A High-Throughput Method to Detect Privacy-Sensitive Human Genomic Data

Vinicius V. Cogo[1], Alysson Bessani[1], Francisco M. Couto[1], and Paulo Verissimo[2]
[1]LaSIGE, Faculdade de Ciências, Universidade de Lisboa, Portugal
[2]SnT, University of Luxembourg, Luxembourg
{vvcogo,anbessani,fjcouto}@ciencias.ulisboa.pt, paulo.verissimo@uni.lu

ABSTRACT

Finding the balance between privacy protection and data sharing is one of the main challenges in managing human genomic data nowadays. Novel privacy-enhancing technologies are required to address the known disclosure threats to personal sensitive genomic data without precluding data sharing. In this paper, we propose a method that systematically detects privacy-sensitive DNA segments coming directly from an input stream, using as reference a knowledge database of known privacy-sensitive nucleic and amino acid sequences. We show that adding our detection method to standard security techniques provides a robust, efficient privacy-preserving solution that neutralizes threats related to recently published attacks on genome privacy based on short tandem repeats, disease-related genes, and genomic variations. Current global knowledge on human genomes demonstrates the feasibility of our approach to obtain a comprehensive database immediately, which can also evolve automatically to address future attacks as new privacy-sensitive sequences are identified. Additionally, we validate that the detection method can be fitted inline with the NGS—Next Generation Sequencing—production cycle by using Bloom filters and scaling out to faster sequencing machines.

Categories and Subject Descriptors

H.4 [**Information Systems Applications**]: Miscellaneous; J.3 [**Computer Applications**]: Life and Medical Sciences—*Biology and genetics*; K.4.1 [**Computers and Society**]: Public Policy Issues—*Privacy*

General Terms

Design; Security; Measurement

WPES'15, October 12, 2015, Denver, CO, USA.

ACM 978-1-4503-3820-2/15/10$15.00.
DOI: http://dx.doi.org/10.1145/2808138.2808139.

1. INTRODUCTION

The increase on sharing human genomic data accelerates medical breakthroughs, while also augments the risks to the privacy of sensitive genomic data [14]. Privacy protection and data sharing are not mutually exclusive. In fact, properly defending the former impels the latter in short-term and extends donors' engagement and trust. A human genome can uniquely identify its owner and reveal information about him/her and his/hers relatives, even for some past and future generations [1, 27]. Additionally, portions of biological data may provide hints on individual's health status with high confidence levels.

Recent studies introduced new attacks to individuals' privacy based on genomic data and other publicly-available information [16, 18, 28, 34]. They have the objective of non-consented disclosure of personal information of individuals from their genomic data, and it can be divided into two classes: threats leading to re-identify donors of anonymized DNA sequences, based on genetic genealogy profiling [16]; and threats leading to the inference of private and sensitive information (e.g., victim's health status) from (re-)identified DNA sequences, based on disease-related genes [28] and genomic variations [18, 34]. These attacks must be efficiently addressed to avoid a rollback on the trend to share DNA sequences, which would hurt genomic studies, or even harden regulations governing genomic data protection [5].

Detecting privacy-sensitive genomic data as soon as it is generated is a long-term ambition from the research and clinical communities [10, 15]. Recent works on privacy-preserving genome processing have been advocating the partitioning of genomic data, but assume this must be done manually [2] or by a tool out of their scope [20]. To the best of our knowledge, our work is the first to provide a comprehensive privacy-aware detection method that enables users to implement such partitioning automatically. We propose a *privacy-sensitivity detection* scheme composed of:

- algorithmic solutions to retrieve privacy-sensitive nucleic and amino acid sequences automatically, with parametric and evolving sensitivity;
- a privacy-sensitivity detection architecture to systematically recognize privacy-sensitive genomic data from the source (e.g., sequencing machines).

Conceptually, the detector has a clear mission: given a DNA segment of predefined size s, detect whether this segment may contain a known privacy-sensitive information or not (see Figure 1). It does so based on a database of published signatures or patterns of privacy-sensitive nucleic and amino

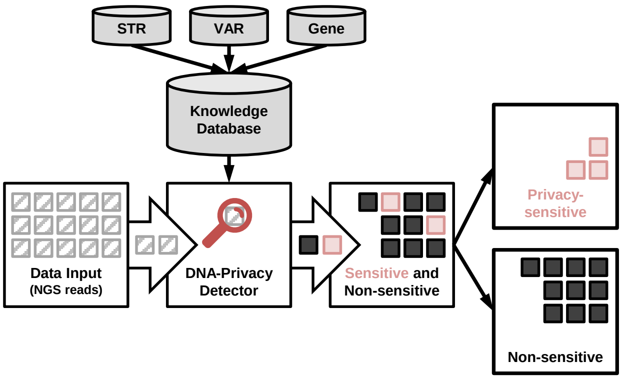


Figure 1: An overview of the method for detecting privacy-sensitive genomic data.

acid sequences (a *knowledge database*). Although conceptually simple, this approach meets at least two feasibility challenges:

(i) *Accuracy:* how to automatically classify DNA segments as privacy-sensitive or non-sensitive with high sensitivity and specificity?

(ii) *Performance:* how to implement a scalable detection solution that supports the high-throughput of modern sequencing machines?

The output from our solution is divided in two subsets: one with the privacy-sensitive portion of input data and the other containing the non-sensitive part. We initially foresee the following scenarios where our detector can be employed, but others may arise in the future:

- *Data segregation*: one may store and analyze privacy-sensitive data in a local, private infrastructure, while he/she may use external, multi-tenant infrastructures (e.g., public clouds) to work with the non-sensitive data.

- *Data outsourcing*: one may store the whole dataset in an external, multi-tenant infrastructure—if he/she applies strong, expensive security premises in the expectedly smaller privacy-sensitive portion, and applies more affordable security techniques in the non-sensitive portion. Additionally, homomorphic encryption could be used in the smaller privacy-sensitive portion to compute it securely in the external facility.

- *Data masking*: One may filter out one of the portions. For example, filtering out the larger non-sensitive portion allows users to store and process only the smaller privacy-sensitive portion, which is enough for several important analyses (e.g., personalized medicine).

Note the method we propose addresses the challenge of systematically detecting privacy-sensitive DNA sequences, whereas what should be done with the two output subsets is independent from our work and can have different implementations according to each use case.

