Reconciling Utility with Privacy in Genomics

Mathias Humbert Erman Ayday
Jean-Pierre Hubaux
Laboratory for Communications and Applications
EPFL, Lausanne, Switzerland
firstname.lastname@epfl.ch

Amalio Telenti
Institute of Microbiology
University Hospital of Lausanne
Lausanne, Switzerland
amalio.telenti@chuv.ch

ABSTRACT

Direct-to-consumer genetics makes it possible for everyone to obtain their genome sequences. In order to contribute to medical research, a growing number of people publish their genomic data on the Web, sometimes under their real identity. However, this is at odds not only with their own privacy but also with the privacy of their relatives. The genomes of relatives being highly correlated, some family members might be opposed to revealing any of the family's genomic data. In this paper, we study the trade-off between utility and privacy in genomics. We focus on the most relevant kind of variants, namely single nucleotide polymorphisms (SNPs). We take into account the fact that the SNPs of an individual contain information about the SNPs of his family members and that SNPs are correlated with each other. Furthermore, we consider that SNPs may have different significance for medical research and different levels of sensitivity for individuals. We propose an obfuscation mechanism that enables public availability of genomic data for research and, at the same time, protects the genomic privacy of the individuals in a family. Our genomic-privacy preserving mechanism relies upon combinatorial optimization and graphical models to optimize utility and meet privacy requirements. We also present an extension of the optimization algorithm to cope with non-linear constraints induced by the correlations between SNPs. Our results on real data show that the proposed technique maximizes the utility for genomic research, and satisfies family members' privacy constraints.

Categories and Subject Descriptors

C.2.0 [**Computer-Communication Networks**]: General—*Security and protection (e.g., firewalls)*; J.3 [**Life and Medical Sciences**]: *Biology and genetics*; K.4.1 [**Computer and Society**]: Public Policy Issues—*Privacy*

Keywords

Genomic Privacy; Obfuscation; Optimization

1. INTRODUCTION

Genomic research has revolutionized our understanding of medicine: the "one size fits all" approach has already left its place to "personalized medicine" for the treatment of many diseases, for which genetic factors of the individuals play an important role. This is also paving the way to early diagnosis of many serious diseases. The association of genetic factors with diseases and treatments is only possible via large-scale genome-wide association studies (GWAS) that require the availability of a considerably high number of human genomes. Computers are at the core of this endeavor, because (i) high computing power is required to interpret genomic data [11], (ii) hand-held devices are used to visualize this data, and (iii) more and more people tend to upload genomic data (and more generally health-related data) on public websites (e.g, OpenSNP.org and personalgenomes.org). Computing systems facilitate data access and processing for legitimate usage, but sometimes also for purposes that were initially unintended, thus raising privacy concerns.

Genomic data carries much sensitive information about its owner. By analyzing the DNA of an individual, it is now possible to learn about his disease predispositions (e.g., for Alzheimer's or Parkinson's), ancestries, and physical attributes [18]. The threat to genomic privacy is magnified by the fact that a person's genome is correlated to his family members' genomes, leading to interdependent privacy risks. Kin genomic privacy was popularized by the story of Henrietta Lacks whose cells were sequenced and DNA put online without the consent of her descendants [1]. After complaints from the family, essentially due to privacy concerns, Henrietta's genome was taken offline, and in 2013, the National Institutes of Health (NIH) came to an agreement with the Lacks family that gave them some control over her genome [2]. Even though this agreement enables the genomic researchers to use Henrietta's genome again, it also draws attention to the lack of techniques for balancing the benefits of genomic research with personal and kin genomic privacy risks. Richard Sharp, the director of biomedical ethics at the Mayo Clinic, warned that the agreement was only a "one-off solution" rather than a broad policy that addresses the tension between research and relatives' privacy, and he added that a "new policy" was absolutely needed [2].

Anonymization was the first countermeasure proposed to protect genomic privacy, but in many different studies it was proven inadequate [25, 26, 43]. Another protection mechanism is to add noise to aggregate statistical results (to satisfy differential privacy) [19, 30], at the cost of reduced accuracy. The last option proposed in the literature is to rely on cryptographic techniques [4, 6]. Even though these are proven to be effective for using genomic data in healthcare [4, 12], computational complexity becomes very high when it comes to conducting statistical tests on large numbers of encrypted genomes for genomic research [32].

In this work, we present a genomic-privacy preserving mechanism (GPPM) to reconcile people's willingness to share their genomes (e.g. to help research[1]) with privacy. Our GPPM acts at the individual data level, not at the aggregate data (or statistical) level like in [19,30]. Focusing on the most relevant type of variants (the SNPs), we study the trade-off between the usefulness of disclosed SNPs (utility) and genomic privacy. We consider an individual who wants to share his genome, yet who is concerned about the subsequent privacy risks for himself and his family. Thus, we design a system that maximizes the disclosure utility while not exceeding a certain level of privacy loss within a family, considering (i) kin genomic privacy, (ii) personal privacy preferences (of the family members), (iii) privacy sensitivities of the SNPs, (iv) correlations between SNPs, and (v) research utility of the SNPs. Our GPPM can automatically evaluate the privacy risks of all the family members and decide which SNPs to disclose. To achieve this goal, it relies on probabilistic graphical models and combinatorial optimization. Our results indicate that, considering the current data model, genomic privacy of an entire family can be protected while revealing an appropriate subset of genomic data. Our contributions can be summarized as follows:

- We propose a GPPM for enabling genomic research while protecting personal and kin genomic privacy.
- Considering the genomic data model, our obfuscation mechanism maximizes the utility, and meets all the privacy constraints of a given family.
- Using combinatorial optimization, we first compute the optimal solution without considering correlations between SNPs, and then we extend the algorithm to deal with non-linear constraints induced by these correlations.

