UNIVERSITÉ DU LUXEMBOURG

PhD-FSTM-2020-20
Faculty of Science, Technology and Medicine

DISSERTATION

Defence held on 18/05/2020 in Luxembourg
to obtain the degree of

DOCTEUR DE L'UNIVERSITÉ DU LUXEMBOURG
EN INFORMATIQUE

by

Maria Isabel Mou Sequeira Fernandes

Reconciling data privacy with sharing in next-generation genomic workflows

Dissertation defence committee

Dr. Yves Le Traon, Chairman
Professor, University of Luxembourg

Dr. Reinhard Schneider, Vice-chairman
Professor, University of Luxembourg

Dr. Paulo Esteves-Veríssimo, Supervisor
Professor, University of Luxembourg

Dr. Francisco Couto
Associate Professor, University of Lisbon

Dr. Erman Ayday
Associate Professor, Case Western Reserve University

Acknowledgements

I would like to thank several people who contributed to the PhD work in this thesis:

- My supervisor Prof. Paulo Esteves-Veríssimo for all the advice and guidance. It was a pleasure to learn from him.
- My advisors Prof. Francisco M. Couto and Dr. Jérémie Decouchant for their constant availability for helping me and for their advice. It was a pleasure to work with them and to learn from them.
- Prof. Reinhard Schneider for being a member of my thesis evaluation committees and for the constructive comments on my work.
- My colleagues from the CritiX group for the all the TGIF meetings and the other moments that made the PhD path enjoyable.
- My friends for all the fun and complementary social life during all the PhD journey.
- My parents for the constant support during all the journey. Obrigada por me apoiarem sempre. São os melhores!
- My sister for showing and reminding me that the simple things are what make us happy. Não são precisas palavras para comunicar, adoro-te.

Abstract

Privacy attacks reported in the literature alerted the research community for the existing serious privacy issues in current biomedical process workflows. Since sharing biomedical data is vital for the advancement of research and the improvement of medical healthcare, reconciling sharing with privacy assumes an overwhelming importance. In this thesis, we state the need for effective privacy-preserving measures for biomedical data processing, and study solutions for the problem in one of the harder contexts, genomics. The thesis focuses on the specific properties of the human genome that make critical parts of it privacy-sensitive and tries to prevent the leakage of such critical information throughout the several steps of the sequenced genomic data analysis and processing workflow. In order to achieve this goal, it introduces efficient and effective privacy-preserving mechanisms, namely at the level of reads filtering right upon sequencing, and alignment.

Human individuals share the majority of their genome (99.5%), the remaining 0.5% being what distinguishes one individual from all others. However, that information is only revealed after two costly processing steps, alignment and variant calling, which today are typically run in clouds for performance efficiency, but with the corresponding privacy risks. Reaping the benefits of cloud processing, we set out to neutralize the privacy risks, by identifying the sensitive (i.e., discriminating) nucleotides in raw genomic data, and acting upon that.

The first contribution is DNA-SeAl, a systematic classification of genomic data into different levels of sensitivity with regard to privacy, leveraging the output of a state-of-the-art automatic filter (SRF) isolating the critical sequences. The second contribution is a novel filtering approach, LRF, which undertakes the early protection of sensitive information in the raw reads right after sequencing, for sequences of arbitrary length (long reads), improving SRF, which only dealt with short reads. The last contribution proposed in this thesis is MaskAl, an SGX-based privacy-preserving alignment approach based on the filtering method developed.

These contributions entailed several findings. The first finding of this thesis is the performance $\times$ privacy product improvement achieved by implementing multiple sensitivity levels. The proposed example of three sensitivity levels allows to show the benefits of mapping progressively sensitive levels to classes of alignment algorithms with progressively higher privacy protection (albeit at the cost of a performance tradeoff). In this thesis, we demonstrate the effectiveness of the proposed sensitivity levels classification, DNA-SeAl. Just by considering three levels of sensitivity and taking advantage of three existing classes of alignment algorithms, the performance of privacy-preserving alignment significantly improves when compared with state-of-the-art approaches. For reads of 100 nucleotides, 72%

have low sensitivity, 23% have intermediate sensitivity, and the remaining 5% are highly sensitive. With this distribution, DNA-SeAl is 5.85× faster and it requires 5.85× less data transfers than the binary classification – two sensitivity levels.

The second finding is the sensitive genomic information filtering improvement by replacing the per read classification with a per nucleotide classification. With this change, the filtering approach proposed in this thesis (LRF) allows the filtering of sequences of arbitrary length (long reads), instead of the classification limited to short reads provided by the state-of-the-art filtering approach (SRF). This thesis shows that around 10% of an individuals genome is classified as sensitive by the developed LRF approach. This improves the 60% achieved by the previous state of the art, the SRF approach.

The third finding is the possibility of building a privacy-preserving alignment approach based on reads filtering. The sensitivity-adapted alignment relying on hybrid environments, in particular composed by common (e.g., public cloud) and trustworthy execution environments (e.g., SGX enclave cloud) in clouds, gets the best of both worlds: it enjoys the resource and performance optimization of cloud environments,while providing a high degree of protection to genomic data. We demonstrate that MaskAl is 87% faster than existing privacy-preserving alignment algorithms (Balaur), with similar privacy guarantees. On the other hand, Maskal is 58% slower compared to BWA, a highly efficient non-privacy preserving alignment algorithm. In addition, MaskAl requires less 95% of RAM memory and it requires between 5.7 GB and 15 GB less data transfers in comparison with Balaur.

This thesis breaks new ground on the simultaneous achievement of two important goals of genomics data processing: availability of data for sharing; and privacy preservation. We hope to have shown that our work, being generalisable, gives a significant step in the direction of, and opens new avenues for, wider-scale, secure, and cooperative efforts and projects within the biomedical information processing life cycle.

Keywords: Raw genomic data, privacy, reads alignment, sensitivity levels, information sharing.

Contents

Abstract **vii**

1 Introduction **1**
1.1 Genomic information processing 2
1.1.1 The human genome 2
1.1.2 DNA sequencing and analysis 5
1.2 Towards cloud-based federated processing 6
1.3 Problem statement 7
1.4 Contributions 9
1.5 Overview 11

2 Related work **13**
2.1 Biomedical infrastructures for large-scale collaboration 13
2.2 DNA alignment and variant calling 14
2.3 Short reads versus long reads 15
2.4 Threats to genomic privacy 16
2.4.1 Genomic privacy attacks 16
2.4.1.1 Re-identification attacks 17
2.4.1.2 Membership attacks 18
2.4.1.3 Inference attacks 19
2.4.1.4 Recovery attacks 20
2.4.2 System exploits 21
2.5 Privacy protection methods for genomic data 23
2.6 Privacy-preserving approaches for genomic data 26
2.7 Short reads filtering 28
2.7.1 Bloom filters 29
2.7.2 Short reads filtering 31
2.8 Genomic data repositories 32
2.8.1 1000 Genomes Project 33

2.8.2 iDASH contest . . . 34

3 Sensitivity levels for genomic data 35
3.1 Methods . . . 37
3.1.1 Sensitivity levels for genomic data . . . 37
3.1.1.1 Sensitivity levels disconnection . . . 38
3.1.1.2 Cloud diversity to prevent inference . . . 42
3.1.2 Classification of reads into sensitivity levels . . . 43
3.1.2.1 Reads classification . . . 44
3.1.2.2 Filters conflict management . . . 45
3.2 Evaluation Setup . . . 46
3.2.1 System model . . . 46
3.2.2 Threat model . . . 46
3.3 Results . . . 47
3.3.1 Sensitivity levels statistics . . . 47
3.3.2 SNP promotions . . . 49
3.3.3 Performance improvement . . . 50
3.3.4 Privacy improvement . . . 53
3.4 Summary . . . 56

4 Long reads filtering 57
4.1 Sensitive short reads detection limitations . . . 58
4.2 Methods . . . 62
4.2.1 Sensitive nucleotides excision for genomic reads . . . 62
4.2.1.1 Sensitive sequences generation . . . 64
4.2.1.2 Long reads filtering . . . 67
4.2.2 Multiple Bloom filters to improve accuracy . . . 68
4.2.3 Tolerating sequencing errors . . . 69
4.3 Evaluation Setup . . . 70
4.3.1 System model . . . 70
4.3.2 Threat model . . . 71
4.3.3 Hardware . . . 71
4.3.4 Software . . . 71
4.3.5 Data . . . 72
4.4 Results . . . 72
4.4.1 Sensitive nucleotides detection . . . 72
4.4.2 False positives rate . . . 73
4.4.3 Tolerating sequencing errors . . . 74
4.4.4 Memory consumption . . . 76

4.4.5 Performance comparison 77
4.5 Summary 78

5 Alignment of masked reads **79**
5.1 Enclaves 80
5.2 Methods 81
5.2.1 Reads filtering and masking 83
5.2.2 Masked reads alignment 85
5.2.2.1 Plaintext alignment algorithm selection 85
5.2.3 Alignment score refinement 86
5.2.3.1 Score refinement and alignment extension 86
5.2.3.2 Smith-Waterman algorithm 87
5.2.3.3 Running SW algorithm in SGX enclaves 88
5.3 Evaluation Setup 89
5.3.1 System model 89
5.3.2 Threat model 90
5.3.3 Hardware 90
5.3.4 Software 91
5.3.5 Enclave initialization and encrypted communications 91
5.3.6 Simulated reads 91
5.3.7 Algorithms evaluation criteria 92
5.3.7.1 Alignment algorithms selection 92
5.4 Results 93
5.4.1 Masked reads alignment time 93
5.4.2 Score refinement time 94
5.4.3 Computation time per MaskAl step 94
5.4.4 Memory consumption 95
5.4.5 Network communications 97
5.5 Summary 98

6 Conclusions and future work **99**
6.1 Conclusion 99
6.2 Future work 101

Terminology **106**

References **107**

List of Figures

1.1 Human DNA. 2
1.2 DNA sequencing and analysis. 6
1.3 Contributions. 10

2.1 A simple example using Bloom filters. 30
2.2 Variant Call Format. 33

3.1 Initial sensitivity levels. 38
3.2 LD-based promotions. 39
3.3 Inference-based promotions. 40
3.4 Proportion of inferred SNPs. 42
3.5 Information stored in two clouds. 43
3.6 Classification of reads in sensitivity levels. 44
3.7 Promotions between sensitivity levels. 47
3.8 Genomic privacy – sensitivity levels. 54
3.9 Genomic privacy – HG03048. 54
3.10 Genomic privacy – HG01086. 54
3.11 Genomic privacy – HG00096. 55
3.12 Genomic privacy – HG01864. 55
3.13 Likelihood ratio value – sensitivity levels. 56

4.1 Proportion of reads containing at least one sensitive nucleotide. . . 61
4.2 DNA reads filter filtering. 63
4.3 Neighbours SNPs study. 64
4.4 Number of missing sensitive nucleotides – genomic variations combination. 65
4.5 Genomic variations combinations. 66
4.6 Non-overlapping window and sliding window approaches. 67
4.7 Partitioning of reads. 69
4.8 Partitioning of reads with errors. 70
4.9 Proportion of sensitive nucleotides – GVs only. 73

4.10 Proportion of sensitive nucleotides – STRs only. 73
4.11 False positive proportion comparison. 74
4.12 Proportion of reads filtered. 74
4.13 Proportion of sensitive using multiple Bloom filters (2% error rate). 75
4.14 Proportion of sensitive using multiple Bloom filters (30-mers). . . . 75
4.15 Bloom filter size comparison. 76
4.16 Throughput comparison. 77

5.1 SGX overview. 81
5.2 MaskAl overview. 83
5.3 Alignment time – BWA vs LAST. 86
5.4 Alignment accuracy – BWA vs LAST. 87
5.5 Alignment time comparison. 93
5.6 Score refinement time comparison. 94
5.7 Computation times per step of MaskAl. 95
5.8 Memory usage comparison. 96
5.9 Network communications comparison. 97

List of Tables

2.1 Genomic privacy attacks – goal and information used. 22
2.2 Advantages and limitations of existing privacy protection methods. 26

3.1 SNPs per sensitivity level. 48
3.2 Number of inferred SNPs per inference and promotion. 49
3.3 Privacy, performance and communication overheads of the alignment algorithms used. 51
3.4 Computation overhead of existing privacy-preserving approaches. . 52
3.5 Communication overhead of existing privacy-preserving approaches. 52

4.1 Proportion of sensitive nucleotides detected per error rate. 62

5.1 Smith-Waterman Score Matrix and Traceback Matrix. 88

Chapter 1

Introduction

Privacy refers to an individual's right to keep his/her personal matters secret. In a biomedical context, in particular in genomics, privacy is enforced by protecting biomedical data, since they might reveal sensitive information about its donor or his/her relatives. Genomic information processing involves intensive computation steps in order to reconstruct an individual's genome from his biological sample. Traditionally, the sensitive parts of genomic information are only determined after these analysis steps. However, those steps being computation intensive, they are commonly offloaded to public clouds to achieve high performance at affordable costs.

Therefore, privacy-preserving methods applied during these computations must treat all genomic data alike. Either as all sensitive, which incurs significant performance costs, or as all non-sensitive, which leaves genomic data unprotected or weakly protected during the analysis. During the window of vulnerability opened by the latter alternative, adversaries may access genomic data and learn about the sensitive information it contains.

The present thesis aims at overcoming this dilemma, allowing high performance processing whilst protecting genomic information throughout the whole workflow: right after its translation from a human's biological samples into digital sequences through sequencing technologies, and during its analysis steps, mainly alignment and variant calling.

The thesis evaluates the following hypothesis, creating the mechanisms to demonstrate it:

The detection and classification of sensitive information using a multi-level scale, and subsequent and immediate excision of the sensitive sequences from raw genomic data, allows fast and secure processing using traditional alignment al-

gorithms, at the cost of a slight decrease in accuracy. However, such accuracy decrease can be fully recovered by later processing of the missing sensitive information in a protected environment, by authorised users.

1.1 Genomic information processing

Let me start by providing the biomedical background necessary for the understanding of the contributions of this thesis.

1.1.1 The human genome

The human genome encodes all the information a human body requires to live, develop and reproduce, organized in 23 pairs of chromosomes. Each chromosome is a double helix of DNA composed of four subunits called *nucleotides*. There are four nucleotides in DNA, each designated by the first letter of the name: A for Adenine, T for Thymine, C for Cytosine, and G for Guanine. Nucleotides pair together in a predetermined way in what is called a ***base pair*** (A links to T and C links to G) to create the double helix (see Figure 1.1(a)). The human genome has a total length of approximately 3 thousand million nucleotides.

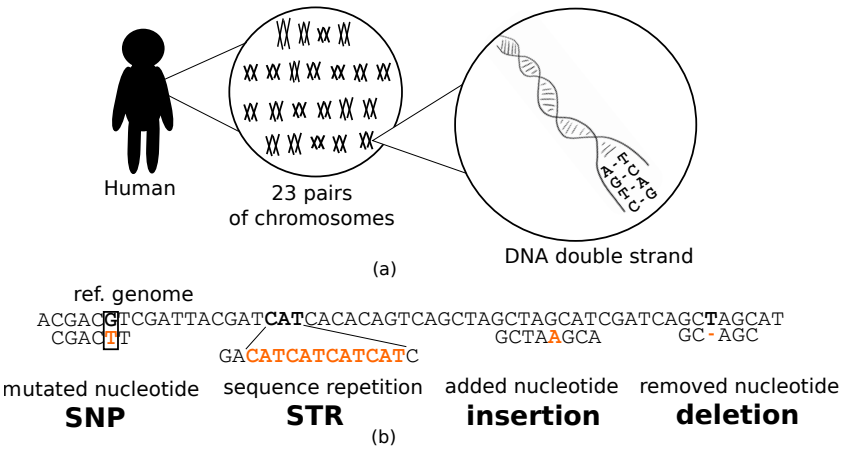

Figure 1.1: **Human DNA**. (a) DNA organization. (b) Genomic variations types.

The *genome sequence* is the sequence of nucleotides in an individual's genome. Any two individuals share approximately 99.5% of their genome sequence. The remaining 0.5% is what makes each individual unique [Ayd+15].

A *genomic variation*, or mutation, is identified when a newly sequenced genome differs form all previous ones in a precise location (i.e., locus). From the 1000

Genomes Project, it is known that an individual's genome contains between 2100 and 2500 genomic variations [Con15]. At each locus, considering a single DNA strand each genome has one of the possible alleles (i.e., an individual can have a 'T', or a 'G' in chromosome X at position 155,259,899), which are the possible nucleotide sequences at that locus. The majority of the loci has two possible alleles, which are then called major allele and minor allele. The major allele is the allele that appears the most frequently in the population. On the other hand, the minor allele is the rarest sequence, typically appearing in less than 5% of the population. Minor alleles are the most relevant for studies since they can influence, for example, the health status of an individual [Con05]. Genomic variations can be classified into three types (see Figure 1.1(b)):

(i) A single nucleotide polymorphism (SNP) occurs when a single nucleotide differs in a genome locus between an individual and the reference genome;

(ii) A short tandem repeat (STR) defines a region where a sequence of 1 to 6 nucleotides is repeated a variable number of times for distinct individuals;

(iii) Insertions, respectively deletions (Indels) occurs when one or more nucleotides are added, respectively removed or present, (or present, resp. absent), as compared to the reference genome.

The genome is transmitted hereditarily and this transmission occurs in segments, which causes genomic variations in neighbouring locations to often be correlated. In humans, the size of the blocks varies between few kilobases[1] (kb) and 100kb; however, the majority of the correlations is found in a smaller range (5–20kb) [WP03]. *Linkage disequilibrium* (LD) measures one type of correlation found in the human genome, and it defines the non-random cooccurrence of alleles from different loci. In other words, LD defines the correlation between genomic variations. LD can be measured using the correlation coefficient[2] r^2, which is often used to infer genomic variations from a set of observed variations. Markov Chains have also been used by some methods for genetic imputation, which consists in the inference of hidden (or non-observable) genomic variations based on a set of observed variations and population statistics [Li+10].

DNA is privacy sensitive information due to its identifying nature. It contains long-lived, static, and identifying information, which is sensitive for the lifetime of

[1] 1 kilobase corresponds to 1,000 nucleotides.

[2] Correlation coefficient r^2 – is a measure commonly used for LD, which represents the probability of a particular allele is present or absent in a first locus and another particular allele is present or absent in a second locus.

its owner. Furthermore, some portions of genomic data stay sensitive even longer, since they are transmitted to descendants. This second property – heredity – is what makes DNA sensitive for an extremely long duration linking individuals to their relatives. In addition, DNA encodes other critical information, such as genetic diseases predisposition. This can be used for valuable purposes, such as personalized medicine or to help solving crime cases. For example the Golden State Killer case was solved in 2018, 42 years after the crime was committed. The killer was captured based on DNA collected from the crime scene through a genealogy search using GEDmatch [HT]. Nevertheless, unintended leakage of genomic information fragilizes individuals, e.g., being used for health insurance denial based on disease predisposition deducted from unauthorized DNA analysis. There was a case of an insurance company that refused a woman's application due to her high predisposition for developing breast cancer [Kni]. Finally, DNA is irreplaceable, which makes it even more privacy critical and valuable, since once disclosed the harm done is irreversible.

The primary goal of privacy breaches on genomic data is to discover sensitive information contained on, or inferred from, the observed genomic information. In order to do so, the adversary obtains partial information, such as the genome sequence of a target individual, for example by accessing it from a public cloud. Then, the adversary can combine that genomic data with additional information from other sources, mostly available in public databases, social networks and so forth, such as population statistics, and participants details (e.g., surname, age, ZIP code). By crossing these informations the adversary can perform privacy attacks. The privacy attacks on genomic data can be classified into four classes: re-identification attacks, membership attacks, inference attacks, and recovery attacks. The classification of the attacks depends on the information the adversary wants to obtain, such as identity, participation in studies, hidden genetic information or discovery of the genome sequence of target individuals [Swe00; Hom+08; NYV09; Kon+08]. For instance, in re-identification attacks, the goal of an adversary is discover the identity of a target individual though the correlation between personal identifiable information and the genomic information [Gym+13; Hum+15]. Such attacks highlighted the privacy issues associated to genomic data and alerted the research community to develop privacy-preserving solutions [Con+15; DFT13]. Therefore, the prevention of privacy attacks was a main tenet of the design of the privacy-preserving approach proposed in this thesis, in particular, inference and re-identification attacks.

1.1.2 DNA sequencing and analysis

In the human body, DNA encodes all the information needed for life, which includes inheritance, coding for proteins and the instructions for the biological processes occurring in the body. In order to decode the information on the DNA and understand its role, the research community developed sequencing techniques. In the traditional short reads sequencing, a physical DNA sample is first cut in smaller chunks through chemical reactions. Then, the obtained chunks are submitted to an amplification step[3] were they are multiplied. In order to be detected by short reads sequencing technologies, the sample to be sequenced needs to contain multiple copies of the DNA region to be sequenced since DNA is microscopic. Then, all these chunks are introduced into a sequencing machine, which reads the sequences of nucleotides therein and converts them into their digital equivalent, called reads(indeed, as the reader may guess, sequences of letters, combinations from the A,T,C, G set, as explained before).

Since the sequencing of the first human genome – finished in April 2003, the production and study of biomedical data, in particular human genomic data, have grown at an unprecedented rate. Such data supports clinical studies and research procedures, including personalized medicine [MM08; Gup08; DB10]. This was possible due to the first sequencing method developed by Sanger [SC75]. Starting from 2005, high-throughput Next-Generation Sequencing (NGS) technologies, also referred to as second-generation sequencing machines, have been rapidly evolving. Those NGS technologies are mainly characterized by an affordable cost (about 1,000$ per genome) and high throughput production of short reads (30–400 nucleotides) with low error rates (0.1–1%). Due to those features, the raw genomic data production increased from initially megabytes [ODS13] to terabytes. Genomic data processing rapidly turned into a big data challenge, and consequently performance bottlenecks during the genomic data analysis have appeared. This highlighted the need for high performance analysis workflows. Later, around 2011, a third generation of sequencing technologies (e.g., PacBio SMRT, Oxford Nanopore [Ip+15]) was released with the goal of producing longer reads ($>$ 1,000 nucleotides), although at the cost (in current technology) of higher error rates (around 15%) [KGE17]. Long reads improve some analyses, such as de novo assembly and structural variants detection. In addition, long reads sequencing eliminates of the amplification step which is time consuming and is a bias source [Ama+20].

In order to discover genomic variations from raw genomic data produced by the above sequencing technologies, two computationally-intensive analysis steps

[3] Amplification step – called polymerase chain reaction (PCR) is a process used in laboratories to produce thousand millions of copies of a source DNA region.

are required as a follow-up: *alignment*, to obtain the location of the reads in the genome by matching them with a reference genome; and *variant calling*, to discover the nucleotides in the reads that differ from the reference genome. Figure 1.2 shows the DNA sequencing and analysis workflow up to the variant calling step.

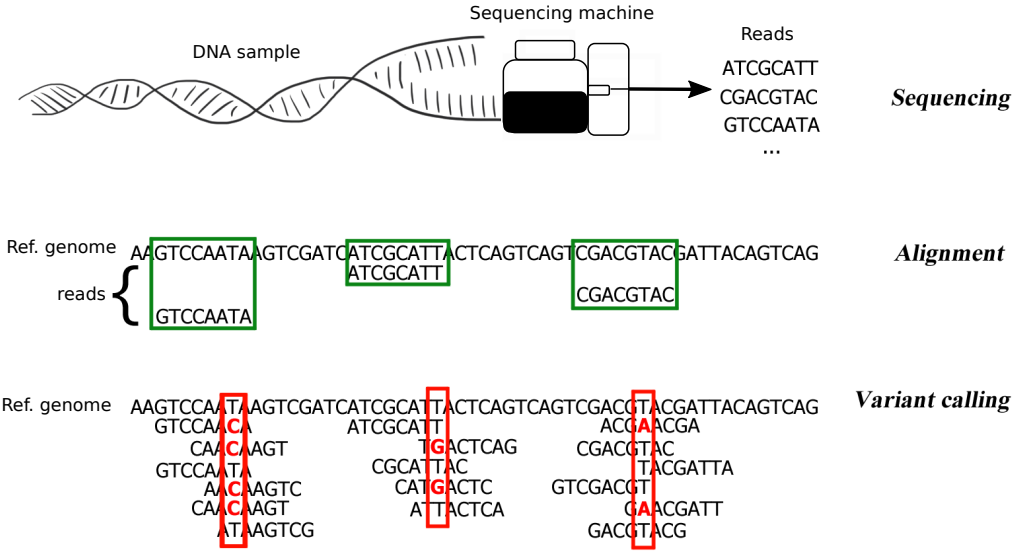

Figure 1.2: **DNA sequencing and analysis**.

1.2 Towards cloud-based federated processing

Sequencing technologies allowed the production of huge amounts of reads, generating terabytes of raw genomic data. That data needs to then be analysed through alignment and variant calling steps. Since the amount of raw genomic data is huge and its processing is computationally costly, often this kind of data analysis is outsourced to cloud environments. Such environments, provide high available and cost efficient computational resources which often are not existent or available among the local resources of biomedical centers.

In addition to genomic data processing in the cloud to achieve high performance, data sharing is essential in genomic data field. Data sharing accelerates data collection and it promotes new discoveries [Kay+09].

Despite all the benefits of migrating genomic data processing to the cloud and data sharing, they also bring new challenges: privacy issues. In general, privacy breaches lead to confidential data disclosure that can be used for re-identification. In answer to this, the research community has been advocating the need for a

secure collective environment where institutions (even competing ones) can collaborate to achieve more complete datasets to promote significant discoveries and personalized healthcare. Esteves-Veríssimo and Bessani called it the *e-biobanking vision* [VB13], an efficient privacy-preserving cooperative environment whose purpose is storing, sharing, and processing digital biomedical data, whilst preserving the interests, rights and needs of the individual stakeholders. The main goal of such environment is to promote use of genomic data in its many application fields, such as personalized medicine and forensics. However, genomic data contains a high level of sensitive information about its owner. When sensitive information is leaked, privacy is violated. Building efficient privacy-preserving processing environments is about preventing sensitive information leaks of genomic data, as well as providing practical operational performance.

1.3 Problem statement

Several privacy attacks on genomic data have been reported in the past decade. Those attacks alerted the research community to the existing privacy issues in current biomedical processing, such as information leakage (e.g., attributes, identity). In order to be able to prevent information leaks, we must understand the properties of the human genome that make it particularly privacy sensitive and the privacy attacks reported on genomic data. The human genome is particularly privacy-sensitive due to the following characteristics: (i) DNA can be used for the re-identification of individuals, since it barely evolves over an individual's life and it is not replaceable; (ii) it reveals the individual's predisposition to genetic diseases; (iii) it records familiar relations, from ancestors to descendants, including siblings' information; (iv) it can reveal the individual's response to medicines which, if misused, can lead to bad consequences; and (v) genomic data can be modified or forged, which, in case it occurs, can result in the arrest of innocent people. To give and example, one could replicate the genome of an individual, introduce it in a crime scene, and, therefore, influence an ongoing investigation. The misuse of sensitive genomic information can cause unwanted harm, such as discrimination, insurance denial and employment refusal [Kni; KAC14]. If the genomic data is acquired by some unauthorized entity, it can be exploited for the interests of that same entity. Employment, social and educational discrimination based on genetic tests for Huntington's disease have also been reported [Goh+13]. Furthermore, the harm caused is irreversible and it can also affect relatives, since they are genetically linked.

In addition to the data-based genomic privacy attacks, which target the data

properties, privacy-preserving solutions should also take into account the system dimension: *there is no trusted data over untrustworthy systems*. As such, system-based attacks, leveraging vulnerable computing systems supporting biomedical information processing workflows, may completely defeat the most sophisticated privacy-preserving algorithm. For example, once an adversary performs a successful intrusion in such a system, he can gain unconstrained access to confidential biomedical and genomic data. This confidential data can, for example, be sold on the black market [San; Gar], or used directly by bio savvy attackers, to perform data-based genomic privacy attacks themselves.

This serious problem has often been minimized in the prior state of the art in data-based privacy-preserving solutions for biomedical/genomic data. Therefore, we additionally claim as a problem to solve, the need for the design of *system-aware privacy-preserving* solutions, that is, holistic solutions that encompass both the data plane privacy (resilience to privacy attacks) and the system plane security and dependability (resilience to faults and attacks on the system structure).

The work developed in this thesis attempts to address the above-mentioned privacy and security issues of biomedical data, specifically genomic sequences. As a first attempt to the early detection and segregation of the sensitive parts of the human genome, the work of Cogo et al. [Cog+15] classified genomic data in a binary scale (sensitive or insensitive) segregating the sensitive part in an automatic way, to highly protected parts of the processing data center.

However, distinct regions of the genome provide different kinds of information, from simple physical traits to disease predisposition. From the privacy point of view, this information can lead to different privacy breach levels with different consequences. Therefore, it would be interesting to define more levels to classify genomic data by level of sensitivity. A specific goal of this thesis is then to improve the s.o.t.a., by designing a privacy-preserving approach for raw genomic data providing sensitivity-adapted alignment and storage, while guaranteeing data privacy protection and practical performance. That is, for each sensitivity level, different methods can be applied to address the protection and performance requirements. The first step towards this goal is the development of a sensitive information classification method, which builds on and refines the filtering approach described in [Cog+15]. The intuition is that the introduction of a filtering step in the traditional analysis workflow could allow early stage protection of genomic data. The main idea of the filtering approach is the identification and excision of sensitive information from the (digital) human genome.

In addition, since only the known-sensitive information requires extra care with regard to privacy, and it is a small percentage of the genome, we can strike a per-

formance/protection trade-off, in terms of the methods and of the environments to execute the workflow in. For example, the most protective methods (e.g., heavy cryptography-based methods), which are usually more costly, can be applied to the most sensitive information instead of the full genome. Likewise, it is possible to adjust the cost, efficiency and privacy guarantees of different analysis environments (e.g., public clouds, private clouds, enclave clouds and trusted execution environments) accordingly to the data sensitivity.

Finally, in order to include the filtering step in the analysis workflow, the assessment of the impact of this step on the performance and accuracy of the analysis is required. This thesis proposes a sensitivity-adapted alignment approach relying on a hybrid clouds environment and it evaluates the privacy protection improvement, while providing high performance for the whole genomic data analysis workflow.

The holistic system-aware privacy-preserving approach we advocate comes evident from the combination enunciated above, between data processing methods and computational environments.

1.4 Contributions

The contributions of this thesis are the following:

- Contribution 1: Definition of sensitivity levels to classify and protect genomic data.
- Contribution 2: Development of a filtering approach for sensitive information compatible with the existing sequencing data (both short and long reads).
- Contribution 3: Development of a privacy-preserving reads alignment approach using masked reads.

Figure 1.3 represents the contributions of this thesis along the genomic analysis workflow. The main steps of the workflow are connected by the horizontal arrows, while the contributions are represented inside the rectangles. As the figure demonstrates this thesis focuses on the filtering of raw genomic data, right after it is sequenced (Contributions 1 and 2) and on the alignment step (Contribution 3). The path to protecting genomic information during the variant calling step is unveiled by our present work, and envisioned as part of future work.

Following the described research plan, the developed works have been published in major international conferences and journals of the area. Here is the complete list of publications related to this project:

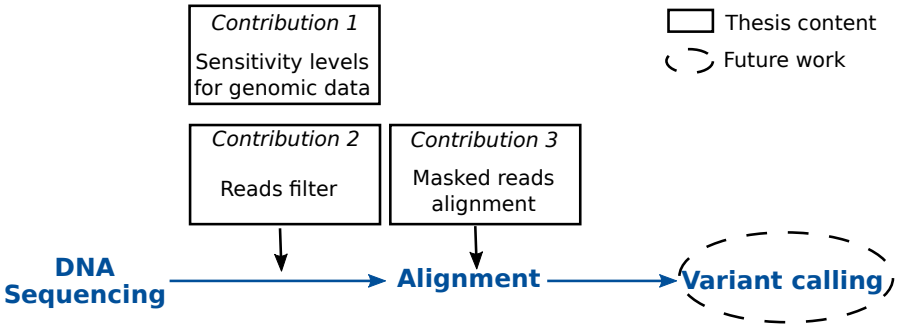

Figure 1.3: **Contributions**. Representation of the contributions along the genomic data analysis workflow.

2017 **Cloud-assisted read alignment and privacy**
Maria Fernandes, Jérémie Decouchant, Francisco M. Couto, Paulo Veríssimo
In Proceedings of the 11th International Conference on Practical Applications of Computational Biology & Bioinformatics (PACBB), pages 220–227

2017 **How can photo sharing inspire sharing genomes?**
Vinícius Vielmo Cogo, Alysson Bessani, Francisco M. Couto, Margarida Gama-Carvalho, Maria Fernandes, Paulo Veríssimo
In Proceedings of the 11th International Conference on Practical Applications of Computational Biology & Bioinformatics (PACBB), pages 74–82

2017 **Enclave-based privacy-preserving alignment of raw genomic information**
Marcus Völp, Jérémie Decouchant, Christoph Lambert, Maria Fernandes, Paulo Veríssimo
In Proceedings of the 2nd Workshop on System Software for Trusted Execution (SysTEX), article 7

2018 **Accurate filtering of privacy-sensitive information in raw genomic data**
Jérémie Decouchant, Maria Fernandes, Marcus Völp, Francisco M. Couto, Paulo Veríssimo
In Journal of Biomedical Informatics, volume 82, pages 1–12

2018 **MaskAl: Privacy preserving masked reads alignment using Intel SGX**
Christoph Lambert, Maria Fernandes, Jérémie Decouchant, Paulo Veríssimo
In Proceedings of the 37th Symposium on Reliable Distributed Systems (SRDS), pages 113–122

2020 **DNA-SeAl: Sensitivity levels to optimize the performance of privacy-preserving DNA alignment**
Maria Fernandes, Jérémie Decouchant, Marcus Völp, Francisco M. Couto, Paulo Veríssimo
In IEEE Journal of Biomedical and Health Informatics, volume 23, issue 3, pages 907–915

1.5 Overview

This section summarizes the content of the remaining chapters of the present thesis.

Chapter 2 reports the related work. First, it introduces the genomic background, which compiles the main biological terms used and the properties of the human genome. Then, it describes the traditional workflow for sequenced DNA analysis. The following three sections of this chapter are devoted to the privacy related background, which includes privacy attacks on genomic data, privacy protection methods and it also presents some examples of privacy-preserving approaches for specific genomic data processing tasks. After the genomic privacy overview, this chapter introduces the approach used as starting point for the work in this thesis. Finally, the last section introduces the data sets and repositories used on the experiments.