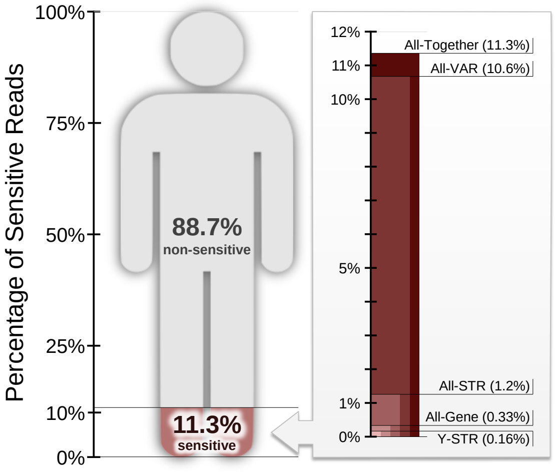


Figure 2: The percentage of privacy-sensitive sequences in entire human genomes considering different detector knowledge databases. The most complete data set (All-Together) lead to the detection of only 11.3% **of the sequences as privacy-sensitive.**

As a proof of concept of the accuracy and completeness of our approach, we have built a knowledge database from known short tandem repeats (small repeated strings), disease-related genes, and genomic variations currently available in public databases. Using all this information and being conservative about what can be considered private, only 11.3% of the sequences of a human genome are detected as privacy-sensitive (see Figure 2).

The originality of our work is in adapting existent enterprise privacy-preserving solutions and intrusion detection techniques to the genomics area, and combining them with different known privacy-sensitive information to protect individuals. By identifying the privacy-sensitive sequences using our solution and protecting them, one neutralizes the existent threats of re-identifying individuals [16], and of inferring private information about them [18, 28, 34]. Moreover, by using the large body of knowledge about the privacy sensitivity of human genomes available today, it was possible to create a reasonably complete privacy detector. Despite this completeness, the detector knowledge database can be automatically updated to address future attacks as new privacy-sensitive sequences are identified, making it generic and evolvable, i.e., it does not become outdated, since public databases can be automatically tracked for updates as they evolve. From the performance viewpoint, we also show that our system can withstand the evolution of NGS and scale out.

The remainder of this paper is organized as follows. In Section 2, we describe our privacy-sensitive detection mechanism, its internal methods to construct a knowledge database, and the implementation details. Section 3 evaluates and shows the efficiency of the proposed method, and Section 4 discusses the completeness of the used knowledge databases. Finally, Sections 5 and 6 present some related work and concluding remarks regarding our techniques and results, respectively.

2. THE DETECTION METHOD

The detection of privacy-sensitive genomic sequences is important both for research and clinical communities. In this section, we present a mechanism to systematically detect privacy-sensitive genomic sequences and its internal methods for creating a knowledge database with this type of sequences.

2.1 Overview

We propose to enhance the NGS production cycle with a mechanism that we call *DNA-Privacy Detector*, which, taking short DNA sequences as input, automatically decides which ones represent privacy-sensitive information. An overview of our detector architecture is shown in Figure 1.

The detector decides based on a database of published signatures or patterns of privacy-sensitive nucleic and amino acid sequences (a *knowledge database*), and forwards input DNA sequences alternatively to the privacy-sensitive output or to the non-sensitive one. The knowledge is defined by statistical heuristics and previous data about a context or a population. Obtaining the privacy-sensitive sequences is not trivial since biological databases provide unharmonized formats and interfaces. Additionally, we need to add all combinations of size s from a sequence to the database, including all their known DNA flanking sequences and mutations (as explained in the next section). The obtained sequences are considered privacy-sensitive and must never be sent to the non-sensitive output stream. Similarly to the signature lists of computer intrusion detection systems (IDSs) [9], the knowledge database can be updated as new patterns are discovered.

Our mechanism can analyze input data sets at any time, but the most powerful and effective configuration would be when input sequences come directly from NGS machines as they are generated. The detector can also be integrated in the NGS machine to automatically detect privacy-sensitive sequences and add a marker with this information in the comment line of a FASTQ entry (the NGS read) [8].

2.2 Privacy-Sensitive Human Genomic Data

Portions of human genomic data are considered privacy-sensitive when they disclose non-consented personal information about an individual. We analyzed recent privacy attacks [16, 18, 28, 34] to obtain the attackers' main goals and the threats they exploit. First, attackers' methods can be divided in two categories depending on their main goal: obtaining the identity of genome donors [16]; or donors' sensitive personal information (e.g., health-related data) [18, 28, 34]. Second, attacks can be divided in three categories regarding the threats they exploit: short tandem repeats [16], disease-related genes [28]; or genomic variations [18, 34]. The studied attacks use only public data (including genomic sequences), however they can also be applied to any sequence obtained from private or external infrastructures.

The next three subsections introduce the threats, describe how they are exploited by attackers, and explain how our detection method, with the proper security techniques, prevents them from succeeding. We opted for a very conservative approach in our solution since we store small sequences of size s in the knowledge database instead of single nucleotides, and detect them as privacy-sensitive sequences independently from their positions in the genome. Configuring the size s and using sequence alignment may interfere on the conservativeness and accuracy of the method.

2.2.1 Short Tandem Repeats

Short tandem repeats (STRs) are small repeated strings comprised of A, C, G and T characters. For instance, the STR called DYS392 is represented by $[\mathrm{TAT}]_n$, and an individual who contains a sequence like *cgacTATTATTATTATcgca* in his DNA will score 4 for DYS392 in his profile. Genetic genealogy profiles are employed in forensic identification, paternity tests, missing people investigations, among others. A profile from paternal lineage is a set of counters of how many times each selected *short tandem repeat* from the Y chromosome (Y-STR) appears in an individual's DNA.

In the United States, a core set of 13 STR markers are being used to generate a nationwide database for forensic identification [6], called FBI Combined DNA Index System (CODIS). Other countries and organizations such as EU, UK, DE and Interpol also selected their sets of core STR markers to identify individuals. There are registers of several known STRs available in public databases, such as the STRBase [29] and the TRDB [12]. These databases have thousands of registered STRs, many more than those few core STRs. Since STRs can also contain mutations, the respective databases must store all known variations.

Attack.