The rest of the paper is organized as follows. In the next section, we provide a brief background on genomics, describe the adversary model and present our genomic-privacy quantification framework. In Section 3, we present our GPPM in detail. Next, in Section 4, we evaluate its performance. In Section 5, we summarize the related work, before discussing the limitations and future work in Section 6.

2. PRELIMINARIES

In this section, we briefly introduce the revelant genomic background, the adversary model and our quantification framework for measuring genomic privacy.

2.1 Genomic Background

Genetic information is encoded on the DNA as a sequence of nucleotides, represented by four letters A, T, G, and C. For every two randomly selected individuals, approximately 99.9% of their DNA is similar, and the remainder is the genetic variation. Single nucleotide polymorphism (SNP) is the most common DNA variation in human population. A SNP occurs when a nucleotide at a specific position on the DNA varies between individuals of a given population. Almost all common SNPs have two different nucleotides (called alleles): (i) the major allele is the most frequently observed nucleotide, and (ii) the minor allele is the rare nucleotide. From here on, we represent the major allele as B for a SNP position, and the minor allele as b (both B and b are from the set $\{A, T, G, C\}$).

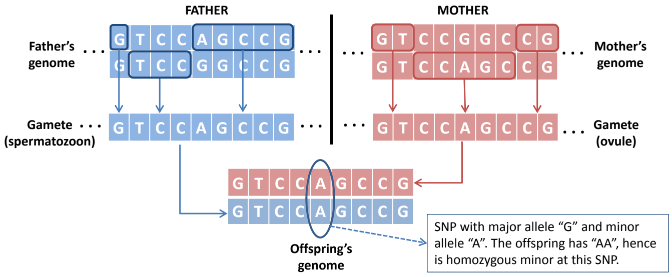


Figure 1: Principle of human reproduction. Each parent produces gametes that are derived from his or her genome. The offspring's genome is the combination of these two gametes. As an example, the circled SNP contains a homozygous-minor SNP (bb) with major allele G and minor allele A.

The risk of an individual developing certain diseases can be computed from his SNPs [31] (and from his clinical and environmental factors). Hence, SNPs carry privacy-sensitive information about individuals (and their family members). In order to find the associations of the SNPs with the diseases, genome-wide association studies (GWAS) are required on a very large number of genomes.

For each SNP position, a child inherits one allele from his mother and one from his father. Each allele of a parent is inherited by a child with equal probability of 0.5. As each SNP position contains two nucleotides (one inherited from the mother and one from the father), the content of a SNP position can be in one of the following states: (i) BB (*homozygous-major* genotype), if an individual receives the major allele from both parents; (ii) Bb (*heterozygous* genotype), if he receives a different allele from each parent (one minor and one major); or (iii) bb (*homozygous-minor* genotype), if he inherits the same minor allele from both parents. We illustrate a SNP with homozygous-minor genotype in Fig. 1 (on the offspring's genome). In this example, we illustrate a SNP (*rs11200638* on the *HTRA1* gene) that increases the risk for *macular degeneration* (this disease is the leading cause of blindness).[2] This SNP (with major allele G and minor allele A) increases the disease risk by 2.243 times, when it is in heterozygous form (i.e., (G, A), as in the father and mother), and by 8.669 times when it is in homozygous-minor form (i.e., (A, A), as in the offspring).[3] In the same figure, we also illustrate how a child inherits his SNPs from his parents.

We represent the content of a SNP position as x_j^i for SNP j at individual i, where $x_j^i \in \{BB, Bb, bb\}$. For simplicity of presentation, we denote BB as 0, Bb as 1, and bb as 2 (i.e., $x_j^i \in \{0, 1, 2\}$). Furthermore, SNPs on the DNA have

[1] http://opensnp.wordpress.com/2011/11/17/first-results-of-the-survey-on-sharing-genetic-information/

[2] http://snpedia.com/index.php/Rs11200638

[3] Risk increase is with respect to the homozygous-major genotype (G, G).

correlations with each other. Linkage disequilibrium (LD) is a correlation that appears between pairs of SNP positions in the whole genome, due to the population's genetic history.

2.2 Adversary Model

The objective of the adversary is to infer the values of hidden SNPs of one or more members of a given family. To do so, he relies on some background knowledge, essentially the minor allele frequencies (MAFs) of the SNPs, the pairwise LD values between the SNPs, and the basic rules of Mendelian inheritance. Note that LD values are only expressed between pairs of SNPs by geneticists, thus only pairwise correlations are available to the adversary. This knowledge represents the genomic data model available to the adversary. Second, the adversary observes a subset of the SNPs of the family members, typically those who have disclosed their genomic data (or part of them).