Chapter 3 introduces *DNA-SeAl* – a sensitivity levels classification based on the privacy risk of DNA information. Guidelines for the creation of a stratification system based on qualitative and quantitative features of genomic data are provided. This chapter also shows how to create the levels based on quantitative features (e.g., allele frequency, linkage disequilibrium links); the creation of the levels based on qualitative features (e.g., disease genes, physical appearance) shall be left for a future project. DNA-SeAl method regarding performance based on the computational time and communication cost, as well as genomic privacy metric and a likelihood-ratio test were also evaluated. Genomic privacy measures the weighted risk of re-identification based on the minor allele frequency of the observed genomic variations, while the likelihood-ratio test compares case and control populations and it represents the upper bound of the membership detection of a case individual. The method proposed in this chapter improves privacy while providing a practical performance, with decreased computational and communication costs in comparison with previous approaches [Sch09; Bar+12; Cog+15]. Using three sensitivity levels instead of the previous two levels already improves privacy while ensuring a feasible performance.

Chapter 4 describes a sensitive filtering approach for human DNA reads. Section 4.1 describes the limiations of the previous filter approach [Cog+15] and the possible solutions. Improvements, such as the compatibility to long reads, which are produced by the most recent sequencing machines, are also showcased. In addition, this chapter details how to build the database of sensitive information and how to initiate the filter. Finally, this thesis provides a comparative evaluation of previous and proposed approach, which shows that the filter classifies 10% of the information as sensitive with a true percentage of 3% of sensitive information. However, it does not present false negatives.

Chapter 5 introduces *MaskAl* – a masking and alignment methodology for DNA reads using Intel Software Guard Extensions (SGX). Section 5.1 introduces enclaves and why they are considered secure. Then, Section 5.2 introduces MaskAl approach, which has two main steps, the first is the detection of sensitive information, followed by the sensitivity-adapted alignment of the reads. MaskAl is a hybrid approach which relies on public and enclave clouds. The last section of this chapter evaluates the performance of MaskAl regarding the alignment accuracy, memory consumption, alignment time and network communications. The evaluation was performed considered error free reads, and reads with an error rate of 1% and 2%.

Chapter 6 summarizes the conclusions and future work.

Chapter 2

Related work

This chapter is organized as follow. In the first section, this chapter describes the ideal privacy-preserving cooperative environment for biomedical data that several works envisioned, highlighting its main properties and giving some examples that aim at developing such an environment. In the next section, it provides a detailed description of two important DNA analysis steps: alignment and variant calling. The third section of this chapter introduces the main threats to genomic data, which can be of two types: privacy attacks and system exploits. The following section gives an overview of the privacy protection methods commonly applied on genomic data and it presents some practical application examples. A brief comparison of short and long reads features and applications is then provided, highlighting the main features and applications of each type of reads. Thereafter, the state-of-the-art filtering approach for short reads is described in detail, including the description of the data structure used, Bloom filters. This approach serves as basis for the work in Chapter 4. In the last section, this chapter briefly introduces existing genomic data repositories and describes the data sets used for the work in this thesis.

2.1 Biomedical infrastructures for large-scale collaboration

The *e-biobanking vision* [Kay+09] defines an efficient privacy-preserving cooperative environment whose purpose is to store, share, and process biomedical data. In this environment, different entities share their data and findings, contributing to larger datasets and extensive studies. Ideally, this system ensures that biomedical data is protected and it offers tools to securely analyse them [VB13;

Bes+15]. The key properties foreseen for this system include: (i) easy and secure access to the data; (ii) resistance to cloud outage and other failures; and (iii) data integrity and data privacy. Besides this list, some authors also consider data quality, confidentiality, security, access control as essential properties for ensuring privacy-preserving data sharing [Kno14].

Several publicly available repositories aimed at concretizing the e-biobanking vision, resulting from collaborations between researchers and institutions (e.g., International Sequence Database Collaboration (nucleotide sequences), UniProt Consortium (protein sequences), 1000 Genomes Project (human genomes)), which promote high data availability [LM16]. However, data availability increases the vulnerability to privacy attacks, which must be prevented. NGS-Logistics [Ard+14] defines a federated ecosystem for privacy-preserving rare genomic variations analysis across geo-distributed data. More recently, the Beacons Project [GH16], developed by the Global Alliance for Genomics and Health, emerged as a privacy-preserving genetic variations sharing framework, which allows queries to be executed on a large network of genomic information repositories and datasets. The answers are restricted to "Yes" or "No" to prevent the inference of identity or additional information, whose goal is to limit the knowledge a query gives to a potential adversary. Some privacy attacks have highlighted some vulnerabilities of this platform, and by consequence of the early stage privacy-preserving collaborative environments [SB15]. These attacks have been addressed and the vulnerabilities mitigated [Rai+17a]. Keeping in mind the performance challenges associated with the singular nature of DNA and the reported privacy breaches, developing efficient privacy-preserving methods for large-scale genomic data sharing is a non trivial problem.

2.2 DNA alignment and variant calling

Alignment is the step where the reads produced by the sequencing machine are matched with a reference genome. This step allows the determination of the original position of the sequences contained in the reads. Alignment algorithms are mainly based on hash tables (e.g., BLAST) [Alt+90] and on suffix arrays (e.g., BWA, Bowtie) [LD09; Lan+09]. The first category of alignment algorithms follows the seed-and-extend paradigm, where a K-mer[1] is used as a seed[2] to find an exact match in a hash table (seed phase). For each match found, the neighbour nucleotides of the seed are consecutively compared to the reference in order to

[1]K-mer – is a sequence of K nucleotides.
[2]Seed – is a small subsequence of the reads to be aligned.

find the best original location for the full read (extend phase). The second category performs the alignment using a binary search[3] on the suffix arrays. The main two performance advantage of this approach are that it requires less memory and repeated regions only need to be aligned once [LH10]. More recently, dynamic programming have been used by alignment algorithms to achieve better performance [SW81].

Aligning reads in untrusted environments, such as public clouds, is now common due to their affordable price and great computational power that allows high performance computations [Ser19]. However, public clouds require the user to trust that the cloud provider has implemented privacy protection mechanisms, and that an intruder would never gain access to its data. Some works reported the risk that using such infrastructures for biomedical data processing entails [MPG14; FEJ15]. Taking this into account, alignment can be considered the first privacy critical step of the genomic analysis workflow after sequencing, since an adversary might obtain sensitive information if the data is not adequately protected. In the last years, some privacy-preserving alignment algorithms emerged. The main goal of those algorithms is to prevent an adversary from observing reads during their alignment. Section 2.6 provides a more detailed discussion of those algorithms.

Variant calling aims at finding the differing genomic sites when comparing a reference genome and the sequenced genome. This thesis focuses on the alignment step. More precisely, this thesis proposes, MaskAl, a hybrid excision and alignment method, which combines processing in an untrusted environment – for insensitive information – with processing in a trusted execution environment, – for sensitive information. The accuracy and performance comparison between MaskAl and state-of-the-art plaintext alignment algorithms is the object of Chapter 5. Considering how to extend our approach to the variant calling step is part of future work.

2.3 Short reads versus long reads

The year of 2005 marked the development of high-throughput next-generation sequencing technologies, which produce thousands of short reads (digital sequences with a length comprised between 30 and 400 nucleotides) with low error rate (0.1-1%). This generation of sequencing technologies enabled the existing rich ecosystem of short reads sequencing technologies and algorithms.

With the constant evolution of the sequencing technologies, the third-generation of sequencing machines was released around 2011 [KGE17], introducing long-reads

[3]Binary search – is an efficient searching algorithm used to find a value in a sorted array.

(sequences of thousands of nucleotides), which nowadays can already contain upwards of 60,000 nucleotides [PAC]. However, these long-reads show a significantly higher error rate than their short reads predecessors (around 15% against the 1% in short reads, both upper bounds) [KGE17]. In addition, long reads require a shorter sample preparation time. Currently, both short and long-reads are used by researchers as their applications are complementary. Short reads sequencing technologies are more precise than the ones producing long reads, with error rates between 1% and 0.1% [KGE17; Met10], they are inexpensive, and they generate massive quantities of reads. In addition, short reads are widely used, representing over 90% of the market share [Cam16]. Furthermore, short and long-reads have distinct applications, which contribute to their complementarity. For example, long reads are particularly useful for the resolution of high repetitive and complex regions of the genome and for the detection of structural variants (i.e., segmental duplications, gene loss) [Pol+18]. While short reads can be used for fragmented DNA – for which long reads are not suitable. The introduction of long reads raises new challenges for genomic data analysis, such as the creation of adapted algorithms able to process such long reads, the development of error correction methods to balance the high error rate characteristic of long-reads, and the secure storage of long-reads which contain more genetic information then their precedents. Some alignment algorithms have been created for long reads; however, they are still in earlier stage of development [Sov+16]. Nowadays, there are only a few algorithms for alignment of long reads [Che+12; PB17].

2.4 Threats to genomic privacy

Genomic data is subject to two types of threats: privacy attacks, and system exploits. Briefly, the first type of threats focuses on properties of genomic data and their goal is to discover sensitive information about one or more target individuals and/or their relatives. The second type refers to system vulnerabilities that are used to access sensitive information, which then can be used to perform privacy attacks. The following subsections describe in detail the main features and goals of each threat type.

2.4.1 Genomic privacy attacks

In the last years, several privacy attacks on genomic data have been reported in the literature, demonstrating the inefficiency of simple anonymization methods. The main goal of a privacy attack is to discover some sensitive information based

on the observation of the complete or partial genomic sequence of an individual. To obtain such sensitive information, several techniques have been applied, such as inference methods and quasi-identifiers exploitation for identity discover [EN14]. An attack can be classified into one of the following classes depending on the kind of sensitive information the attack wants to achieve: re-identification attacks, membership attacks, inference attacks, and recovery attacks. Those classes of genomic privacy attacks are described in the following sections, and for each class some literature examples are presented.

2.4.1.1 Re-identification attacks

Re-identification attacks are designed to link anonymous sequences to their owner. Sweeney [Swe00] showed that demographic information can be used to identify people. In this particular study, the author showed that 53% of the American population could be uniquely identified using only three attributes (place, gender and data of birth). Malin et al. [Mal06] developed IdentiFamily, a software to link de-identified genomic data and named individuals using genealogical relations. Goodrich et al. [Goo09] took advantage of genetic tests leaking information and showed that it is possible to determine the identity of a person in a limited number of attempts. Gymrek et al. [Gym+13] showed that it is possible to link genomic data and surnames based on the Y chromosome's genetic information and genealogical information publicly available. The authors de-anonymized 131 genomes from the 1000 Genomes Project by corresponding genetic information and surnames. Humbert et al. [Hum+15] demonstrated how to de-anonymize genomes combining SNPs information and phenotypic features (e.g. gender, blood type, skin color). Lippert et al. [Lip+17] developed an algorithm based on maximum entropy, which matches genomic samples with phenotype, assessing which ones can be originated by the same individual. They showed that this method is able to re-identify individuals from populations composed of different ethnicities. Zaaijer et al. [Zaa+17] proposed MinION sketching, a fast and inexpensive strategy for human DNA samples re-identification based on MinION sequencing. This work showed that with 60-300 randomly selected SNPs and using Bayesian inference methods, it is possible to infer the identity of anonymized genomic samples. During the inference, reference population statistics and known SNPs statistics are taken into consideration.

This thesis prevents re-identification attacks by: (i) disconnecting the information from different levels in order to avoid inference between levels, preventing inference of more sensitive information from least sensitive observed information;

(ii) excising sensitive information in raw genomic reads, which could reveal the identity of its owner and other privacy sensitive information; and (iii) aligning sensitive information inside a trusted environment, which content is assumed to be not observable by an adversary.

Trail attacks

Trail attacks are a variant of re-identification attacks performed in distributed networks, where the goal is to re-identify a target individual through the combination of his unique features, collected from different repositories in the network (e.g., independent hospitals) [MS04]. For example, consider an individual that participates in different studies, in which he contributes with different genomic regions. If an adversary collects data about that individual from several studies, he might obtain enough information to re-identify him. Malin [Mal02] described the REIDIT algorithms, which use deterministic methods to compute how re-identification can be performed assuming independent releases of data. The approach assumes that each release is a subset of a full dataset, which contains different attributes. In addition, DNA sequenced data must not be released jointly with identifying attributes, e.g., name and social security number. The power of trail attacks is lower than that of other attacks described in this section, since they are restricted to data collected from studies data releases, which are very controlled. For example, if the genomic data is not present in at least one of the collected releases, it is not possible to deploy such kind of attacks.

2.4.1.2 Membership attacks

Membership attacks try to guess the presence of an individual in a study. In general, the majority of the attacks in this class correlates information about a target individual's genotype or SNPs profile with population statistics of two groups: a reference population and a target group (e.g., the case group of a particular diseases study). Homer et al. [Hom+08] proposed a statistical method that compares the proximity of an individual to a target group and to the general population. The performed computations use SNP expression or allele frequencies as input data. Wang et al. [Wan+09] improved Homer's attack adding alleles correlations (e.g., linkage disequilibrium links) to the statistics computed. They showed that with few hundred SNPs it is possible to determine the presence of an individual in the case group. One similar attack was proposed by Braun et al. [Bra+09], who used empirical tests to determine the participation of an individual in a particular group. The knowledge required for this attack includes the genotype of

the target individual and the marginal allele frequencies of the tested group. Jacobs et al. [Jac+09] used a likelihood-based statistical framework to determine membership based on genotype frequencies and individual genotypes. With this approach the authors were able to determine if an individual or some close relative participated in a particular genome-wide association study (GWAS) study. The used statistic compares the logarithm of the genotypes frequencies. Sankararaman et al. [San+09] proposed a tool – SecureGenome – and showed that summary statistics of a GWAS can be used to discover the participation of an individual in that study. The tool assesses if an individual's genotype is part of a pool of genotypes (membership attack) based on the allele frequencies of the pool and on the allele frequencies of a reference independent dataset. The goal of the tool is to advise the release of SNPs information, preventing membership attacks. Clayton et al. [Cla10] perform a membership attack based on a Bayesian approach. Cai et al. [Cai+15] show how to determine the membership of an individual in a GWAS study, using his genetic information and the study statistics for sets of less than 25 SNPs. This showed that from the Beacons Project framework, it is possible to discover the membership of individuals by querying a set of 250 SNPs [SB15]. This attack was later improved by von Thenen et al. [TAC18], who achieved the same result with less than 0.5% of the queries used by the precedent work. The key of this improvement is the use of SNPs correlations, such as linkage disequilibrium and higher order SNPs relations. The higher order relations are inferred using high-order Markov chains. A re-identification attack using long range familial relations (relatives information) was performed by Erlich et al. [Erl+18]. This attack shows that familiar relations can strongly contribute for individuals re-identification. Backes et al. [Bac+16] demonstrated that further information, such as expression data, can also be used to perform membership attacks. In this attack the authors used microRNA expression data to determine the participation of a target individual in a study group.

In the evaluation of the classification of sensitive information into multiple levels – DNA-SeAl – we have considered the partition of each sensitivity level information, followed by the distribution of the data partitions over multiple locations. The partitioning was designed so that individuals from two populations, i.e., case and control, could not be distinguished.

2.4.1.3 Inference attacks

Inference attacks focus on discovering hidden or non-observed genomic information, based on partial or incomplete genomic information. These attacks proved the inefficiency of gene hiding methods to protect sensitive information. Nyholt et

al. [NYV09] determined Prof. Watson's susceptibility of developing Alzheimer's disease based on its neighbouring regions of the APOE gene, even though the gene information was removed from Prof. Watson's released sequence. Wang et al. [Wan+09] showed that it is possible to recover SNPs information using integer programming, which can then be linked with diseases. Gitschier [Git09] demonstrated how to infer the Y chromosome haplotypes[4] of two men based on genealogical information. After, inferring the haplotypes, the author discovered that using a similar method it is possible to infer the surnames of some of the CEU[5] participants. He et al. [He+18] combined publicly available genomic information and traits released by the individual or their relatives, for example shared by a GWAS, to predict hidden genotypes and traits. Samani et al. [Sam+15] described an attack using high-order correlations, such as recombination rate, diploid genotypes, to disclose hidden SNPs. Ayday et al. [AH17] proposed an inference attack where the partial genome of an individual or genomic data from his relatives are combined with phenotypic information to discover hidden or non-observable genomic information. A different attack, based on gene expression data instead of the genomic information, was proposed by Schadt et al. [SWH12]. In their attack, the authors show that the predicted genome, obtained from the gene expression data, can be used to identify individuals in large populations.

This thesis contributes to the prevention of a specific kind of inference attack, called **amplification attack**, which was taken into account during the development of the stratification method for sensitive information (Chapter 3). In an amplification attack, an adversary has access to some information about a target individual (e.g., SNPs related to the physical appearance), for instance through a successful attack of another kind, which is then used to infer more sensitive information (e.g., SNPs related to diseases predisposition risk). In addition, the filtering approach proposed in this thesis prevents inference attacks by hiding the regions of a genomic sequence which are commonly used as input information for such attacks.

2.4.1.4 Recovery attacks

The goal of recovery attacks is to reconstruct the genomic sequence of a target individual. For this kind of attacks, the adversary uses statistics and allele frequency information combined with publicly available data. Once the adversary

[4]Haplotype – is the group of alleles that are inherited together from one parent. It refers to the genetic information contained in one of the two double helices of the DNA.

[5]CEU – is a particular population of the HapMap project, which is composed by Utah residents with Northern and Western European ancestry.

knows the sequence, re-identification and membership attacks could also be performed [Zho+11]. Kong et al. [Kon+08] showed that it is possible to infer haplotypes of individuals based on the genetic information of their relatives. Fredrikson et al. [Fre+14] showed that it is possible to infer an individual's genetic markers from its pharmacogenetic model (used in personalized medicine for development of drugs based on the patients genetics) in combination with demographic information from the individual. Deznabi et al. [Dez+18] described how to discover hidden genomic regions based on familial relationships, special features of genomic data and publicly available phenotype information, such as physical traits and disease related information. Recovery attacks are out of the scope of the present thesis, however, when executed they can possibly enable an adversary to perform the other described classes of attacks. This thesis does not consider this last type of attacks because we assume that the publicly available data does not include the individual's genome, who is being sequenced, or the genome of one of his relatives.

To conclude this section, Table 2.1 summarizes the goal and the information required by each class of attack.

2.4.2 System exploits

In addition to the genomic privacy attacks, biomedical data systems are also subject to exploits and intrusions, which might lead to unwanted users' data exposure. Computer security is the field responsible for the prevention of such exploits and intrusions.

Computer systems security is associated to three main areas: confidentiality, integrity, and authentication. Confidentiality is achieved by preventing unauthorized access to the information in the system. Integrity ensures that the information is not modified by unauthorized entities in a way those modifications are not detected. Authentication allows the verification of the identity of the users. Other concepts often considered important in computer security include: access control, availability, and privacy. Briefly, access control guarantees that the users access only the information and resources they are entitled to. Availability ensures that the system is available in a certain moment. This is often achieved with redundancy. Privacy assures that the users information is controlled by themself. In other words, the user is aware of what information is shared and he/she grants the assess to it, knowing the purpose of its collection and how it is processed [Rus94].

In the biomedical data field, the computers used by the hospitals to support data analysis and storage are often considered isolated systems. However, in reality they are not as often isolated as they are assumed to be. Therefore, during the

Table 2.1: Genomic privacy attacks – goal and information used.

Re-identification attacks	
Goal	Establish the link between an individual's genomic information (e.g., SNPs set, genotype, full genome) and his identity.
Information used	Reference population statistics, SNPs statistics, identifiers and quasi-identifiers (e.g., surname, social security number, ZIP code), genomic information of the target individual, demographic information, long range familial relations, phenotypic information, mitochondrial DNA.
Membership attacks	
Goal	Determine if a target individual participates in a particular group, e.g., the case group in a study of a particular disease.
Information used	Reference population statistics, group of interest statistics, genomic information of a target individual.
Inference attacks	
Goal	Infer hidden or non-observed information based in partial or incomplete genomic information.
Information used	Partial genomic information of a target individual, publicly available genomic information (e.g., SNPs statistics, reference genome), gene expression data.
Recovery attacks	
Goal	Discover genomic information from a target individual.
Information used	Partial genomic information of a target individual, relatives genomic information, familial relations, phenotypic information.

development of privacy-preserving solutions the risk of exploits and intrusions to which the systems are subject must be considered.

As referred in the introduction, biomedical data processing is commonly outsourced to clouds in order to achieve high performance. This refers to another kind of systems, called cloud systems, which can be considered during the solutions design. The main idea of cloud systems is resources and data sharing through the internet. In particular for biomedical data, cloud systems are often used to store data and to perform data analysis.

Although cloud environments present several benefits, they are not completely safe. For example, they do not ensure that the service provider or an intruder do

not access the data [RC11]. Zubairu [Zub19] analysed and discussed the risks of storing biomedical data in clouds. The main advantages of processing biomedical data in cloud environments are the following: high resources availability, which allows parallelization and consequently fast processing and great efficiency; cost effective computational resources; and the delegation of maintenance and management to the cloud provider. However, on the other hand, the use of clouds to process biomedical data has some drawbacks, including, but not limited to: both cloud provider and client are responsible for the management, control and security of the data; authorization should be strong and adequate since cloud resources and information can be accessed remotely; and cloud failure can happen since cloud resources are shared, therefore, the client should take precautions (e.g., create a backup copy in other location) to avoid data losses.

Given the challenge of providing privacy-preserving solutions for biomedical data storage in the cloud, some proposed solutions include DepSky [Bes+13] and CHARON [Men+19], which are two cloud-of-clouds systems designed for secure data storage. These approaches benefit from public clouds for data storing and sharing while ensuring data security and dependability.

Trusted execution environments (TEEs), discussed later in this chapter, are solutions to ensure confidentiality and integrity of the code or data that is processed inside them, by isolating them from the main processor. The privacy-preserving alignment approach proposed in this thesis uses TEEs to prevent unauthorized data access to sensitive information.

2.5 Privacy protection methods for genomic data

In recent years, several privacy protection strategies have been developed to address the privacy requirements of genomic data highlighted by the privacy attacks. However, achieving a good level of protection while ensuring practical performance is not a trivial task, since privacy and performance are often contradicting goals, e.g., privacy protection inevitably adds a performance overhead. In other words, privacy-preserving algorithms are slower than plaintext algorithms, which achieve high performance by not providing any protection of the data. Existing genomic privacy protection methods can be classified into one of four categories: (i) access control [EN14; MMC19]; (ii) de-identification methods [MS00; MS04]; (iii) data augmentation methods [LHA02; Mal05b]; and (iv) cryptography-based methods [Goo09; Kan+08]. Table 2.2 summarizes the advantages and limitations of the four categories of privacy protection methods.

Access control allows the customization of the access rights by defining who can access the data and what can do with it [EN14]. Used alone it does not really protect data, since its main role is to manage data accesses. However, it can be used as a complementary method.

De-identification methods, such as anonymization and pseudonymization, encrypt, remove, or hide sensitive information (i.e., quasi-identifiers, such as name, zip code, social security number) to protect the identity of the data donor [Mal05b]. k-anonymity is a commonly used anonymization principle, which states that the information of an individual is shared by at least $k-1$ individuals in the dataset, making them indistinguishable [Swe02; ED08]. Such methods cannot guarantee sufficient protection due to their inability to deal with re-identification problems [Mal05a; Hay13; SB15].

Data augmentation methods commonly rely on the generalization or obfuscation to protect DNA reads. The main goal of these methods is to make each record indistinguishable from others. Generalization lattices, in the context of genomics, can be defined as the generalization of pairs of DNA reads by representing them with a common sequence [Mal05b; LWS12]. Differential privacy is often used in the publication of aggregate information. This method adds noise to the data before its release. The resulting statistics cannot be distinguished whether or not a single individual is removed from the dataset [VG16; Nav+15; EN14]. This is an efficient method for genomic privacy protection, where the higher the privacy protection, the lower the accuracy and data utility. Therefore, they imply significant data utility loss, which can compromise the analysis. The privacy-preserving approach proposed in this thesis is classified into this category given that the proposed approach excises the sensitive regions of DNA reads, which can be assimilated to obfuscation.

The privacy protection methods described until now have been considered inefficient, since they do not completely prevent re-identification attacks described in the literature or they compromise the data utility. Therefore, other privacy protection methods, cryptographic privacy-preserving methods, have been developed. These are the most secure methods up to date, since they prevent the access to the plaintext data.

Cryptography-based methods encrypt the genomic information, not disclosing the raw data while they allow data mining. Methods in this category include secure multi-party computations (SMC), homomorphic encryption and garbled circuits. SMC are the basis of secure collaborations between institutions (e.g., hospitals, universities, research centers). They consist of the joint computation of some function using as input the data of all participants, while the data is kept private [Hua+11; JKS08]. Homomorphic encryption is a cryptographic scheme that

allows performing operations on cyphertexts without decrypting. The operations performed generate encrypted results, which need to be decrypted in order to be interpreted. Atallah et al. [AKD03] compute the edit distance between sequences using homomorphic encryption. Garbled circuits allow two-party secure computations, in which the input data and the intermediate results are not learned by the receiver [Bar+12]. Cryptography-based solutions are considered secure and are, therefore, widely used. However, they do not provide sufficient protection for the complete lifetime of genomic data since ciphers can be broken in a shorter time [Hum+13]. In addition, they allow a limited number of operations on the encrypted data [Bar+12]. Thus, cryptographic methods are restricted to some applications, such as disease susceptibility tests [NTP16], computations of genomic frequencies and χ^2 [KL15], and secure datasets querying [Çet+17]. Furthermore, cryptography-based methods are limited by their long computation time and high communication costs [Bar+12]. The described limitations of cryptography-based solutions, indeed restrict the acceptance by researchers, making them continue relying on the state-of-the-art algorithms, i.e., genomic data processing in plaintext.

In addition to the aforementioned methods, hardware and software solutions have also been considered for privacy protection. **Trusted execution environments** (TEEs) are used to securely perform some operations without revealing intermediate results. They can be used, e.g., to decrypt sensitive genomic data, perform operations on it, and re-encrypt the data and the results before revealing the results. TEEs create a secure section of the main processor. The goal of this component is to provide confidentiality and ensure integrity of the processing done inside it. Intel's Software Guard eXtensions (SGX), also called enclaves, are one common kind of TEE used nowadays – they create an isolated area, like a "black box", whose content is private and isolated from the outside environment. Thus, an enclave allows privacy- preserving data processing. The main drawback of enclaves is the limited secure memory available (128MB per CPU for Intel SGX enclaves). However, other options are available in the marked which could be used, such as ARM TrustZone. The ARM TrustZone allows the creation of two environments (trusted and untrusted), which run simultaneously in a single core [ARM]. Regarding the protection provided by TEE, several side channel attacks have exposed vulnerabilities of those components [Sch+17; Bra+17; Göt+17]. However, mitigation methods for those side channel attacks were also proposed in [Sch+17] and [Göt+17].

This thesis uses SGX enclaves as the selected trusted execution environment. In particular, SGX enclaves are used to protect sensitive information that is stored inside them, as well as, the intermediate results and final results of processes run

inside them.

Method	Features	Limitations
Access control	Limits the access to authorized users.	Data remains unchanged.
	Collects the access logs for audit purposes.	Does not provide privacy guarantees.
De-identification	Removes personal sensitive information.	Applied alone does not provide sufficient privacy protection.
Data augmentation	Obfuscates or hides sensitive information.	Genomic links need to be considered during the obfuscation.
	Introduces randomness/noise to protect sensitive information.	Significant data utility loss.
Cryptography-based	Ensures privacy of input and output data.	Allows only limited operations on the encrypted data (addition or multiplication).
	Operations can be performed on the encrypted data.	Long computation time.
		High communication costs.

Table 2.2: Advantages and limitations of existing privacy protection methods.

2.6 Privacy-preserving approaches for genomic data

This section provides some practical examples of privacy-preserving algorithms that apply the methods described in the previous section. In the biomedical field, such algorithms have been developed for applications such as reads obfuscation, secure reads alignment, information leakage measurement, and secure genetic testing, among others.

Reads obfuscation: The method in this category focus on the early detection of privacy sensitive information combined with information hiding methods. Existing works in the area describe the manual partitioning of genomic data into sen-

sitive and insensitive [Ayd+14], or by using a hypothetical external tool [Zha+11]. Ayday et al. [Ayd+14] proposed a distributed architecture for aligned reads querying, which masks the non-queried reads regions to avoid releasing further information. Cogo et al. [Cog+15] proposed a classification for short reads (30 nucleotides long), which distinguishes reads with genomic variations from those without, called respectively, sensitive and insensitive reads. This classification method uses a dictionary with the known genomic variations (i.e., SNPs, STRs and disease genes) and it is useful for the development of adapted DNA analysis frameworks. This thesis builds upon and extends this work by presenting the first automated reads classifier adapted to existing sequencing technologies (both short and long reads). Existing sequence classifiers distinguish only sensitive from insensitive information. However, this classification is not sufficient [Ard+14; DDK16], due to several limitations that we discuss in Chapter 4. The classification presented in this thesis, therefore, distinguishes multiple sensitivity levels (Chapter 3).

Secure alignment: The main goal of approaches in this category it to perform the alignment step while protecting data privacy. Existing solutions rely on one of the protection methods discussed in the previous section or a combination of them. Zhang et al. [Zha+11] presented a hybrid clouds approach, which assumes a hypothetical tool for classifying privacy sensitive and insensitive data. This approach aims at benefiting from public and private clouds while mitigating their drawbacks. Chen et al. [Che+12] proposed a privacy-preserving alignment approach using public and private clouds. The main principle of this approach is the conversion of sensitive plaintext data into hash values, which can be processed safely in public clouds. Balaur [PB17] is a privacy-preserving alignment approach based on locality sensitive hashing. In order to work, this approach requires large amounts of memory and significant network bandwidth (Gigabytes). The most efficient alignment algorithms that have been proven secure are based on cryptographic techniques, such as garbled-circuits [Hua+11] and homomorphic encryption [CKL15; KL15]. To conclude, the application of cryptography-based alignment algorithms is limited due to their high computational and communication costs. Chapter 5 shows how efficient but unsecure alignment algorithms can be used for privacy-preserving genomic information processing by exposing only insensitive information to this algorithm.

Impact of information leakage: Measuring the impact and associated risk of information leakage associated with genomic data is important to design efficient privacy-preserving solutions. However, this subject is still in its infancy. Although privacy metrics for genomic data have been reported and compared, they are in their majority not intuitive and/or based on refined genomic data (e.g., SNPs) [Wag15]. GenoShare [Rai+17b] is a risk measurement framework to

support genomic data sharing, which assesses the information leakage associated to inference attacks. When a query is made, the engine runs the three most relevant attacks on genomic data based on the information the adversary can access in addition to the information he gets in case the query is granted. Then, it computes the risk associated with that query. Humbert et al. [Hum+17] assessed the privacy loss of an inference attack due to kin privacy[6], comparing metrics for the quantification of genomic and health privacy. Recently, a study assessed the sensitive information leakage associated to expression genomic data [HG18].

Secure genetic testing: The approaches in this group focus on privacy-preserving testing of genomic data to assess the predisposition to genetic diseases. PRINCESS [Che+17b] is a privacy-preserving rare diseases analysis framework that uses Intel SGX enclaves for secure computations across geo-distributed data. This framework uses lightweight cryptographic primitives and data compression to achieve secure computations and communications. PRESAGE [Che+17a] is a hybrid framework for genetic testing supported by SGX. The main techniques used are cryptographic protocols and minimal perfect hashing scheme.

Existing privacy-preserving solutions for genomic data enable several applications as described in this section, however, they have some limitations. In addition, all privacy-preserving solutions proposed in the literature protect the data either when it is stored or during its processing, such as alignment and sharing. Before this thesis work, to the best of our knowledge, there is only one approach that protects genomic reads as soon as they are produced [Cog+15] (described in Section 2.7), which serves as basis for the work in Chapter 4 of this thesis. Regarding privacy-preserving processing of genomic data, this thesis also presents a stratification system for sensitive information (described in Chapter 3), which enables the adequate attribution of different kinds of alignment algorithms according to the sensitivity of the data. Moreover, the proposed stratification system is a way to obtain better performance.

2.7 Short reads filtering

This thesis proposes a filtering approach based in a previous work developed by Cogo et al. [Cog+15]. This section describes this previous work, a method to detect sensitive DNA in reads of 30 nucleotides based on a Bloom filter. This section first introduces the Bloom filters as a data structure and then describes

[6]Kin privacy – refers to the familiar privacy, which can be compromised by the disclosure of the genomic data of any member of the family.

the short reads filtering approach propsoed in [Cog+15].

2.7.1 Bloom filters

A Bloom filter [Blo70] is a probabilistic data structure which is space and time efficient due to the association of keys or indices to each value introduced in the Bloom filter. The keys are generated using hash functions which take as input the element to be inserted in the filter. This property of Bloom filters make them efficient regarding the search task, since the same hashes are used when an element is searched in the filter. Therefore, the searching time of Bloom filters is O(1), that is constant complexity. For more details on Bloom filters, we refer the reader to a comprehensive survey of Bloom filters and their different applications written by Broder and Mitzenmacher [BM04].

Bloom filters [Blo70] do not produce false negatives, however, they can produce false positives. In other words, when a Bloom filter does not detect an element, it means that the element is absent of the set – no false negatives. On the other hand, if a Bloom filter detects an element, it means that the element might be in the set, and there is a probability that the element is – false positives. The main principle of the Bloom filters is the implementation of a bit table and hash functions for mapping of elements into the bit indices. The false positive rate, p, is usually set by the user in the Bloom filter initialization step. This parameter defines the maximum number of incorrect results the Bloom filter can return. Another parameter defined in the Bloom filter initialization is n, the number of elements to be inserted in the filter. This parameter is used to compute the space (in bits) required for the Bloom filter, in order to contain all the elements the user wants to insert. The two key equations of bloom filters are the following: $p = \left(1 - e^{-\frac{kn}{m}}\right)^k$ and $k = \frac{m}{n}\ln(2)$, where p is the false positive rate, m is the size in bits of the Bloom filter, k is the number of hash functions the filter uses, and n is the number of elements to be inserted in the filter.

Looking into Bloom filters in more detail, the two main phases of Bloom filters based processing are the initialization of the Bloom filtering and its querying. The initialization is the step where the elements are inserted in the bloom filter, which uses hash functions to determine the indices in the filter where the element will be added. The search step uses the same hash function(s) used in the initialization to compute the indices where the searched element would have been stored. Then, if the locations in the filter corresponding to the searched hashes are not empty, the element might be in the filter.