Gymrek *et al.* [16] described an attack that re-identifies participants of the 1000 Genomes project (`http://www.1000genomes.org/`) in early 2013. The attack was based on two facts: surnames are paternally inherited in most human societies; and so are Y-STRs in male individuals [21]. It had two goals: obtain the surname of individuals and triangulate their identity. For surname inference, they profiled Y-STRs of individuals, queried them in public recreational genealogy databases and obtained a list of possible surnames for the profiles in question. Each query contained registers of about 30 known Y-STRs in this case. Authors queried the Y-STR profiles in the YSearch (`http://www.ysearch.org/`) and the SMGF (`http://www.smgf.org/`) databases, and recovered the correct surname in 12% of cases (with 82% of confidence). For triangulating identities, authors combined the obtained surnames with age and state, which were considered public information that did not needed to be suppressed in anonymization processes. A query on U.S. census by year of birth and state results in 60,000 U.S. males in 50% of cases. Aggregating the surname to the query shrinks the result to only 12 males on average. Each surname inference breached the privacy of nearly 16 individuals. Although the result of 12% appears to be unimpressive, it means that from the 1,092 participants of 1000 Genomes project, 131 of them will never recover their privacy, nearly 2,100 participant's relatives had their privacy breached [16], and the disclosed data will continue available for their descendants.

Solution.

We can protect genomic data against this attack with our approach by registering information in the knowledge database about all known Y-STRs, detecting privacy-sensitive sequences containing them, and segregating them from the non-sensitive output stream. We aggregate the following information for each known Y-STR:

1. The STR regular expression, e.g., $[\text{TAT}]_n$ for DYS392.
2. Minimum and maximum number of repetitions observed so far, e.g., 6-17 (for the above STR).
3. All known mutations of the STR [11].
4. All observed left and right flanking sequences, which are commonly found either before or after the STR. E.g., 5'-TAGAGGCAGTCATCGCAGTG-3' is a primer sequence observed before DYS392 and 5'-AAGGAATGGGATTGGTAGGTC-3' after it.

Since an STR is a repetition of a small string, a sequence can start with each different letter from that small string. For example, in the case of DYS392 we have three possibilities for base sequences, with strings starting with TAT, ATT or TTA. Considering sequences with $s = 10$ base pairs, any read with this size matching entirely this entry should be only TATTATTATT, ATTATTATTA or TTATTATTAT, which we call base sequences. For each known mutation we have to create the respective base sequences, i.e., three for each mutation of DYS392. Left and right flanking sequences are concatenated with each base sequence, creating all possible combinations, which we call long sequences. Each long sequence is composed by a left flanking sequence, a base sequence and a right flanking sequence, that finally gives place to small sequences of size s (the size of our entries) created by sliding a window of the same size, which we call privacy-sensitive sequences.

While attacks based on paternal lineage use only Y-STRs, a hypothetical attack employing forensic identification methods may use STRs from all chromosomes, for example, when comparing the victim's genome from her blood sample to a database with identified genomes. Thus, we could also register STRs from all chromosomes in the detector's knowledge by using the same algorithm as for Y-STRs.

2.2.2 Disease-Related Genes

Some portions of a genome, called exons, are translated to RNA and subsequently to amino acid sequences, which finally encode proteins. Proteins provide the functional elements of a biological system, which can have an important role in many human diseases. The presence or absence of specific gene mutations is a tell-tale of the predisposition to or actual contraction of certain diseases. Masking disease-related genes is thus a viable approach to protect the privacy of individuals that had their genomes (re-)identified. This solution focuses mainly on protecting individuals' privacy when they have DNA sequences leaked by unauthorized disclosure (for example, through the first attack [16]) to preclude attackers from obtaining extra information about individuals, namely their health status.

This method is very important in cases where sample donors consent to analyze or store their genome in external infrastructures, but wish to mask or to apply stronger protection on information about some specific diseases. One real example is the Dr. Jim Watson's case. He is a co-discoverer of the double-helix structure of DNA and his complete genome was sequenced and published in 2008 [35]. However Dr. Watson requested the retraction of all information about the *APOE*, a gene associated to the Alzheimer disease, before the public release of his genome.

Attack.

The adversaries, after obtaining an identified genome or portions of it, may learn additional information about the victim's health by using the presence or absence of specific disease-related genes in it. The attack process is similar to existent direct-to-consumer health-related genetic profiling [13], which informs about an individual's probabilities of contracting a disease or any other gene-related information. These tests do not provide diagnosis, but in the wrong hands, they still may cause harm to a victim's reputation or otherwise disadvantage her.

Solution.

Detecting all known disease-related genes is a possible solution after obtaining the complete list of these genes and their sequences. There are dedicated databases that correlates genes and diseases (e.g., GeneCards [30]). These databases allow users to retrieve the names of all currently studied genes, or the few genes related with some specific disease.

Again, the detector is capable to handle data sets obtained from any set of genes present on any database, we use the GeneCards database [30] only as a proof-of-concept. From the estimate of existing $20k$ – $25k$ human genes (the exact number is not yet know [7]), the GeneCards database currently contains the name and data about disease correlation of 19,231 genes. A more conservative approach may detect all known human genes as privacy-sensitive independent if they were already correlated to a disease or not. For example, one may use the OMIM database (`http://omim.org/`) to get the sequences of all known human genes (approximately $23k$). In this work, we retrieve the genes from GeneCards, obtain their accession numbers, and retrieve their sequences from UniProt [32]. For each gene sequence, we must break it into smaller sequences of size $\frac{s}{3}$ (e.g., 10 amino acids) and insert them into the knowledge database.

Masking disease-related genes is sensitive to imputation methods based on linkage disequilibrium between genomic variations. In 2009, Nyholt *et al.* [28] described the method that allowed them to recover Dr. Watson's masked *APOE* status. They respected Dr. Watson's request for *APOE* anonymity in the public manuscript, but the main goal of the authors was to highlight the challenges concerning the privacy and the complexities of informed consents. Note the next method is based on genomic variations, the very same information type used in Nyholt *et al.* [28], and thus we will be able to prevent also that attack from succeeding.

2.2.3 Genomic Variations

Humans are 99.5% genetically similar to one another, however small portions of the remaining 0.5% can uniquely identify who a DNA belongs to [1]. There are numerous studies about *genomic variations* present on individuals. Allele-frequency analysis [31] roughly identifies how common or rare the sequence variants of an individual are, in comparison to a specific population. Genome-wide association studies (GWAS) correlate several traits with these genetic variants common in a population [17].

Attack.