Thousands of genomes are already publicly available on the Internet (e.g., 1000 Genomes Project, or OpenSNP). Even though these genomes are generally anonymized, it has been shown that an attacker can re-identify a genome's owner by (i) linking the owner's demographics to publicly available records such as voter lists [43], or (ii) searching part of the owner's DNA, such as his Y-chromosome[4] or her mtDNA[5], in online databases (e.g., Ysearch or Family Tree DNA) and linking DNA to last names [25]. Once the genome's owner is de-anonymized, the attacker can easily rely on genealogy websites[6] or online social networks (e.g., Facebook or 23andMe) in order to obtain the (partial) family tree of the genome's owner. As shown in Appendix A, multiple owners from the same family can also be identified, reinforcing the attacker's inference power. Thus, in this paper, we assume that the attacker has de-anonymized the genome(s) owner(s) and knows (part of) the familial relationships.

The attacker's ultimate goal is then formally defined as computing posterior marginal probabilities of unknown variables from the global posterior probability distribution $P(\mathbf{X_H}|\mathbf{X_O},\mathcal{K})$, where $\mathbf{X_H}$ represents the set of hidden SNPs, $\mathbf{X_O}$ the set of observed SNPs, and $\mathcal{K}$ the background knowledge or data model. We show how this inference attack can be efficiently carried out in the next subsection, and present a real-world example in Appendix A. Note that this is currently the best known *attribute-inference attack* [27].

2.3 Genomic-Privacy Metrics

We briefly summarize our framework for quantifying kin genomic privacy. In order to quantify the genomic privacy of an individual, we mimic an attacker willing to infer some *targeted SNPs* of a family member (or multiple family members) of a *targeted family*. The resources of such an attacker are the observed genomic data (of members of the targeted family), the family tree, and public genomic knowledge (background knowledge). We define $\mathbf{F}$ to be the set of family members in the *targeted family* and $\mathbf{S}$ to be the set of SNP IDs (i.e., positions on the DNA sequence), where $|\mathbf{F}| = n$ and $|\mathbf{S}| = m$. We also let x_j^i be the value of SNP j ($j \in \mathbf{S}$) for individual i ($i \in \mathbf{F}$), where $x_j^i \in \{0, 1, 2\}$ (as introduced in Section 2.1). Furthermore, we let $\mathbf{X}$ be the set of SNPs of all family members, hence $|\mathbf{X}| = n \cdot m$. Some elements of $\mathbf{X}$ might be observed by the adversary (the observed genomic data of one or more family members) and others might be hidden. We denote the set of SNPs from $\mathbf{X}$ whose values are hidden as $\mathbf{X_H}$, and the set of SNPs from $\mathbf{X}$ whose values are observed by the adversary as $\mathbf{X_O}$.

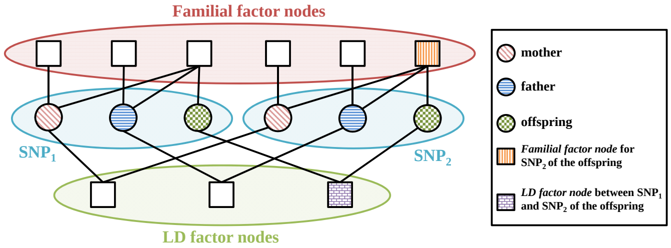

Familial factor nodes
SNP$_1$
SNP$_2$
LD factor nodes
mother
father
offspring
Familial factor node for SNP$_2$ of the offspring
LD factor node between SNP$_1$ and SNP$_2$ of the offspring

Figure 2: Factor graph representation of genomic data (i.e. SNPs), familial relationships, and linkage disequilibrium (LD) values between the SNPs.

The attacker carries out a reconstruction attack to infer $\mathbf{X_H}$ by relying on the background knowledge $\mathcal{K}$ (or data model) and on his observation $\mathbf{X_O}$. To efficiently infer hidden SNPs, the attacker can make use of the belief propagation algorithm [35, 38], a message-passing algorithm for performing inference on graphical models. The belief propagation algorithm can be run on a factor graph, a bipartite graph containing two sets of nodes (corresponding to variables and factors) and edges connecting these two sets. In this particular attack, the *variable nodes* on the factor graph are the SNPs of the family members. Furthermore, there are two types of *factor nodes*: (i) the *familial factor nodes*, representing the familial relationships and reproduction rules, and (ii) the *LD factor nodes*, representing the LD relationships between the SNPs. Such a factor graph is illustrated in Fig. 2 for a nuclear family. Message passing takes place between the variable nodes and the factor nodes, possibly during multiple iterations if the factor graph contains loops (which is the case in our inference problem). The inference algorithm converges when the marginal probabilities stabilize. These probabilities are used by the attacker as the inferred values of the targeted SNPs. Note that this algorithm provides exact posterior probabilities when SNPs are independent of each other (case without LD correlations) and approximate probabilities, but in practice very accurate, when SNPs are in LD with each other [27].

Once the targeted SNPs are inferred by the adversary, genomic and health privacy of the family members are evaluated based on the adversary's success and certainty about the targeted SNPs and the diseases they reveal. We propose two different metrics for quantifying genomic and health privacy: expected estimation error (incorrectness) and uncertainty [27].

The inferred marginal probabilities can be expressed as $P(\hat{x}_j^i|\mathbf{X_O},\mathcal{K})$, for all $i \in \mathbf{F}, j \in \mathbf{S}$. *Incorrectness* quantifies the adversary's error in inferring the targeted SNPs. In other words, it quantifies the expected distance between the adversary's estimate on the value of a SNP, $\hat{x}_j^i$ and the true value of the corresponding SNP, x_j^i. This expected estima-

[4] Y-chromosome is directly passed from father to son.