Figure 2.1 presents an example of a Bloom filter based processing using a single

hash function. Here, the Bloom filter is represented as a bit vector with twelve cells and corresponding indices. Initially, all the cells are empty (with value 0), in this example represented in white. Figure 2.1(a) shows the initialization of a Bloom filter, where the element 'ATCTCGCAC' is added to the Bloom filter. The initialization starts with the computation of the index at which the sequence is added, using a hash function, $h()$. Once the index is computed, the element is added by setting to 1 the vector at index defined by the hash (in the figure the cells with value 1 are represented in green). Figure 2.1(b) represents the search step. First, the hash of the searched element 'ATCTCGCAC' is computed using the same hash function $h()$, which gives the index that should be checked in the filter. Secondly, the computed index is searched in the vector and if it is filled (green), the element might be in the filter. Otherwise, if the searched index is empty (white), the element is definitely not in the filter.

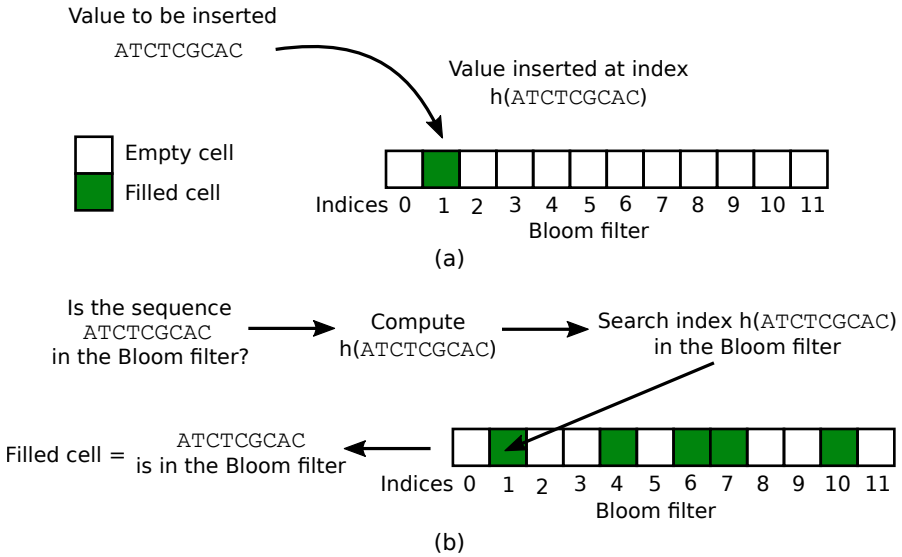

Figure 2.1: **A simple example using Bloom filters.** (a) Bloom filter initialization. h() represents a hash function. (b) Bloom filter search.

From the privacy point of view, Bloom filters are considered unsecure since they might allow the inference of the original data based on the encoded data they contain [Kuz+11]. However, some works relied on Bloom filters combined with data protection methods to build privacy-preserving approaches [SB17]. Schenell et al. [SBR09] combined Bloom filters and identifiers encryption to provide a privacy-preserving record linkage system. Durham et al. [Dur+14] proposed a system based on Bloom filters for secure record linkage, which is resistant to frequency-based cryptanalysis attacks. Randall et al. [Ran+14] also developed a

privacy-preserving record linkage approach for large datasets of hospital admissions data.

2.7.2 Short reads filtering

This section describes the existing short reads filtering (SRF) approach [Cog+15], which served as starting point for the work in this thesis.

The human genome contains genomic variations whose unique combination distinguish each individual from all the others. The proposed filtering approach focuses on this property of human DNA. Since only a small portion of the genome is privacy critical, it would be advantageous if one could apply higher protection methods, which are more costly, just to the privacy sensitive information. The SRF approach allows the classification of 30 nucleotides reads as sensitive or insensitive, using a Bloom filter.

In the SRF approach, the Bloom filter is initialized with all sequences of 30 nucleotides that contain one or more genomic variations. Those sequences are created using the human reference genome and the genomic variations present in known publicly available databases. The authors considered three types of genomic variations: single nucleotide polymorphisms (SNPs), short tandem repeats (STRs), and disease-associated genes. Then, during the filtering process, if a sequence is detected by the Bloom filter, it is classified as sensitive.

Since the SRF can produce false positives, the authors evaluated the accuracy of the method in order to quantify its specificity. The proposed SRF classified about 10% of the human genome as sensitive information.

The computation time and the memory consumption of the proposed solution were also evaluated, to ensure that it did not create a bottleneck. Regarding the computations time, the SRF was already considered efficient, since the filtering time using a single-core was 44× to 200× faster than the 0.3 millions base-pairs per second of the NGS machines. Moreover, the SRF performance can be even higher using multi-cores. The use of Bloom filters allows a reduction of 6× of the memory consumption, with a false positive rate of 10^{-6}, compared to the space required to store the original data (35.1GB) and the Bloom filter structure with the same information (5.6GB). In addition, this parameter can be modified by adapting the false positive rates, which would also reduce the Bloom filter size.

To summarize, the SRF is a systematic sensitive reads detection method, whose performance and reproducibility allow the integration of such method inline with the NGS machines. This method can be used to detect sensitive information (i.e., SNPs, STRs and disease-associated genes) and, for example, to remove it from genomic data before sharing it.

2.8 Genomic data repositories

The value of genomic data and its wide range of applications (e.g., personalized medicine, research, forensics), in combination with the decrease of sequencing cost, supported several projects whose goal is to create significant collections of human genomic data. Those projects include, for example, the 1000 Genomes Project (GP)[7], which was the first effort to obtain a significant dataset of human genomes, and the 100,000 GP[8], which was launched in 2012 in England with the goal of supporting rare disease and cancer research by sequencing patients and their families. Even though human genomes sequencing has increased in the last years, with a projection of more than 100 million human genomes sequenced by 2025 [Ste+15], the current number of open access genomes collections is still limited. Genome sharing allows more significant studies (based in bigger datasets), and supports the reproducibility of experiments. However, genome sharing can raise several privacy issues, as illustrated by published privacy attacks.

The work in this thesis was mainly tested on data sets made available by the 1000 Genomes Project (1000 GP) and by the 2017 iDASH contest, which are described later in the this section. In addition, the human reference genome considered to build the genomic sequences used on the experiments was the GRCh38.p11 version (available at https://www.ncbi.nlm.nih.gov/assembly/GCF_000001405.37/). This reference version differs from the one described in the publication resulting from the work described in Chapter 5 (GRCh37 version), because since the paper's submission the results were updated using the most up-to-date reference version. The changes between reference sequences are small and they do not affect significantly the experiments results. The reference genome is the most common sequence of nucleotides that is present in a human population. The genome reference used in this thesis was collected from the 1000 GP population.

The experiments were run on simulated reads, in order to assess the methods performance with different sizes and error rates, mimicking the reads generated by different sequencing technologies. The reads generation was made using wgsim [Li11], a software that simulates reads with the properties of sequencing technologies. Whenever necessary, the length and the error rate of the reads used for the experiments are specified.

[7] 1000 Genomes Project (GP) – http://www.internationalgenome.org/

[8] 100,000 GP – https://www.genomicsengland.co.uk/about-genomics-england/the-100000-genomes-project/

2.8.1 1000 Genomes Project

The 1000 GP (http://www.internationalgenome.org/) was the first effort to create a large genomes repository to support research. It includes genomes from 5 different populations: African, American, East Asian, South Asian, and European. Although the project ran from 2008 until 2015, it is still maintained and extended. The current version contains information from 2504 individuals. In 2015, the project had already discovered and annotated more than 88 millions genomic variations [Con15]. This is currently the biggest open access dataset of human genomes available, which allows the users to download the genetic data freely and without registration.

```
##fileformat=VCFv4.0
##fileDate=20090805
##source=myImputationProgramV3.1
##reference=1000GenomesPilot-NCBI36
##phasing=partial
##INFO=<ID=NS,Number=1,Type=Integer,Description="Number of Samples With Data">
##INFO=<ID=DP,Number=1,Type=Integer,Description="Total Depth">
##INFO=<ID=AF,Number=.,Type=Float,Description="Allele Frequency">
##INFO=<ID=AA,Number=1,Type=String,Description="Ancestral Allele">
##INFO=<ID=DB,Number=0,Type=Flag,Description="dbSNP membership, build 129">
##INFO=<ID=H2,Number=0,Type=Flag,Description="HapMap2 membership">
##FILTER=<ID=q10,Description="Quality below 10">
##FILTER=<ID=s50,Description="Less than 50% of samples have data">
##FORMAT=<ID=GT,Number=1,Type=String,Description="Genotype">
##FORMAT=<ID=GQ,Number=1,Type=Integer,Description="Genotype Quality">
##FORMAT=<ID=DP,Number=1,Type=Integer,Description="Read Depth">
##FORMAT=<ID=HQ,Number=2,Type=Integer,Description="Haplotype Quality">
#CHROM POS     ID        REF ALT    QUAL FILTER INFO
FORMAT      NA00001        NA00002        NA00003
20     14370   rs6054257 G      A       29   PASS   NS=3;DP=14;AF=0.5;DB;H2
GT:GQ:DP:HQ 0|0:48:1:51,51 1|0:48:8:51,51 1/1:43:5:.,.
20     17330   .         T      A       3    q10    NS=3;DP=11;AF=0.017
GT:GQ:DP:HQ 0|0:49:3:58,50 0|1:3:5:65,3   0/0:41:3
20     1110696 rs6040355 A      G,T     67   PASS   NS=2;DP=10;AF=0.333,0.667;A
A=T;DB GT:GQ:DP:HQ 1|2:21:6:23,27 2|1:2:0:18,2   2/2:35:4
20     1230237 .         T      .       47   PASS   NS=3;DP=13;AA=T
GT:GQ:DP:HQ 0|0:54:7:56,60 0|0:48:4:51,51 0/0:61:2
20     1234567 microsat1 GTCT   G,GTACT 50   PASS   NS=3;DP=9;AA=G
GT:GQ:DP    0/1:35:4       0/2:17:2       1/1:40:3
```

Figure 2.2: **Variant Call Format**. From http://www.internationalgenome.org/wiki/Analysis/vcf4.0/

Genomic information in the 1000 GP is available in the variant call format (VCF), per chromosome, with the genotype per position for each individual. Figure 2.2 shows an example of the content of a VCF file. The line initiated with '##' are headers, which contain meta-information such as the file format, the date, the source, the reference genome used, and additional information about the

content of the file. The other lines are called data lines, and each line contains the chromosome, the position where the variation occurs, the ID of the variation, the reference allele, the alternative allele, the quality score, the filter results (if passed all filters or not), and additional information. The VCF is a compressed format, where instead of having the full genomic sequence, it contains only the differences (i.e., genomic variations) between the reference genome considered and the genome of an individual or more.

Other repositories have been created, however, they present some limited data. For example, Genome in a Bootle[9] contains a data set composed of seven human genomes, characterized using 12 different sequencing technologies. This work aims at providing reference human genome sequenced data to help genome comparison and benchmarking [Zoo+16]. It provides reference standards, methods and data for the development of technologies and analysis validation. Other example is the 100,000 Genomes Project[10] which includes 100,000 genomes from patients affected by a rare disease or cancer, with the intuit of promoting the research of those health conditions. The 100,000 Genomes Project is the biggest repository of genomes which supports the diagnosis, the treatment and the study of rare diseases and cancer.

2.8.2 iDASH contest

The iDASH contest[11], which is organized on a yearly basis, is a community effort to promote the development of new privacy-preserving solutions for biomedical data, thus addressing practical problems in the area.

The competition usually happens between May and October, ending with the presentation of the winning solutions at the Security Workshop. For instance, the 2019 tasks included: (i) the development of gene-drug interaction data sharing based on Blockchain and smart contracts; (ii) genotype imputation using homomorphic encryption; (iii) performing machine learning inside SGX enclaves to ensure privacy protection; and (iv) privacy-preserving collaborative machine learning models training. For each task a dataset is provided in addition to an explanation of the experimental setting, requirements, and evaluation criteria.

The experiments in this thesis (Chapter 3) use the dataset provided by the 2017 iDASH contest. The dataset is composed by 1000 case and 1000 control individuals selected from the 1000 GP population, with information about 2,014,777 SNP loci.

[9] Genome in a Bottle – https://jimb.stanford.edu/giab
[10] 100,000 Genomes Project –https://www.genomicsengland.co.uk/
[11] iDASH – http://www.humangenomeprivacy.org/[Year]/

Chapter 3

Sensitivity levels for genomic data

In the last decades, researchers have studied and discovered multiple genomic variations, which they use in several scientific domains, for example, health. A human genome contains sensitive information, the set of genomic variations it contains, which is unique for each individual. Hereupon, a question arises: "Are all the genomic variations equally sensitive?". From the biological perspective, the answer is no, since different health conditions, with different frequency and severity, are caused by different genomic variants. Similarly, from the privacy perspective, rare genomic variations represent more privacy concerns, since they contribute strongly to re-identification and/or membership. Considering this, a new question arises: "Can we define sensitivity levels for genomic data?". The answer to this question is yes, and the need for sensitivity levels has been underlined by some authors in the field [DDK16]. However, the benefits of such a classification are still unclear.

This chapter addresses the challenge of improving privacy while providing practical performance of sequenced data analysis, based on the classification of reads into sensitivity levels. This thesis proposes DNA-SeAl, a three level classification. DNA-SeAl defines three levels due to identification of three main classes of alignment algorithms (discussed later). However, the number of levels can be defined according to user preferences. The first step of DNA-SeAl is the classification of sequenced data intro sensitivity levels based on quantitative and qualitative features of the human genome. The levels are created in a way that prevents the inference of more sensitive information when an adversary can observe a subset of the information stored at a given level. The inference prevention is based on MaCH, a state-of-the-art haplotype inference software that relies on associations between genomic variations (e.g., linkage disequilibrium). This process is called levels disconnection. After the creation of the different sensitivity levels, the raw

reads are classified into the different sensitivity levels. Cogo et al. [Cog+15] developed a classification method that distinguish the regions of the genome that are sensitive (i.e., that contain genomic variations) from the insensitive. This previous work, which this thesis extends, serves as basis for our classification method, which this thesis extended. After the reads have been classified in the sensitivity levels, they can be aligned using different algorithms, with different privacy guarantees, for the different levels of sensitivity. There are three main types of alignment algorithms: (i) non-secure but fast algorithms – also called plaintext algorithms; (ii) secure but slow algorithms – also known as cryptography-based algorithms; and (iii) intermediate protection algorithms. The *non-secure but fast algorithms*, also known as plaintext algorithms, are the most efficient algorithms, nevertheless they do not provide any protection to the data being analysed. Some of the state-of-the-art cloud-based plaintext algorithms include CloudBurst [Sch09] and DistMap [PS13]. These two algorithms can perform plaintext alignment in public clouds, either with or without encrypted transfers of data. *Secure but slow algorithms*, also called cryptographic algorithms, present the higher data protection, however, they have high bandwidth and computational costs. This occurs since the priority of this class of algorithms is to prevent data leakage, which can be achieved by sacrificing performance. Garbled circuits [Hua+11], homomorphic encryption [DFT13; Bar+12] and secure multi-party computations [Hua+11; JKS08] have been used to perform alignment on encrypted reads. Finally, the *intermediate protection algorithms* provide limited protection and a reasonable performance. Chen et al. [Che+12] proposed a hashed K-mers approach for reads alignment in hybrid clouds. Balaur [PB17] is another approach in this category, which uses locally sensitive hashing, secure k-mer voting, and a MinHash algorithm to align reads. For the performance evaluation, the Chen et al. approach was selected due to its lighter memory consumption in comparison with Balaur.

In respect to the attacks, the work described in this chapter aims at preventing re-identification, inference and membership attacks. **Re-identification attacks:** in this family of attacks, an adversary aims at relating genomic information to its owner, for example by linking genotypes or SNPs information with names or surnames. Several attacks of this type were described in the literature [Mal06; Gym+13; Hum+15]. **Inference attacks:** the goal of the adversary is, given partial genomic data, to infer further genomic information, such as hidden or non observed SNPs. DNA-SeAl considers the subclass of inference attacks that targets sensitive levels called amplification attack. In this particular case, the goal of the adversary is to infer more sensitive information based on information from a lower sensitivity level. The proposed sensitivity levels classification aims at preventing this kind of attacks. In other words, once an adversary observes a set of raw reads,

he is not able to infer further information, since more sensitive levels information is assumed to be securely stored. **Membership attacks:** in this kind of attacks, the goal is to determine the participation of an individual or more in a specific study group. This class of attacks are considered in the privacy evaluation of DNA-SeAl, where it is demonstrated that they can be prevented by splitting the sensitivity levels.

The alignment method based on sensitivity levels for genomic data, named DNA-SeAl, limits the adversary access to the plaintext reads, while providing a high performance. In addition, it also leverages the diversity of alignment algorithms, while adapting the privacy protection to the sensitivity of the data. To demonstrate the improvements achieved by using three sensitivity levels, we evaluated the performance and privacy of DNA-SeAl.

3.1 Methods

3.1.1 Sensitivity levels for genomic data

This thesis presents DNA-SeAL, a sensitivity levels classification based on quantitative features (e.g., alleles frequency and linkage disequilibrium). An example with three distinct levels is defined and its performance and privacy guarantees evaluated. The presented approach uses three levels, however, the number of levels can be adapted to the user preferences and to the availability of more diverse algorithms and execution environments. Nonetheless, relying on more levels can help to increase performance for a given security level, or to increase security while maintaining a practical performance.

The first step of DNA-SeAl is the definition of the sensitivity levels based on the alleles frequency. This classification relies on the fact that rare genomic variations should be considered more sensitive since they concern a smaller subset of the population and therefore they can strongly contribute for re-identification attacks. The created sensitivity levels are represented in Figure 3.1(a). The first sensitivity level, defined as level 1, is bounded by the minor allele frequency that defines rare genomic variations (0.05) [San+09]. Level 2 contains the genomic variations with frequency between 0.05 and 0.2. Finally, level 3 contains all the remaining genomic variations, considered common and consequently less sensitive (frequency>0.2). Later, this chapter discusses the distribution of the alleles among the different sensitivity levels (see Section 3.3.1).

Finally, this chapter presents an example of a sensitivity level classification based on qualitative properties of the human genome (see Figure 3.1b), which can

include drug response, disease predisposition, and physical appearance, to name some. In this example, the most sensitive level (level 1) contains information about disease genes, ethnicity related variants and other regions of the genome that can lead to the re-identification of individuals. The work in this thesis does not include the definition of levels based on such qualitative properties since it would requires further data which are not fully available, such as medical records and extensive phenotype information. Consequently, this topic is not covered by this thesis and it is part of the future work.

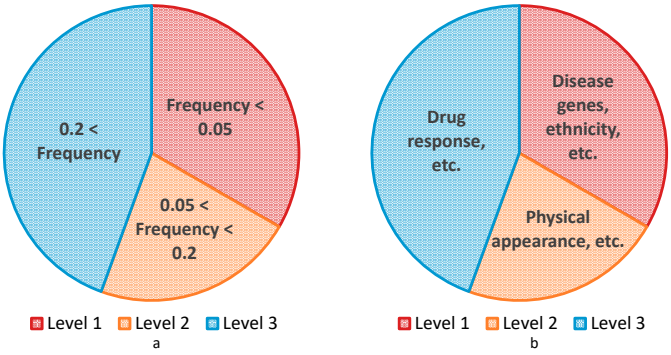

Figure 3.1: **Initial sensitivity levels**. The sensitivity levels built using different properties of the genomic information. (a) Alleles frequency-based sensitivity levels (quantitative classification). (b) Manual declaration-based sensitivity levels (qualitative classification).

3.1.1.1 Sensitivity levels disconnection

After the initial sensitivity levels definition based on quantitative properties, biological relations (e.g. linkage disequilibrium (LD)) and inference correlations are taken into consideration to modify the levels, since they can lead to information leaks if not adequately managed. In order to avoid those links between levels with different sensitivities, alleles that are linked are promoted to the highest sensitivity level found among the linked alleles set.

SNP promotions based on linkage disequilibrium associations

Linkage disequilibrium (LD) defines the non-random transmission of genomic variations [MS04], which comes from the transmission of genomic variations in blocks. Some privacy attacks take advantage of these non-random associations [Wan+09].

This section describes how the sensitivity levels are disconnected based on known LD relations, which we compute from a database of genomes. First, the LD is computed between genomic variants present in the human genome, within a maximum distance of 1000 nucleotides between a pair of variants. This range was chosen considering that LD relations occur mainly within few kilobases (kb), and occasionally can extend up to 100 kb [AKS02]. Figure 3.2(a) shows some examples of LD linked SNPs. For example, SNP 1 is linked with SNP 3, which is linked to SNP 6. Those links represent direct inference connections, which should be removed to prevent inference between different levels. In case the inference links are not removed, if an adversary observes SNP 3, he would be able to learn SNPs 1 and 6. In Figure 3.2(a), two other linked blocks are represented, one is composed by SNPs 2 and 5, and the other represents the link between SNP 4 and SNPs 7 and 8.

Figure 3.2(b) represents the sensitivity levels after the promotions based on LD relations. The promotions are made by moving the SNPs that are linked to the most sensitive level found among the different levels the SNPs are located. In this process, SNPs 3 and 6 were promoted to level 1 since they are linked to SNP1 that is from level 1 (the most sensitive one). Following the same rule, SNPs 5, 7 and 8 are promoted to level 2. Keeping the linked SNPs in the same sensitivity levels prevents attacks based on LD relations, including amplification attacks.

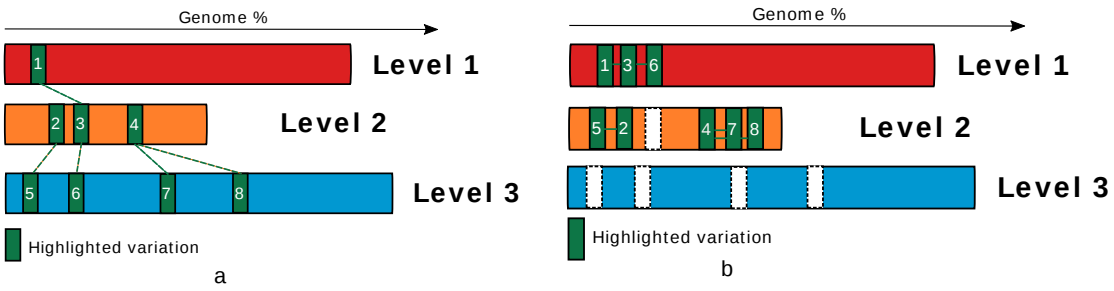

Figure 3.2: **LD-based promotions**. (a) Identification of the LD links between SNPs of different levels. (b) Refined levels after LD-based promotions.

SNP promotions through haplotypes inference

In order to completely isolate each sensitivity level, this thesis applied the MaCH [Li+10] haplotype inference software. This software is able to infer hidden or non-observed SNPs of an individual based on a reference population. Briefly, MaCH software receives two input sets: (i) a list of biomarkers and the respective SNPs information for a reference population; and (ii) a list of biomarkers

and respective SNPs which are observed by an adversary. MaCH is executed for each sensitivity level and for each inference cycle. The complete process of sensitivity levels disconnection through haplotypes inference requires four steps per sensitivity level, which are represented in Figure 3.3 and described in the following paragraphs.

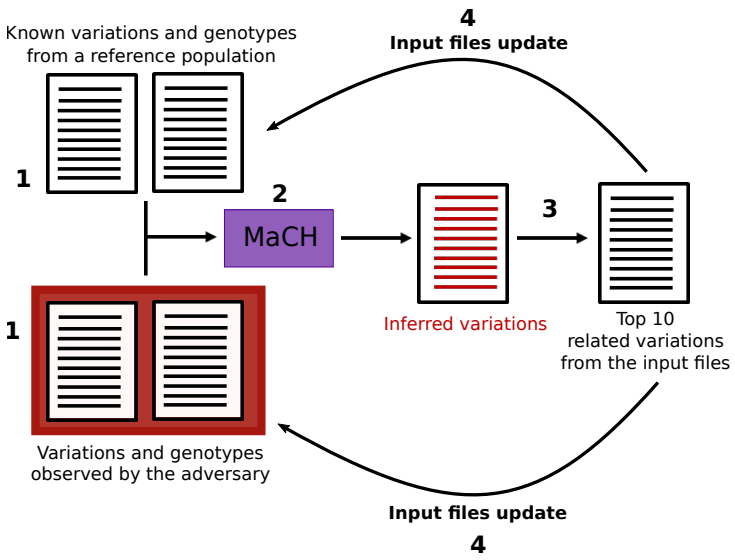

Figure 3.3: **Inference-based promotions**. Step 1: Creation of input files for MaCH. Step 2: Haplotypes inference with MaCH. Step 3: Identification of the statistically correlated SNPs in the MaCH output set. Step 4: Deletion of the correlated SNPs from the input files (i.e., promotions).

Step 1: Creation of input files for MaCH. To start, sets of four files are created, corresponding to the input files used by MaCH. To create those files, subsets of 20 000 SNPs from chromosome 1 were collected from the 1000 Genomes Project. Since the correlation between SNPs decreases with their separating distance, a range of 20 000 SNPs was considered sufficient to include all the LD relations. Two of the files created correspond to the reference population information, one file with the biomarkers of the genomic variants and the other with the respective genotypes in the reference population. The biomarkers file has the extension .snps, while the genotypes file has the extension .haplos. The other two files contain the genomic variations the adversary observes from a given sensitivity level, while the remain genomic variations in the level are excised since they are assumed to be stored in a secure location. One of the files (extension .dat) contains the biomarkers for the observed SNPs, while the other file (extension .ped) contains the genotypes information for those observed SNPs.

Step 2: Haplotypes inference with MaCH. After creating the input files, the next step is to run MaCH on the input files to obtain the set of SNPs that can be inferred given the provided reference population and the set of observed SNPs. Since the goal of this process is to disconnect sensitivity levels, only the SNPs belonging to more sensitive levels are considered, i.e., inferred SNPs from the same level as the observed ones are ignored. In addition, only the SNPs with good accuracy are considered, i.e., SNPs with an r^2 value higher than 0.3 (value recommended by the MaCH software's authors). This step demonstrates the information that could be inferred if some information from a sensitivity level were to be leaked.

Step 3: Identification of the statistically correlated SNPs in the MaCH output set. The top 10 SNPs, which are observed by the adversary, correlated with each inferred SNP are collected from MaCH's output. Since this output only contains the list of inferred SNPs, the relations between inferred and observed SNPs are discovered computing the linkage disequilibrium between the SNPs in those sets. After obtaining the top 10 SNPs that might lead to the inference of each inferred SNP, that subset is removed from the MaCH input file. In other words, the top 10 SNPs are promoted to higher sensitivity levels to prevent inference between levels.

Step 4: Deletion of the correlated SNPs from the input files. Removing the top 10 SNPs from the set of SNPs observed by the adversary, might prevent the inference of more sensitive SNPs. Promoting less SNPs per inference cycle would lead to a higher number of inference cycles required until all the links between levels are cancelled. Moreover, more cycles would require more time.

The four steps are repeated until no more infeSNP promotions through haplotypes inferencerences are possible or until the number of inferred SNPs becomes constant.

The number of haplotypes inference cycles required to avoid the inter-sensitivity levels inference was determined by running consecutive inference cycles until a significantly small proportion of inferred SNPs is reached (ideally 0%). In each consecutive inference cycle, the SNPs that allow the inference of others were promoted to the highest sensitivity level found.

Figure 3.4 shows the proportion of SNPs inferred in consecutive inference cycles. The results show that before applying any inference cycle (0), it is possible to infer 9.97% of the SNPs in level 1 (most sensitive level) using the SNPs of level 2 (intermediate level) and 0.42% of SNPs in levels 1 and 2 using SNPs from level 3 (least sensitive level). After two cycles of inference (Number of inference cycles = 2), it is possible to infer only 0.046% of the SNPs from level 1 using information

from level 2 and 0.13% from levels 1 and 2 based on information from level 3.

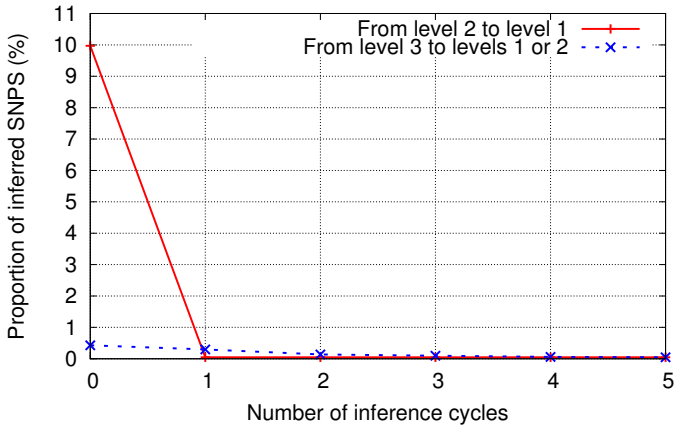

Figure 3.4: **Proportion of inferred SNPs.**

After five inference cycles with MaCH, very few genomic variations could still be inferred (e.g., less than 0.04% of the SNPs from level 2, less than 0.05% of the SNPs from level 1). This results can be due to the limited number of genomes used in the 1000 Genomes Project, or by the unique combinations of statistically unlikely SNPs some specific individuals can present (since the few inferred SNPs can still be inferred after more inference cycles). We believe that in a larger population it would not be possible to infer those SNPs any more because the number of unique combinations of several SNPs would be much smaller. To summarize, with 5 inference cycles almost all the inference links are removed and, therefore, we used this number of inference cycles in the experiments presented in this chapter. The number of cycles to run is a user-defined parameter. When deciding, the user needs to consider that this process is time consuming, however, if it is underestimated it can allow some links between sensitivity levels which can be used for amplification attacks. To conclude, the SNP promotions through haplotypes inference is a one time cost process.

3.1.1.2 Cloud diversity to prevent inference

Another solution to prevent amplification attacks consists in splitting the input sets during MaCH experiments between different clouds. This way the amount of data observed if a single cloud were to be attacked and its content exposed would be limited, making amplification attacks even more difficult. Precisely determining how to split the sensitivity levels on different clouds could increase performance by reducing the number of promotions across sensitivity levels. Figure 3.5 represents the division of level 3 – the least sensitive level – in two sections, which are stored

in two different clouds (A and B). The division is made based on pairs of variation that are linked with each others (e.g. linkage disequilibrium, inference relations), and it ensures that they are separated and stored in distinct clouds. This strategy prevents an adversary that obtains the content of one cloud (in the example, Cloud A) from inferring variations from higher levels.

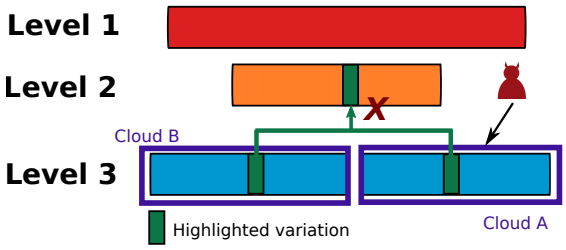

Figure 3.5: **Information stored in two clouds**. Storing the data in two different clouds makes alleles inference more difficult. Once an adversary can observe the information in cloud A, he only has access to partial data and he is not able to infer more sensitive information.

3.1.2 Classification of reads into sensitivity levels

This section describes the reads classification into levels of sensitivity, which extends the previous filtering method proposed by Cogo et al. [Cog+15]. The main two steps of the proposed method are the definition of sensitivity levels for reads and the conflict management that comes with use of several filters. First, the sensitivity levels are created based on the privacy risk of genomic variations – the probability of an adversary to extract personal sensitive information and the corresponding negative impact. Then, a conflicts management step is required to produce the final result when using several filters. In the end, DNA-SeAl allows the adjustment of the storage cost, performance, and access control accordingly to the privacy requirements of the data. Considering that more sensitive data require stronger privacy protection and that privacy-preserving alignment algorithms are slower, DNA-SeAl, by classifying the genomic data into sensitivity levels and by applying the adequate protection to each level, improves the performance $\times$ privacy product. The same applies to the storage and data access control, the higher the sensitivity of data, respectively, the stronger should be the protection method used and the more restrict should be the access.

Figure 3.6 presents the overview of the reads classification process described in this section. A more detailed explanation about the classification and conflict management steps of DNA-SeAl is included in the following sections.

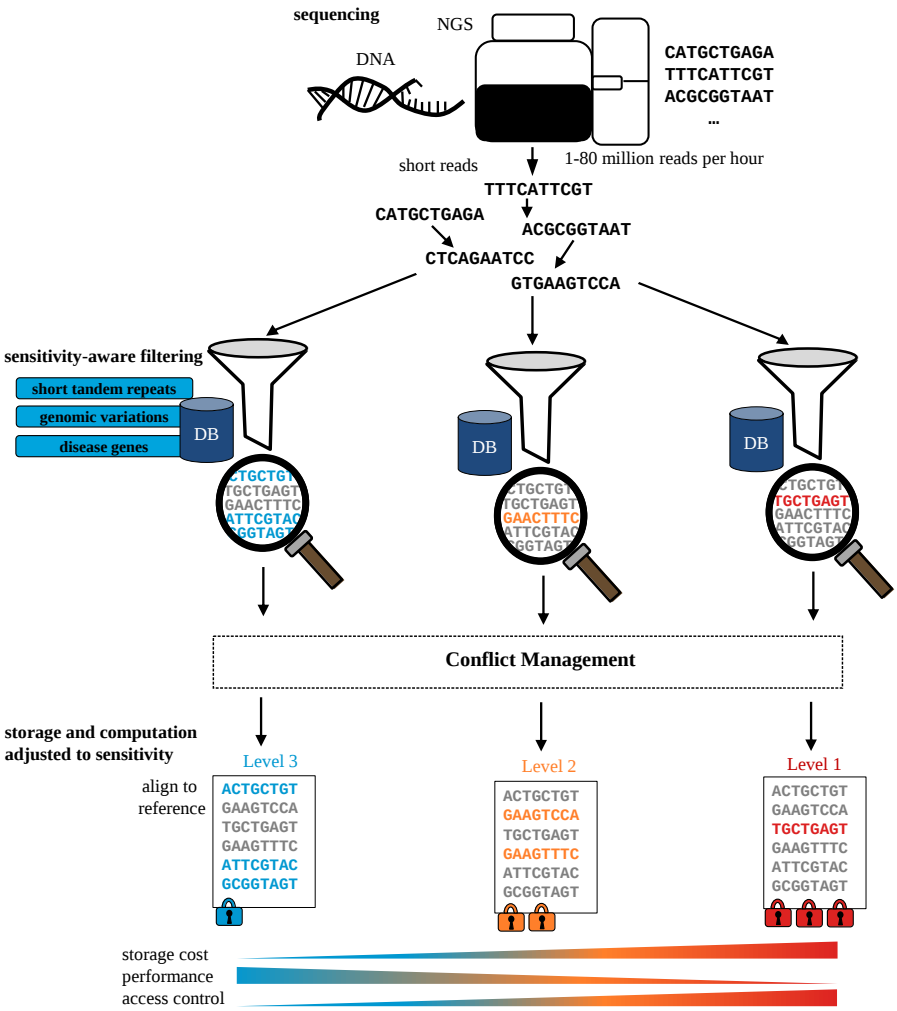

Figure 3.6: **Classification of reads in sensitivity levels and adjusted storage, algorithms, and access control per sensitivity level**.