In 2009, Wang *et al.* showed that it is possible to acquire knowledge about targeted individuals from statistical results publicly released by GWAS studies [34]. More precisely, the attacker is assumed to have a blood sample of the victim and genotyped as few as a couple of hundreds of her variations, for example, single nucleotide polymorphisms (SNPs). Then, the attacker goes on to determine the victim's presence in the GWAS' case group, which indicates her contraction of a disease. This result extended another work [18], published one year before, which shows a similar attack to other common techniques employed in genetic studies, for example microarrays. Another study [23] states that obtaining 30 to 80 statistically independent SNP positions is enough to uniquely identify a single person. Finally, the study from Nyholt [28] described a genetic imputation method based on linkage disequilibrium between genomic variations, which allowed them to infer a masked gene of a genome by interpreting neighboring variations present on it.

Solution.

Detecting all known genomic variations of an individual is a feasible approach to prevent such attacks. The knowledge of all genomic variations (SNPs, indels, substitutions, etc) within a population is as complete as the allele frequency (AF) analysis performed in this population. A file in the Variation Call Format (VCF) contains a table with all variations resulting from the AF analysis and the occurrence of them on each individual from the studied population.

We employed the AF analysis of 1000 Genomes project [31], since it is one of the most important AF studies freely available on the internet. It contains 39.7 million variations of 1,092 individuals from 14 populations. We employed it as a significant use case, though it still does not cover 100% of variations of all those samples [31] – there are several extremely rare mutations that are yet to be documented. Note however that our methodology is generic and evolvable: the detector is again not limited to this specific data set since it supports any other AF analysis of any population, independent of coverage of variations. Additionally, our framework could be regularly used in biobanks and hospitals, which could create AF analyses of their private sample collections and use the resulting data to create the knowledge for the detector. It is important to remark that the more complete the sample collection and AF analysis, the better privacy protection our solution provides.

The knowledge construction in this algorithm starts with obtaining a VCF file. For each genomic variation present in this file we extract some data fields about it, for example, the chromosome and position in which it appears, as well as the reference and variation alleles. After extracting this information, we search the chromosome position in the reference genome used by the AF study and concatenate the $s-1$ left flanking nucleotides with the variation allele and the $s-1$ right flanking neighbors. Finally, we add each sequence of s base pairs to the knowledge database.

Detecting and protecting all known genomic variations with this method neutralizes also attacks by genetic imputation using those variations neighboring the masked genes [28], as Dr. Watson's case, allowing the safe use of the previous method: detecting disease-related genes.

2.3 Implementation

After obtaining the knowledge database of nucleic and amino acid sequences to the privacy-sensitivity detector, a second challenge that needs to be addressed is how to implement this component efficiently and effectively. The technical challenge is three-fold:

- Questions such as *"Is GCTAGCTAGCTAGCGGGGC-CCTAGCTAGCT privacy-sensitive?"* cannot be answered immediately as there is no available data label or pattern to detect privacy-sensitive DNA sequences, which differs genomic data from enterprise data (e.g., `SocialSecurityNumber= 123-45-6789`).

- Obtaining the privacy-sensitive sequences is not trivial, and we need also to lookup large amounts of data—including DNA flanking regions and all combinations of size s from a sequence.

- Useful solutions in the genomics area must support the high throughput of NGS machines and scale out, while searching input sequences in the whole database (i.e., tens of GBs).

We explored several data structures to address the mentioned challenges, and Bloom filters [4] presented the most adequate results. Thus, we use a Bloom filter data structure to store the knowledge database of privacy-sensitive sequences. Bloom filters (BF) are probabilistic space-efficient data structures that can be used to represent sets in a compact way. In a nutshell, when performing lookups, BFs return `false` if an entry *definitely does not belong to the set* or `true` if it *probably belongs to the set*. Notice that false positives do not affect the detector's effectiveness (its privacy guarantees), only its efficiency (wrongly classifying a sequence as privacy-sensitive overloads the respective output stream, vs. the cheaper and more available non-sensitive).

Our knowledge database may contain both nucleic sequences (i.e., composed of As, Cs, Gs, and Ts) and amino acid sequences (i.e., composed of letters from the IUPAC nomenclature), but lookups in the detector receive only DNA sequences of size s as input (30 nucleotides in our case). We first search the original DNA sequence, and if it is not found in the Bloom filter, then we translate the 30-bp sequence to the correspondent ten amino acids sequence and lookup again, returning the respective result. If the knowledge contains only DNA sequences, then we only do the first lookup and return. If the knowledge contains only amino acid sequences, then we directly translate each received DNA sequence to the correspondent amino acid sequence, lookup and return.

An interesting aspect of our Bloom filter implementation is that, due to the large number of entries in the knowledge database, we will be using an unconventionally big Bloom filter, sizing several gigabytes. We chose and improved a BF implementation called `Java-LongFastBloomFilter`, which is a bigger and faster Java solution than most BF implementations. Bigger because it uses numbers of `long` type (64-bits) to index the bit set of BFs, while others still use numbers of `int` type (32-bits). This implementation is faster for two reasons. One, it uses a 64-bit Murmur hash, which is one of the fastest non-cryptographic hash functions with good random distribution of regular keys. Two, it has an algorithmic optimization that allows reducing (by configuration) the number of hash keys needed to index an entry by increasing the BF size up to a configurable size (in terms of percentages). We modified two aspects from the mentioned BF implementation: we made the *add* and *contain* methods thread-safe, and added an argument in the constructor to configure the value considered on the performance optimization.

The source-code of our implementation is available under the Apache License (v2.0) in GitHub (`https://github.com/vvcogo/dna-privacy-detector`) together with additional descriptions for reproducibility on the steps performed to obtain the privacy-sensitive data sets to the knowledge database. Additionally, the first author may provide the complete data sets (more than 60GB) used in this paper upon request.

3. EXPERIMENTAL EVALUATION

This section presents results showing that the proposed detection method is efficient in terms of *privacy-sensitivity of genomes*, *memory space required for our Bloom filter*, and *throughput performance.*

3.1 Experimental Setup

The implementations of the previously described methods, henceforward referred as STR-, Gene-, and VAR-based, generate different data sets of sequences. The STR-based method generates two data sets containing sequences of 30 nucleic acids each, namely: the Y-STR and All-STR, which contain short tandem repeats from Y chromosome and from all chromosomes respectively. The Gene-based method contains one data set (All-Gene) of sequences composed of ten amino acids each, where all published disease-related genes are added to the knowledge database. The VAR-based method generates one data set (All-VAR) of sequences also composed of 30 nucleic acids each, where it contains all genomic variations available in the AF analysis of 1000 Genomes project [31]. Table 1 contains the number of entries from each data set to the privacy-sensitivity detector, and the respective size as a text-based input file with one entry per line (considering one byte per character, e.g., UTF-8).