[5] Mitochondrial DNA (mtDNA) is directly passed from mother to son or daughter.

[6] For instance, http://www.genealogy.com/.

tion error can be expressed as follows:

$$E_j^i = \sum_{\hat{x}_j^i \in \{0,1,2\}} P(\hat{x}_j^i | \mathbf{X_O}, \mathcal{K}) ||x_j^i - \hat{x}_j^i||. \quad (1)$$

Privacy can also be represented as the adversary's *uncertainty* [15, 40], that is the ambiguity of $P(\hat{x}_j^i|\mathbf{X_O}, \mathcal{K})$. This definition of uncertainty can be quantified as the (normalized) entropy of $P(\hat{x}_j^i|\mathbf{X_O}, \mathcal{K})$ as follows:

$$H_j^i = \frac{-\sum_{\hat{x}_j^i \in \{0,1,2\}} P(\hat{x}_j^i|\mathbf{X_O}, \mathcal{K}) \log P(\hat{x}_j^i|\mathbf{X_O}, \mathcal{K})}{\log(3)}. \quad (2)$$

We can also rely upon a similar metric, based on the mutual information $I(x_j^i; \mathbf{X_O}) = H(x_j^i) - H(x_j^i|\mathbf{X_O})$, that measures the mutual independence between the targeted and the observed data:

$$I_j^i = 1 - \frac{I(x_j^i; \mathbf{X_O})}{H(x_j^i)} = \frac{H(x_j^i|\mathbf{X_O})}{H(x_j^i)}. \quad (3)$$

The above metrics are useful for quantifying the genomic privacy of individuals. To quantify something more tangible, these genomic-privacy metrics can be converted into health-privacy metrics. To quantify an individual's health privacy, one can focus on the predisposition to different diseases. Let $\mathbf{S}_d$ be the set of IDs of the SNPs that are associated with a disease d. Then, a metrics quantifying the health privacy for an individual i regarding the disease d can be defined as follows:

$$D_d^i = \frac{1}{\sum_{k \in \mathbf{S}_d} c_k} \sum_{k \in \mathbf{S}_d} c_k G_k^i, \quad (4)$$

where G_k^i is the genomic privacy of a SNP k for individual i, computed using (1) or (2), and c_k is the contribution of SNP k to disease d.

3. PROPOSED SOLUTION

In order to mitigate attribute-inference attacks and protect genomic and health privacy, the GPPM relies upon an *obfuscation mechanism.* In practice, obfuscation can be implemented by adding noise to the SNP values, by injecting fake SNP values, by reducing precision, or by simply hiding the SNP values. In this paper, we choose SNP hiding, essentially because other options would not be tolerated by the genomic research community. Indeed, genetic researchers are very reluctant to adding noise or fake data, notably because of the huge investment they make to increase (sequencing) accuracy. We assume one family member at a given time willing to disclose his SNPs (e.g., for the research community) while guaranteeing a minimum privacy level for him and his family. Fig. 3 provides an overview of our genomic-privacy preserving framework.

3.1 Settings

For clarity of presentation, we focus on one family, whose members are defined by the set $\mathbf{F}$ ($|\mathbf{F}| = n$). We assume that there is only one donor D making a decision to share his genome at a given time. His relatives might have already publicly shared some of their genomic data on the Internet. D takes this into account when he makes his own disclosure decision. We let $\mathbf{S}$ ($|\mathbf{S}| = m$) be the set of SNP IDs. Its cardinality m can go up to 50 million, as this is currently the approximate number of SNPs in the human population [3]. In practice, however, people put online (e.g., on OpenSNP) up to 1 million most significant SNPs. We let $\mathbf{X}^D = \{x_j^D : j \in \mathbf{S}\}$ represent the set of SNPs of D (x_j^D is the value of SNP j of the donor D), that are all initially undisclosed, i.e. $\mathbf{X}^D \subseteq \mathbf{X_H}$ (where $\mathbf{X_H}$ denotes the set of SNPs from $\mathbf{X}$ whose values are hidden, as discussed before). Finally, we let $\mathbf{y}^D = \{y_j^D : j \in \mathbf{S}\}$ represent the decision vector of D, where $y_j^D = 1$ means the corresponding SNP will be disclosed, and $y_j^D = 0$ means x_j^D will remain hidden. Note that the decision of disclosing a SNP j could be probabilistic, transforming y_j^D into a continuous variable in $[0, 1]$. We leave the study of the continuous case for future work.