3.1.2.1 Reads classification

Briefly, the reads filtering method [Cog+15] DNA-SeAl uses relies on Bloom filters. The main advantages of this approach are the following: (i) high throughput – Bloom filters are at least 40 times faster than current sequencing machines and they can be parallelized, and (ii) Bloom filters can produce some false positives, but they never produce false negatives. The filters are initialized with a database of known sensitive information. The idea here is to protect all the sensitive information used in re-identification attacks described in the literature. Such sensitive information includes, but is not limited to, the three main kinds of sensitive sequences: (i) genomic variations, including single nucleotide polymorphism (SNPs)

and structural variations (SVs); (ii) disease related genes; and (iii) short tandem repeats (STRs).

The classification is made using a short read filter per sensitivity level, which has been previously defined in order to prevent amplification attacks, as described in Section 3.1.1. The dictionaries used, one per sensitivity level, for the initialization of the Bloom filters were built based on the genomic variants publicly available and on the human reference genome. In the end, each Bloom filter is initialized with short genomic sequences that contain all the genomic variations of a sensitivity level and after the Bloom filers are ready to filter reads.

Finally, the filters do not detect genomic variations which have not be included in their initialization (e.g., de novo SNPs). However, this is not a critical feature since the rate of new genomic variations discovery have dropped along the years [Cog+15]. Consequently, the risk of missing the detection of a sensitive nucleotide is low. Furthermore, the filters can be easily updated with the new variations discovered, working similar to the anti-virus update schemes.

3.1.2.2 Filters conflict management

This section explains the conflict management step represented in Figure 3.6. The possible conflicts that can occur are the following: (i) each variation can be classified into multiple sensitivity levels; and (ii) false positives can lead to information leakage.

Using several Bloom filter, one per sensitivity level, can produce different results for a single read, which need to be composed to obtain the final classification. When several Bloom filters are used, a read can match multiple sensitivity levels. If it happens, the read is classified with the most sensitivity level it matches to.

Finally, Bloom filters might produce some false positives that can lead to information leaks if not handled adequately. False positives are nucleotides that are detected as sensitive but in fact are insensitive – they do not participate in any genomic variation – although they have been classified as sensitive in the filtering process. It is important to ensure that false positives do not leak any further information. To do so, DNA-SeAl classifies each read with the higher sensitivity level it matched with. In this way, even if the read was classified with a higher sensitivity than it really is, information leakage is prevented.

3.2 Evaluation Setup

3.2.1 System model

DNA-SeAl assumes a system composed of public and private clouds. Since private clouds are much more costly in comparison to public clouds, it is assumed that there is more availability of public clouds than private clouds. In particular for the clouds environment, DNA-SeAl assumes that no more than $f = 1$ public clouds can be compromised. In other words, an adversary is only able to compromise one cloud. Considering this, the privacy guarantees increase with N/f, where N denotes the total number of clouds used. The larger the number of available clouds, the more compromised clouds it is possible to tolerate without privacy breaches. Extending to the $f > 1$ scenario is part of future work, because preventing privacy breaches becomes more complex, since it is not trivial to determine the information each compromised cloud has. In addition, for $f > 1$, compromised clouds can also communicate and exchange information, and consequently, increase the criticality of the privacy breaches.

The system described emulates a biocenter that relies on public and private clouds to process its data. Furthermore, the biocenter takes the necessary precautions which allows it to prevent full compromise (of functionality and data) if one of the participating clouds is attacked.

3.2.2 Threat model

Regarding the threat model, the proposed sensitivity levels classification presented in this chapter assumes an honest-but-curious adversary, which is able to observe some reads during the alignment step and whose goal is to infer further information from the observed reads. The adversary can observe the reads when the alignment is executed in a public cloud using plaintext alignment algorithms. In addition, it is also assumed that the adversary has access to the reference genome, so that he is able to align the observed reads to the reference genome and discover the information they contain, and has access to the statistical relationships between genomic variations. Finally, the adversary has the necessary knowledge to combine the observed information and the discovered statistics to perform existing privacy attacks.

3.3 Results

This section presents the statistics of the sensitivity levels, and the results of the performance and privacy improvements evaluation. First, this section describes the proportion of SNPs per sensitivity levels, with three levels. Those statistics are computed for the initial levels, after the levels disconnection, and for reads of 100 and 1000 nucleotides. Later, this section also shows the results for the performance and privacy improvement evaluations. The performance evaluation assesses the alignment task, using alignment algorithms with distinct privacy protection guarantees for the different sensitivity levels. Finally, for the privacy improvement, the evaluation is based on two privacy metrics: the likelihood-ratio and the genomic privacy metrics.

3.3.1 Sensitivity levels statistics

This section shows the results regarding the number of variations per sensitivity level, which were defined as described in section 3.1.1 before and after promotions. Then, the percentage of reads of two different sizes (100 and 1000 nucleotides) per sensitive level, considering sets of 1000 reads, is also reported.

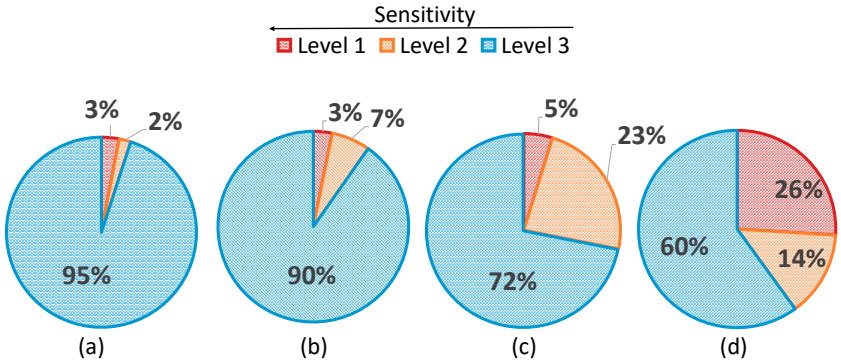

Figure 3.7: **Promotions between sensitivity levels**. (a) Proportion of an individual genomic variations based on alleles frequency. (b) Proportion of an individual genomic variations after levels disconnection. (c) Distribution of 100-nucleotides reads by the sensitivity levels. (d) Distribution of 1000-nucleotides reads by the sensitivity levels.

Figure 3.7 shows the evolution of the levels, in proportion of SNPs, after the different promotions steps. First, the SNPs were classified among the sensitivity levels based on the allele frequencies thresholds. From this classification, the obtained distribution was: 95% of the SNPs on the least sensitive level (level 3), 2% on the middle level (level 2), and 3% of highly sensitive SNPs (level 1) (see

Figure 3.7(a)). Figure 3.7(b) reports the distribution after links between sensitivity levels were removed, thanks to promotions based on linkage disequilibrium and based on inference (using MaCH). After this promotion step, 5% of the SNPs previously in level 3 were transferred to level 2, since they could leak information about this level, and level 1 remained with 3% of the SNPs. Figure 3.7(c) reports the proportion of 100-nucleotides (short) reads per sensitivity. Level 3 presents 72% of the reads, 23% are classified as level 2, and the remaining 5% belong to level 1. Finally, Figure 3.7(d) shows the distribution over the sensitivity levels of 1000-nucleotides (long) reads. In this case, level 1 is substantially larger than in the previous cases, with 26% of the reads, while 14% belong to level 2, and the remaining 60% are in the lower sensitivity level. Globally, the sensitivity of

Table 3.1: SNPs per sensitivity level.

Chr.	**Total SNPs**	**Level 1**	**Level 2**	**Level 3**
1	12854384	3.0%	2.1%	94.9%
2	14072346	3.0%	2.1%	94.9%
3	11588182	3.0%	2.3%	94.7%
4	11387566	3.4%	2.5%	94.1%
5	10460934	3.0%	2.1%	94.9%
6	9979424	3.7%	2.4%	93.9%
7	9370502	3.2%	2.4%	94.4%
8	9131088	3.4%	2.3%	94.3%
9	7073218	3.3%	2.3%	94.4%
10	7931058	3.1%	2.5%	94.4%
11	8037004	3.2%	2.2%	94.5%
12	7684622	2.9%	2.3%	94.8%
13	5677874	3.1%	2.4%	94.5%
14	5274600	3.1%	2.2%	94.7%
15	4817318	3.2%	2.3%	94.5%
16	5356960	3.5%	2.4%	94.1%
17	4626788	2.9%	2.2%	94.9%
18	4504308	3.0%	2.4%	94.6%
19	3637726	3.6%	2.8%	93.6%
20	3602154	2.8%	2.2%	95.0%
21	2194746	3.4%	2.7%	93.9%
22	2190790	3.4%	2.5%	94.1%
X	6864096	44.6%	1.9%	53.5%
All	6864096	4.9%	2.3%	92.8%

long reads is higher, which is justified by the larger number of variations they can contain.

In addition to the distributions in Figure 3.7, Table 3.1 shows the distribution of SNPs per sensitivity level for an individual genome – HG03556, per chromosome and globally (last line). In general, around 4.9% of the SNPs of the genome are classified in level 1 (most sensitive level), 2.3% in level 2 (intermediate sensitivity), and the remaining 92.8% in level 3 (least sensitive level). It is interesting to observe that for all the chromosomes except for chromosome X, more than 93% of the SNPs are in level 3, around 2% of the SNPs are in level 2, and the remaining approximately 3% are in level 1.

This distribution supports the implementation of sensitivity levels since they demonstrate that the majority of the genomic variations of an individual's genome have low sensitivity. Therefore, there are benefits in classifying the genomic variations into different sensitivity levels and adapting the applied privacy protection methods.

As a remark, this distribution could vary depending on the ethnicity of the individual. The population-based study is out of the scope of this thesis, however, it could enrich the definition of the sensitivity levels proposed in this thesis.

3.3.2 SNP promotions

Overall, 5% of the SNPs were promoted though SNPs relations (i.e., LD relations and/or inference relations). The majority of those promotions were made due to LD relations since around 3.5% of all the SNPs were promoted from one level to a more sensitive one based on LD relations.

Table 3.2: Number of inferred SNPs per inference and promotion.

Inference cycles	0	1	2	3	4	5
Inferred SNPs (%) from level 2 to level 1	9.97	0.05	0.05	0.05	0.05	0.05
Inferred SNPs (%) from level 3 to level 2	0.43	0.30	0.14	0.10	0.06	0.04

The SNPs promotions though haplotype inference is described in section 3.1.1.1, and relies on MaCH. In the end of the first inference cycle, 1.6% of the SNPs in level 3 and 18% of the SNPs in level 2 were promoted to a higher sensitivity level. Overall, 1.5% of all SNPs were promoted from one level to a more sensitive one based on haplotype inference relations. Table 3.2 shows the number of inferred SNPs for each inference cycle performed. Before any promotion, it is possible to

infer 9.97% of the SNPs in level 1 (the most sensitive) using SNPs information from level 2 (intermediate level). At this point, based on information of level 3 (the least sensitive) it is possible to infer 0.43% of the SNPs in level 2. After the first inference cycle, only few SNPs could still be inferred, for example, less than 5 SNPs inferred from level 2. Due to the identification nature of human genome, which contains unique combinations of SNPs for each individual, it is not possible to completely mitigate inference. In addition, the population used – 1000 Genomes Project – can be statistically limited, since a larger dataset might change the frequency of those unique combinations of SNPs.

3.3.3 Performance improvement

For the performance evaluation, we provide a comparison between the sensitivity levels approach and standard alignment strategies. The standard strategies considered include the alignment exclusively in a public cloud, exclusively in a private cloud, and an hybrid approach using private and public clouds. *Private cloud only:* describes the case where a biocenter performs reads alignment only using its private cloud infrastructure and plaintext algorithms. *Public cloud only:* considers the alignment in a non-secure environment using proven secure algorithms, such as encryption-based alignment algorithms. In this case, an adversary might be able to observe unencrypted data during computations and communications. *Public-private sensitivity adapted alignment:* reads considered high sensitive and low sensitive are aligned in private and public clouds, respectively. This scenario corresponds to a conscious use of the the computing resources for sensitive computations, including the secure usage of public clouds.

We selected representative of each class of alignment algorithms, in order to assess the performance of the alignment task using the adequate protection to the sensitivity level. For highly sensitive information, the protection provided should be higher. Table 3.3 summarizes the privacy level, computation time and communication costs of the selected alignment algorithms. Since the selected algorithms provide different privacy levels and distinct sensitivity levels require differentiated privacy protection, each algorithm was used for one sensitivity level matching the privacy requirements. The information from the three sensitivity levels, with very high, high, and low sensitivity, was respectively processed using homomorphic encryption, hashed K-mers and Cloudburst alignment algorithms. The presented values represent the cost of aligning a single 100-nucleotides read to the full human genome, using a single core. Comparing the computation and communication costs of the three alignment algorithms, one can easily observe the already discussed pattern where higher protection means a higher cost (i.e., slower algorithms and

heavier data transfers).

Aligning reads in a cloud implies to assume that the cloud provider is trustful and that the cloud will not suffer from any attack. However, those are risky assumptions that DNA-SeAl does not make. Consequently, for the alignment task, three categories (already introduced in the beginning of this chapter) of algorithms are used depending on the standard alignment strategy. Cloudburst [Sch09] is the representative of the *non-secure but fast algorithms*, which offers no protection, although it only requires 0.41 CPU seconds if the read is encrypted before it is transferred to the cloud server. The representative of the *secure but slow algorithms* is a homomorphic encrypted based approach called 5PM [Bar+12], which takes 22 CPU days. Finally, the *intermediate protection algorithms* are represented by a hashed k-mer algorithm, proposed in [Che+12], which presents a higher efficiency requiring only 1.3 CPU seconds. However, its security have not been proven yet, and therefore it possibly leaks some information regarding equal K-mers. All the described times are relative to the alignment of one read of 100 nucleotides to the reference human genome, using a single core.

Table 3.3: Privacy, performance and communication overheads of the alignment algorithms used.

Method	**Privacy**	**Computation (CPU time)**	**Communication volume**
Homom. encr. [Bar+12]	Very high	22.08 days	3.75×10^8 KB
Hashed K-mers [Che+12]	High	1.3 sec.	5.22 KB
Cloudburst [Sch09]	Low	0.41 sec.	2.3 KB

The computation overhead and communication costs were evaluated depending on the proportion of public and private clouds available. To give an example, a configuration of 10/1 means that the public cloud is 10 times powerful than the private cloud. In Table 3.3 three configurations are evaluated (1/1, 2/1, and 10/1), and they aim at representing the gains associated to a more powerful public power. The power of public clouds is incremented and the private cloud power remains constant, since public clouds are cheaper and easier to access. For each configuration the following three read alignment possibilities were assessed: (i) on the public cloud only using secure but slow algorithm (5PM [Bar+12]); (ii) on the private cloud only using non-secure but fast algorithm (Cloudburst [Sch09]); or (iii) on both using Cloudburst [Sch09] for the alignment of sensitive reads on the private cloud and using 5PM [Bar+12] for the alignment of insensitive reads on the public cloud.

Table 3.4: Computation overhead of existing privacy-preserving approaches.

Proportion **Pub./Priv.**	**Our approach**	**Previous approaches** **Pub.*** (3×10^8s)	**Priv.**† (0.41s)	**Pub.*****/Pri.**† **with [Cog+15]** (0.29s)
1/1	0.29s	10^6x	1.39x	1x (0.29s)
2/1	0.097s	10^6x	4.20x	1.51x (0.11s)
10/1	0.019s	10^6x	20x	5.85x (0.11s)

† – Private cloud running [Sch09].
* – Public cloud running [Bar+12].

Table 3.4 compares the computation overhead of the DNA-SeAl sensitivity-adapted alignment with the previous approaches and depending on the proportion of public and private clouds available. The results in this table show that it is not possible to rely only on public clouds to align a single read using a secure but slow algorithm, which would require 3.0×10^8 CPU seconds to align one read. Relying on a private cloud only using a non-secure but fast algorithm is the most efficient solution, requiring only 0.41 CPU seconds to align a single read, but it requires a powerful private cloud to scale with the number of reads. The performance of the previous two levels classification [Cog+15] (sensitive or insensitive) can be improved considering that the public cloud has at least the same power as the private cloud (0.29 CPU seconds with [Cog+15] for 1/1 configuration). To conclude, DNA-SeAl improves the performance, reaching the 0.019 seconds when relying on a configuration of 10/1 of public/private clouds.

Table 3.5: Communication overhead of existing privacy-preserving approaches.

Proportion **Pub./Priv.**	**Our approach**	**Previous approaches** **Pub.*** (16.8GB)	**Priv.**† (2.3KB)	**Pub.*****/Pri.**† **with [Cog+15]** (1.6KB)
1/1	1.6KB	10^7x	1.39x	1x (1.6KB)
2/1	0.55KB	10^7x	4.20x	1.51x (0.83KB)
10/1	0.11KB	10^7x	20x	5.85x (0.65KB)

† – Private cloud running [Sch09].
* – Public cloud running [Bar+12].

Table 3.5 compares the communication overhead of the DNA-SeAl sensitivity-adapted alignment with the previous approaches and depending on the proportion of public and private clouds available. The results obtained for the communication

overhead follow patterns similar to those of the computation overhead. The data transfer required while relying only on a public cloud is huge (16.8GB), while relying only on a private cloud requires 2.3KB. Again, considering the sensitive and insensitive classification the communications can be lowered, starting at 1.6KB for public and private clouds with the same power. Finally, DNA-SeAl is at least as good as the sensitive and insensitive classification (1/1 configuration) and it further decreases the communications down to 0.11KB for the 10/1 configuration, i.e., using a public cloud 10 times more powerful than the private cloud.

3.3.4 Privacy improvement

In order to evaluate the privacy improvement due to the use of sensitivity levels two metrics were considered for the privacy evaluation: the genomic privacy and the likelihood ratio (LR) value. The genomic privacy metric was defined by Ayday et al. and it quantifies the weighted risk of re-identification based on the adversary estimates of the minor allele frequencies for the observed SNPs [ARH13; Wag15]. The LR value defines the upper bound of the detection power of a case individual [Jia+14].

Genomic privacy

Figure 3.8 shows the values of the genomic privacy metric for the HG03556 genome, considering the genomic variations distributed among the sensitivity levels defined per chromosome. For the genomic privacy metric, higher values correspond to higher privacy. Considering this, the obtained results show that level 1 (circles line), containing the most sensitive information, has high genomic privacy value for all the chromosomes (between 7.5×10^4 and 7.5×10^5). In addition, this level contains the variations that participate the most to the privacy metric on an individual – since this value is close to the overall genomic privacy value (crosses line), which is computed with all the genomic variations presented by the individual per chromosome. Level 2 (squares line) presents values between 3.0×10^4 and 2.0×10^5, while, level 3 (vertical dashes line) has values bellow 1.0×10^5. Based on these results, it is possible to conclude that more sensitive SNPs contribute more to the privacy metric. Consequently, information classified into levels 2 and 3 is less critical (lower genomic privacy value), and, therefore, it can be processed using a lighter protection when compared to information classified into level 1.

The same study was performed for more individuals of the 1000 Genomes Project. Figures 3.9, 3.10, 3.11 and 3.12 show the results for the genomic privacy analysis for four other individuals.

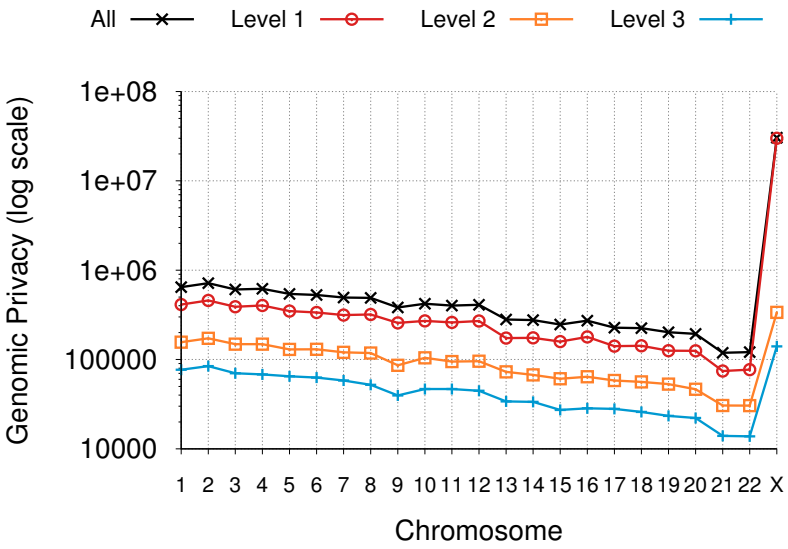

Figure 3.8: **Genomic privacy – sensitivity levels**.

Figure 3.9 shows similar values to the ones described for Figure 3.8. The other three, Figures 3.10, 3.11 and 3.12 present some oscillation on the genomic privacy value of the most sensitive level (level 1) and the intermediate level (level 2). The differences verified between figures 3.9, 3.10, 3.11 and 3.12 show that studying further could help on the definition of the different sensitivity levels, which could be more adapted to distinct genomes. The genomes in the mentioned figures have distinct heritage: HG03048 has African ancestry, HG01086 has American ancestry, HG00096 has European ancestry, and HG01864 has Asian ancestry. Therefore, one possible explanation for the differences in the distributions is that population specific variations might contribute differently in each sensitivity level.

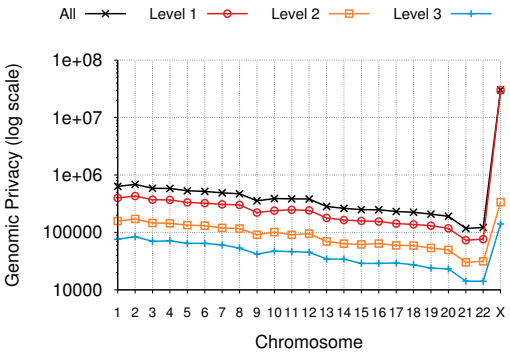

Figure 3.9: **Genomic privacy – HG03048**.

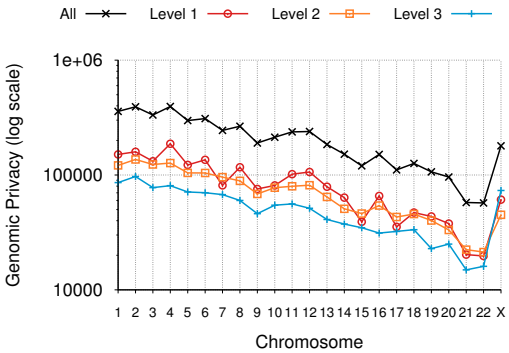

Figure 3.10: **Genomic privacy – HG01086**.

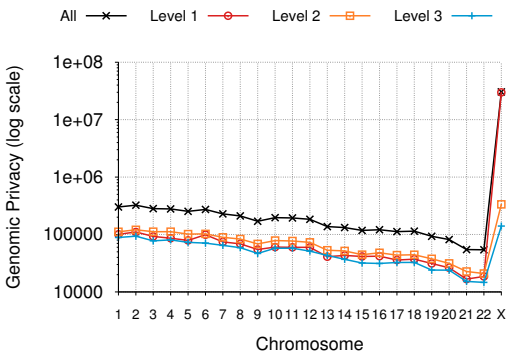

Figure 3.11: **Genomic privacy – HG00096**.

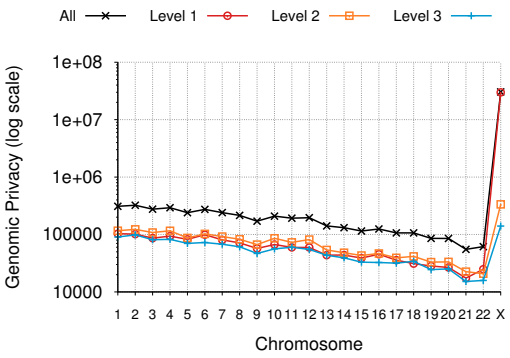

Figure 3.12: **Genomic privacy – HG01864**.

Likelihood ratio value

Figure 3.13 shows the likelihood ratio (LR) value results, which were computed considering a total of 2,014,777 SNP locations. Overall, for each subject, level 1 contained between 130,000 and 158,000 alleles, level 2 contained between 111,000 and 125,000 alleles, and level 3 contained between 1,733,000 and 1,773,000 alleles. Then, random partitions of each sensitivity level were tested regarding the identification of case individuals from the set. The goal was to find the bigger partition that would not allow the identification of any individual as member of the case population. For these experiments, the following three scenarios were compared: (i) without sensitivity levels (in black); (ii) with the full sensitivity levels; and (iii) larger partitions of the sensitivity levels that prevent the link of case individuals with the disease. For scenario (i), the results show that it is possible to link 31.3% of the case individuals with the disease. Scenario (ii), which considers the full sensitivity levels, allows the link of 32.9%, 6.2%, and 3.7% of the case individuals with the disease, based on the information from level 1, 2, and 3, respectively. To conclude, the scenario (iii) prevents the identification of case individuals when considered partitions of 50% for levels 1 and 2, and partitions of 20% for level 3. With these results. it is possible to conclude that randomly partitioning the level enables the processing of sets of variations while preventing linkage attacks. The division of the sensitivity levels presented in Figure 3.13 supports the cloud diversity described earlier in this chapter (see section 3.1.1.2).

Finally, genomic privacy metric supports the use of sensitivity levels for reducing the re-identification risk, and LR metric for reducing the membership detection risk by splitting and protecting adequately each sensitivity level.

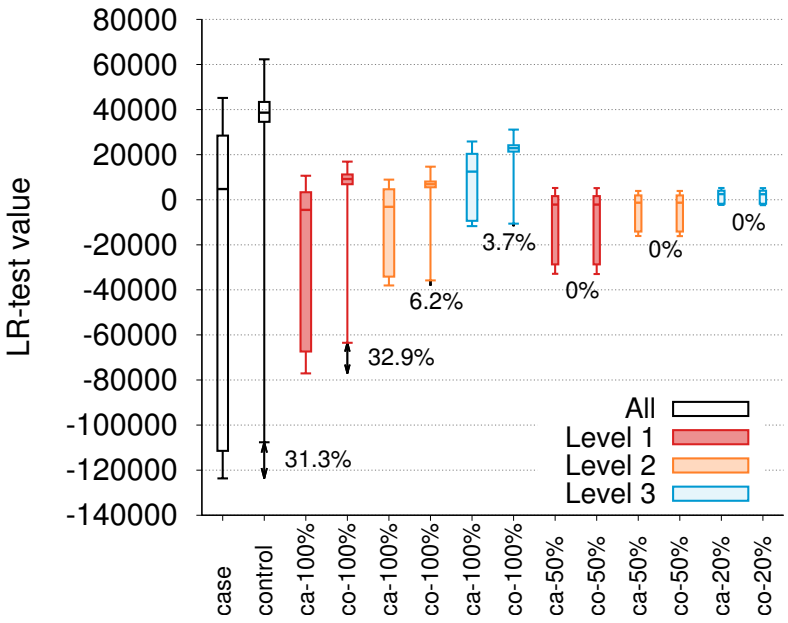

Figure 3.13: **Likelihood ratio test – sensitivity levels**.

3.4 Summary

The sensitivity levels classification proposed in this chapter, DNA-SeAl, is designed to classify unaligned reads. The different levels are created based on qualitative and quantitative properties of the human genome. The qualitative properties refer to biological insights, while the quantitative properties include the allele frequencies in a population and the linkage disequilibrium relations. The presented evaluation considers only levels defined quantitatively, since levels defined based on qualitative features are subjective and they present a small size. After defining the levels based on allele frequencies, DNA-SeAl disconnects the sensitivity levels based on linkage disequilibrium relations in order to prevent amplification attacks. DNA-SeAl classification allows the privacy adapted alignment of reads by using the three existing classes of alignment algorithms: plaintext, encryption-based, and intermediate protection algorithms. The combination of such algorithms enables privacy protection adapted to the sensitivity of the reads, while optimizing performance. The evaluation section shows that DNA-SeAl's approach improves the privacy $\times$ performance product in comparison to the state of the art alignment algorithms, as well as when compared with previous binary classification.

Chapter 4

Long reads filtering

In the previous chapter, DNA-SeAl used the reads classification method, short reads filter (SRF) approach proposed in [Cog+15], which is described in Section 2.7. During the development of the work described in Chapter 3, we found out that this method could be improved in several aspects. The first observation was that this method could not be used for the longer reads produces by the third generation of sequencing machines, since the classification is made per read and a dictionary of sensitive long reads would be huge. In addition, since longer reads are likely to contain more genomic variations than short reads, almost all long reads would be classified as sensitive by the filter described in [Cog+15]. A more detailed analysis of the limitations and the possible improvements of the SRF is presented.

The ideal privacy-preserving workflow would protect sensitive genomic data during all its stages, starting from the moment it is sequenced, during its storage, during its analysis or processing, and while sharing it when it is required. Considering this ideal workflow and addressing the high performance challenge, some interesting questions arise: "How could we enforce privacy protection before the reads alignment?", "Can we detect sensitive information in raw reads in order to protect it before analysing it?". The work in this chapter targets those two questions and addresses the need for classification methods for genomic data [Zha+11].

This chapter addresses the challenge of protecting privacy of raw reads providing high throughput by proposing a long reads filtering (LRF) approach. The approach proposed in this chapter detects sensitive nucleotides in reads, i.e., nucleotides that participate in a genomic variant or short tandem repeat. High filtering throughput is achieved thanks to the use of Bloom filters, which allow fast membership testing and space efficiency. Section 4.1 discusses the limitations of the SRF and explains how the LRF addresses those limitations, and the evalu-

ation section of this chapter (Section 5.4) compares the obtained results with the ones of the previous approach, SRF. The LRF approach is compatible with the sensitivity levels proposed in the previous chapter. However, as represented in Figure 4.2, only two sensitivity levels (sensitive and insensitive) are used to provide a fair comparison with the previous approach (SRF).

Nowadays the research community relies on clouds for the processing of genomic data in order to achieve high performance at an affordable price. Nonetheless, the privacy guarantees provided by the clouds are limited, since they cannot ensure that the hosted data is not accessed by the cloud service provider or by an intruder [RC11]. Therefore, the use of clouds is usually associated to privacy risks when processing biomedical data in the cloud without the adequate protection [MPG14; FEJ15]. The LRF approach classifies the content of the reads as sensitive or insensitive and works on a per-nucleotide basis. While designing this approach the goal was to provide a lightweight, accurate and fast method for reads obfuscation, to enable safe reads alignment relying on plaintext algorithms running in clouds. The LRF does not require parallelization or massive computational resources to achieve high performance. Despite of that the LRF can still benefit from being parallelized. At the end of the filtering process the LRF produces two distinct output files, one with the sensitive nucleotides and the other with the insensitive nucleotides, also called masked reads. Masked reads only contain the nucleotides that are shared by any two individual in the studied population. Since these nucleotides do not leak any personal detail, masked reads can be aligned in the cloud. The sensitive genomic data stays protected in the private environment (e.g., private machine). The feasibility of masked read alignment is assessed in the next chapter of this thesis. Relying on the proposed filter to develop a privacy-preserving alignment strategy relying on plaintext alignment algorithms can contribute to a higher performance in comparison with cryptography-based alignment algorithms, since aligning in plaintext is much faster.

4.1 Sensitive short reads detection limitations

This section presents the limitations of the short reads filtering approach (SRF) proposed by Cogo et al. [Cog+15], which are addressed by the long reads filter (LRF) approach proposed in this chapter. Each limitation is discussed to clarify its importance and a resolution proposal paragraph explaining how the limitation was solved. Then, this section provides a more detailed discussion regarding the long reads filtering and the filtering in the presence of errors, using SRF.

Limitation 1: The SRF can only classify reads with 30 nucleotides (30-mers).

Discussion: SRF receives as input sequences of 30 nucleotides, and consequently, it also uses this length for the sequences used during the Bloom filter initialization. However, nowadays 3rd generation NGS sequencing machines are producing much longer reads (i.e., a few thousands nucleotides), and the SRF is not adapted to such long reads. The per-read classification is not suitable for long reads, since longer reads have higher probability of containing at least one sensitive nucleotide. SRF produces a high proportion of misclassification of insensitive nucleotides when classifying long reads, since if a single nucleotide is present in the read, the full read is classified as sensitive.

Resolution proposal: The LRF approach is not limited to 30 nucleotides as the previous approach. In fact, its best K-mers length is 34 as discussed later in this chapter. In the LRF approach the K-mers length is a parameter completely characterizable by the user.

Limitation 2: SRF uses a non-overlapping window based classification.

Discussion: This approach saves computational time since each nucleotide is only read once, in a window of 30 nucleotides. However, this non-overlapping window approach affects the accuracy of the filter and the false positive rate (see Limitation 3).

Resolution proposal: LRF uses a sliding window approach to detect the sensitive sequences. With this approach a single nucleotide is assessed multiple times, the window slides one position each time, i.e., removing one nucleotide from the beginning of the window and adding the next one at the end of the window.

Limitation 3: SRF classifies the full read, which contributes to a high percentage (60%) of false positives.

Discussion: Due to the full read classification, even if there is only one sensitive nucleotide, all the 30 nucleotides are classified sensitive. This process leads to an over classification of sensitive nucleotides, even if in practice they are not sensitive.

Resolution proposal: The LRF uses a nucleotide-based classification, instead of the full read-based classification performed by SRF. In other words, for each window match, the LRF classifies a single nucleotide within that window

as sensitive, per Bloom filter. The index of the sensitive nucleotide is defined during the dictionaries' initialization (see section 4.2.1.1).