The knowledge data is a set of small sequence entries comprised of 30 base pairs or ten amino acids each. The number of entries in Table 1 can be directly translated to the amount of storage space needed for them. For example, if each base pair requires one byte to be stored, the Y-STR data set would require $1 \times 30 \times 0.5 \times 10^6$ bytes, or 15MB. The Gene-based would require $1 \times 10 \times 8.7 \times 10^6$ bytes, or 87MB. The All-Together would require $1 \times (30 \times 1169 + 10 \times 8.7) \times 10^6$ bytes, or 35.1GB.

We have randomly selected ten donors identifiers from the 1000 Genomes project, which compose the input data to our experiments. The resulting donors were those identified by the numbers NA19788, HG00173, NA20810, HG00339, HG00619, NA20339, HG00475, HG01390, NA19449, and NA12546. We describe the steps performed to obtain the entire genome sequences from these donors in the detector's GitHub page. Each input genome of 3GB is the contiguous genomic sequence in the FASTA format, which is the resulting data from assembling a 30× coverage sequencing data (e.g., FASTQ files with 180GB).

Our experimental environment is one physical machine that runs all components of our system architecture. This machine is a Dell PowerEdge R410 server, equipped with two Intel Xeon E5520 (quad-core, HT, 2.27Ghz), 32GB of RAM and a hard disk with 146GB (15k RPM). The operating system is an Ubuntu Server Lucid Lynx (10.04 LTS, 64-bits), running with a kernel 2.6.32-21-server, and the Java version is the 1.7.0_25 (64-bits).

3.2 Privacy-Sensitivity of Human Genomes

Our first analysis calculates how much of an assembled genome is considered privacy-sensitive for each knowledge data set and false positive rate. We picked the ten mentioned genomes to execute the test, where each genome was split in approximately 103 million sequences of 30-bp each, the equivalent to 3GB. Figure 3(a) shows the average percentages of sensitive entries from these genomes for different false positive rates in our Bloom filter.

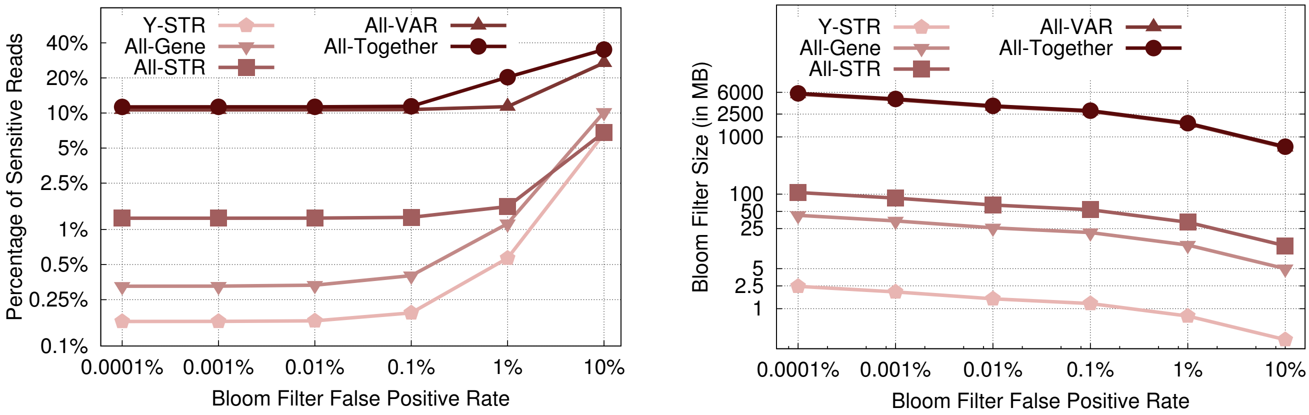
There is a minimal percentage of privacy-sensitive reads that is independent of the false positive rate of the Bloom filter, which is in the similar results from probabilities 10^{-6} to 10^{-3}. It means that one needs to enforce strong security premises in at least 0.16% (4.8 MB—without compression) of each assembled genome if using the Y-STR knowledge database, and in at least 11.3% (345 MB) if the All-Together data set is used instead. Segregating 11.3% of each human genome to the privacy-sensitive portion leads to the reduction of almost 90% of data that must be maintained under strong security premises. Due to the high similarity in human genome sequences (more than 99.5%), increasing the number of input samples will not affect significantly the obtained privacy-sensitive percentage.

3.3 Space Efficiency

This second analysis estimates the size of the knowledge database in main memory when using the different data sets from Table 1. Since we use a Bloom filter, theoretically, the filter size depends only on the expected number of entries and the expected false positive rate [4]. The expected number of entries is a constant for each configuration/data set (from All-Together to Y-STR), which appears in Table 1. The false positive rate can be defined by the system administrator to fit the Bloom filter size in the server's memory

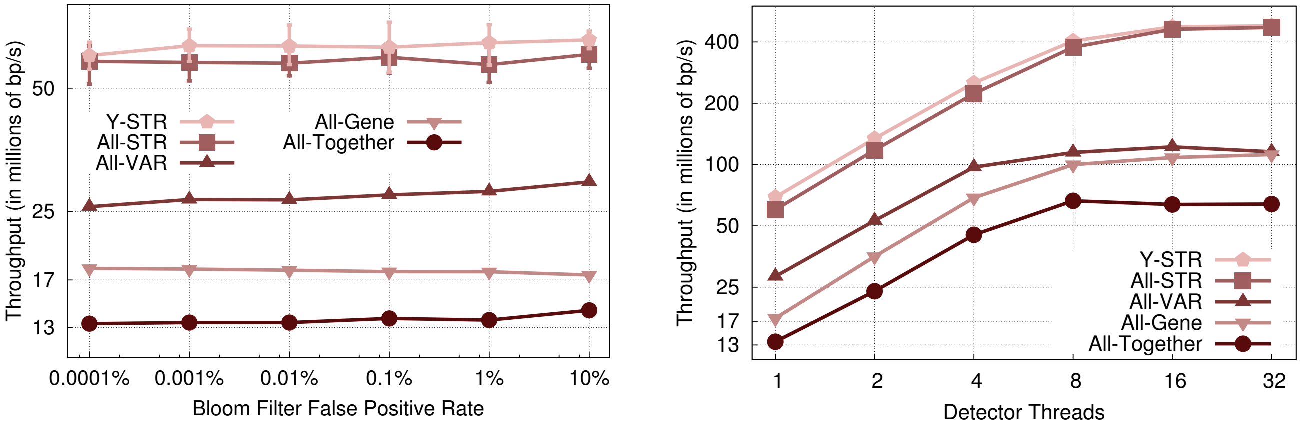
Data set	Method	Acid type	Entry size	Number of entries	Text file size
Y-STR	STR-based	Nucleic	30	0.5 M	15MB
All-STR	STR-based	Nucleic	30	22 M	660MB
All-Gene	Gene-based	Amino	10	8.7 M	87MB
All-VAR	VAR-based	Nucleic	30	1,147 M	34.4GB
All-Together	All	Both	10 and 30	1,178 M	35.1GB