We express the privacy constraints of a family member both in terms of genomic and health privacy. Our framework can account for different privacy preferences for different family members, SNPs, and diseases. In practice, we can set the privacy sensitivities s_j^i's of all SNPs to be equal by default. Then, an individual willing to personalize his privacy preferences may further define his own privacy sensitivities regarding specific SNPs based on his privacy concerns regarding, e.g., certain phenotypes. The most well-known example of such a scenario is the case of James Watson, co-discoverer of DNA, who made his whole DNA sequence publicly available, with the exception of one gene known as Apolipoprotein E (ApoE), one of the strongest predictors for the development of Alzheimer's disease.[7] We let the sets $\mathbf{P_s^i}$ and $\mathbf{P_d^i}$ include the privacy-sensitive SNP IDs and privacy-sensitive diseases of individual i, respectively. We represent the tolerance to genomic-privacy loss of individual i as $\text{Pri}(i, \mathbf{P_s^i})$, and the tolerance the health-privacy loss of individual i regarding disease $d \in \mathbf{P_d^i}$ as $\text{Pri}(i, d)$. These tolerance values represent the maximum privacy loss (after the disclosure of D's SNPs) that an individual would bear. By considering the privacy losses instead of the absolute privacy levels, we ensure that the donor will more likely reveal a SNP whose value is already well inferred by the attacker before donor's disclosure (e.g., by using SNPs previously shared by the donor's relatives). Note that these tolerance values can always be updated for any new family member willing to disclose his genome. Finally, the utility function is a non-decreasing function of the norm of $\mathbf{y}^D$ as the knowledge of more SNPs can only help genomic research. As a first step towards enhanced genomic privacy, we assume linear contribution of SNPs on utility.[8] Formally, we define u_j to be the utility provided by SNP j to genomic research. Note that, in practice, the utility of the SNPs can be determined by the (trusted) research authorities and can vary based on the type of study.

3.2 Linear Optimization

3.2.1 Optimization Problem

The donor is facing an optimization problem: maximizing research utility while protecting his own and his relatives' genomic and health privacy. First, the objective function is formally defined as $\sum_{j \in \mathbf{S}} u_j y_j^D$. Then, privacy constraints are defined, for each individual, as the sum of privacy losses induced by the donor's disclosure over all SNPs. This sum

[7] Later researchers have used correlations in the genome to unveil Watson's predisposition to Alzheimer's [37]. In this work, we also consider such correlations.

[8] We intend to study non-linear utility functions in future work.

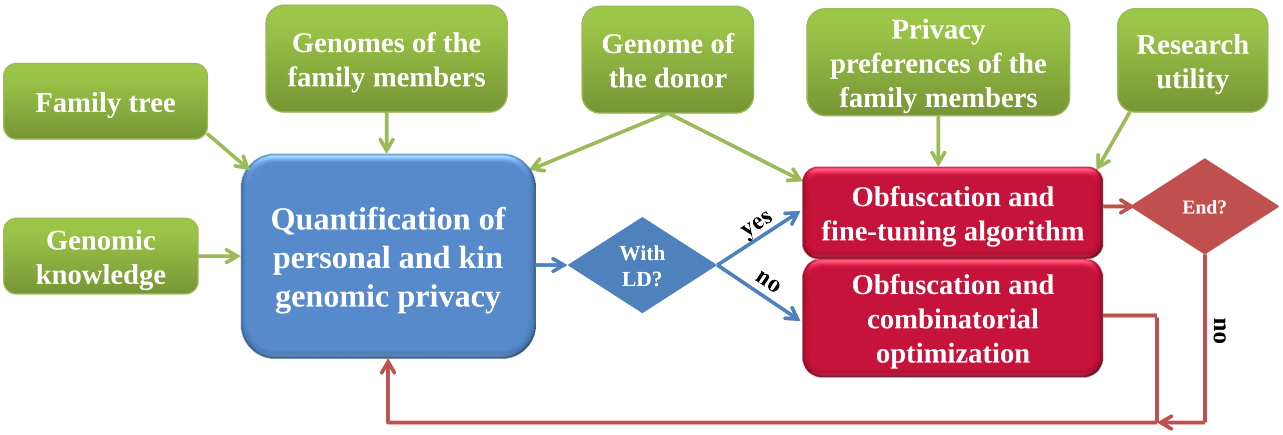

Family tree
Genomes of the family members
Genome of the donor
Privacy preferences of the family members
Research utility
Genomic knowledge
Quantification of personal and kin genomic privacy
With LD?
yes
no
Obfuscation and fine-tuning algorithm
Obfuscation and combinatorial optimization
End?
no

Figure 3: General protection framework. The GPPM takes as inputs (i) the privacy levels of all family members, (ii) the genome of the donor, (iii) the privacy preferences of the family members, and (iv) the research utility. First, LD is not considered in order to use combinatorial optimization (see Subsection 3.2). Note that we go only once through this box. Then, LD is used and a fine-tuning algorithm is used to cope with non-linear constraints. See Subsection 3.3 for details on the end criterion. The algorithm outputs the set of SNPs that the donor can disclose.

must be capped by the respective privacy loss tolerances of all family members. Formally, for all individuals $i \in \mathbf{F}$ and SNPs $j \in \mathbf{S}$, the privacy loss induced by the disclosure of x_j^D is defined as $(E_j^i(y_j^D = 0) - E_j^i(y_j^D = 1))$.[9] Note here that the privacy loss at a given SNP j for any relative is only affected by the donor's decision y_j^D regarding SNP j but no other SNP $k \neq j$, meaning that LD correlations are not taken into account. We make this assumption here in order to define linear constraints. We show how to extend the linear optimization problem to encompass LD correlations in Subsection 3.3. Finally, note that if an individual i has already revealed his SNP j, i.e. $x_j^i \in \mathbf{X_O}$, the privacy loss at this SNP for i is zero, because $E_j^i(y_j^D = 0) = E_j^i(y_j^D = 1) = 0$.