Limitation 4: The SRF fails to detect a great percentage of sensitive nucleotides, since they are not included in the initialization of the filter. First, the SRF approach does not consider that multiple genomic variations can occur in the same K-mer, since the creation of each sensitive sequence consists on introducing a single genomic variation in the reference genome sequence and then collect the all possible 30 nucleotides sequences containing that variation. In addition, the SRF excludes genomic variation locations that are not bi-allelic and it only considers SNPs, i.e., genomic variations that mutate a single nucleotide. Finally, the SRF only considers genomic variations where the two alleles are the same for the double DNA strand of the individuals from the 1000 Genomes Project.

Discussion: The SRF generates sensitive sequences without considering that multiple genomic variations can appear in the same sensitive sequence, even for short reads with 30 nucleotides. Therefore, since Bloom filters work in a exact match basis, they are not detected since they are not present in the set of sensitive sequences during the initialization step. If one SNP is discovered after the creation of the sensitive sequences, then it will not be detected until it is added to the sensitive sequences dictionary. However, the dictionary can be updated at any point in time.

Resolution proposal: The LRF approach considers all the possible combinations of up to 8 genomic variations within a sensitive sequence of length K (K is a parameter defined by the user during the creation of the sensitive sequences to insert in the Bloom filter). Furthermore, when creating those sensitive sequences, the LRF does not apply the exclusions described above for the SRF. Those restrictions would make the dictionaries of sensitive reads incomplete, which might make the filter miss the detection of some sensitive nucleotides.

Limitation 5: The SRF does not consider the detection of sensitive nucleotides in the presence of errors since short reads have small error rates. This mainly occurs because Bloom filters work on an exact match base, and are, therefore, limited to the detection of the sequences their dictionaries were initialized with.

Discussion: Even though short reads present a small error rate, errors still occur and the SRF is not designed to deal with them. Therefore, some

sensitive nucleotides can be missed, and consequently, they might lead to sensitive information leakage.

Resolution proposal: The LRF approach combines several filters, which does not result in a high false positive rate, since it uses per-nucleotide classification.

Detection of sensitive nucleotides in long reads using SRF

Long reads are emerging and becoming a promising area, with several advantages on their study, as discussed in Section 2.3. To evaluate the limitations of the short reads filter (SRF) when processing long reads, we studied the proportion of reads with at least one sensitive nucleotide. The purpose of this study is to demonstrate that the short reads filter (SRF), which classifies sequences of 30 nucleotides as sensitive or insensitive, cannot be extended for long reads.

Figure 4.1 shows the proportion of reads with at least one sensitive nucleotide. This proportion increases quickly with the reads size. For example, for reads of 100 nucleotides 88% of the reads are classified sensitive, and for reads of 1000 nucleotides 95% are sensitive. Therefore, SRF would classify nearly all the filtered long reads as sensitive, which is not practical.

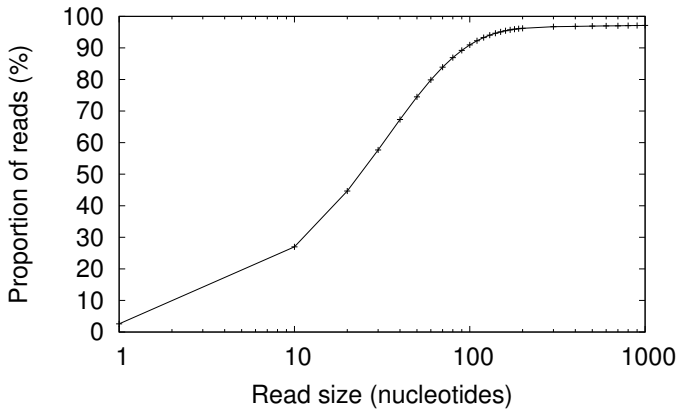

Figure 4.1: **Proportion of reads containing at least one sensitive nucleotide depending on the reads size (logarithmic size)**.

These results highlight the importance of the long reads filter proposed in this thesis, which is able to filter reads of arbitrary length (short and long reads).

Detection of sensitive nucleotides in the presence of errors using SRF

Using SRF, if a 30 nucleotides sequence matches with the sequences in the Bloom filter then it is classified as sensitive. However, if some nucleotide is modified, deleted or inserted in the sequence, there is a high probability that it will not match with the Bloom filter sequences, since Bloom filters perform exact matching. Considering P_m the probability that a nucleotide is modified by the sequencing machine, the probability that all 29 nucleotides in the SRF window are correct is represented by $(1-P_m)^{29}$. Only 29 nucleotides are considered as possibly modified, since it is assumed that at least one sensitive nucleotide is correct. Otherwise, if a sequence contains a single sensitive nucleotide and this nucleotide is modified by the sequencing machine, the sequence becomes insensitive.

Table 4.1 shows the proportion of sensitive nucleotides detected for different error rates (0.1%, 1%, 2% and 4%). As the results show, increasing the error rate quickly degrades the proportion of sensitive nucleotides detected. Those missed sensitive nucleotides are part of genomic variations, which might become available for performing attacks.

Table 4.1: Proportion of sensitive nucleotides detected per error rate.

Error rate	0.1%	1%	2%	4%
Sens. nucleotides	97%	75%	56%	31%

To conclude, errors in the reads highly compromise the detection of sensitive nucleotides performed by SRF. This supports the importance of the long reads filter proposed in this thesis, since it is able to tolerate sequencing errors, as explained later in Section 4.2.3.

4.2 Methods

4.2.1 Sensitive nucleotides excision for genomic reads

This chapter proposes a filtering method for long reads which extends the filtering approach for short reads proposed by Cogo et al. [Cog+15].

Figure 4.2 shows the overview of the process from the sequencing of DNA until the step where the filtering output is obtained. This figure represents a filtering example considering two levels of sensitivity for the information (sensitive and insensitive, respectively, in red and grey).

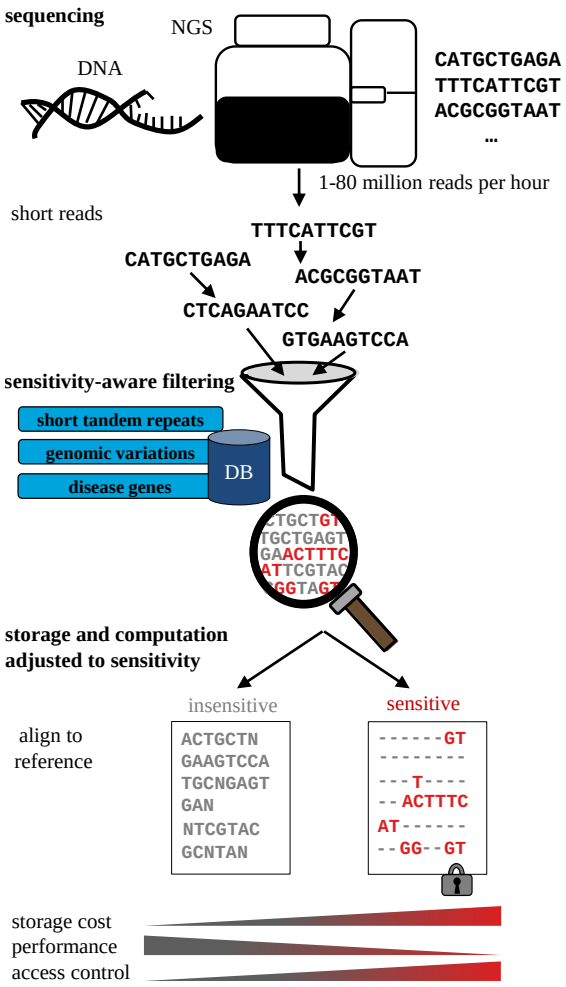

Figure 4.2: **DNA reads filtering process**.

At first a DNA sample is sequenced, and raw reads produced by sequencing machines. These reads are the input of the filtering method. Second, the raw reads are filtered, which consist in classifying each nucleotide in the read as sensitive (red) or insensitive (grey). After the filtering, the storage cost and computational power can be adapted to the sensitivity of the data. Sensitive information requires higher protection, which in general is more costly and requires more computational resources for processing. In addition, this division of the data also allows the application of distinct access control settings.

The main steps of this approach are the following: generation of the sensitive sequences, Bloom filter initialization, and long reads filtering. The next sections describe the generation of the sensitive sequences and the long reads filtering step. The Bloom filter initialization does not differ from the previous approach (see

Section 2.7), therefore, it is not detailed in this chapter.

This approach is compatible with the sensitivity levels proposed in the previous chapter. As represented in Figure 4.2, this chapter considers only two sensitivity levels (sensitive and insensitive) in order to provide a fair comparison with the previous approach (SRF), which uses those same levels.

4.2.1.1 Sensitive sequences generation

The very first step of the proposed long reads filter is the creation of dictionaries of sensitive sequences. The collection of sensitive sequences should be as complete as possible, i.e., it should contain all the known genomic variants and STRs for the human genome, since this factor influences the accuracy of the filter. The used nomenclature defines $dict_i$ as the dictionary containing (K, i)-sequences, i.e., sequences with K nucleotides where the i^{th} nucleotide is sensitive. A nucleotide is considered sensitive if it participates in a genomic variation or STR.

In order to build a complete database with the known genomic variations and since LRF does not limit the reads length, it is important to consider that a single read can contain multiple genomic variations, independently of its size. Hence, it is important to determine the number of variations that can be found in a single read to avoid the misclassification of sensitive nucleotides due to incomplete sensitive sequence sets. Therefore, we studied the cumulative proportion of genomic variations (GVs) that contain from 0 to 10 neighbour GVs at a maximum distance of 30 nucleotides, located either before or after. Figure 4.3 presents the results of this study, which show that combinations of up to 8 variations in a single read allow the detection of 99.96% of the genomic variations.

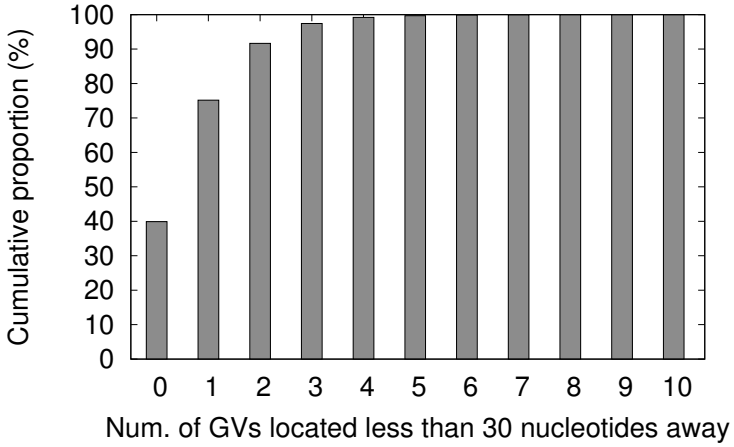

Figure 4.3: **Neighbours SNPs study**.

Genomic variations combinations

Each human individual has a unique combination of genomic variations with at least two possible alleles for each genomic variation location. Therefore, in sequenced reads it is possible to find multiple variations. SRF considers only one variation per sequence. In case multiple genomic variations are present within the range of 30 nucleotides, only one possible combination was considered: the alternative allele for one genomic variation and the reference allele for all the other genomic variations in the sequence.

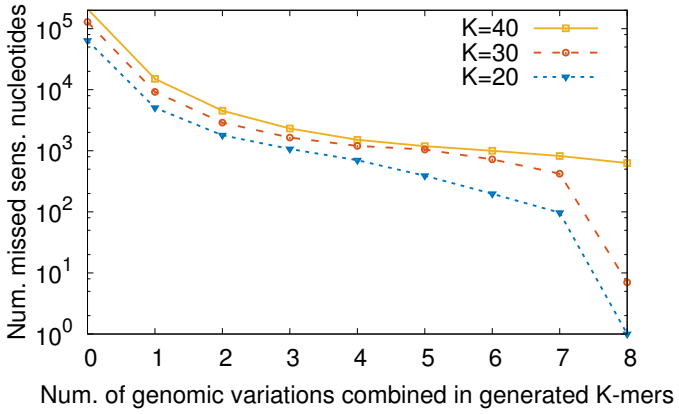

Figure 4.4: **Number of missing sensitive nucleotides – genomic variations combination**.

To evaluate the importance of combining genomic variations present in a sequence for the detection of sensitive nucleotides, we studied the number of missed sensitive nucleotides considering different maximum number of genomic variations (up to 8) combined in each sequence inserted in the Bloom filter. The maximum of 8 combinations was defined by the study of the number of variations within a 30 nucleotides sequence (Figure 4.3). In these experiments three different K-mer lengths (20, 30 and 40) were studied. Figure 4.4 represents the results of the study. SRF results are represented by the $K = 30$ line, in particular the point considering 0 variations combined. Therefore, it misses 128 860 sensitive nucleotides. For shorter reads ($K = 20$), combining 8 genomic variations drastically decreases the number of missed sensitive nucleotides. From 63 773 missed sensitive nucleotides with 0 variations combined to 1 missed with 8 variations combined. Finally, for longer reads ($K = 40$), combinations considering 8 genomic variations also improves the sensitive nucleotides detection, however, the number of missed nucleotides is higher than the one obtained for the other k-mer lengths. From 210 789 missed sensitive nucleotides with 0 variations combined to 631 missed with 8

variations combined. This study highlights the importance of combining genomic variations within each sequence to maximize the sensitive nucleotides detection.

In order to prevent missing the detection of sensitive nucleotides, the LRF proposed in this thesis considers combinations of genomic variations when creating the dictionaries of sensitive sequences. Regarding the resources requirements, combining two genomic variations in sequences of 30 nucleotides increases the required memory from 5GB (without genomic variations combinations) to 7GB each sensitive sequences file, which is not a great difference. Naturally, the size of the sensitive sequences file increases with the number of genomic variations considered for the combinations. Computing all the possible combinations without an upper bound will lead to the creation of a huge number of sensitive sequences and consequently the dictionary files would grow to tens of Gigabytes. Therefore, an upper bound for the the number of combinations needs to be defined. This thesis considers 8 variations for the combinations, supported by the results in Figure 4.4.

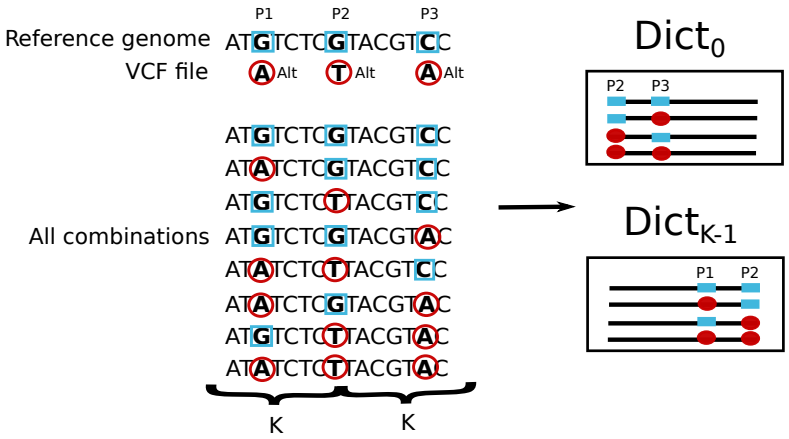

Figure 4.5: **Genomic variations combinations**. This figure illustrates how to create the sensitive sequences of length K for regions of the genome with multiple genomic variations.

Figure 4.5 illustrates the process of creating the all the possible combinations of genomic variations for sequences of length K = 8. The example in the figure focuses on genomic variation at the position P_2 as the sensitive nucleotide of the sequence introduced in the dictionaries, either located in the first position (index 0) for $Dict_0$ or in the last position (index $K-1$) for $Dict_{K-1}$. In this case there are two genomic variations (P_1 and P_3) located within the range of K nucleotides from variation P_2. For all three variations there are two alleles, one called reference allele (represented by blue squares) and the other called alternative allele (represented by red circles). Since three genomic variation with two possible alleles each are

considered, creating all the combinations results in $2^3 = 8$ sequences. Since the dictionaries include sequences of K nucleotides where P_2 is either in the first or in the last position, each dictionary contains only four sequences.

4.2.1.2 Long reads filtering

The long reads filtering approach uses one or multiple filters, initialized with the dictionaries of (K, i)-sensitive sequences generated as described in the previous section. A sensitive sequence of K nucleotides where the i^{th} nucleotide is sensitive is called (K, i)-sensitive. Then, the reads are filtered by assessing all the K-mers in a sliding window fashion. For each K-mer the Bloom filter checks if it was inserted in the Bloom filter and if yes, it contains a sensitive nucleotide at position i. For each match with a K-mer in the Bloom filter, a single nucleotide is detected sensitive.

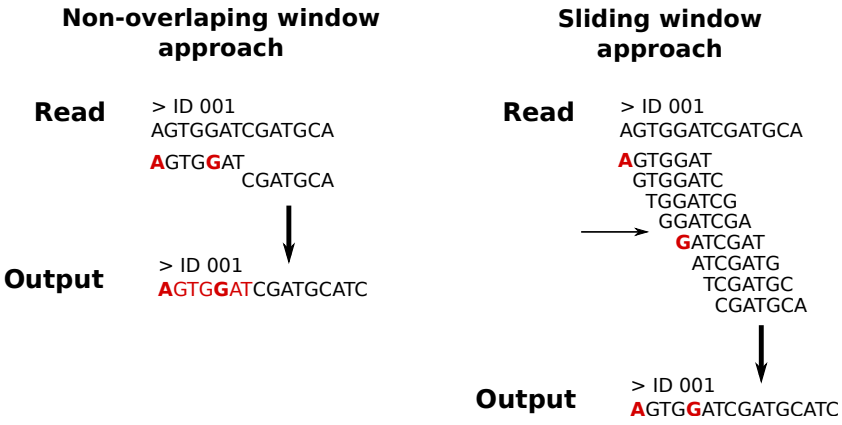

Figure 4.6: **Non-overlapping window and sliding window approaches**.

Figure 4.6 highlights one of the main differences between the filtering approach proposed in this thesis, which uses sliding window approach, and the previous filtering approach, which uses non-overlapping approach. In this figure, the truly sensitive nucleotides are represented in red with the truly sensitive nucleotides in bold, while the insensitive nucleotides are in black. Briefly, the non-overlapping window approach (SRF approach) studies sequences (windows) of length K (in the figure $K = 7$). The first window studied contains the first 7 nucleotides of the read, and the next window ($w = 1$) starts at the first non-studied nucleotide, whose index is $K \times w + 1$, where w represents the window number and K is the window size. Then in case of match, the full window is classified as sensitive (red) even though only two nucleotides ('A' and 'G') are truly sensitive.

In the sliding window approach, the sequences (windows) studied overlap in $K-1$ nucleotides, considering K the window length. Then, as explained previously in this section, in case of match a single nucleotide is classified as sensitive.

Illustrated by the figure, the sliding window approach has a more accurate detection of sensitive nucleotides. Contrary, the non-overlapping approach classifies as sensitive several nucleotides which in practice are not, since for each match all the K nucleotides are detected sensitive. In addition, longer reads have higher probability of containing at least one sensitive level (see Figure 4.1) and they are more likely to contain more variations than short reads. The comparison of the false positive rate for those two approaches is discusses in more detail later in Section 4.4.2.

4.2.2 Multiple Bloom filters to improve accuracy

The nucleotide detection used by LRF, using a single Bloom filter, can lead to the misclassification of sensitive nucleotides, as insensitive.

Figure 4.7 illustrates the partitioning of reads according to their sensitivity. The Bloom filters are initialized with all the K-mers (K=10, in this example) where there is a sensitivity nucleotide (represented in red) at a certain position. In this example are represented three Bloom filters that present the sensitive nucleotide in the first, middle and last position of their sequences, being called, respectively, BF1, BF2 and BF3. For each read the three Bloom filters check if there is a match between the sequences they contain and the read subsequence of the same size, in order to define the sensitivity of each nucleotide in the read. In this case the non-studied parts problem does not apply, since all the nucleotides of each read are covered by at least one Bloom filter. In figure 4.7 only the sequences in each Bloom filter that match with the read are represented, to simplify. However, they contain more information. The filter outputs two files, one with a suffix '.pri' and the other with '.pub', containing, respectively, the privacy sensitive and insensitive nucleotides – also referred to as masked reads, since the reads are in their majority composed by insensitive nucleotides. For the insensitive the sensitive nucleotides are excised by replacing them with a 'N' symbol. However, in case of consecutive sensitive nucleotides they are all replaced by a single 'N', as represented in the figure.

The first sensitive nucleotide (red 'A') is only detected by BF1, while the last sensitive nucleotide (red 'G') is only detected by BF3. This supports the use of multiple Bloom filters with the sensitive nucleotide locate at different indexes for improving the detection. Therefore, the combination of multiple Bloom filters helps to catch all the sensitive nucleotides present in the reads. Nonetheless, the

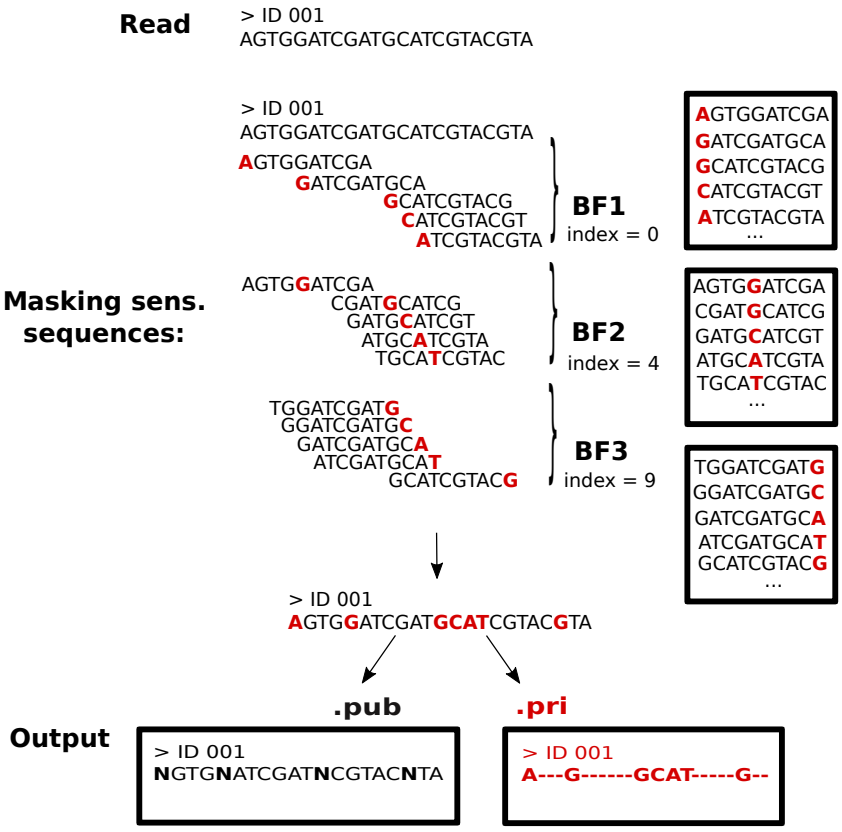

Figure 4.7: **Partitioning of reads according to the sensitivity of their nucleotides**.

use of multiple Bloom filters implies an increase on the memory requirements and a decrease in the throughput. Therefore, when deciding the number of Bloom filters to use, the accuracy gains and the resources requirements need to be considered.

4.2.3 Tolerating sequencing errors

Until now, this chapter focused on the detection of sensitive nucleotides in error-free reads. However, even with the constant evolution of sequencing technologies, it is important to be able to tolerate errors in the reads. This section discusses how to use the error-free detection approach to tolerate some sequencing errors. Mainly, our strategy is to combine several Bloom filters which have been initialized with different (K, i)-mers, where K is the length and i is the index of the sensitive nucleotide. Normally, iterating the detection process, i.e., use multiple Bloom filters, increases the probability that a sensitive nucleotide is detected in the presence of sequencing errors. Even with the small error rate presented by

current sequencing machines (around 1% for 2^{nd} generation), errors in the reads might affect the sensitive nucleotides detection. To achieve good performance, the several Bloom filters can be run in parallel, for example using several threads. This is possible since all the filters fit in the memory of the machine used for the experiments. Nevertheless, depending on the computational resources available, the implementation of the several filters can be done differently. For example, they can be run sequentially if a single machine is available (low throughput) or they can be run in parallel when several machines are available (high throughput).

Each Bloom filter tags a subset of sensitive nucleotides in the filtered read. When all the Bloom filters finish, the union of the subsets of nucleotides classified as sensitive represents the final set of sensitive nucleotides.

Figure 4.8 demonstrates how the use of multiple Bloom filters allow the detection of sensitive in the presence of sequencing errors. The figure presents an example using three Bloom filters, where two of them fail on detecting the sensitive nucleotide in the read (BF_{10} and BF_0) and one is able to detect it (BF_5).

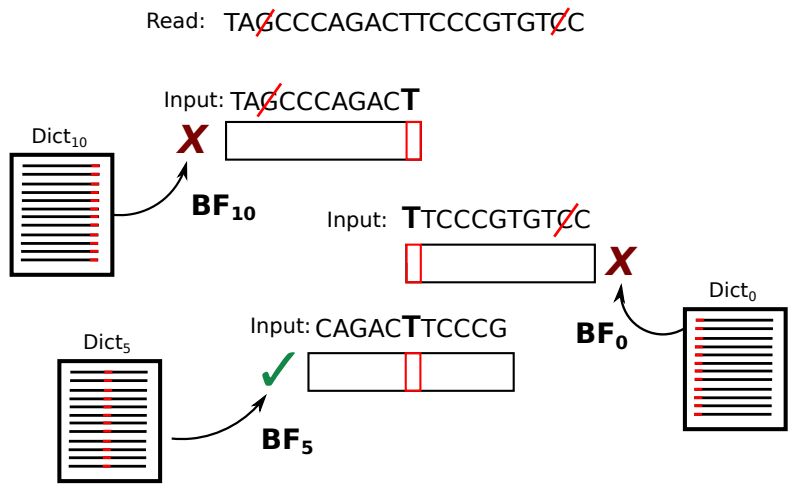

Figure 4.8: **Partitioning of reads with errors according to the sensitivity of their nucleotides**.

4.3 Evaluation Setup

4.3.1 System model

Nowadays, the reads come out from the sequencing machine directly to a private computer (e.g. lab computer or researcher laptop), for example using MinION sequencing device from Nanopore [Tec20], or they are sent directly to a cloud

environment, as allowed by BaseSpace Sequence Hub provided by Illumina [Ill19]. Since the main purposed of the work in this chapter is to protect raw reads as soon as they are sequenced, for the first case, it is assumed that the reads transmission to the computer is not compromised and the filtering is run in the private computer. For the second case, depending on the type of cloud environment used (private or public), the filtering step can be run in the private cloud directly, as soon as it is transferred, or for the public cloud case, the filtering should be performed in a trusted environment and only after the reads are sent to the cloud. As discussed later in the chapter, when using public clouds some reads information might not be sent to the cloud.

The system described in this section emulates a biocenter that applies the filtering method using their trusted resources and then relies on cloud environments to achieve high performance during the data processing.

4.3.2 Threat model

The proposed method focuses on hiding sensitive information contained in raw reads from any unintended adversary, which is assumed to be honest-but-curious. The goal of the adversary is to perform privacy attacks using raw reads observed before or during the alignment run in the cloud.

4.3.3 Hardware

The experiments related to the work in this chapter were executed on a quad-socket Intel Xeon E5-4650 v3 processor with 12 cores machine running at 2.10 GHz. Regarding the memory, the used machine was equipped with 190 GB of RAM and 15 TB of disk. The described machine configuration allows the parallelization of the experiments and the creation of different sensitive databases which assume different parameters of the filters. Even with the described computational resources, it is not possible to create intermediate files considering the combination of more than 8 genomic variations within the same sequence. However, this does not significantly affect the performed experiments since only a minimal ($< 0.05\%$) proportion of genomic variations have more than 8 other genomic variations in the neighbouring regions (up to 30 nucleotides away left and right) (see Figure 4.3).

4.3.4 Software

The the generation of the dictionaries of sensitive sequences was coded in Python. The long reads filter and the implementation of the short reads filter used in this

work was codded in C++ for performance optimization. The long reads filter comprises the following steps: reading the input file containing the raw reads, initialization of the Bloom filter(s) using the sensitive sequences, and reads filtering.

4.3.5 Data

The sensitive sequences dictionaries used in the experiments were built as described in Section 4.2.1.1, using genomic variations and individual genomes from the 1000 Genomes Project [IGS]. The STRs were collected from the Tandem Repeats Database [GRB07]. The sensitive sequences were reconstructed base on the GRCh38 Phase 3 20130502 version of the human reference genome from the 1000 Genomes Project.

4.4 Results

This section presents the comparison of the proposed approach (LRF) and the previous filtering approach (SRF). For the proposed approach the evaluation includes the use of one, two, and three Bloom filters, respectively, named LRF-1, LRF-2, and LRF-3. In addition, different K-mer sizes ($15 \leq K \leq 50$) were tested for the accuracy and memory consumption evaluations. An important parameter of the evaluation is the false positive rate which refers to the proportion of nucleotides detected as sensitive that in fact are not since they do not participate neither in a genomic variation neither in a short tandem repeat (STR). The performed evaluation considered different false positive rates (between 1×10^{-6} and 0.2).

4.4.1 Sensitive nucleotides detection

Figures 4.9 and 4.10 represent the proportion of nucleotides classified as sensitive (either true positive or false positive) considering, respectively, genomic variations only or STRs only. In both figures, the short reads filtering (SRF) and the long reads filter using one (LRF-1), two (LRF-2) or three (LRF-3) filters are compared. In addition, in both figures is also represented the minimum theoretical percentage of sensitive nucleotides (yellow line), being approximately 3% for the genomic variations and approximately 0.5% for the STRs. Focusing on the genomic variations only (Figure 4.9), LRF improves significantly the accuracy on detecting the true sensitive nucleotides, reaching percentages of 6.3% using LRF-1, 8.6% using LRF-2, and 9.6% using LRF-3 (results for 34-mers), while SRF is not able to go bellow 50% for all K-mers evaluated. Comparing the three LRF approaches,

LRF-1 is the closest to the minimal theoretical proportion of sensitive nucleotides, which corresponds to the smallest false positive rate. However, LRF-2 and LRF-3 combine a smaller false positive rate (around 10% for 34-mers) and a smaller false negative rate, i.e., sensitive nucleotides missed (less than 10 sensitive nucleotides for 34-mers). The next section compares the false positives of SRF and LRF approaches.

The analysis using only STRs information produces similar results (Figure 4.10).

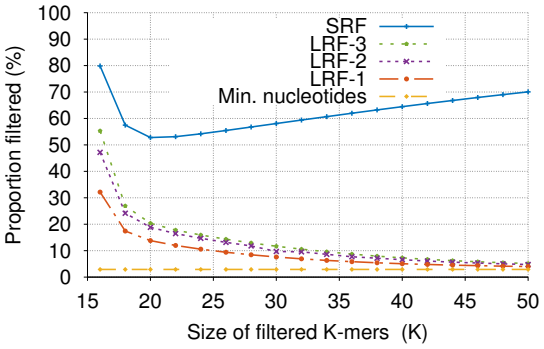

Figure 4.9: **Proportion of sensitive nucleotides – GVs only**.

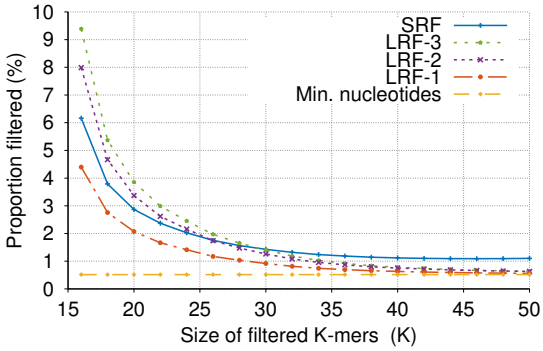

Figure 4.10: **Proportion of sensitive nucleotides – STRs only**.

4.4.2 False positives rate

The false positive rate is an important feature of the filtering approach, since it can influence the performance of downstream workflow steps, i.e., alignment and variant calling.

Figure 4.11 shows the proportion of nucleotides classified as sensitive which in practice do not participate in any genomic variation or STR (false positive). To reduce the number of sensitive nucleotides missed by the filters, the size of the dictionaries used was increased due to the inclusion of all the combinations of genomic variations. This leads to an increase of the false positive rates. Filtering short reads (16 nucleotides) produces high false positive rates (between 90% and 100%), independently of the filtering approach used. For longer reads, SRF still produces a very high false positive rate (around 95%), while for LRF-1, LRF-2, and LRF-3 the false positive rate is significantly lower, respectively, 30%, 40% and 43% (values for reads of 50 nucleotides). These results support the increase of the K-mers length to decrease the false positive rate. Although, larger K-mers increase the probability of missed sensitive nucleotides as shown in Figure 4.4. Decreasing the false positive rate is very important, since more insensitive information can be

used for the alignment. However, if not detected before, after the alignment those false positives are detected and removed from the set of sensitive nucleotides.

Comparing the LRF with the SRF, the false positive rate of LRF is considerably smaller. The main reason of this difference is the use of a non-overlapping approach by the SRF, since if at least one sensitive nucleotide is present in a K-mer, all the nucleotides in that K-mer are classified sensitive.

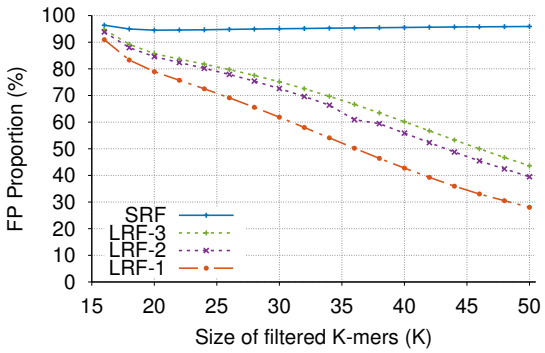

Figure 4.11: **False positive proportion comparison**. False positive rate for a full genome using the SRF and LRF per K-mers size.

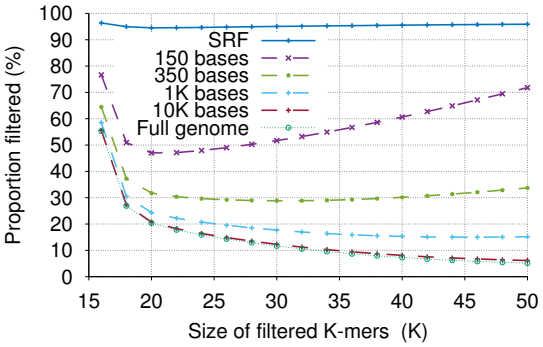

Figure 4.12: **Proportion of reads filtered**. SRF is limited to reads of 30 bases, while for LRF reads of 150, 350, 1000, 10K nucleotides and the full genome were filtered.