Table 1: The different privacy-sensitive data sets considered in this study. Our partitioning method uses them as a knowledge database to decide if a DNA segment is privacy-sensitive or not. For each dataset, we present the approach used to obtain it, the acid type, size and number of entries, and the dataset size when using a text-based representation.



(a) (b)

Figure 3: (a) The percentage of privacy-sensitive reads for different false positive rates and knowledge data sets. (b) Bloom filter size for different false positive rates and knowledge data sets. Both axes from (a) and (b) are in logarithmic scale.



(a) Single-core performance (b) Multi-core performance

Figure 4: The throughput of our method using Bloom filters. (a) Considering different false-positive rates (single-core) and (b) multiple threads (with 0.1% false-positive rate). Both axes from (a) and (b) are in logarithmic scale.

capacity. Figure 3(b) presents the resulting database size (in megabytes) based on these two properties.

The Bloom filter size using input from STR- or Gene-based methods is 50- to 130-fold smaller than the size if using the VAR-based method. The largest size corresponds to the use of the largest data set (All-Together) and a false positive rate of 10^{-6} (one in a million), which leads to a data structure as big as 5.6GB. This size easily fits on the main memory of current commodity servers, and is 6× smaller than the original size of the entries of this data structure (35.1GB).

3.4 Throughput Performance

Our final experiment aims at identifying how many base pairs per second our detector is able to analyze. Figure 4(a) shows the results considering different false positive rates in a single core deployment, whereas Figure 4(b) considers different number of threads in our multi-core system.

As expected, the higher the false positive rate, the higher the throughput (Figure 4(a)). This happens because higher rates lead to smaller Bloom filters, which require less hash operations to test whether or not a sequence belongs to the set. There is a difference in terms of performance between methods that use knowledge about nucleic acids and those that use amino acid sequences as entries. This occurs because we need to translate each 30-bp sequences from the testing genome to 10-aa sequences when the knowledge database contains entries composed of amino acids.

Notably, even with our largest and most complete data set (All-Together with a false positive probability of 10^{-6}), the detector is still able to analyze more than 13.2 million bp (i.e., 0.44 million operations) per second *with a single core*. Additionally, the detector evaluates 60 million bp (i.e., two million operations) per second using the other data set of interest (Y-STR). Considering that high-throughput NGS machines produce 0.3 million bp/sec [24], our solution works 44× to 200× faster. It means the detector could be integrated directly in NGS machines with the addition of minimal hardware. In this way, the machine could generate different FASTQ files containing either privacy- or non-privacy-sensitive reads, or, just add one character to the comment line of the FASTQ entry with its privacy sensitivity.

Obtaining higher throughput is possible when parallelizing the detection process. Figure 4(b) shows a scalability test of up to 32 threads in our test machine, which is equipped with two quad-core processors with hyper-threading, i.e., 16 hardware threads. This scalability test considers a false positive rate of 0.1% (i.e., 10^{-3}), which we consider to be the sweet spot of our design since it minimizes the BF size at the same time that it maintains similar percentages of sensitive reads (see Figures 3(a) and 3(b)). The throughput scales up to 480 million bp per second when testing only nucleic acids, up to 110 million for amino acids, and up to 66 million bp per second when using a knowledge database with both acids. This corresponds to a speedup of up to eight times, which is the number of cores in use. Our solution works 200× to 1600× faster than current high-throughput NGS machines (0.3 million bp/sec) [24]. Therefore, the detector is not a bottleneck for current or near-future machines.

4. COMPLETENESS OF THE METHOD

The approach used in our detection method adapts the blacklisting from passive knowledge-based intrusion detection systems (IDS) [9], where network message sequences are continuously filtered by comparison with entries in a database of known attack signatures. We refer our knowledge database as a blacklist because it stores all known privacy-sensitive sequences (dangerous in the sense of IDSs) that we want to prevent from being sent to the non-sensitive output stream.

The detector's effectiveness is complete for all "signatures" existent in the database, but only for them. Sequences made vulnerable by new, previously unknown attacks, will not be recognized without updating the knowledge database, pretty much like what happens in IDSs with the notorious zero-day vulnerabilities. In the following, we discuss the risks and implications of new discoveries related with the three knowledge sources employed by our technique.

New STR Sequences.

As discussed before, STR is a prime method to identify individuals in forensic analysis. Our detector uses the database containing all know STR sequences, but it might fail to detect a sequence containing a yet to be discovered STR. To understand the window of vulnerability posed by newly appeared STRs, we analyzed the annual evolution of the number of entries in TRDB [12], the main database for short tandem repeats. From 2003 to 2014, this database evolved from $237k$ to $238k$ STRs. This means that in 11 years, the TRDB registered $1k$ novel STRs, which represent a growth of only 0.42% in its entries. This suggests finding many novel STRs is unlikely, even with the explosive growth on the number of whole genome studies due to the introduction of NGS methods. Additionally, the genetic genealogy databases employed in the re-identification attack [16] use a static small set of STRs (i.e., few dozens) to profile individuals. They probably will not increase the number of profiled STRs since it would require the reanalysis of all participants. Finally, if an attacker discovers novel STRs, he will be able to harm victims only after these databases reanalyze the victim's DNA or profile victim's relatives with the newly identified STRs, which is also unlikely.

New Disease-Related Genes.