For all $i \in \mathbf{F}$ and SNP $j \in \mathbf{S}$, the privacy weight p_j^i is defined as

$$p_j^i = s_j^i \times (E_j^i(y_j^D = 0) - E_j^i(y_j^D = 1)), \tag{5}$$

where s_j^i is the privacy sensitivity of individual i regarding SNP j. Clearly, p_j^i at a given SNP j can be different for each family member, depending on how close he is from the donor in the family tree, on the actual values x_j^i and x_j^D of his and the donor's SNPs, and on his sensitivity. Note that $s_j^i = 0 \ \forall j \notin \mathbf{P_s^i}$.

We can now define the donor's linear optimization problem as:

$$\begin{aligned}
\underset{\mathbf{y}^D}{\text{maximize}} \quad & \sum_{j\in\mathbf{S}} u_j y_j^D \\
\text{subject to} \quad & \sum_{j\in\mathbf{P_s^i}} p_j^i y_j^D \leq \mathrm{Pri}(i, \mathbf{P_s^i}), \forall i \in \mathbf{F} \\
& \sum_{k\in\mathbf{S}_d} p_k^i y_k^D \leq \mathrm{Pri}(i, d), \forall d \in \mathbf{P_d^i}, \forall i \in \mathbf{F} \\
& y_j^D \in \{0,1\}, \forall j \in \mathbf{S},
\end{aligned} \tag{6}$$

where $\mathbf{S}_d$ is the set of SNPs that are associated with disease d. Note that, for the last inequality, we replace the sensitivity s_k^i in p_k^i by the contribution c_k of SNP k to disease d described in (4), and we embed the normalization factor $\sum_k c_k$ of (4) in $\mathrm{Pri}(i, d)$.

3.2.2 Optimization Algorithm

Our optimization problem is very similar to the multidimensional knapsack problem [33] discussed in Appendix B. We decide to follow the branch-and-bound method proposed by Shih [41] because it finds the optimal solution, represents a good trade-off between time and storage space, and allows for the extension of the algorithm to null and negative (privacy) weights. A branch-and-bound algorithm is a systematic enumeration of all candidate solutions, where large subsets of candidate solutions are pruned by using upper bounds on the quantity being optimized. A branch-and-bound method generally relies on two main rules: (i) the estimation of the upper bound at any node (state of assigned variables) in the search tree, and (ii) a choice criterion for the selection of a branching variable at the node selected for further partitioning.

In order to find (i), Shih suggests to treat the C-constraint knapsack problem as C single-constraint knapsack problems with the same objective function, and then compute the value associated to the optimal fractional solution (thus relaxing $y_j^D \in \{0,1\}$ into $y_j^D \in [0,1]$) of all of these C problems separately. The fractional optimal solution is easier to solve than the integer solution, as it allows us to sort the items (SNPs) with respect to their ratios between utility and privacy weights $r_j^i = u_j/p_j^i$, from the highest to the lowest ratios, and then to select all the highest ones that can fit under the constraint, with the last SNP to fit partially included (based on the remaining room). Note that, in our setting, one can have different orderings of SNPs for different constraints, based on the p_j^i values of the family members. The computation of the fractional optimal solution is repeated C times, for the C different optimization problems, leading to C optimal values. Then, the upper bound at the given

[9] Even though we use the expected estimation error E_j^i as the privacy metrics here, other metrics, such as H_j^i (discussed in Subsection 2.3), could also be relied upon.

node is defined as the minimum among all these C values.

The node selected for next branching is defined as the one in the search frontier whose upper bound is the highest among all nodes in the frontier, and where the solution associated with this upper bound is infeasible (some variables being different than 0 and 1, or some constraints being not satisfied). The branching variable is the one whose ratio is the smallest among all the non-zero free variables (variables not explicitly assigned to 0 or 1 at a node) in this infeasible solution. If the solution at this node happens to be feasible (all decision variables being assigned to 0 or 1 and all constraints being satisfied), then it is optimal, and the algorithm stops.

Let us mention that our optimization problem has two main differences with the multidimensional knapsack problem. First, the privacy metrics, hence weights, are expressed in real values between 0 and 2 for E_j^i, whereas the knapsack problem assumes integer numbers only. In order to obtain integer values, we merely multiply all our privacy weights p_j^i's and tolerance values Pri(.) by 10^k, where $k \in \mathbb{N}^+$ depends on the precision we want to attain, and then round the weights to the closest greater integer and the tolerance values to the closest smaller integer. This ensures that all privacy constraints in the space of real numbers are still satisfied. Second, the privacy weight p_j^i can be equal to zero (e.g., if $x_j^i \in \mathbf{X_O}$) or even negative (when the donor reveals a SNP whose value increases the privacy of his relative(s) at the same SNP).[10] Thus, the ratios r_j^i might not be defined or be negative. In order to resolve this issue, we give a higher ranking in the ordering of SNPs to ratios with null weights with respect to those with positive weights, and we give even a higher ranking to those with negative weights. We furthermore give higher ranking to negative weights with absolute values higher than others. To enforce this ranking in practice in Section 4, we set $r_j^i = u_j/0.1$ for null p_j^i's, and $r_j^i = u_j|p_j^i|/0.01$ for negative p_j^i's. Note that, due to the requirement of integer values for weights, all other (positive) weights p_j^i belong, after the aforementioned multiplication by 10^k and rounding, to $\mathbb{N}^+$.