Figure 4.12 shows the proportion of filtered nucleotides depending on the length of the K-mers ($15 \leq K \leq 50$). The proposed LRF approach was compared with the previous approach (SRF) considering different reads lengths (150, 350, 1000, 10,000, and full genome). As the results show, SRF has a proportion of filtered nucleotides above 90% for all the K-mer sizes, which corresponds to an high rate of false positives. In contrast, our approach decreases the proportion of filtered nucleotides with the increase of the length of K-mers used by the filter.

4.4.3 Tolerating sequencing errors

Considering that the human genome mutates over the time which happens with a probability of mutation per nucleotide, P_m. The probability P_{detect} that a sensitive nucleotide is detected by at least one of the B Bloom filters used can be represented by the following expression:

$$P_{detect} = \sum_{mut=0}^{F} P_{mut} \times P_{detect|mut} \tag{4.1}$$

where P_{mut} represents the probability that the sequence contains M mutations, and $P_{detect|mut}$ represents the probability that a sensitive nucleotide is detected when the sequence contains M mutations. Furthermore, the probability that precisely M mutations occur in a sequence of $2F$ nucleotides, P_{M2F}, follows a binomial distribution and it can be represented by the following expression:

$$P_{M2F} = C_{2F}^{M}.P_{m}^{M}.(1 - P_{m})^{2F-M} \tag{4.2}$$

Both P_{mut} and $P_{detect|mut}$ decrease quickly with the increase of the number of mutation M. Therefore, it is appropriated to evaluate the probability that a sensitive nucleotide is present in the presence of mutations, depending on the number of Bloom filters used for detection.

Figure 4.13 shows the proportion of nucleotides classified as sensitive depending on the K-mers length, assuming a error rate of 2%. The 2% error rate considered in these experiments represents the worse case, since 2^{nd} generation sequencing machines have currently lower error rates. The same error rate was assumed for long reads, since technologies are under development and it is expected that they reach lower error rates. The results in the figure show that shorter the sequence, higher is that probability that the sensitive nucleotide will be detected. For the different K-mer lengths studied, 20, 24, 30 and 34, the probability of detections is around 94%, 92%, 88% and 84%, respectively. As the slope in Figure 4.13 demonstrates, using more than three Bloom filters improves slowly the probability of detection. Consequently, using three Bloom filters is the ideal setup for a good detection and overhead balance. Regarding the K-mers size, shorter K-mers were not considered since they would increase the proportion of false positives, as discussed in a previous section (see Section 4.4.2).

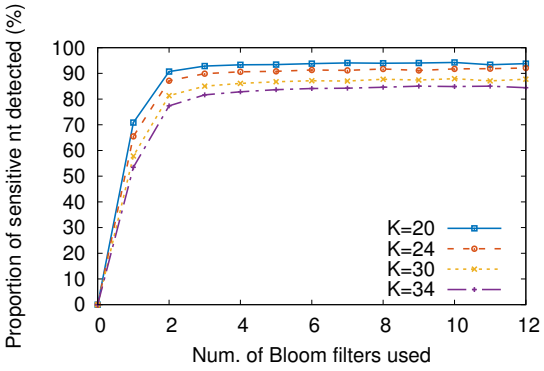

Figure 4.13: **Proportion of sensitive using multiple Bloom filters (2% error rate) depending on the K-mers size.**

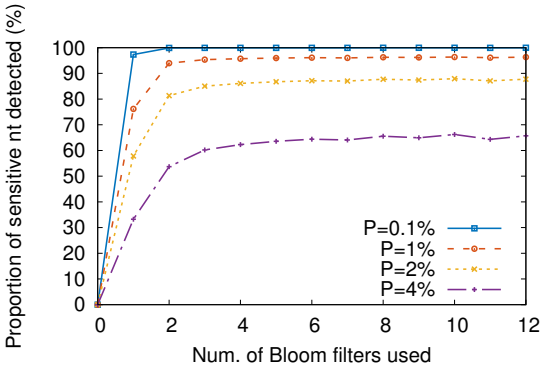

Figure 4.14: **Proportion of sensitive using multiple Bloom filters (30-mers) depending on the error rate.**

Figure 4.14 presents the proportion of nucleotides classified as sensitive depending of the error rate, for 30-mers. Even though the optimal K-mers length for LRF is 34, the experiments were run for 30-mers for a fair comparison with the SRF approach that only works with 30-mers. As represented in the figure, higher the error rate, lower is the percentage of sensitive nucleotides detected. For the different error rates studied, 0.1%, 1%, 2% and 4%, the proportion of sensitive nucleotides was 99.97%, 96%, 86% and 62%, respectively. These values need to be compared with the SRF results (see Table 4.1). Taking the 2% error rate as example, LRF is able to detect 86% of the sensitive nucleotides whiel SRF detects only 56%.

4.4.4 Memory consumption

The memory consumption of SRF and LRF were compared accordingly to the length of the sensitive sequences (K-mers) inserted in the Bloom filters.

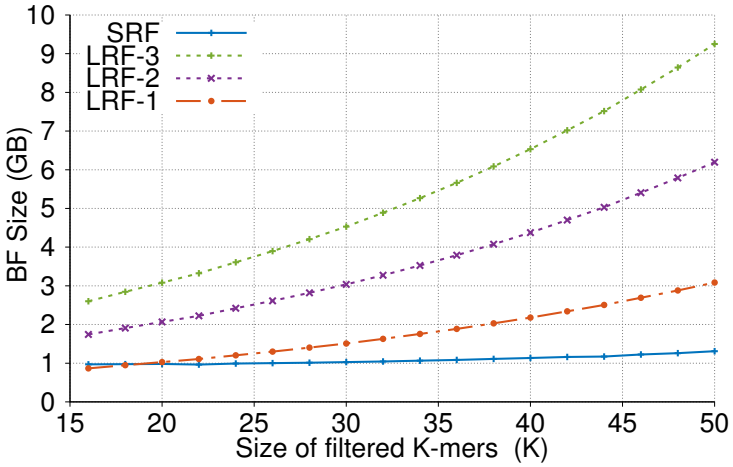

Figure 4.15: **Bloom filter size comparison**.

Figure 4.15 shows the Bloom filter size in GB for each filtering approach. The first observation from the figures is that the Bloom filter size increases with the length of the sensitive sequences used in their initialization. For the LRF is considered a sliding window approach which requires the generation of more sensitive sequences than the non-overlapping window approach adopted by SRF. In addition, longer reads have a higher probability of containing more sensitive nucleotide than short reads. Consequently, for the LRF more combinations of genomic variations need to be considered, producing more sensitive sequences. For sensitive sequences of 50 nucleotides (50-mers), the Bloom filter requires 3 GB of memory

for the LRF-1 approach, while SRF's Bloom filter requires only 1.5 GB. This difference is mainly due to the combinations of genomic variations generated in the LRF approach.

Finally, comparing the use of a single Bloom filter (LRF-1) with the use of multiple Bloom filters (LRF-2 and LRF-3), the memory required increases linearly with the number of Bloom filters used. For example, considering 50-mers, LRF-3 uses approximately 9 GB, which is three times the memory used by LRF-1. Finally, the 9 GB of memory required by LRF-3 are plausible for the current memory hardware capacity.

4.4.5 Performance comparison

The throughput of SRF and LRF were compared considering sensitive sequences of 34 nucleotides. This length was selected since for the LRF it is the smallest size for which LRF-3 produces less than 10% of false positives.

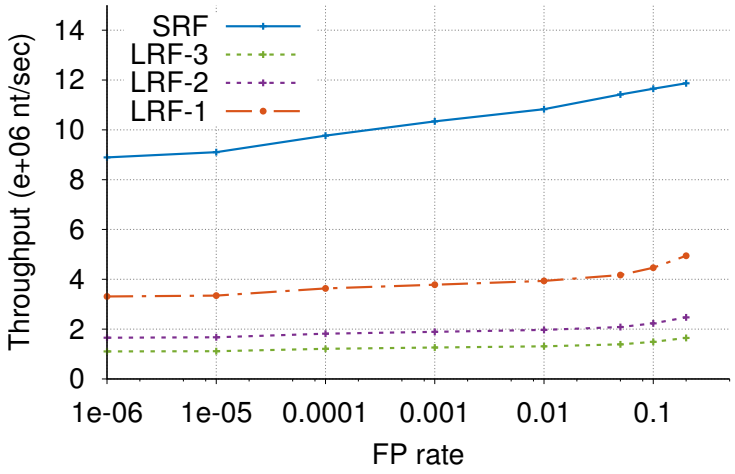

Figure 4.16: **Throughput comparison**.

Figure 4.16 compares the throughput in 10^6 nucleotides per second for the different filtering configurations, depending on the defined false positives rate. The figure also shows that higher the false positive rate considered, higher is the throughput. This occurs since higher false positive rates are associated to less hash functions, and consequently the Bloom filter size is smaller. Therefore, the searching space is also smaller, which leads to a higher throughput. Comparing the filtering approaches, the LRF-1, the fastest setting of LRF using a single Bloom filter is in average 3× slower than SRF. This occurs since SRF studies the sequences using non-overlapping window approach, while LRF uses sliding window approach.

Briefly, LRF-1 sacrifices some performance compared to SRF, with a throughput between 9.1 and 12 million nucleotides per second, to provide better false positive and false negative rates. Finally, LRF-1, with a throughput between 3.3 and 4.9 millions nucleotides per second depending on the false positive rate, is still ten times faster than the sequencing machines throughput, which is approximately 0.3 millions of bp/sec [Liu+12]. Thus introducing it in the traditional pipeline does not create any throughput bottleneck.

4.5 Summary

The long reads filter (LRF) proposed in this chapter allows the detection of sensitive nucleotides in long reads, which are produced by most recent generation of sequencing machines. The LRF improves the state-of-the-art solution, the short reads filter (SRF) porposed in [Cog+15] regarding the false negatives – missing less than 10 sensitive nucleotide in comparison with the 100 000 nucleotides missed by the SRF – and false positives – 10% of an individual genome is classified as sensitive instead of the 60% classified by SRF approach. The LRF is also able to tolerate some errors, which was not possible with the previous work. For a error rate of 2%, LRF is able to detect 86% of the sensitive nucleotides instead of 56%. Although the memory and CPU requirements of the LRF approach are not negligible, its throughput is higher than the current sequencing machines. In addition, it can be easily parallelized, which allows the proposed approach to adapt to the throughput evolution of the sequencing machines. LRF is able to filter reads of any size and it was designed to detect all kinds of genomic variations, missing less than 10 sensitive nucleotides per genome.

The LRF can be integrated in the DNA-SeAl approach, replacing the used SRF from [Cog+15]. Similarly, it would apply one filter per sensitivity level.

Finally, the proposed long reads filter opens the path for the introduction of the filtering approach into the traditional workflow, in order to provide early protection to the raw sequenced data. Furthermore, it allows the distinct processing of sensitive and insensitive information, the former being more privacy critical and, therefore, stored in a separate and more secure location, while the insensitive information can be processed with plaintext alignment algorithms in a cloud environment to achieve high performance. The next chapter introduces an alignment approach using the LRF and it demonstrates how this approach can be included on the genomic data analysis workflow, providing a the same time an evaluation of its impact in the workflow, regarding the performance.

Chapter 5

Alignment of masked reads

After the sequencing step, genomic data is analysed to disclose its biological content. Due to the high throughput of sequencing machines, this analysis demands high performance. As a consequence, computing environments that provide powerful resources for a low price, such as public clouds, are commonly used by researchers. However, the use of these environments raise privacy concerns [RC11; Ayd+13]. Considering the high performance need, the resources provided by cloud environments, and the privacy protection method described in Chapter 4, which targets the protection of raw genomic data, we identified the following questions: "How does the filtering impact the analysis workflow?", and "How to maintain a high performance and enforce privacy during the genomic data analysis in cloud-based environments?".

This chapter addresses the challenge of developing a practical privacy-preserving high-performance alignment approach. The alignment of reads is the first step of the sequenced data analysis workflow and it identifies the original location of the sequenced fragments in the genome. The proposed approach, MaskAl, takes as input the output masked reads – reads from which the sensitive information have been excised – produced by the (LRF) filtering method previously described (see chapter 4). MaskAl relies on Intel Software Guard eXtensions (SGX) enclaves to enforce privacy of genomic reads and on public clouds to achieve high performance. The proposed approach can be described in two main steps: (i) masked reads alignment, and (ii) alignment score refinement. The first step consists on the alignment of masked reads, i.e, reads whose sensitive information have been excised, relying on public clouds since this step is the most computationally intensive. The second step uses the complete reads (including their sensitive information), consequently, it is performed using the Smith-Waterman algorithm inside an SGX enclave (trusted environment). Both steps can be parallelized in the corresponding

environments. Besides, MaskAl can be defined as practical and efficient, since it executes its most intensive tasks in public clouds, and it is faster and requires less resources than state-of-the-art privacy-preserving methods. Furthermore, MaskAl shows that excising sensitive nucleotides from raw reads is compatible with the traditional analysis workflow steps, with adaptations that we describe later in this chapter.

Regarding the privacy threats considered in this chapter, MaskAl's goal is to prevent trail attacks, a kind of re-identification attack, which combines identifiable features from an individual's genome collected from different studies or institutions to match the genome to its donor [MS04].

Finally, the performance and accuracy evaluation compares MaskAl with state-of-the-art privacy-preserving and plaintext alignment algorithms. We selected one representative for each of the alignment algorithms classes (plaintext and privacy-preserving) for performance comparison. This comparison included the following metrics to support our high performance claim: computational time, memory consumption, and network communications. The filtering step allows the separation of sensitive and insensitive information from raw reads. Then, sensitive and insensitive information can be processed using different environments targeting the best performance, while addressing their distinguished privacy requirements. MaskAl improves privacy since it excises and protects the sensitive information in the aligned reads, by the initial excision step and by processing it inside SGX enclaves.

5.1 Enclaves

A trusted execution environment (TEE) is a secure area created inside a main processor, which isolates the code and data transferred to the TEE. Therefore, using a TEE guarantees confidentiality and integrity of the processed data. Some examples of TEE include SGX (developed by Intel) and TrustZone (developed by ARM). The work in this chapter relies on an Intel SGX enclave. However, the methods that we run in this environment could be implemented in another TEE solution.

Figure 5.1 shows a simplistic representation of an enclave. Briefly, an enclave isolates a section of the processor. Therefore, it can be used to create a secure environment inside an untrusted environment (e.g., public cloud). In general, an enclave communicates with a storage location (DB) and a service provider. Besides, the function(s) to be run and the data to be processed are transferred to the enclave. The dynamics of an enclave processing can be described in the

following steps: (i) an untrusted application initializes an enclave; (ii) the enclave and the service provider communicate for attestation and after verification, it is granted the access to the secure data; and (iii) data is securely transferred to the enclave through a secure channel.

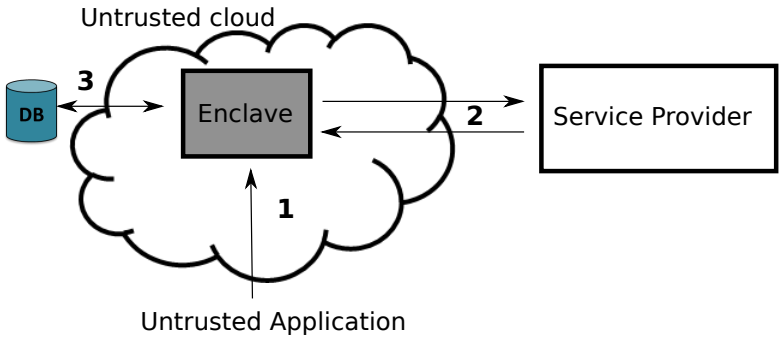

Figure 5.1: **SGX overview**. 1 – Untrusted application initializes the enclave. 2 – Attestation step. 3 – Data encryption and transfer to the enclave through a secure channel.

Regarding the privacy guarantees of TEE, some attacks have been described in the literature (e.g., side channel attacks) [Sch+17; Bra+17; Göt+17]. Some examples of attacks performed on enclaves include prime + probe attacks, and spectre attacks. **prime+probe attacks** allow the collection of privacy sensitive information from the cache. **Spectre attacks** explore a vulnerability of modern microprocessors, which leads to private information leakage. Such attacks allow an adversary to obtain confidential information from the cache. However, solutions to mitigate those side channel attacks have already been proposed and they can be implemented by the enclave developers. Considering this, the work in this chapter assumes that the TEE used are secure.

To conclude, in the context of biomedical data processing, several approaches have been proposed based on SGX enclaves to protect data privacy. Those approaches focus on genomic variants search, genetic testing and rare disease studies [Che+17b; Che+17a; Man+18].

5.2 Methods

Addressing the need for privacy-preserving high-performance alignment algorithms, this chapter proposes MaskAl, a privacy-preserving alignment based on reads filtering, which relies on a SGX enclave for the privacy sensitive information processing. The alignment is the first step of the genomic data workflow which consists in

finding the location in the genome of each sequenced read. Without this step it is not possible to discover the genomic variations present in the sequenced genome, neither to perform further studies, such as discover the links between health conditions and genotypes.

To achieve high performance scientists use plaintext alignment algorithms and rely on public clouds, due to their computational power benefits for a low cost. However, human genome is particularly privacy critical since it is unique for each individual. Therefore, processing and storing genomic data in plaintext in public clouds can leak private information, such as genomic variations. The inadequate use of public clouds for biomedical data processing can harm the privacy of the data donor [Akg+15; Ton+14; FEJ15].

MaskAl first relies on the filtering approach described in Chapter 4 to classify the reads information as sensitive or insensitive. The main idea of MaskAl is to use masked reads – reads from which the sensitive information have been excised – to perform privacy-preserving alignment relying in public and enclave clouds to achieve high performance while ensuring privacy.

Figure 5.2 represents the MaskAl overview, which main steps are described in the following paragraphs.

Step 0: Reads filtering and masking. Anonymized reads provided by the sequencing center are filtered and their sensitive content is excised using the reads filter described in Chapter 4. The filtering process consists in the classification of the nucleotides in the reads either as sensitive or insensitive. At the end of the filtering, two distinct files are created, one containing only the sensitive nucleotides, which we call sensitive reads, and the other with the insensitive nucleotides, which we call masked reads. In the file containing the masked reads the length of the removed sensitive information is hidden, preventing the prediction of that information. This first step is performed in a secure environment, either in the sequencing machine or in a private cloud after sequencing.

Step 1: Masked reads alignment. The sensitive reads are stored securely in the private cloud. In order to analyse the sequenced data, the masked reads are transferred to the public cloud using cryptography-based method to ensure data integrity. The masked reads alignment is performed in the public cloud. In the end of the alignment, a single file with the masked reads reads and their corresponding positions in the genome is transferred from the public cloud to the biocenter. To clarify, in this step the data transfers are limited to a single file, i.e., one input file (FASTA or FASTQ) and one output file (SAM format).

Step 2: Alignment score refinement. This step is only performed if the masked read alignment fails to find the position in the genome for the masked reads. Therefore, step 2 is performed when the masked reads cannot be aligned

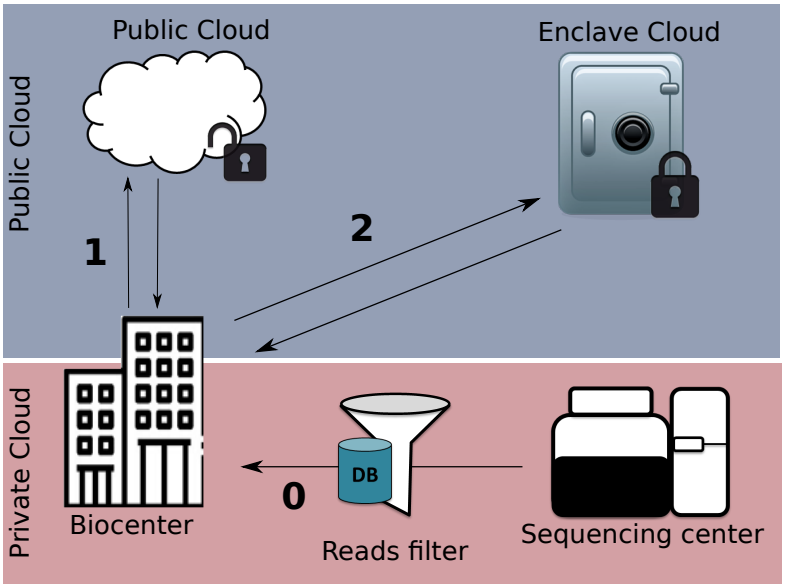

Figure 5.2: **MaskAl overview**. 0 – Raw reads filtering and sensitive nucleotides excising. 1 – Masked reads alignment in a public cloud and results transmission back to the biocenter. 2 – Transmission of the encrypted reads to the enclave cloud and refinement of the alignment score.

or are aligned with an insufficient alignment score. In this case, the biocenter encrypts and sends the full reads (plaintext) and their corresponding candidate positions, which were obtained in step 1, to the enclave cloud. Then, the validation and alignment score refinement are performed in the enclave using the full reads, since the enclave is considered a trusted environment. In addition, the alignment refinement also helps fixing incorrect candidate positions resulting from the masked reads alignment. Such incorrect alignment can occur due to two main reasons: (i) too many sequencing errors in the reads; and (ii) too many sensitive nucleotides are excised in the masked reads. In the end of the refinement step, the enclave sends back to the biocenter the read ID, refined position and refined alignment score. We describe in detail each MaskAl step in the following sections.

5.2.1 Reads filtering and masking

This section summarizes the main aspects of the reads filtering and sensitive nucleotides excising step, which is performed using the long reads filter (LRF) method proposed in Chapter 4.

The filtering and sensitive nucleotides excising method is used in MaskAl to obtain in separate output files the sensitive and insensitive information contained

in raw reads. The filter receives reads as input and classifies their nucleotides as sensitive or insensitive, producing two distinct output files, one with the sensitive nucleotides and the other containing masked reads – reads from which the sensitive nucleotides have been excised. This filtering process can be divided in three main steps: (i) dictionaries creation; (ii) Bloom filters initialization; and (iii) reads filtering and sensitive nucleotides excising.

Dictionaries creation: The dictionaries of sensitive information are created based on the genomic variations reported in the 1000 Genomes Project, by reconstructing the sequences in which they are inserted. Such sequences are obtained by inserting each known genomic variation in the corresponding position in the human reference genome sequence. Since individuals have multiple genomic variations, all the possible combinations are considered. The dictionaries are then built using sequences of K nucleotides from the reference with the combinations of variations. A dictionary d_i represents the dictionary containing the sequences of K nucleotides where the i^{th} nucleotide is sensitive. Depending of the selected position i, some nucleotides at the extremity of the reads cannot be detected during the filtering. In this case, those nucleotides are automatically classified as sensitive to prevent sensitive information leakage. However, such effect is mitigated with the use of several dictionaries with different sensitive nucleotide position (i).

Bloom filters initialization: After the creation of the dictionaries of sensitive sequences, the LRF initializes one or multiple Bloom filters (usually one per dictionary created). The number of Bloom filters is a parameter decided by the user. However, the initialization of several Bloom filters is useful for sequencing error detection in the reads.

Reads filtering and sensitive nucleotides excision: Once the Bloom filters are ready, the filtering and sensitive nucleotides excision can be performed. The filtering process consists on a reading sliding window of size K, starting at the beginning of a read and reading until its end. When a window matches with a sequence in the Bloom filter initialized with the dictionary d_i, the i^{th} nucleotide of that window is classified as sensitive in the read. For each match to a sequence in the Bloom filter only one nucleotide is classified as sensitive.

In the end of the filtering, the file with the masked reads can be outsourced to public clouds without privacy concerns, while the file with the sensitive information is stored in a private and secure environment, such as a private cloud, and it is encrypted before any data transfer in order to prevent leaks.

5.2.2 Masked reads alignment

The filtering step produces a file with the input reads from which the sensitive nucleotides were removed – masked reads– which do not contain any privacy concerning information, since it is the common information shared by all the human individuals. The sensitive information is also an output of the filtering process, however, it is returned in a different file.

The two complementary parts of the original reads can be processed in different environments, accordingly to the sensitivity of the information they carry. In the plaintext reads – reads containing only the insensitive nucleotides – file, the consecutive sensitive nucleotides are replaced by a single 'N'. This replacing methodology prevents sensitive information leakage based on the length of the excised sequence. Similarly, in the sensitive file, the insensitive nucleotides are replaced by '-'. In any phase of the masked alignment process MaskAl is able to recover the full reads, if it is required.

5.2.2.1 Plaintext alignment algorithm selection

Nowadays, there is a wide range of alignment algorithms, which can be classified in three main classes (plaintext, hybrid and cryptography-based) depending on the methods they use. For the evaluation presented in this chapter, we selected one representative algorithm per category. For alignment algorithms, high performance is required since they are expected to align a great number of reads to determine their position in the genome. Considering the high performance requirement, the BWA and LAST algorithms were compared (computation time and accuracy) in order to select the plaintext representative algorithm.

Figure 5.3 summarizes the alignment time comparison between BWA and LAST algorithms. In general, BWA is from 5× to 7× faster than LAST. For example, aligning a read of 150 nucleotides with an error rate of 1% takes 14.83 seconds with BWA, while LAST requires 71 seconds. The alignment time of BWA slightly increases with the length of the reads and with their error rate. For LAST, the alignment time also increases with the length of the reads, however, it slightly decreases with the increase of the error rate.

Figure 5.4 compares the alignment accuracy of the BWA and LAST algorithms for masked and plaintext reads. BWA (green bars) is able to align both masked and plaintext reads with an accuracy above 95.7%. For LAST, the accuracy achieved for masked and plaintext reads is considerably different. In addition, for masked reads the accuracy also varies with the length of the reads. For plaintext reads LAST is able to achieve accuracy above 95.6% (the same as BWA), while for

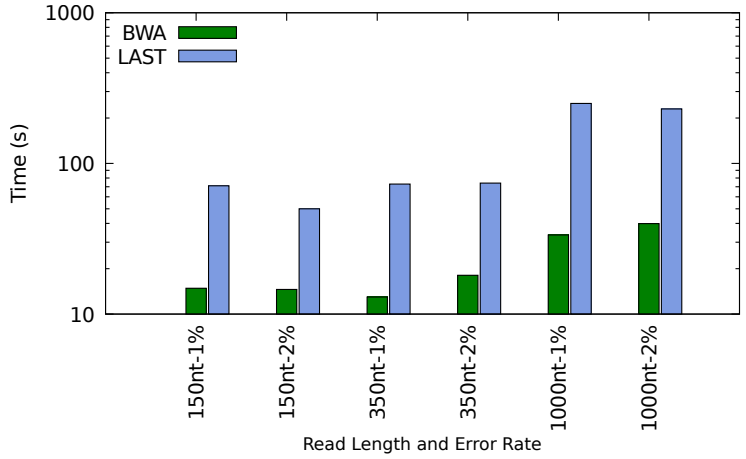

Figure 5.3: **Alignment time – BWA vs LAST**.

masked reads the accuracy varies between 82.7% and 92.2% depending on the reads length.

To conclude, we selected BWA as the representative algorithm for the plain-text alignment class since it showed a faster alignment and a higher accuracy (in particular for masked reads) compared to LAST.

5.2.3 Alignment score refinement

This section describes the last step of MaskAl approach, which includes the alignment score refinement and the adapted implementation of the Smith-Waterman (SW) algorithm run in the enclave. Due to the limited amount of memory available inside the enclave, the original SW algorithms needed some changes.

5.2.3.1 Score refinement and alignment extension

After the masked reads alignment, MaskAl determines the most accurate position among the set of candidate positions. This process is made by aligning the read to each candidate position and compare their similarity. The computational overhead of this step is reasonable, since this second alignment is only performed against the neighbouring regions of the candidate positions. It works similarly to the extension step of seed-and-extend algorithms, and thus can replace it in these algorithms. To prevent attacks on correlated encrypted messages, the masked alignment and the extension steps are performed in different machines.

The extension uses the full read, and, therefore, it is performed in a secure environment, e.g., an Intel SGX enclave.

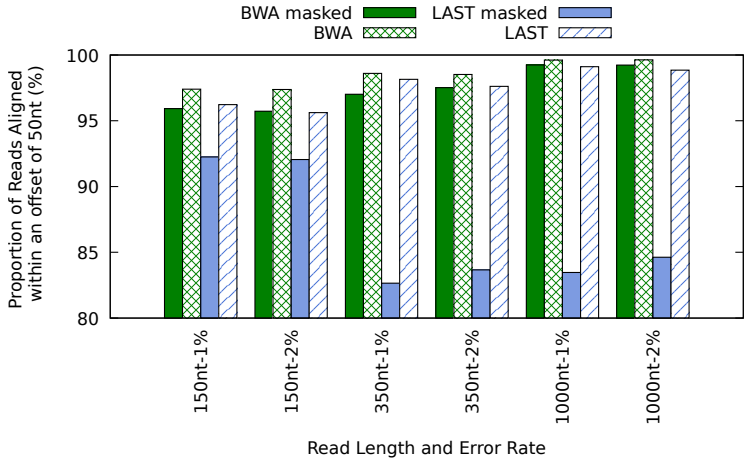

Figure 5.4: **Alignment accuracy – BWA vs LAST**.

5.2.3.2 Smith-Waterman algorithm

The Smith-Waterman (SW) algorithm [SW81; Got82] performs local alignment for two sequences. This means that the two sequences are compared character by character. In addition, the matches, mismatches, insertions and deletions are weighted. In general, a local alignment algorithm computes the longest section shared by the two compared sequences.

The local alignment performed by SW uses a score matrix M (Table 5.1 left matrix), where the first row and the first column contain the two compared sequences, respectively, the reference sequence and the read to be aligned. In the beginning of each sequence an empty character is added (represented as ε). Then, the matrix is initialized with 0 in each cell. In parallel, a traceback matrix T (Table 5.1 right matrix) is used and helps to define the sequence alignment corresponding to the score given by M. This matrix T is also initialized with 0 in all cells. The score produced by the SW algorithm considers five possible edit distance score cases: match, mismatch, insertion, deletion and 0 (particular of SW algorithm). The value of each case is computed accordingly to the function in equation 5.1. For the local alignment score computations, MaskAl attributes the value +2 to matches and the value −1 to mismatches, insertions and deletions.

The local alignment is defined in the traceback matrix, which contains the alignment directions. There are three possible directions: diagonal (D), left (L), and up (U). If the $M(i, j)$ is the cell of the matrix M selected due to a match or mismatch case accordingly to the scoring function $Score_{i,j}$, then $T(i, j)$ will contain a D (diagonal direction). This direction is called diagonal since it is computed using the value of $M(i-1, j-1)$, which is the precedent cell in its upper right diagonal.

Table 5.1: Smith-Waterman Score Matrix (left) and Traceback Matrix (right). The weights used are: match/mismatch 2/-1 and gap penalty: -1. The alignment result according to both tables is "ACGT" and "A-GT"

	ε	A	C	G	T	A
ε	0	0	0	0	0	0
A	0	2	1	0	0	2
G	0	0	0	3	2	1
T	0	0	0	2	4	3
C	0	0	2	1	3	3

	ε	A	C	G	T	A
ε	0	0	0	0	0	0
A	0	D	L	0	0	0
G	0	0	0	D	0	0
T	0	0	0	0	D	0
C	0	0	0	0	0	0

Following the same principle, an insertion corresponds to U and a deletion to L, respectively, insertion uses the upper neighbour value and deletion uses the left neighbour value for the score computation (see equation 5.1). The alignment corresponding to the highest score can be found following the directions in the traceback matrix T, starting from the cell with highest score in M. Then, the directions are followed towards the left upper cell and it terminates when a cell populated with a 0 is found.

Table 5.1 represents an example of a score matrix and corresponding traceback matrix. In the end the alignment has a score equal to 4 and the the longest alignment matches the sequences "ACGT" (from the reference) and "A-GT" (read aligned).

$$Score_{i,j} = max \begin{cases} 0 & \\ Score_{i-1,j-1} + 2 & : u = v \\ Score_{i-1,j-1} + (-1) & : u \neq v \\ Score_{i,j-1} + (-1) & : insertion \\ Score_{i.1,j} + (-1) & : deletion \end{cases} \tag{5.1}$$

Equation 5.1 represents the scoring function for the SW algorithm, where u and v represent the characters of the two sequences that are compared.

5.2.3.3 Running SW algorithm in SGX enclaves

Traditionally, SW algorithm aligns a read to the full reference genome, which is a 3 Gigabases long sequence. Due to the limited memory of the enclave, instead of the full reference genome, the adequate subsection of the reference genome is sent to the enclave. The selection of the adequate subsection is made based on the candidate positions obtained in the masked alignment. Using BWA for the masked alignment, one can expect an alignment result within a range of 50 nucleotides (nts)

around the origin of a read for 96% of the reads aligned (see Figure 5.4). Therefore, the section from the reference genome used for the alignment inside the enclave corresponds to 50 nts before the candidate position plus the length of the aligned read and extra 50 nts.

Due to the small memory limitation of enclaves – 128 MB from which 90 MB can be used for computations –, the memory is quickly used by the SW algorithm. In order to circumvent the memory limitation, the reads to be aligned are limited to a length of 1000 nts. The two reasons that support this threshold length are the following: (i) 1000 nts reads easily fit in an enclaves and it allows the parallelization of two threads; and (ii) there are a great amount of reads with a length smaller than 1000 nts, produced by the majority of the sequencing technologies, except by the third generation sequencers. As TEE technologies are still evolving, in the future the reads length limit may disappear with a memory extension of enclaves.

Regarding the network communications, encrypted overlapping chunks of the reference genome with 2000 nts are kept in each enclave to avoid unnecessary data transfers. The use of overlapping chunks of the reference genome avoids extra encryption and decryption steps on the enclave, since it prevents that a read aligns on two chunks. This validation step runs in the enclave and uses an hashmap to select the adequate encrypted chunk of the reference genome. Inside the processor's reserved memory of the enclave, the reference chunk and full read are decrypted. Then, the SW algorithm is executed, followed by the the score refinement and in the end the encrypted result is transmitted to the biocenter.