The main risk of detecting privacy-sensitive sequences using the disease-related genes knowledge database is that it might not identify genes that were not yet linked to some disease. As with STRs, we also analyzed the annual evolution on the number of entries in the GeneCards database [30] (the database we used in this paper to obtain the known disease-related genes). Our analysis shows an impressive evolution from $3k$ disease-related genes in 2005, to more than $19k$ in 2014, which corresponds to a growth of 83.3% in nine years (mostly due to NGS). Given that researchers estimate that humans have 20–$25k$ genes in their genomes [7], we have that there are at most 4–23% of genes to be correlated with diseases yet (and thus are not included in our detector database). This means that a database such as GeneCards can only grow up to $25k$ entries. Additionally, the remaining genes do not determine alone the contraction of a disease, may have no relation with any disease, or are related with rare diseases that affect very few people. Finally, assuming discoveries directly translate into novel privacy-sensitive sequences is a misconception. Most discoveries correlate diseases with already known genes and genomic variations, not new ones. Our method detects a sequence as privacy-sensitive independently on how many diseases it is correlated with (one correlation is enough).

New Rare Genomic Variations.

The sequences generated by identifying variations are as complete as the size and coverage of the population considered in the allele frequency study over that population. Currently, there are already countries storing and conducting studies on the genome of its whole population [19]. A detector using a database like this will be able to detect most (if not all) privacy-sensitive sequences, since the VAR-based detection is by far the method that generates more entries in our detection database (see Table 1). As the size and representativeness of the population on the database increase, the detector accuracy increases, accounting for the exact frequency of a given variant in a population, being thus capable of detecting any privacy-sensitive sequence.

These discussions suggest that there is already a large body of knowledge about the privacy sensitiveness of the human genome. More specifically, it is possible to have a *reasonably complete* privacy detector since (1) the rate of discovery of new STRs was extremely low in the last decade—suggesting we probably know most of them, (2) disease-related genes are being discovered at an incredible pace—exhausting the maximum number of human genes, and (3) rare variations can be accurately covered by increasing population samples in allele frequency studies. This vouches for the usefulness of an evolvable tool that can be used together with standard security techniques to dramatically improve the *status quo* of robustness of genomic data repositories, much in the lines of what intrusion detection has achieved in the protection of IT data servers.

5. RELATED WORK

Detecting privacy-sensitive genomic data as soon as it is generated is a long-term ambition from the research and clinical communities [10, 15]. Recent works on privacy-preserving genome processing have been advocating the partitioning of genomic data, but assume this is done manually [2], or by a tool that is out of their scope [20]. To the best of our knowledge, the present paper is the first work implementing an automatic method to this task.

For example, Sedic [20] is a privacy-aware platform that modifies the Apache Hadoop MapReduce to work in a hybrid cloud environment. They compute all privacy-sensitive data in a private cloud or cluster, and the non-sensitive portion in a public cloud provider. However, they forward the burden of labeling sensitive data to the user-side, and do not propose any solution on how to do that in genomic data. Our mechanisms can automatically detect and label privacy-sensitive genomic data, complementing Sedic's approach and increasing the appeal of their solution.

In another work [2], the authors propose a privacy-preserving method to process mapped short reads in a scenario of personalized medicine. They encrypt all genomic data before storing it, mask a few genomic variations based on donor's preferences, and limit the access of medical units to portions of these aligned short reads. There are at least three main issues of their approach. First, they do not provide an automatic selection of the genomic variations that will be masked. Second, they suggest the selection of only a few variations to be masked, while this protection is insufficient to preserve donor's privacy completely [23]. Third, their masking mechanism roughly filters out the selected genomic variations, which makes several medical analyses impractical. Our approach contrasts to theirs since we allow the automatic detection and masking of all known genomic variations and many other privacy-sensitive information from human genomes (e.g., disease-related genes). Furthermore, we increase the usefulness of data by allowing users to improve the protection and control of the privacy-sensitive portion of genomes (e.g., by maintaining it in sites with increased control and security premises) rather than throwing it away.

Notice that manually marking parts of the genomic data as private (as in previous works) *does not provide any additional benefit* to using our detector. This happens because our algorithm uses all the knowledge available now and is insusceptible to false negatives. On the contrary, doing such classification manually may lead to mistakes and misguided decisions that may cause leaks on already known privacy-sensitive genomic data.

6. CONCLUSION

We described a novel efficient solution to detect privacy-sensitive DNA sequences automatically from an input stream, using as reference a knowledge database of privacy-sensitive sequences. The assessment of the privacy-sensitivity detector demonstrated its feasibility to address the challenges imposed by some recently published attacks and to establish the basis for future developments in this field. More specifically, the idea of having a detection method for privacy-sensitive DNA is important for four main reasons.

First, the severity of threats to privacy of genomic information will be amplified by the explosive growth in DNA sequences resulting from NGS, bound to be stored and analyzed in external multi-tenant infrastructures [25]. If little is done, a severe leak may reverse the public opinion trend to make DNA sequences public or shareable and hurt genomic studies, or even harden state laws for genomic data protection. Researchers have been alerting about the inevitability of a major leak of genome information in the near future, and that we should start defining the steps that need to be taken to avoid a public outcry a genome breach might incite [5]. We believe that the systematic detection, with the proper protection, of DNA sequences as they are generated can dramatically reduce the risk of such leaks, and thus is an important step towards a sound semantic data ecology by improving privacy protection whilst enabling adequate mechanisms to promote trusted and secure data sharing [33].

Second, we demonstrated the robustness and high performance of the three methods with which we built the knowledge database of privacy-sensitive sequences: short tandem repeats, disease-related genes and genomic variations present on individuals. These methods were shown to be enough for avoiding all known attacks we are aware of [16, 18, 28, 34].

Third, our solution is reproducible and evolving: the same database may be re-used with different data sets, and the knowledge database can continually and transparently be updated as new privacy-sensitive sequences are revealed, without affecting the workflow. In essence, a protection ecosystem built in accordance to the principles proposed here would exhibit similar effectiveness as continuously updatable industry from the intrusion detection systems.

Finally, we have shown that the privacy-sensitivity detector can easily be fitted inline with the NGS production cycle and fulfill the systematic detection promise, exhibiting adequate performance and scalability, by using Bloom filters. However, we note again that besides the effectiveness of our solution, the proposed computational framework and architecture can be reused and evolved with new privacy-sensitive sequences being identified.

Despite these motivating results, the work presented here by no means makes standard security methods such as access control and encryption obsolete, neither does it settle the issue of privacy protection of genomic information. Future directions encompass further exploring privacy in human genomes to improve the global knowledge about privacy-sensitive sequences, exploring cases that might benefit from the proposed detector method (e.g., hybrid storage using single and multi-clouds [3, 26]) and using different techniques (e.g., text mining [22]) to recognize novel privacy-sensitive sequences directly from the scientific literature even before they are added to existing databases.