The output of the above optimization algorithm is an optimal solution $\mathbf{y}^{*D}$ that represents the set of SNPs the donor could disclose and an optimal value u^* representing the maximum research utility. We represent the set of the optimal candidate SNPs to be shared as $\tilde{\mathbf{X}}^D \subseteq \mathbf{X}^D$. This is the output we see in state 2 of Fig. 4. We give $\tilde{\mathbf{X}}^D$ as input to the non-linear algorithm described in Subsection 3.3 to eventually reach state 3.

3.3 Non-Linear Extension

3.3.1 Non-Linear Optimization Problem

The LD correlations between the SNPs are not considered in the above optimization problem in order for the constraints to remain linear. In this subsection, we propose an extension of the branch-and-bound algorithm in order to deal with non-linear constraints.

Whereas in the case without LD, the privacy loss at a given SNP j of individual i depended only on the donor's decision y_j^D regarding SNP j, we have here to consider all the SNPs in LD with j to evaluate the privacy loss at j. Defining $\tilde{E}_j^i$ to be the privacy level of individual i at SNP j quantified by including LD correlations, the privacy loss at SNP j of individual i induced by the disclosure of $\tilde{\mathbf{X}}^D$ is equal to $(\tilde{E}_j^i(\mathbf{y}^D = \mathbf{0}) - \tilde{E}_j^i(\mathbf{y}^{*D}))$. This leads to the following updated privacy weights

$$\tilde{p_j^i} = s_j^i \times (\tilde{E}_j^i(\mathbf{y}^D = \mathbf{0}) - \tilde{E}_j^i(\mathbf{y}^{*D})). \tag{7}$$

Note that now the argument of $\tilde{E}_j^i$ is the entire vector $\mathbf{y}^D$ and not only y_j^D, because of LD. The optimization problem in (6) is reformulated as a non-linear optimization problem:

$$\begin{aligned} \underset{\mathbf{y}^D}{\text{maximize}} \quad & \sum_{j\in\mathbf{S}} u_j y_j^D \\ \text{subject to} \quad & \sum_{j\in\mathbf{P_s^i}} \tilde{p_j^i}(\mathbf{y}^D) \leq \text{Pri}(i, \mathbf{P_s^i}), \forall i \in \mathbf{F} \\ & \sum_{k\in\mathbf{S}_d} \tilde{p_k^i}(\mathbf{y}^D) \leq \text{Pri}(i, d), \forall d \in \mathbf{P_d^i}, \forall i \in \mathbf{F} \\ & y_j^D \in \{0,1\}, \forall j \in \mathbf{S}. \end{aligned} \tag{8}$$

Instead of solving this very complex optimization problem, we rely on the optimal solution $\mathbf{y}^{*D}$ computed in Subsection 3.2, embed it into (8), and check whether the privacy constraints are still met with the updated privacy weights $\tilde{p_j^i}$'s. Let us first study the case when no SNP has been disclosed by any relative before the donor's decision.[11] If $\mathbf{X_O} = \varnothing$, then

$$\sum_{j\in\mathbf{P_s^i}} \tilde{E}_j^i(\mathbf{y}^D = \mathbf{0}) = \sum_{j\in\mathbf{P_s^i}} E_j^i(\mathbf{y}^D = \mathbf{0}) \tag{9}$$

and, because of LD correlations,

$$\sum_{j\in\mathbf{P_s^i}} \tilde{E}_j^i(\mathbf{y}^{*D}) \leq \sum_{j\in\mathbf{P_s^i}} E_j^i(\mathbf{y}^{*D}). \tag{10}$$

Embedding (9) and (10) in (5) and (7), we get

$$\sum_{j\in\mathbf{P_s^i}} \tilde{p_j^i}(\mathbf{y}^{*D}) \geq \sum_{j\in\mathbf{P_s^i}} p_j^i y_j^D, \tag{11}$$

meaning that, for the same value of $\text{Pri}(i, \mathbf{P_s^i})$ in (6) and (8), the privacy constraint of family member i in (8) will be violated with high likelihood once LD is taken into account. If $\mathbf{X_O} \neq \varnothing$, then two scenarios can happen. If

$$\underbrace{\sum_{j\in\mathbf{P_s^i}} E_j^i(\mathbf{y}^{*D}) - \tilde{E}_j^i(\mathbf{y}^{*D})}_{\substack{\text{Privacy difference using LD or not} \\ \text{when } \tilde{\mathbf{X}}^D \subset \mathbf{X_O}}} \geq \underbrace{\sum_{j\in\mathbf{P_s^i}} E_j^i(\mathbf{y}^D = \mathbf{0}) - \tilde{E}_j^i(\mathbf{y}^D = \mathbf{0})}_{\substack{\text{Privacy difference using LD or not} \\ \text{when } \tilde{\mathbf{X}}^D \subset \mathbf{X_H}}},$$

then we get the same inequality (11), leading to the same consequences of constraint violation. If , on the contrary,

$$\sum_{j\in\mathbf{P_s^i}} E_j^i(\mathbf{y}^{*D}) - \tilde{E}_j^i(\mathbf{y}^{*D}) < \sum_{j\in\mathbf{P_s^i}} E_j^i(\mathbf{y}^D = \mathbf{0}) - \tilde{E}_j^i(\mathbf{y}^D = \mathbf{0}), \tag{12}$$

[10] For example, assume a child to be homozygous-major at a given SNP and his father to be heterozygous. Then, the estimation error for the child's SNP, thus the privacy of the child for this SNP, increases when the father's SNP is observed by the attacker (compared to the case when it is unknown, when only the MAF is used, and this MAF is close to 0).