5.3 Evaluation Setup

5.3.1 System model

Our system model makes some assumptions that we describe in this section. MaskAl was developed assuming a biocenter that processes sequenced reads and has limited computer resources. As a consequence, the biocenter relies in public, private and enclave clouds to provide privacy protection to the sensitive information contained in the reads while achieving high performance. *Public clouds* are server farms available at an affordable price, which provide computational resources that are usually managed by outsourced third-parties (e.g., Amazon AWS, Google Cloud). Although, public clouds allow high performance, they do not prevent the access to data by the cloud service provider or an intruder [RC11]. Moreover, several works demonstrated the harm associated to the inadequate use of clouds for biomedical data procession [MPG14; Ton+14; FEJ15; YKÖ17]. *Private*

clouds have the power of public clouds, however, the full control of the hardware and software is given to the owner. Due to this, private clouds provide a higher security level than public clouds. The filtering step performed in MaskAl is run in this environment type given that this step is faster and it requires less computational resources than the reads alignment performed in public clouds. Finally, the *enclave clouds* considered in this thesis consist on SGX enclaves running inside public clouds. Weighing the enclave attacks and mitigation methods, the enclaves used in MaskAl are considered secure. Since MaskAl uses SGX enclaves implemented in public clouds the data transfers to and from the enclave need to be encrypted. In addition, it is also assumed that one hour of computations in the public and enclave clouds have the same cost. The main limitation of enclave clouds is the protected memory available, which is restricted to 128 MB (96 MB usable) per SGX CPU. Due to this memory limitation, it is not possible to run the filtering and reads alignment steps inside the enclave environment.

5.3.2 Threat model

MaskAl assumes a honest-but-curious adversary operating in the cloud that intent to infer private information related to the sequenced data donor(s). In addition, it is assumed that the adversary is able to observe raw reads during the alignment step, which is performed in plaintext. Then, the adversary has the knowledge to align the raw reads and discover the genomic variations on those reads. The adversary then has enough information to perform privacy attacks, for example, trail attacks. A trail attack consists in combining personal identifying traces of an individual's genome from different studies or institutions to relate the genome to its donor [Mal02; MS04]. This scenario occurs, for example, if an individual donates his genome to a study for a particular condition, and years later the same individual participates in other study.

5.3.3 Hardware

The experiments executed for the performance evaluation of MaskAl use three different environments, corresponding to three cloud environments (public, private and enclave). The private cloud is assumed to have a quad socket Intel Xeon E5-4650 v3 processor with 12 cores running at 2.10 HGz. Regarding the memory, 190 GB of RAM and 15 TB of hard disk are available. The public cloud is represented by a HPC shared equipment with Intel Xeon E7-4850 processors from which 30 threads were used for the experiments. The memory available in this equipment was 1 TB of RAM and 150 GB of hard disk. The Balaur experiments were run in

this settings, due to the memory incapacity of the available local resources. Finally, a desktop computer was used to mimic the enclave cloud environment. The used computer was equipped with Intel i7-6700 processors and 16 GB of RAM. For the communications between the different cloud environments, it was assumed a common bandwidth of 1 Gbit/s.

5.3.4 Software

MaskAl was implemented in C/C++. For the enclave cloud implementation, MaskAl uses the official Intel SGX SDK in version 1.9 and the code was run on Ubuntu (16.04 LTS). The secure Smith-Waterman implementation uses a ECALL which receives the encrypted reads, applies decryption inside the processor reserved memory (PRM) and then runs the Smith-Waterman algorithm in the plaintext reads inside the enclave cloud.

The alignment is a process that can be easily parallelized since the reads can be aligned independently. Consequently, MaskAl is assumed to scale linearly with the number of enclave clouds used. In the present analysis, the proposed implementation is evaluated considering a single thread.

5.3.5 Enclave initialization and encrypted communications

All the data used by the Smith-Waterman alignment algorithm inside the enclave is transferred encrypted and stored on the enclave in order to prevent inference attacks. In particular, MaskAl uses AES128-GSM since it is integrated in the SGX SDK to encrypt the plaintext reads and reference chunks – the reference genome is transmitted in chunks since it is too big to fit all at once inside the enclave. Furthermore, the communications between the different clouds are also encrypted, using RSA2048, and it is assumed that the adversary is not able to decrypt the communications.

5.3.6 Simulated reads

The evaluation experiments were run using simulated reads, which were created using wgsim [Li11] and the GRCh38 version of the human reference genome. The reads simulation assumes reads the length and error rate of existing sequencing technologies. Nowadays, the different generations of sequencing machines are able to produce short (up to 400 nucleotides [Jün+13]) and long reads ($>$ 1000 nucleotides). Therefore, reads of 150 and 350 nucleotides were created to represent the short reads and reads of 1000 nucleotides are the representative of the long

reads category. For the error rate, the reads were simulated with error rates of 1% and 2%. Finally, each reads category was composed by 100000 reads from the whole human genome.

Currently, long reads present a high error rate (around 11-14%) [Pol+18], however, it is expected that such error decreases with the optimization of the sequencing techniques over time or through the development of error correction methods. For this reason, the error rates of 1% and 2% were assumed for all the simulated reads (short and long).

5.3.7 Algorithms evaluation criteria

The main goal of MaskAl is to improve privacy in the alignment step without significant changes in its accuracy. Therefore, the evaluation of the proposed method includes the measurement and comparison with state-of-the-art alignment algorithms, regarding the following parameters: memory consumption, network communication volume, and computational overhead. The results of each parameter evaluation are described in section 5.4. Aligning algorithms with a high performance are more likely to be accepted by the research and biomedical community, since big data processing is one of the main challenges of sequenced data analysis [Sae+12].

5.3.7.1 Alignment algorithms selection

There are three main classes of alignment algorithms, as already mentioned in Chapter 3: plaintext algorithms, cryptographic algorithms and intermediate protection algorithms. Plaintext algorithms are the faster, however, they do not provide any privacy protection. The cryptographic algorithms, on the other hand are located at the other extreme of the spectrum, as they are the most secure but also slowest algorithms. Finally the intermediate protection algorithms provide some privacy guarantees, which have not been proven secure, and they have a practical computation time.

In order to compare our solution, MaskAl, with previous work, one representative of each algorithms class was selected. In practice, the comparison was only performed against plaintext and intermediate protection algorithms. The cryptographic algorithms class is mainly represented by solutions based on garbled circuits [Bar+12; Hua+11] and homomorphic encryption [AKD03; CKL15]. Such algorithms have a low throughput, which make them unpractical, and for that reason they are not included in the comparison with MaskAl.

Regarding the plaintext alignment algorithms, BWA and LAST were compared and selected based on their accuracy and alignment time, as described in section 5.2.2.1. BWA [LD09] is a plaintext algorithm that performs alignment using suffix tree. LAST [Kie+11] is a plaintext alignment algorithm based on hashmap. Due to its higher accuracy with masked reads and the short time used in the alignment, BWA was selected to represent the plaintext alignment algorithms category.

Regarding the algorithms that provide an intermediate protection level, MaskAl was compared to Balaur [PB17], a recent privacy-preserving alignment algorithm that relies on hybrid clouds using locality sensitive hashing and K-mer voting. In addition, this solution also uses data encryption to ensure privacy protection during data transfers to the public cloud.

5.4 Results

5.4.1 Masked reads alignment time

The performance evaluation included the comparison of the time each algorithm takes to perform the alignment of reads of different sizes and error rates. Figure 5.5 summarizes the alignment time of three algorithms: BWA, Balaur, and the proposed approach - MaskAl. The results shown in the figure shows that BWA is the fastest algorithm, in average 58% faster than MaskAl. In comparison with Balaur, the representative of existing privacy-preserving alignment algorithms, MaskAl is in average 87% faster.

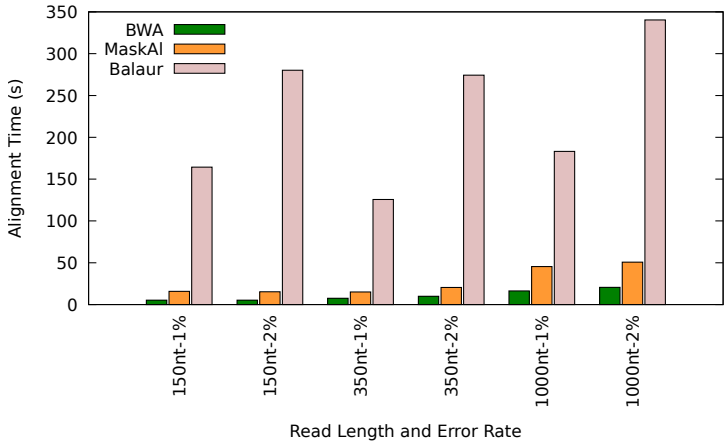

Figure 5.5: **Alignment time comparison**. We compare the alignment time of BWA, Balaur, and MaskAl.

5.4.2 Score refinement time

The score refinement is the step after the masked alignment, run in the enclave, that determines from the candidate positions which one is the most accurate. This step can be divided in some parts: (i) encryption and decryption of the reads; (ii) decryption of the reference chunk; (iii) adaptation of the reference chunk size accordingly to the reads size; (iv) run the Smith-Waterman (SW) algorithm; and (v) encryption and transmission of the result back to the biocenter. Considering that the complexity of SW algorithm is cubic, the workload needs to be reduces in order to run inside the enclave. Therefore, the usual SW table size of $readLenght \times referenceLenght$ was limited to $readLength \times (readLength + 100)$. Figure 5.6 compares the score refinement time for reads of 150 and 350 nucleotides. As the results in the figure show, the score refinement time increases linearly with the increase of either the read size or the reference chunk size.

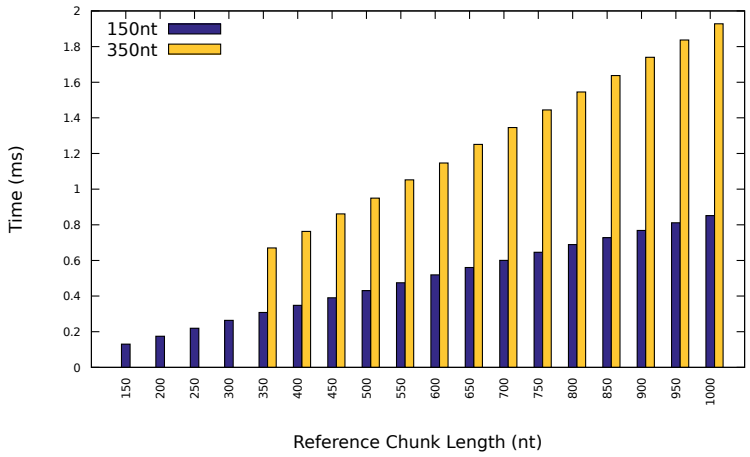

Figure 5.6: **Score refinement time comparison**. Score refinement time for reads of 150 and 350 nucleotides.

5.4.3 Computation time per MaskAl step

In order to access the most time critical steps of MaskAl approach, the alignment time per step and the cumulative time was computed for reads of 150, 350 and 1000 nucleotides, with error rates of 1% and 2%.

Figure 5.7 summarizes the computational time required per MaskAl step. The data transferring time is computed assuming a 1 Gbit/s network bandwidth and a enclave cloud composed by 96 SGX enclaves. The most time consuming step is the alignment of masked reads (blue section) independently of the reads length

and error rate. For longer reads (1000 nucleotides) the alignment time is higher than the time required to align shorter reads (150 and 350 nucleotides). Following the same tendency, the secure alignment score refinement time (purple section) and filtering time (yellow section) also increase with the reads length. Finally, for all cases, the network costs and cutting time are negligible in comparison with the times the other three referred steps.

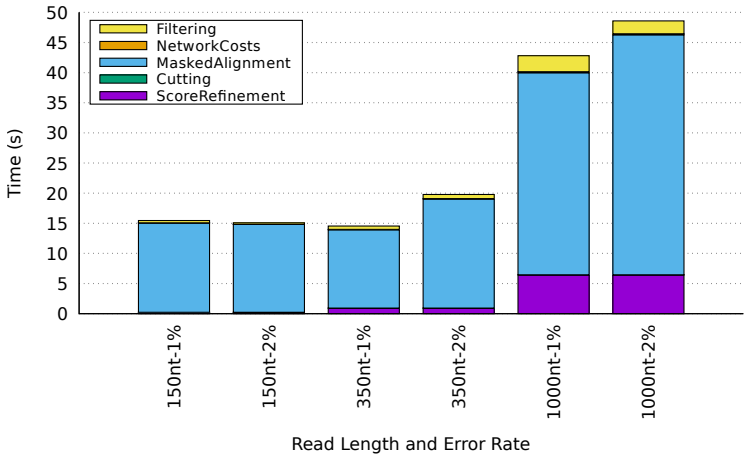

Figure 5.7: **Computation times per step of MaskAl**. Computational time of each MaskAl step: reads filtering, network communication times, masked reads alignment, cutting, and score refinement.

Regarding the computation time, it is important to discuss the case where the masked reads alignment fails. First, the filtering approach classifies the content of the reads as sensitive or insensitive. From the filtering approach evaluation (Chapter 4), 10% of an individual's genome is classified sensitive. Considering a similar percentage of sensitive reads, 90% of the reads are insensitive (masked reads) and they can be aligned in the public cloud environment. From those 90% aligned reads, around 4% can fail (see Figure 5.4, BWA alignment accuracy), for example due to the excision of long regions of the reads, and need to be re-aligned in the enclave using the full read. Therefore, 3.60% ($90\% \times 4\%$) of the reads aligned need a second round of alignment, however, since this percentage is small, is is not expected a significant increase in the computation time due to this process.

5.4.4 Memory consumption

BWA, LAST and Balaur alignment algorithms were compared regarding the peak memory consumption. Understanding precisely how much memory an alignment

algorithm requires helps choosing the adequate characteristics of the public cloud. Therefore, the memory consumption is an important property, allowing the optimization of cost and resources used, in this particular case, for the alignment. Figure 5.8 presents the results of the memory consumption evaluation for the three compared alignment algorithms. Balaur (red bars) requires between 67 GB and 153 GB of RAM during the alignment, LAST (blue bars) requires between 16 GB and 23 GB, while BWA (green bars) requires less than 6 GB, which is considerably less memory.

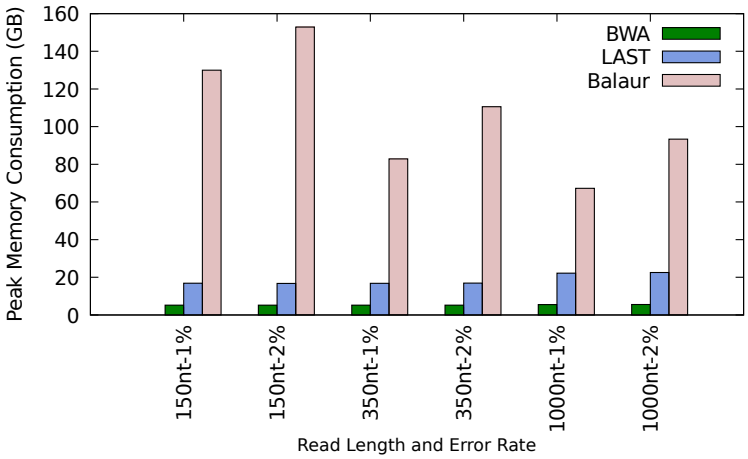

Figure 5.8: **Memory usage comparison**. We compare BWA, LAST and Balaur memory requirements.

The presented peak memory comparison do not include the alignment score refinement, since it is performed in the enclave, and, therefore, it is limited to 128 MB. This bound corresponds to the Processor Reserved Memory the Intel SGX enclave allows, without artificial extension, which can be done using a SWAP-like technique.

Finally, comparing Balaur and MaskAl, both requires some pre-processing memory, which is not represented in Figure 5.8. The alignment performed by Balaur requires the reference genome indexing, and this step depends on the length of the reads to be aligned. For reads between 150 and 1000 nucleotides, Balaur requires between 62 GB and more than 189 GB, with longer reads requiring less memory. Contrarily, MaskAl only requires tens of GBs of memory during the filtering and sensitive nucleotides excising process.

5.4.5 Network communications

The sensitive nucleotides excising step transforms the plaintext reads hiding the sensitive nucleotides they contain. This transformation reduces the size of the reads – masked reads are smaller. Therefore, the transmission costs of masked reads are smaller when compared to the transmission cost of plaintext reads.

MaskAl has two network transfers, since it relies in different cloud systems (public and enclave clouds). The first data transfer considers the transference of masked reads to the public cloud where they are aligned and the transference of the SAM-files with the aligned masked reads. The second data transfer refers to the transference of encrypted plaintext reads and their aligned positions to the enclave cloud in order to perform the score refinement and then the transference of the encrypted results back to the biocenter.

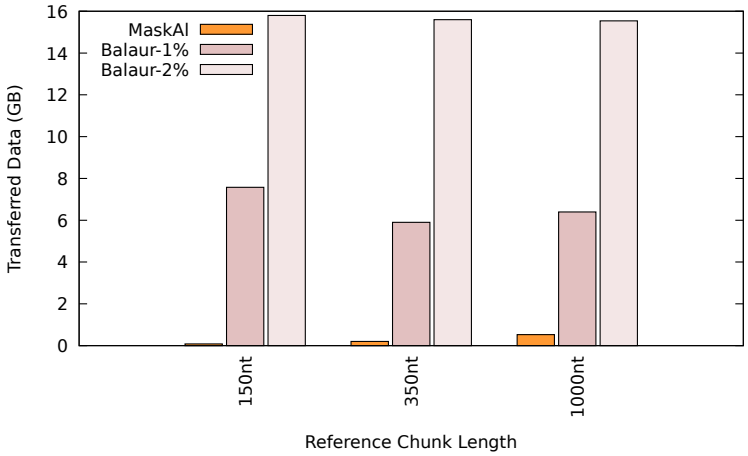

Figure 5.9: **Network communications comparison**.

MaskAl requires significantly less networking resources than Balaur. Balaur uses keyed-hashes of the reads, which have a constant length in order to ensure data security. Consequently, Balaur has nearly constant data transfer when compared different reads lengths. However, Balaur requires between 5.7 GB and 15 GB more data transfers than MaskAl. Figure 5.9 compares the communication costs per 100,000 reads of different lengths (150, 350 and 1000 nucleotides) using Balaur and MaskAl. Two error rates were considered for the alignment with Balaur, 1% and 2%. Regarding the reads length, with the increase of the reads length, bigger are the data transfers. Regarding the error rate comparison, for reads of 150 nucleotides with an error rate of 1% Balaur requires around 7.5 GB of transferred data, while for 2% error rate it requires almost the double of the data transfers (16 GB). In the figure is represented only one value for MaskAl while there are two

values are presented for Balaur, considering two different error rates (1% and 2%). For MaskAl, the network communications are not affected by the reads error rate since the same data transfers are required for reads with, for example, an error rate of 1% and 2%. In this case, the error rate only interfere in the alignment accuracy.

5.5 Summary

MaskAl provides privacy-aware alignment based on the reads partitioning accordingly to their sensitivity – sensitive or insensitive. In order to provide high performance, MaskAl relies on public clouds which can also run enclave clouds. To ensure the adequate privacy, the insensitive reads, i.e., reads containing only information shared by the human population, are aligned on the public cloud environment, while the sensitive information is processed inside enclave clouds. The partitioning step is performed in a private cloud environment since the input of this step are the plaintext reads from the sequencing machines. The main gain on the performance and privacy ratio is due to the partitioning of data and subsequent privacy-preserving algorithm adaptation to the data sensibility. This gain is only evident since the majority of the human genome (around 99.5%) is common among individuals, which allows the outsourcing of alignment of the majority of an individuals reads – 90% of the reads classified as insensitive by the filter approach used – to the public cloud without privacy breaches while providing high performance.

Evaluating the performance of MaskAl, the excision of long regions from the reads can lead to some failed alignment, those reads need to be re-aligned in the enclave environment using the full read. As discussed in this chapter, the percentage of re-aligned reads is small, 3.60%, and consequently it is not expected a significant increase of the computational time. In average, the MaskAl is 87% faster than most recent state-of-the-art privacy-preserving alignment algorithm (Balaur), due to its reduced communication overhead. Regarding the RAM usage, MaskAl requires in average 95% less RAM during data processing. In relation to the network communications, MaskAl requires between 5.7 GB and 15 GB less data transfers than Balaur. These results show how MaskAl provides privacy-preserving alignment and how it scales using public clouds, which allow the achievement of the high throughput imposed by the existing and emerging sequencing technologies.

Chapter 6

Conclusions and future work

6.1 Conclusion

This thesis contributes to the development of privacy-preserving mechanisms for the processing of early genomic data with high performance. More specifically, this thesis presents: (i) methods to define sensitivity levels for raw genomic data and adjust the level of protection per sensitivity level, which we demonstrated with the alignment of reads entirely classified in a sensitivity level (DNA-SeAl); (ii) a filtering method (LRF) that classifies the nucleotides contained in raw genomic reads according to their sensitivity, which allows the genomic data to be partitioned in sensitivity levels and processed according to its sensitivity; and (iii) a privacy-preserving alignment algorithm (MaskAl) that leverages public clouds and trusted execution environments.

The first contribution, DNA-SeAl, focused on the definition of a finer-grained classification of genomic data into sensitivity levels based on its quantitative features, in particular allele frequencies and statistical relationships between genomic variations. The example we presented detailed three sensitivity levels and showed that the implementation of a finer classification allows the user to take advantage of the diversity of algorithms, enforcing privacy protection while achieving high performance. DNA-SeAl improves the state of the art by extending the classical binary classification (sensitive or insensitive) to multiple sensitivity levels. An example of three sensitivity levels was presented, however, the number of levels is completely customizable by the user and it can be adapted to the algorithms and computational resources available.The example of three sensitivity levels presented shows that for short reads (i.e., 100 nucleotides), 72% have low sensitivity, 23% have intermediate sensitivity, and the remaining 5% are highly sensitive. Considering this distribution, the performance comparison with previous approaches,

using different public/private clouds settings, showed that computation time and communications are reduced when adopting the three proposed sensitivity levels. To give an example, DNA-SeAl is 5.85× faster than the approach using the binary classification of reads and it requires 5.85× less data transfers, when using a proportion of public/private clouds of 10/1. Finally, regarding the privacy improvement, the results support the use of sensitivity levels to reduce the membership detection risks by splitting and protecting adequately each sensitivity level. Splitting into two parts and up to five parts the sensitivity levels, we showed that membership attacks could be prevented.

The second contribution, LRF, proposed a long reads filtering method for raw human DNA reads that relies on Bloom filters. The main goal of LRF is to detect the nucleotides of raw reads that participate in genomic variations, called sensitive nucleotides. LRF improves upon the state of the art regarding the detection of sensitive nucleotides in raw reads, independently of their length. In addition, LRF also improves the detection method, reducing significantly the false positives and reducing the false negatives. The previous approach (SRF) did not consider the possibility of having errors in the reads, while LRF is also accurate in the presence of a small error rate. To give an example, with reads generated with an error rate of 2%, 86% of the sensitive nucleotides are correctly detected instead of 56% previously by SRF. Regarding the detection accuracy, LRF missed 10 sensitive nucleotides per genome instead of the 100,000 missed by SRF. In addition, regarding the incorrect classification of insensitive nucleotides (false positives), LRF classifies 10% of an individual's genome as sensitive instead of the previous 60%, while the true sensitive nucleotides percentage is equal to 3%. Regarding the performance, LRF sacrifices some performance in order to achieve better accuracy, however, its performance is still 10× faster than the sequencing machines throughput. The memory used by LRF is larger in comparison to the SRF approach, once again in order to improve accuracy. Finally, to the best of our knowledge, LRF is the first introducing an early protection method for genomic data, which is applied as soon as the data is produced. LRF opens the path for the development of a sensitivity-adapted analysis workflow. Subsequently, it is used as starting point for the third contribution presented in this thesis.

The third contribution of this thesis, MaskAl, describes a excision and alignment methodology for DNA reads using Intel SGX. The main concept behind this contribution is that sensitive and insensitive reads require different privacy protection during alignment. Therefore, sensitive reads are aligned in an enclave (Intel SGX) for high privacy protection and the insensitive reads – referred to as masked reads – are aligned in a public cloud since they do not require particular protection. MaskAl provides a faster and lighter privacy-preserving alignment approach,

where approximately 10 % of the reads (sensitive) are aligned inside an enclave and the remaining 90% (masked reads) are aligned in the public cloud environment. The obtained results show that more than 96% of the masked reads are successfully aligned using the BWA software. For this reason, only a small percentage of the total aligned reads (3.60%) requires a second round of alignment, that is executed in an enclave. Since the proportion of reads is small, one observes a small increase of the computation time due to this second round of alignment. These results support the high alignment accuracy provided by MaskAl. Regarding the performance, MaskAl is 87% faster than existing privacy-preserving alignment algorithms (Balaur). In addition, MaskAl requires 95% less RAM memory and it requires between 5.7 GB and 15 GB less data transfers in comparison with Balaur. These improvements on the memory consumption and computation time over the state of the art algorithms make MaskAl a competitive solution.

Finally, the work presented in this thesis promotes the implementation of a privacy-preserving analysis workflow, which would be the first protection mechanism applied to genomic data, as soon as it is produced. In addition, the compatibility with the traditional analysis workflow, currently used for genomic sequenced data, was tested and demonstrated with the alignment step.

6.2 Future work

The work on this thesis allowed us to identify other challenges that require further study.

In general, a deeper study about the sensitivity of the information in the human genome sequence and the definition of privacy thresholds applicable to genomic reads would be interesting topics. The idea is to provide a better understanding of the risks of information leakage and to promote the development of more secure and privacy-aware systems and algorithms, since it clarifies how difficult it is to protect the privacy of an individual's genomic data. Later, we plan to extend the study from a single genome to sets of genomes, to replicate the settings of practical clinical tests that usually include several individuals.

Related to the LRF method, in future work it would be interesting to explore other approaches to perform the classification task, since in different systems or for other types of data it possibly improves the method. The adaptation of the method for other kinds of biomedical data, or the development of new methods would enrich the analysis environment, since genomic data is usually linked and processed with other kinds of data, for example, metadata and medical data.

The work on this thesis evaluated a novel privacy-preserving genomic data

alignment approach, MaskAl, which focuses on the alignment step, however, the evaluation of further steps of sequenced data analysis workflow, such as variant calling and error correction, would strength its adoption by the research community. After the study of those steps it might be possible to build the full privacy-preserving raw genomic data processing workflow. Once completed this process, our intuition is that such methods can also be integrated on a distributed system model.

Finally, MaskAl claims that the small practical memory made available by an Intel SGX enclave (MB) is not an impediment for the approach it proposes. As an alternative, in future work, exploring alternatives to SGX enclaves could provide new alternatives for the privacy-preserving analysis of genomic data.

Terminology

Concept	Description
Alignment	The process where sequenced reads and a reference genome are compared to determine the original position of the sequenced reads in the genome.
Allele	It is one of two corresponding genes of on a chromosomes pair. For each bi-allelic SNP locus, the possible variants verified in a population are referred to as the major and minor alleles, being represented by 0 and 1, respectively.
Allele frequency	It represents the incidence of a genomic variation in a population. Then allele frequency can be computed by dividing the number of times the allele of interest is observed in a population by the total number of alleles that are observed at that particular location.
Alternative allele	Also called rare allele, it is the allele held by a minority of individuals of a population. The alternative allele is a mutation of the reference allele. In each genomic variant locus, it is possible to have one (bi-allelic case) or more alternative alleles, e.g., population specific variations.
Chromosome	Thread-like structures in which the DNA is packed inside the nucleus of each cell. In particular, the human genome is composed of 23 pairs of chromosomes.
Deoxyribonucleic acid (DNA)	Genetic information carrier which is present in all living creatures.
Gene	The basic physical and functional unit of heredity. A gene encodes the sequence to produce a functional molecule, called protein, which is essential for life.
Genome	The complete set of genetic information in an individual. It contains all the information the individual requires to function. The genome is usually stored in long molecules of DNA called chromosomes.
Genome wide association study (GWAS)	The group of steps researchers run to identify inherited genomic variations linked to some particular disease or trait. In these studies, scientists compare the genomic variations present in case (people with the disease or the studied trait) and control (people without the disease or the studied trait) groups.

Concept	Description
Genotype	The complete unique combination of alleles of an individual. The genome revealed by personal genome sequencing. For each SNP location an individual can have three possible genotypes: (i) homozygote with the major allele (00); (ii) homozygote with the minor allele (11); or (iii) heterozygote (01 or 10). On average, approximately 99.5% of the genotypes of any two humans are identical.
Haplotype	Group of alleles that are inherited together from one parent. It refers to the genetic information contained in one of the two double helices of the DNA.
Indels (insertions and deletions)	They are specific structural variations that can occur in the genome. Insertions define the addition of one or more nucleotides to the genomic sequence, while deletions are the opposite process where one or more nucleotides are deleted from a genomic sequence.
Kinship relations	Relation between individuals within a family. Such relations include ancestors, descendants, siblings and cousins, to name some.
Linkage disequilibrium (LD)	Non-random association of alleles among neighbouring SNP loci in a certain population. Two loci are considered in linkage disequilibrium, if the probability of the association of their different alleles diverges from their random association probability.
Locus (plural is loci)	In genomic context, it denominates a specific place in the genome where some change/mutation can occur.
Masking	In the context of this thesis, masking refers to the removal of one or more nucleotides from a genomic sequence in combination to the replacement of those nucleotides with the undefined base symbol ('N'). This technique is used to hide sensitive information.
Mutation	Also referred to as genomic variation, it defines a change in the reference genome, such as insertion or deletion of nucleotides.

Concept	Description
Nucleotide	This is the basic unit of DNA, which is composed by three components: a sugar molecule, a phosphate group and a nitrogenous base. There are five nitrogenous bases; four of them are present on the DNA: adenine (A), thymine (T), cytosine (C), and guanine (G). The excluded one, uracil (U) is only present on RNA and replaces T. NOTE: Ribonucleic acid (RNA) acts as a messenger of the instructions coded in the DNA which control proteins synthesis.
Phenotype	The observed characteristics of an individual, which result from the combination of the genotype and the environment.
Polymorphism	The occurrence of more than one genetic form in the same population of a certain specie (e.g. human).
Read	It is a subsection of a genomic sequence, which is produced by a sequencing machine.
Reference allele	Allele held by the majority of the individuals in a given population.
Reference genome	Genome representation containing all the most common variations verified in a given population.
Sequencing technology	Method based on chemical reactions that allow the conversion of DNA/RNA molecules into digital sequences of nucleotides. There are different technologies defined by the chemical reaction used, which defines the error rate and reads length.
Single nucleotide polymorphism (SNP)	The smallest polymorphism which is a variation on a single nucleotide (A, T, G, or C) at a specific position in the human genome.
Short tandem repeat (STR)	Also called microsatellite, is a short sequence of DNA, with length between 2 and 6 base pairs, that is repeated a certain number of times in a head-tail manner. The number of repeats can vary between individuals and can be used to uniquely identify an individual.

Concept	Description
Structural variations (SVs)	The general name to refer to the group of modifications in a genomic sequence caused by duplications, insertion, deletions, inversions, and translocations of DNA segments.
Trusted execution environment (TEE)	It is a secure environment for storing data and executing code, which is isolated from the rest of the machine components. Therefore, they can ignore the threats from the other components. Some examples of the available TEE are the Arm TrustZone and the Intel SGX.

References

[AH17] E. Ayday and M. Humbert. "Inference Attacks against Kin Genomic Privacy". In: *IEEE Security Privacy* 15.5 (2017), pp. 29–37.

[AKD03] Mikhail J. Atallah, Florian Kerschbaum, and Wenliang Du. "Secure and Private Sequence Comparisons". In: *Proceedings of the 2003 ACM Workshop on Privacy in the Electronic Society (WPES)*. 2003, pp. 39–44.

[Akg+15] Mete Akgün, A Osman Bayrak, Bugra Ozer, and M Şamil Sağıroğlu. "Privacy preserving processing of genomic data: A survey". In: *Journal of biomedical informatics* 56 (2015), pp. 103–111.

[AKS02] Kristin G. Ardlie, Leonid Kruglyak, and Mark Seielstad. "Patterns of linkage disequilibrium in the human genome". In: *Nature Review Genetics* 3 (2002), pp. 299–309.

[Alt+90] Stephen F Altschul, Warren Gish, Webb Miller, Eugene W Myers, and David J Lipman. "Basic local alignment search tool". In: *Journal of molecular biology* 215.3 (1990), pp. 403–410.

[Ama+20] Shanika L. Amarasinghe, Shian Su, Xueyi Dong, Luke Zappia, Matthew E. Ritchie, and Quentin Gouil. "Opportunities and challenges in long-read sequencing data analysis". In: *Genome Biology* 21.30 (2020), pp. 1–16.

[Ard+14] Amin Ardeshirdavani, Erika Souche, Luc Dehaspe, Jeroen Van Houdt, Joris Robert Vermeesch, and Yves Moreau. "NGS-Logistics: federated analysis of NGS sequence variants across multiple locations". In: *Genome Medicine* 6.9 (2014), p. 71.

[ARH13] E. Ayday, J. L. Raisaro, and J. P. Hubaux. "Personal use of the genomic data: Privacy vs. storage cost". In: *2013 IEEE Global Communications Conference (GLOBECOM)*. 2013, pp. 2723–2729.

[ARM] ARM. *Arm TrustZone Technology*. https://developer.arm.com/ip-products/security-ip/trustzone. Online; Accessed: 09-December-2019.

[Ayd+13] Erman Ayday, Jean Louis Raisaro, Jean-Pierre Hubaux, and Jacques Rougemont. "Protecting and Evaluating Genomic Privacy in Medical Tests and Personalized Medicine". In: *Proceedings of the 12th ACM Workshop on Privacy in the Electronic Society (WPES)*. 2013, pp. 95–106.

[Ayd+14] Erman Ayday, Jean Louis Raisaro, Urs Hengartner, Adam Molyneaux, and Jean-Pierre Hubaux. "Privacy-preserving processing of raw genomic data". In: *Data Privacy Management and Autonomous Spontaneous Security*. Springer, 2014, pp. 133–147.

[Ayd+15] Erman Ayday, Emiliano De Cristofaro, Jean-Pierre Hubaux, and Gene Tsudik. "Whole Genome Sequencing: Revolutionary Medicine or Privacy Nightmare?" In: *IEEE Computer* 48.2 (2015), pp. 58–66.

[Bac+16] Michael Backes, Pascal Berrang, Mathias Humbert, and Praveen Manoharan. "Membership Privacy in MicroRNA-based Studies". In: *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*. 2016, pp. 319–330.

[Bar+12] Joshua Baron, Karim El Defrawy, Kirill Minkovich, Rafail Ostrovsky, and Eric Tressler. "5pm: Secure pattern matching". In: *SCN* (2012), pp. 222–240.

