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 purely theoretical point of view, with the additional cost of $\mathcal{O}(n)$ bits in memory, these two operations can be supported with *constant time*. 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 to track the runs in the run-length encoded implementations of BWT and $PBWT$, where we put one at each head of run.

RLBWT

Speaking of *pangenome*, linear indexing via FM-index is no longer the best solution as it does not handle the large repetitions present in this new type of sequences in the best possible way. 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 rappresentation of consecutive

equal characters, the so-called *runs*. With this change of perspective the algorithms have gone 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, which was then introduced by Gagie et al., was called **r-index** and corresponds to the *RLBWT* and a *suffix array sampling* at the begin and at the end of every run. The algorithm for querying through the *RLBWT* take 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 also make use of **straight-line programs (*SLP*)**, for *random access* in *MONI* and for *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, $|M| = |P|$, 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 but neither $P[i - 1 : i + l - 1]$ nor $P[i : i + l]$ are. Furthermore, using a particular function called φ (and and its "inverse" φ^{-1}), based on the use of the **inverse suffix array (*ISA*)** as well, it was possible to find all the starting positions of all the copies of P in T from the starting positon of a match extracted by *MS*. More formally, given a starting position p, $\varphi(p) = SA[ISA[p] - 1]$, $NULL$ if $ISA[p] = 0$, and $\varphi^{-1}(p) = SA[ISA[p] + 1]$, $NULL$ if $ISA[p] = |T| - 1$, where $ISA[i] = j$ iff $SA[j] = i$.
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.

PBWT

Based on the theories of the *BWT* Durbin, in 2014, devised the **positional Burrows–Wheeler transform (*PBWT*)**, in order to solve the problem of pattern matching on panels (matrices) of haplotype. In detail he defined 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, for each column k, contains, for each position i, the haplotype of index m in the original panel. 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. Note that 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 (which we could also call "MEMs") from a new sequence z to X in $\mathcal{O}(M^2N)$.
Despite the fact that *PBWT* has been poorly regarded in the scientific community in the early years 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, for example, the **dynamic PBWT (*d-PBWT*)**, and in terms of its use to study haplotype panels, for imputations etc...

Studies on haplotypes

For the sake of completeness, it is necessary to cite some examples of studies on haplotypes.
A first example 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. In that case the *PBWT* was not used but the same could allow a better efficiency of the algorithm itself.
Another 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 an 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*.
Both these last two examples are limited in the bi-allelic case but they show the potential of *PBWT* in the study of haplotypes. However, it is important to note that in both cases the authors not only make use of *PBWT* but also other data structures are cited to achieve the same results.

Research goals

For my master's thesis I therefore tried to combine the ideas related to the *RLBWT* with those of the *PBWT*, creating the **RLPBWT**. In order to obtain this result I have to rethink the concept of *Matching Statistics* for *PBWT*, how to compute the *SLP* for the panel, how to use *thresholds/LCE queries*, how to obtain the same behaviour of the φ function etc...
In detail, regarding *MS*, instead of the *pos* we track a *row* of the panel. More formally, given a panel X (where every row is defined by X_j) and a pattern P, $|P| = X_{width} = n$, we define MS as an array of length n such that, for each position $1 \leq i \leq n$, $X_{MS[i].row}[i-MS[i].len+1 : i] = P[i-MS[i].len+1 : i]$ and $P[i - MS[i].len : i]$ is not a suffix of $X_1[1 : i], \ldots, X_{X_{height}}[1 : i]$. Talking about thresholds they are defined by the index of the minimum *LCP value* inside a run, considering also the *LCP value* of the head of the next run, if exists. Regarding the *SLP*, to get it, we stretch the reverse panel (from the right to the left) in in order to make *LCE queries* possible, having that *LCE queries* are made, between two rows, from the right to the left. Instead, for the φ function and its inverse, I have implemented a new simple data structure to identify which row is above and which row is below each row in the *RLPBWT matrix*, that is the panel already permuted via the prefix array and run-length encoded.
Obviously there are some limitations, such as the study of biallelic panels only and the lack of management of any missing data, which are very frequent in real cases. First of all it will be interesting to implement a new version of the *RLPBWT* that can handle multiallelic data, adapting the current use of bitvectors to manage the *LF-mapping* also with more than two alleles. On the other hand, it will be more complicated to manage the missing data. This problem is known to be *hard* so I should probably deal with parametric algorithms or approximate algorithms, based on researches already developed in *BIAS*.
Summarizing some of the research goals during my *Phd* will be:

1. *
2. *
3. *

From a more technical point of view, during both my bachelor and master thesis projects, I mainly focused on the use of the **C++ programming language**, due to the availability of efficient libraries, as the already cited *SDSL*, that have now become standard in the bioinformatics field. In addition to *C++*, however, I had the opportunity to deepen **Python**, with libraries such as *biopython*, and **Rust**, with recently developed libraries such as *bio-rust* (even if not yet complete from an algorithmic point of view).
Another point of interest is the analysis and development of efficient algorithms based on parallel computing, also on GPU, algorithms that are increasingly in use in both *bioinformatics* and *systems biology*.
To conclude this proposal I also point out the intention to maintain contacts with various researchers, initially obtained during the work on the project of the master thesis, including Christina Boucher (University of Florida), Travis Gagie (Dalhousie University) etc... in order to make my PhD program more complete.

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

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

[4] 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.

Essendo una prima bozza probabilmente è tutta da riscrivere, soprattutto in luce del "come vada scritta".
Note:

- il font dovrebbe essere corretto secondo le richieste. Non ho visto specifiche relative ai margini del file
- nell'introduzione mancano alcuni concetti relativi allo studio dei grafi nel caso pangenomico. Una volta chiariti gli scopri precisi del progetto di ricerca bisogna correggere (oltre che aggiungere eventuali aspetti non considerati e toglierne altri)
- nell'introduzione devo capire come ordinare meglio i concetti
- nell'introduzione forse servono esempi di casi d'uso relativi alle varie problematiche. Probabilmente è necessaria una parte più discorsiva relativa anche all'importanza biologica di questo genere di studi
- la conclusione dell'introduzione, dove si parla di cosa verrà eventualmente approfondito durante il PhD, è praticamente un *placeholder*
- l'incipit allo stato dell'arte è sicuramente da modificare
- la struttura dell'intera sezione riguardante lo stato dell'arte penso debba essere rivista in base a standard per la stesura della proposal che non conosco
- nello stato dell'arte la parte relativa alla BWT è commentata in quanto, pur avendola scritta, non la ritengo essenziale (soprattutto avendo a che fare con un limite di 4 pagine)
- nello stato dell'arte la sezione relativa ad alcuni esempi di studio sugli aplotipi deve essere profondamente rivista in luce soprattutto dei research goals. Un esempio potrebbe essere la citazione, ad esempio, di HapCol, qualora i goals si spostassero un po' dalla PBWT e dai suoi usi più immediati
- per ignoranza mia su come vada scritto questo tipo di documento non sono sicuro che alcuni dettagli formali nei vari passaggi dello stato dell'arte siano sensati da mettere. Stesso discorso vale nell'incipit dei research goals
- la citazione di quanto svolto per la tesi magistrale non so se sia necessario, e, qualora lo fosse, se la collocazione all'inizio dei research goals sia corretta, ne tantomeno se sia troppo riassunto/esteso
- mancano le specifiche dei fini della ricerca, da chiarire coi docenti (e da cui dipendono diversi punti della proposal)
- la conclusione del research goals non so se sia necessaria
- per quanto riguarda la scelta delle reference nel file `.bib` ne sono presenti diverse. In ogni caso, per la prima selezione fatta, ritengo le prime due necessarie mentre la terza e la quarta sono per diversi punti di vista superflue. Avendo un bound di cinque citazioni c'è comunque spazio di manovra