7. ACKNOWLEDGMENTS

Authors warmly thank Ulf Leser (HU-Berlin), Margarida Carvalho (ULisboa), and Andreia Fonseca (ULisboa) for their criticisms and suggestions. This work was partially supported by Fundação para a Ciência e a Tecnologia (FCT), through the LaSIGE project (PEst-OE/EEI/UI0408/2014), and by EU FP7, through the BiobankCloud project (ICT-317871).

8. REFERENCES

[1] E. Ayday, E. De Cristofaro, J.-P. Hubaux, and G. Tsudik. Whole genome sequencing: Revolutionary medicine or privacy nightmare? *Computer*, (2):58–66, 2015.

[2] E. Ayday, J. L. Raisaro, U. Hengartner, A. Molyneaux, and J.-P. Hubaux. Privacy-preserving processing of raw genomic data. In *Proc. of the DPM 2014*, pages 133–147. Springer, 2014.

[3] A. Bessani et al. SCFS: a shared cloud-backed file system. In *USENIX ATC'14*, pages 169–180. USENIX Association, June 2014.

[4] B. H. Bloom. Space/time trade-offs in hash coding with allowable errors. *Commun. ACM*, 13(7):422–426, 1970.

[5] S. E. Brenner. Be prepared for the big genome leak. *Nature*, 498(7453):139–139, 2013.

[6] J. M. Butler. Genetics and genomics of core short tandem repeat loci used in human identity testing. *J. Forensic Sci.*, 51(2):253–265, 2006.

[7] M. e. Clamp. Distinguishing protein-coding and noncoding genes in the human genome. *PNAS*, 104(49):19428–19433, 2007.

[8] P. Cock et al. The Sanger FASTQ file format for sequences with quality scores, and the Solexa/Illumina FASTQ variants. *Nucleic Acids Res.*, 38(6):1767–1771, 2010.

[9] H. Debar, M. Dacier, and A. Wespi. Towards a taxonomy of intrusion-detection systems. *Computer Networks*, 31(8):805–822, 1999.

[10] Y. Erlich and A. Narayanan. Routes for breaching and protecting genetic privacy. *Nature Reviews Genetics*, 15(6):409–421, 2014.

[11] H. Fan and J.-Y. Chu. A brief review of short tandem repeat mutation. *Genomics, Proteomics & Bioinformatics*, 5(1):7–14, 2007.

[12] Y. Gelfand, A. Rodriguez, and G. Benson. TRDB – the tandem repeats database. *Nucleic Acids Res.*, 35(suppl 1):D80–D87, 2007.

[13] L. Goldsmith, L. Jackson, A. O'Connor, and H. Skirton. Direct-to-consumer genomic testing: systematic review of the literature on user perspectives. *Eur. J. Hum. Genet.*, 20(8):811–816, 2012.

[14] M. Goodman and A. Hessel. The bio-crime prophecy: DNA hacking the biggest opportunity since cyber attacks. *Wired-UK*, 2013.

[15] D. Greenbaum, A. Sboner, X. J. Mu, and M. Gerstein. Genomics and privacy: Implications of the new reality of closed data for the field. *PLoS Computational Biology*, 7(12):e1002278, 2011.

[16] M. Gymrek, A. L. McGuire, D. Golan, E. Halperin, and Y. Erlich. Identifying personal genomes by surname inference. *Science*, 339(6117):321–324, 2013.

[17] J. N. Hirschhorn and M. J. Daly. Genome-wide association studies for common diseases and complex traits. *Nature Reviews Genetics*, 6(2):95–108, 2005.

[18] N. Homer et al. Resolving individuals contributing trace amounts of DNA to highly complex mixtures using high-density SNP genotyping microarrays. *PLoS Genet.*, 4(8):e1000167, 2008.

[19] R. K. Jakobsen. Sequencing the genome of an entire population. *ScienceNordic*, June 2012.

[20] K. Zhang *et al.* Sedic: privacy-aware data intensive computing on hybrid clouds. In *Proc. of the CCS'11*, pages 515–526, 2011.

[21] T. E. King and M. A. Jobling. What's in a name? Y chromosomes, surnames and the genetic genealogy revolution. *Trends Genet.*, 25(8):351–360, 2009.

[22] A. Lamurias, J. D. Ferreira, and F. M. Couto. Identifying interactions between chemical entities in biomedical text. *J. Integr. Bioinform.*, 11(3):247, 2014.

[23] Z. Lin, A. B. Owen, and R. B. Altman. Genomic research and human subject privacy. *Science*, 305(5681):183, 2004.

[24] L. Liu et al. Comparison of next-generation sequencing systems. *J. Biomed. Biotechnol.*, 2012, 2012.

[25] V. Marx. Biology: The big challenges of big data. *Nature*, 498(7453):255–260, 2013.

[26] V. Marx. Genomics in the clouds. *Nature methods*, 10(10):941–945, 2013.

[27] M. Naveed et al. Privacy in the genomic era. *ACM Comput. Surv*, 48(1), 2015.

[28] D. R. Nyholt, C.-E. Yu, and P. M. Visscher. On Jim Watson's APoE status: genetic information is hard to hide. *Eur. J. Hum. Genet.*, 17:147–149, 2009.

[29] C. M. Ruitberg, D. J. Reeder, and J. M. Butler. STRBase: a short tandem repeat DNA database for the human identity testing community. *Nucleic Acids Res.*, 29(1):320–322, 2001.

[30] M. Safran et al. GeneCards version 3: the human gene integrator. *Database*, 2010:baq020, 2010.

[31] The 1000 Genomes Project Consortium. An integrated map of genetic variation from 1,092 human genomes. *Nature*, 491:1, 2012.

[32] UniProt Consortium. The universal protein resource (UniProt). *Nucleic Acids Res.*, 36(suppl 1):D190–D195, 2008.

[33] P. E. Verissimo and A. Bessani. E-biobanking: What have you done to my cell samples? *IEEE Security&Privacy*, 11(6):62–65, 2013.

[34] R. Wang, Y. F. Li, X. Wang, H. Tang, and X. Zhou. Learning your identity and disease from research papers: Information leaks in genome wide association study. In *Proc. of the CCS'09*, pages 534–544, 2009.

[35] D. A. Wheeler et al. The complete genome of an individual by massively parallel DNA sequencing. *Nature*, 452(7189):872–876, 2008.