[Bes+13] Alysson Bessani, Miguel Correia, Bruno Quaresma, Fernando André, and Paulo Sousa. "DepSky: Dependable and Secure Storage in a Cloud-of-Clouds". In: *ACM Trans. Storage* 9.4 (2013).

[Bes+15] Alysson Bessani, Jörgen Brandt, Marc Bux, Vinicius Cogo, Lora Dimitrova, Jim Dowling, Ali Gholami, Kamal Hakimzadeh, Micheal Hummel, Mahmoud Ismail, Erwin Laure, Ulf Leser, Jan-Eric Litton, Roxanna Martinez, Salman Niazi, Jane Reichel, and Karin Zimmermann. "BiobankCloud: a platform for the secure storage, sharing, and processing of large biomedical data sets". In: *DMAH* (2015).

[Blo70] Burton H Bloom. "Space/time trade-offs in hash coding with allowable errors". In: *Commun. ACM* 13.7 (1970), pp. 422–426.

[BM04] Andrei Broder and Michael Mitzenmacher. "Network applications of bloom filters: A survey". In: *Internet mathematics* 1.4 (2004), pp. 485–509.

[Bra+09] Rosemary Braun, William Rowe, Carl Schaefer, Jinghui Zhang, and Kenneth Buetow. "Needles in the Haystack: Identifying Individuals Present in Pooled Genomic Data". In: *PLOS Genetics* 5.10 (2009), pp. 1–8.

[Bra+17] Ferdinand Brasser, Urs Müller, Alexandra Dmitrienko, Kari Kostiainen, Srdjan Capkun, and Ahmad-Reza Sadeghi. "Software Grand Exposure: SGX Cache Attacks Are Practical". In: *11th USENIX Workshop on Offensive Technologies (WOOT)*. 2017.

[Cai+15] Ruichu Cai, Zhifeng Hao, Marianne Winslett, Xiaokui Xiao, Yin Yang, Zhenjie Zhang, and Shuigeng Zhou. "Deterministic identification of specific individuals from GWAS results". In: *Bioinformatics* 31.11 (2015), pp. 1701–1707.

[Cam16] Todd Campbell. *Is This the Biggest Threat Yet to Illumina?* https://www.fool.com/investing/general/2016/03/18/is-this-the-biggest-threat-yet-to-illumina.aspx. Online; Accessed: 16-April-2019. 2016.

[Çet+17] Gizem S. Çetin, Hao Chen, Kim Laine, Kristin Lauter, Peter Rindal, and Yuhou Xia. "Private queries on encrypted genomic data". In: *BMC Medical Genomics* 10.2 (2017), p. 45.

[Che+12] Yangyi Chen, Bo Peng, XiaoFeng Wang, and Haixu Tang. "Large-Scale Privacy-Preserving Mapping of Human Genomic Sequences on Hybrid Clouds". In: *NDSS*. 2012.

[Che+17a] Feng Chen, Chenghong Wang, Wenrui Dai, Xiaoqian Jiang, Noman Mohammed, Md Momin Al Aziz, Md Nazmus Sadat, Cenk Sahinalp, Kristin Lauter, and Shuang Wang. "PRESAGE: PRivacy-preserving gEnetic testing via SoftwAre Guard Extension". In: *BMC Medical Genomics* 10.48 (2017), pp. 77–85.

[Che+17b] Feng Chen, Shuang Wang, Xiaoqian Jiang, Sijie Ding, Yao Lu, Jihoon Kim, S Cenk Sahinalp, Chisato Shimizu, Jane C Burns, Victoria J Wright, Eileen Png, Martin L Hibberd, David D Lloyd, Hai Yang, Amalio Telenti, Cinnamon S Bloss, Dov Fox, Kristin Lauter, and Lucila Ohno-Machado. "PRINCESS: Privacy-protecting Rare disease International Network Collaboration via Encryption through Software guard extensionS". In: *Bioinformatics* 33.6 (2017), pp. 871–878.

[CKL15] Jung Hee Cheon, Miran Kim, and Kristin Lauter. "Homomorphic computation of edit distance". In: *Financial Cryptography and Data Security*. 2015, pp. 194–212.

[Cla10] David Clayton. "On inferring presence of an individual in a mixture: a Bayesian approach". In: *Biostatistics* 11.4 (2010), pp. 661–673.

[Cog+15] Vinicius V Cogo, Alysson Bessani, Francisco M Couto, and Paulo Verissimo. "A high-throughput method to detect privacy-sensitive human genomic data". In: *Proceedings of the 14th ACM Workshop on Privacy in the Electronic Society*. 2015, pp. 101–110.

[Con+15] Scott D. Constable, Yuzhe Tang, Shuang Wang, Xiaoqian Jiang, and Steve Chapin. "Privacy-preserving GWAS analysis on federated genomic datasets". In: *BMC Medical Informatics and Decision Making* 15.S2 (2015).

[Con05] The International HapMap Consortium. "A haplotype map of the human genome". In: *Nature* 437.7063 (2005), pp. 1299–1320.

[Con15] The 1000 Genomes Project Consortium. "A global reference for human genetic variation". In: *Nature* 526 (2015), pp. 68–74.

[DB10] Adrian V Dalca and Michael Brudno. "Genome variation discovery with high-throughput sequencing data". In: *Briefings in Bioinformatics* 11.1 (2010), pp. 3–14.

[DDK16] Stephanie OM Dyke, Edward S Dove, and Bartha M Knoppers. "Sharing health-related data: a privacy test?" In: *NPJ genomic medicine* 1.16024 (2016), pp. 1–6.

[Dez+18] I. Deznabi, M. Mobayen, N. Jafari, O. Tastan, and E. Ayday. "An Inference Attack on Genomic Data Using Kinship, Complex Correlations, and Phenotype Information". In: *IEEE/ACM Transactions on Computational Biology and Bioinformatics* 15.4 (2018), pp. 1333–1343.

[DFT13] Emiliano De Cristofaro, Sky Faber, and Gene Tsudik. "Secure genomic testing with size-and position-hiding private substring matching". In: *Proceedings of the 12th ACM workshop on Workshop on privacy in the electronic society*. 2013, pp. 107–118.

[Dur+14] Elizabeth A Durham, Murat Kantarcioglu, Yuan Xue, Csaba Toth, Mehmet Kuzu, and Bradley Malin. "Composite bloom filters for secure record linkage". In: *IEEE transactions on knowledge and data engineering* 26.12 (2014), pp. 2956–2968.

[ED08] Khaled El Emam and Fida Kamal Dankar. “Protecting Privacy Using k-Anonymity”. In: *Journal of the American Medical Informatics Association* 15.5 (2008), pp. 627–637.

[EN14] Yaniv Erlich and Arvind Narayanan. “Routes for breaching and protecting genetic privacy”. In: *Nature Reviews Genetics* 15 (2014), pp. 409–421.

[Erl+18] Yaniv Erlich, Tal Shor, Shai Carmi, and Itsik Pe’er. “Re-identification of genomic data using long range familial searches”. In: *bioRxiv* (2018).

[FEJ15] Benjamin Fabian, Tatiana Ermakova, and Philipp Junghanns. “Collaborative and secure sharing of healthcare data in multi-clouds”. In: *Information Systems* 48 (2015), pp. 132–150.

[Fre+14] Matthew Fredrikson, Eric Lantz, Somesh Jha, Simon Lin, David Page, and Thomas Ristenpart. “Privacy in Pharmacogenetics: An End-to-End Case Study of Personalized Warfarin Dosing”. In: *Proceedings of the USENIX Security Symposium* (2014), pp. 17–32.

[Gar] Mackenzie Garrity. *Patient medical records sell for $1K on dark web.* https://www.beckershospitalreview.com/cybersecurity/patient-medical-records-sell-for-1k-on-dark-web.html. Published: 20-February-2019; Accessed: 18-April-2020.

[GH16] The Global Alliance for Genomics and Health. “A federated ecosystem for sharing genomic, clinical data”. In: *Science* 352.6291 (2016), pp. 1278–1280.

[Git09] Jane Gitschier. “Inferential genotyping of Y chromosomes in Latter-Day Saints founders and comparison to Utah samples in the HapMap project”. In: *The American Journal of Human Genetics* 84.2 (2009), pp. 251–258.

[Goh+13] Anita M. Y. Goh, Edmond Chiu, Olga Yastrubetskaya, Cheryl Erwin, Janet K. Williams, Andrew R. Juhl, and Jane S. Paulsen. “Perception, experience, and response to genetic discrimination in Huntington’s disease: the Australian results of The International RESPOND-HD study”. In: *Genetic testing and molecular biomarkers* 17.2 (2013), pp. 115–121.

[Goo09] Michael T Goodrich. “The mastermind attack on genomic data”. In: *Security and Privacy, 2009 30th IEEE Symposium on*. IEEE. 2009, pp. 204–218.

[Göt+17] Johannes Götzfried, Moritz Eckert, Sebastian Schinzel, and Tilo Müller. "Cache Attacks on Intel SGX". In: *Proceedings of the 10th European Workshop on Systems Security (EuroSec)*. 2017, pp. 1–6.

[Got82] Osamu Gotoh. "An improved algorithm for matching biological sequences". In: *Journal of molecular biology* 162.3 (1982), pp. 705–708.

[GRB07] Yevgeniy Gelfand, Alfredo Rodriguez, and Gary Benson. "TRDB–the Tandem Repeats Database". In: *Nucleic acids research* 35.Database issue (2007), pp. 80–87.

[Gup08] Pushpendra K. Gupta. "Single-molecule DNA sequencing technologies for future genomics research". In: *Trends in Biotechnology* 26.11 (2008), pp. 602–611.

[Gym+13] Melissa Gymrek, Amy L McGuire, David Golan, Eran Halperin, and Yaniv Erlich. "Identifying personal genomes by surname inference". In: *Science* 339.6117 (2013), pp. 321–324.

[Hay13] Erika Check Hayden. "Privacy protections: The genome hacker. Yaniv Erlich shows how research participants can be identified from 'anonymous' DNA". In: *Nature* 497.7448 (2013), pp. 172–174.

[He+18] Z. He, J. Yu, J. Li, Q. Han, G. Luo, and Y. Li. "Inference Attacks and Controls on Genotypes and Phenotypes for Individual Genomic Data". In: *IEEE/ACM Transactions on Computational Biology and Bioinformatics* (2018), pp. 1–1.

[HG18] Arif Harmanci and Mark Gerstein. "Analysis of sensitive information leakage in functional genomics signal profiles through genomic deletions". In: *Nature Communications* 9.2453 (2018), pp. 1–9.

[Hom+08] Nils Homer, Szabolcs Szelinger, Margot Redman, David Duggan, Waibhav Tembe, Jill Muehling, John V Pearson, Dietrich A Stephan, Stanley F Nelson, and David W Craig. "Resolving individuals contributing trace amounts of DNA to highly complex mixtures using high-density SNP genotyping microarrays". In: *PLoS Genetics* 4.8 (2008), pp. 1–9.

[HT] Breeanna Hare and Christo Taoushiani. *What we know about the Golden State Killer case, one year after a suspect was arrested*. https://edition.cnn.com/2019/04/24/us/golden-state-killer-one-year-later/index.html. Published: 24-April-2019; Accessed: 09-December-2019.

[Hua+11] Yan Huang, David Evans, Jonathan Katz, and Lior Malka. "Faster Secure Two-Party Computation Using Garbled Circuits". In: *USENIX Security Symposium*. Vol. 201. 1. 2011.

[Hum+13] Mathias Humbert, Erman Ayday, Jean-Pierre Hubaux, and Amalio Telenti. "Addressing the concerns of the lacks family: quantification of kin genomic privacy". In: *Proceedings of the 2013 ACM SIGSAC conference on Computer & communications security*. 2013, pp. 1141–1152.

[Hum+15] Mathias Humbert, Kévin Huguenin, Joachim Hugonot, Erman Ayday, and Jean-Pierre Hubaux. "De-anonymizing Genomic Databases Using Phenotypic Traits". In: *Proceedings on Privacy Enhancing Technologies* 2015.2 (2015), pp. 99–114.

[Hum+17] Mathias Humbert, Erman Ayday, Jean-Pierre Hubaux, and Amalio Telenti. "Quantifying Interdependent Risks in Genomic Privacy". In: *ACM Trans. Priv. Secur.* 20.1 (2017), pp. 1–31.

[IGS] International Genome Sample Resource (IGSR). *1000 Genomes Project: A Deep Catalog of Human Genetic Variation*. Available at: http://www.1000genomes.org/.

[Ill19] Illumina. *BaseSpace Sequence Hub – Suport Resources*. https://support.illumina.com/sequencing/sequencing_software/basespace.html. Online; Accessed 03-November-2019. 2019.

[Ip+15] CLC Ip, M Loose, JR Tyson, M de Cesare, BL Brown, M Jain, RM Leggett, DA Eccles, V Zalunin, JM Urban, P Piazza, RJ Bowden, B Paten, S Mwaigwisya, EM Batty, JT Simpson, TP Snutch, E Birney, D Buck, S Goodwin, HJ Jansen, J O'Grady, HE Olsen, and null null null. "MinION Analysis and Reference Consortium: Phase 1 data release and analysis [version 1; referees: 2 approved]". In: *F1000Research* 4.1075 (2015).

[Jac+09] Kevin B Jacobs, Meredith Yeager, Sholom Wacholder, David Craig, Peter Kraft, David J Hunter, Justin Paschal, Teri A Manolio, Margaret Tucker, Robert N Hoover, et al. "A new statistic and its power to infer membership in a genome-wide association study using genotype frequencies". In: *Nature genetics* 41.11 (2009), pp. 1253–1257.

[Jia+14] Xiaoqian Jiang, Yongan Zhao, Xiaofeng Wang, Bradley Malin, Shuang Wang, Lucila Ohno-Machado, and Haixu Tang. "A community assessment of privacy preserving techniques for human genomes". In: *BMC Medical Informatics and Decision Making* 14.S1 (2014), pp. 1–10.

[JKS08] Somesh Jha, Louis Kruger, and Vitaly Shmatikov. "Towards practical privacy for genomic computation". In: *IEEE Symposium on Security and Privacy*. 2008, pp. 216–230.

[Jün+13] Sebastian Jünemann, Fritz Joachim Sedlazeck, Karola Prior, Andreas Albersmeier, Uwe John, Jörn Kalinowski, Alexander Mellmann, Alexander Goesmann, Arndt von Haeseler, Jens Stoye, and Dag Harmsen. "Updating benchtop sequencing perfomance comparison". In: *Nature Biotechnology* 31 (2013), pp. 294–296.

[KAC14] Robert Klitzman, Paul S Appelbaum, and Wendy Chung. "Should Life Insurers Have Access to Genetic Test Results?" In: *JAMA* 312.18 (2014), pp. 1855–1856.

[Kan+08] Murat Kantarcioglu, Wei Jiang, Ying Liu, and Bradley Malin. "A cryptographic approach to securely share and query genomic sequences". In: *IEEE Transactions on information technology in biomedicine* 12.5 (2008), pp. 606–617.

[Kay+09] Jane Kaye, Catherine Heeney, Naomi Hawkins, Jantina de Vries, and Paula Boddington. "Data sharing in genomics re-shaping scientific practice". In: *Nature Reviews Genetics* 10.5 (2009), pp. 331–335.

[KGE17] Mehdi Kchouk, Jean-Francois Gibrat, and Mourad Elloumi. "Generations of Sequencing Technologies: From First to Next Generation". In: *Biology and Medicine* 9 (2017).

[Kie+11] Szymon M Kiełbasa, Raymond Wan, Kengo Sato, Paul Horton, and Martin C Frith. "Adaptive seeds tame genomic sequence comparison". In: *Genome research* 21.3 (2011), pp. 487–493.

[KL15] Miran Kim and Kristin Lauter. "Private genome analysis through homomorphic encryption". In: *BMC Medical Informatics and Decision Making* 15.S3 (2015), pp. 1–12.

[Kni] Meredith Knight. *Life insurance companies deny coverage to those with cancer genes like BRCA*. https://geneticliteracyproject.org/2016/05/09/life-insurance-companies-deny-coverage-cancer-genes-like-brca/. Published: 09-May-2016; Accessed: 28-May-2018.

[Kno14] Bartha Maria Knoppers. “Framework for responsible sharing of genomic and health-related data”. In: *The HUGO Journal* 8.3 (2014), pp. 1–6.

[Kon+08] Augustine Kong, Gisli Masson, Michael L Frigge, Arnaldur Gylfason, Pasha Zusmanovich, Gudmar Thorleifsson, Pall I Olason, Andres Ingason, Stacy Steinberg, Thorunn Rafnar, Patrick Sulem, Magali Mouy, Frosti Jonsson, Unnur Thorsteinsdottir, Daniel F Gudbjartsson, Hreinn Stefansson, and Kari Stefansson. “Detection of sharing by descent, long-range phasing and haplotype imputation”. In: *Nature Genetics* 40.9 (2008), pp. 1068–1075.

[Kuz+11] Mehmet Kuzu, Murat Kantarcioglu, Elizabeth Durham, and Bradley Malin. “A constraint satisfaction cryptanalysis of Bloom filters in private record linkage”. In: *International Symposium on Privacy Enhancing Technologies Symposium*. Springer. 2011, pp. 226–245.

[Lan+09] Ben Langmead, Cole Trapnell, Mihai Pop, Steven L Salzberg, et al. “Ultrafast and memory-efficient alignment of short DNA sequences to the human genome”. In: *Genome biol* 10.3 (2009), pp. 1–10.

[LD09] Heng Li and Richard Durbin. “Fast and accurate short read alignment with Burrows–Wheeler transform”. In: *Bioinformatics* 25.14 (2009), pp. 1754–1760.

[LH10] Heng Li and Nils Homer. “A survey of sequence alignment algorithms for next-generation sequencing”. In: *Briefings in bioinformatics* 11.5 (2010), pp. 473–483.

[LHA02] Zhen Lin, Michael Hewett, and Russ B Altman. “Using binning to maintain confidentiality of medical data”. In: *Proceedings of the AMIA Symposium*. American Medical Informatics Association. 2002, pp. 454–458.

[Li+10] Yun Li, Cristen J Willer, Jun Ding, Paul Scheet, and Gonçalo R Abecasis. “MaCH: Using Sequence and Genotype Data to Estimate Haplotypes and Unobserved Genotypes”. In: *Genetic epidemiology* 34.8 (2010), pp. 816–834.

[Li11] Heng Li. *wgsim-Read simulator for next generation sequencing*. https://github.com/lh3/wgsim. Github Repository. 2011.

[Lip+17] Christoph Lippert, Riccardo Sabatini, M. Cyrus Maher, Eun Yong Kang, Seunghak Lee, Okan Arikan, Alena Harley, Axel Bernal, Peter Garst, Victor Lavrenko, Ken Yocum, Theodore Wong, Mingfu Zhu, Wen-Yun Yang, Chris Chang, Tim Lu, Charlie W. H. Lee, Barry Hicks, Smriti Ramakrishnan, Haibao Tang, Chao Xie, Jason Piper, Suzanne Brewerton, Yaron Turpaz, Amalio Telenti, Rhonda K. Roby, Franz J. Och, and J. Craig Venter. "Identification of individuals by trait prediction using whole-genome sequencing data". In: *Proceedings of the National Academy of Sciences* 114.38 (2017), pp. 1–6.

[Liu+12] Lin Liu et al. "Comparison of next-generation sequencing systems". In: *J. Biomed. Biotechnol.* 2012.251364 (2012), pp. 1–11.

[LM16] Shawn E. Levy and Richard M. Myers. "Advancements in Next-Generation Sequencing". In: *Annual Review of Genomics and Human Genetics* 17.1 (2016), pp. 95–115.

[LWS12] Guang Li, Yadong Wang, and Xiaohong Su. "Improvements on a privacy-protection algorithm for DNA sequences with generalization lattices". In: *Computer Methods and Programs in Biomedicine* 108.1 (2012), pp. 1–9.

[Mal02] Bradley Malin. *Compromising privacy with trail re-identification: the REIDIT algorithms*. Carnegie Mellon University. Center for Automated Learning and Discovery, 2002.

[Mal05a] Bradley A Malin. "An evaluation of the current state of genomic data privacy protection technology and a roadmap for the future". In: *Journal of the American Medical Informatics Association* 12.1 (2005), pp. 28–34.

[Mal05b] Bradley A. Malin. "Protecting DNA Sequence Anonymity with Generalization Lattices". In: *Methods of Information in Medicine* 44 (2005), pp. 687–692.

[Mal06] Bradley Malin. "Re-identification of familial database records". In: *AMIA Annual Symposium proceedings* 2006 (2006), pp. 524–528.

[Man+18] Avradip Mandal, John C. Mitchell, Hart Montgomery, and Arnab Roy. "Data Oblivious Genome Variants Search on Intel SGX". In: *Proceedings of the DPM and CBT workshops, ESORICS*. 2018, pp. 296–310.

[Men+19] R. Mendes, T. Oliveira, V. V. Cogo, N. F. Neves, and A. N. Bessani. "CHARON: A Secure Cloud-of-Clouds System for Storing and Sharing Big Data". In: *IEEE Transactions on Cloud Computing* (2019), pp. 1–1.

[Met10] Michael L Metzker. "Sequencing technologies—the next generation". In: *Nature reviews Genetics* 11.1 (2010), pp. 31–46.

[MM08] Olena Morozova and Marco A. Marra. "Applications of next-generation sequencing technologies in functional genomics". In: *Genomics* 92.5 (2008), pp. 255–264.

[MMC19] Alexandros Mittos, Bradley Malin, and Emiliano De Cristofaro. "Systematizing Genome Privacy Research: A Privacy-Enhancing Technologies Perspective". In: *Proceedings on Privacy Enhancing Technologies* 2019.1 (2019), pp. 87–107.

[MPG14] Antonis Michalas, Nicolae Paladi, and Christian Gehrmann. "Security aspects of e-health systems migration to the cloud". In: *e-Health Networking, Applications and Services (Healthcom), 2014 IEEE 16th International Conference on*. IEEE. 2014, pp. 212–218.

[MS00] Bradley Malin and Latanya Sweeney. "Determining the identifiability of DNA database entries." In: *Proceedings of the AMIA Symposium*. 2000, pp. 537–541.

[MS04] Bradley Malin and Latanya Sweeney. "How (not) to protect genomic data privacy in a distributed network: using trail re-identification to evaluate and design anonymity protection systems". In: *Journal of biomedical informatics* 37.3 (2004), pp. 179–192.

[Nav+15] Muhammad Naveed, Erman Ayday, Ellen W. Clayton, Jacques Fellay, Carl A. Gunter, Jean-Pierre Hubaux, Bradley A. Malin, and Xiaofeng Wang. "Privacy in the Genomic Era". In: *ACM Computing Surveys* 48.1 (2015), pp. 1–44.

[NTP16] Mina Namazi, Juan Ramón Troncoso-Pastoriza, and Fernando Pérez-González. "Dynamic Privacy-Preserving Genomic Susceptibility Testing". In: *Proceedings of the 4th ACM Workshop on Information Hiding and Multimedia Security (IH&MMSec)*. 2016, pp. 45–50.

[NYV09] D R. Nyholt, C.-E. Yu, and P. M. Visscher. "On Jim Watson's APOE status: genetic information is hard to hide". In: *European Journal of Human Genetics* 17 (2009), pp. 147–149.

[ODS13] Aisling O'Driscoll, Jurate Daugelaite, and Roy D. Sleator. "'"Big data", Hadoop and cloud computing in genomics". In: *Journal of Biomedical Informatics* 46.5 (2013), pp. 774–781.

[PAC] PACBIO. *SMRT Sequencing: Read lenghts*. https://www.pacb.com/smrt-science/smrt-sequencing/read-lengths/. Online; Accessed: 16-April-2019.

[PB17] Victoria Popic and Serafim Batzoglou. "A hybrid cloud read aligner based on MinHash and kmer voting that preserves privacy". In: *Nature Communications* 8.15311 (2017), pp. 1–7.

[Pol+18] Martin O Pollard, Deepti Gurdasani, Alexander J Mentzer, Tarryn Porter, and Manjinder S Sandhu. "Long reads: their purpose and place". In: *Human Molecular Genetics* 27.2 (2018), pp. 234–241.

[PS13] Ram Vinay Pandey and Christian Schlötterer. "DistMap: a toolkit for distributed short read mapping on a Hadoop cluster". In: *PLoS One* 8.8 (2013), e72614.

[Rai+17a] Jean Louis Raisaro, Florian Tramèr, Zhanglong Ji, Diyue Bu, Yongan Zhao, Knox Carey, David Lloyd, Heidi Sofia, Dixie Baker, Paul Flicek, Suyash Shringarpure, Carlos Bustamante, Shuang Wang, Xiaoqian Jiang, Lucila Ohno-Machado, Haixu Tang, XiaoFeng Wang, and Jean-Pierre Hubaux. "Addressing Beacon re-identification attacks: quantification and mitigation of privacy risks". In: *Journal of the American Medical Informatics Association* 24.4 (2017), pp. 799–805.

[Rai+17b] Jean Louis Raisaro, Carmela Troncoso, Mathias Humbert, Zoltan Kutalik, Amalio Telenti, and Jean-Pierre Hubaux. "GenoShare: Supporting Privacy-Informed Decisions for Sharing Exact Genomic Data". In: *EPFL infoscience* (2017), pp. 1–19.

[Ran+14] Sean M Randall, Anna M Ferrante, James H Boyd, Jacqueline K Bauer, and James B Semmens. "Privacy-preserving record linkage on large real world datasets". In: *Journal of biomedical informatics* 50 (2014), pp. 205–212.

[RC11] Francisco Rocha and Miguel Correia. "Lucy in the Sky Without Diamonds: Stealing Confidential Data in the Cloud". In: *IEEE/IFIP 41st International Conference on Dependable Systems and Networks Workshops (DSN-W)* (2011), pp. 129–134.

[Rus94] John Rushby. “Critical system properties: survey and taxonomy”. In: *Reliability Engineering & System Safety* 43.2 (1994). Special Issue on Software Safety, pp. 189–219.

[Sae+12] Fahad Saeed, Alan Perez-Rathke, Jaroslaw Gwarnicki, Tanya Berger-Wolf, and Ashfaq Khokhar. “High Performance Multiple Sequence Alignment System for Pyrosequencing Reads from Multiple Reference Genomes”. In: *Journal of parallel and distributed computing* 72.1 (2012), pp. 83–89.

[Sam+15] S. S. Samani, Z. Huang, E. Ayday, M. Elliot, J. Fellay, J. P. Hubaux, and Z. Kutalik. “Quantifying Genomic Privacy via Inference Attack with High-Order SNV Correlations”. In: *2015 IEEE Security and Privacy Workshops*. 2015, pp. 32–40.

[San] Lauren Santye. *The Deep Dark Web: Medical Records Sold on the Black Market*. https://contemporaryclinic.pharmacytimes.com/news-views/the-deep-dark-web-medical-records-sold-on-the-black-market. Published: 16-November-2016; Accessed: 18-April-2020.

[San+09] Sriram Sankararaman, Guillaume Obozinski, Michael I Jordan, and Eran Halperin. “Genomic privacy and limits of individual detection in a pool”. In: *Nature genetics* 41.9 (2009), pp. 965–967.

[SB15] Suyash Shringarpure and Carlos Bustamante. “Privacy risks from genomic data-sharing beacons”. In: *The American Journal of Human Genetics* 97.5 (2015), pp. 631–646.

[SB17] Rainer Schnell and Christian Borgs. “Secure privacy preserving record linkage of large databases by modified Bloom filter encodings”. In: *International Journal for Population Data Science* 1.1 (2017), pp. 1–10.

[SBR09] Rainer Schnell, Tobias Bachteler, and Jörg Reiher. “Privacy-preserving record linkage using Bloom filters”. In: *BMC medical informatics and decision making* 9.41 (2009), pp. 1–11.

[SC75] F. Sanger and A.R. Coulson. “A rapid method for determining sequences in DNA by primed synthesis with DNA polymerase”. In: *Journal of Molecular Biology* 94.3 (1975), pp. 441–448.

[Sch+17] Michael Schwarz, Samuel Weiser, Daniel Gruss, Clémentine Maurice, and Stefan Mangard. "Malware Guard Extension: Using SGX to Conceal Cache Attacks". In: *Detection of Intrusions and Malware, and Vulnerability Assessment*. 2. Springer International Publishing, 2017, pp. 3–24.

[Sch09] Michael C Schatz. "CloudBurst: highly sensitive read mapping with MapReduce". In: *Bioinformatics* 25.11 (2009), pp. 1363–1369.

[Ser19] Amazon Web Services. *Illumina Saves Nearly $400,000 Monthly, Speeds Genomics Analysis Using Amazon EC2 Spot Instances*. https://aws.amazon.com/solutions/case-studies/illumina/. Online; Accessed 09-December-2019. 2019.

[Sov+16] Ivan Sović, Mile Šikić, Andreas Wilm, Shannon Nicole Fenlon, Swaine Chen, and Niranjan Nagarajan. "Fast and sensitive mapping of nanopore sequencing reads with GraphMap". In: *Nature Communications* 7.11307 (2016), pp. 1–11.

[Ste+15] Zachary D. Stephens, Skylar Y. Lee, Faraz Faghri, Roy H. Campbell, Chengxiang Zhai, Miles J. Efron, Ravishankar Iyer, Michael C. Schatz, Saurabh Sinha, and Gene E. Robinson. "Big Data: Astronomical or Genomical?" In: *PLOS Biology* 13.7 (2015), pp. 1–11.

[SW81] Temple F Smith and Michael S Waterman. "Identification of common molecular subsequences". In: *Journal of molecular biology* 147.1 (1981), pp. 195–197.

[Swe00] Latanya Sweeney. "Simple demographics often identify people uniquely". In: *Health (San Francisco)* 671 (2000), pp. 1–34.

[Swe02] Latanya Sweeney. "k-anonymity: A model for protecting privacy". In: *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems* 10.05 (2002), pp. 557–570.

[SWH12] Eric E Schadt, Sangsoon Woo, and Ke Hao. "Bayesian method to predict individual SNP genotypes from gene expression data". In: *Nature Genetics* 44.5 (2012), pp. 603–608.

[TAC18] Nora von Thenen, Erman Ayday, and A Ercument Cicek. "Re-identification of individuals in genomic data-sharing beacons via allele inference". In: *Bioinformatics* 35.3 (2018), pp. 365–371.

[Tec20] Oxford Nanopore Technologies. *MinION*. https://nanoporetech.com/products/minion. Online; Accessed: 03-November-2019. 2008-2020.

[Ton+14] Yue Tong, Jinyuan Sun, Sherman SM Chow, and Pan Li. "Cloud-assisted mobile-access of health data with privacy and auditability". In: *IEEE Journal of biomedical and health Informatics* 18.2 (2014), pp. 419–429.

[VB13] Paulo E. Verissimo and Alysson Bessani. "E-biobanking: What Have You Done to My Cell Samples?" In: *Security Privacy* 11.6 (2013), pp. 62–65.

[VG16] Effy Vayena and Urs Gasser. "Between Openness and Privacy in Genomics". In: *PLoS Medicine* 13.1 (2016), pp. 1–7.

[Wag15] I. Wagner. "Genomic Privacy Metrics: A Systematic Comparison". In: *2015 IEEE Security and Privacy Workshops*. 2015, pp. 50–59.

[Wan+09] Rui Wang, Yong Fuga Li, XiaoFeng Wang, Haixu Tang, and Xiaoyong Zhou. "Learning your identity and disease from research papers: information leaks in genome wide association study". In: *Proceedings of the 16th ACM conference on Computer and communications security*. 2009, pp. 534–544.

[WP03] Jeffrey D. Wall and Jonathan K. Pritchard. "Haplotype blocks and linkage disequilibrium in the human genome". In: *Nature Review Genetics* 4.8 (2003), pp. 587–597.

[YKÖ17] Buket Yüksel, Alptekin Küpçü, and Öznur Özkasap. "Research issues for privacy and security of electronic health services". In: *Future Generation Computer Systems* 68 (2017), pp. 1–13.

[Zaa+17] Sophie Zaaijer, Assaf Gordon, Daniel Speyer, Robert Piccone, Simon Cornelis Groen, and Yaniv Erlich. "Rapid re-identification of human samples using portable DNA sequencing". In: *eLife* 6.e27798 (2017), pp. 1–17.

[Zha+11] Kehuan Zhang, Xiaoyong Zhou, Yangyi Chen, XiaoFeng Wang, and Yaoping Ruan. "Sedic: Privacy-aware Data Intensive Computing on Hybrid Clouds". In: *Proceedings of the 18th ACM conference on Computer and communications security*. 2011, pp. 515–526.

[Zho+11] Xiaoyong Zhou, Bo Peng, Yong Fuga Li, Yangyi Chen, Haixu Tang, and XiaoFeng Wang. "To release or not to release: Evaluating information leaks in aggregate human-genome data". In: *Computer Security–ESORICS 2011*. Springer, 2011, pp. 607–627.

[Zoo+16] Justin M. Zook, David Catoe, Jennifer McDaniel, Lindsay Vang, Noah Spies, Arend Sidow, Ziming Weng, Yuling Liu, Christopher E. Mason, Noah Alexander, Elizabeth Henaff, Alexa B.R. McIntyre, Dhruva Chandramohan, Feng Chen, Erich Jaeger, Ali Moshrefi, Khoa Pham, William Stedman, Tiffany Liang, Michael Saghbini, Zeljko Dzakula, Alex Hastie, Han Cao, Gintaras Deikus, Eric Schadt, Robert Sebra, Ali Bashir, Rebecca M. Truty, Christopher C. Chang, Natali Gulbahce, Keyan Zhao, Srinka Ghosh, Fiona Hyland, Yutao Fu, Mark Chaisson, Chunlin Xiao, Jonathan Trow, Stephen T. Sherry, Alexander W. Zaranek, Madeleine Ball, Jason Bobe, Preston Estep, George M. Church, Patrick Marks, Sofia Kyriazopoulou-Panagiotopoulou, Grace X.Y. Zheng, Michael Schnall-Levin, Heather S. Ordonez, Patrice A. Mudivarti, Kristina Giorda, Ying Sheng, Karoline Bjarnesdatter Rypdal, and Marc Salit. "Extensive sequencing of seven human genomes to characterize benchmark reference materials". In: *Scientific Data* 3.1 (2016).

[Zub19] Babangida Zubairu. "Security Risks of Biomedical Data Processing in Cloud Computing Environment". In: *Cloud Security*. 2019, pp. 1748–1768.