[11] Without loss of generality, we focus here on the genomic-privacy constraints.

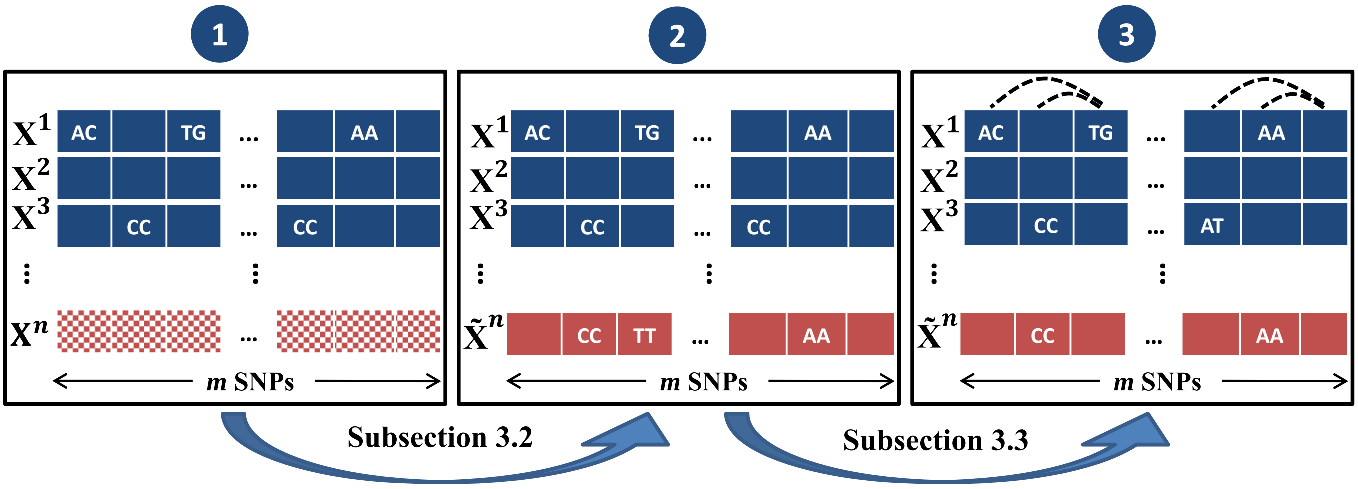


Figure 4: Main steps of the optimization algorithm. Without loss of generality, the donor D is assumed to be the n-th member of the family, thus $\mathbf{X}^D = \mathbf{X}^n$. First, the donor selects a subset $\tilde{\mathbf{X}}^D$ of candidate SNPs to be shared using the optimization algorithm of Subsection 3.2.2, and then reveals less or more SNPs depending on the updated privacy weights computed with LD by relying upon the fine-tuning step of Subsection 3.3.

then we get

$$\sum_{j \in \mathbf{P}^i_{\mathbf{s}}} \tilde{p}^i_j(\mathbf{y}^{*D}) < \sum_{j \in \mathbf{P}^i_{\mathbf{s}}} p^i_j y^D_j, \qquad (13)$$

which might allow the donor to reveal more of his SNPs without violating any of his relatives' privacy constraints. At a first glance, Inequality (13) looks counterintuitive. However, in order to understand it, let us look at Inequality (12), which states that the difference in privacy levels if LD is used or not is smaller after the observation of the donor's SNPs $\tilde{\mathbf{X}}^D$. This means that, by revealing his own SNPs, the donor reduces the importance of using LD correlations to correctly infer some of the SNPs of his relatives. For instance, let us assume the donor to be the father of a child i whose mother has already revealed SNP j, in LD with another SNP k revealed by the child. Furthermore, assume that the father, mother and child are homozygous major at SNPs j and k. Now, before the father reveals his SNP j, there is some uncertainty on the child's SNP j, but observing SNP k of the child, the attacker improves his estimation if he uses LD correlations and thus reduces his estimation error, meaning $\tilde{E}^i_j(\mathbf{y}^D = \mathbf{0}) < E^i_j(\mathbf{y}^D = \mathbf{0})$. However, once the father decides to reveal his homozygous major SNP j ($y^{*D}_j = 1$), the attacker is certain that the child's SNP j is homozygous major, no matter if LD is used or not, i.e. $E^i_j(\mathbf{y}^{*D}) = \tilde{E}^i_j(\mathbf{y}^{*D}) = 0$. Thus, we have $E^i_j(\mathbf{y}^{*D}) - \tilde{E}^i_j(\mathbf{y}^{*D}) < E^i_j(\mathbf{y}^D = \mathbf{0}) - \tilde{E}^i_j(\mathbf{y}^D = \mathbf{0})$, leading by extension to Inequality (12).

3.3.2 *Fine-Tuning Algorithm*

Let us first describe how we proceed if one or multiple constraints are violated once LD correlations are considered in the privacy quantification. In this case, we first select the privacy constraint that is not met anymore with the highest difference between $\text{Pri}(i, \mathbf{P}^i_{\mathbf{s}})$ (or $\text{Pri}(i, d)$) and the newly computed privacy losses. Focusing on the set of genomic-privacy constraints, we thus select the constraint of the family member k, where

$$k = \arg\max_{i \in \mathbf{F}} \{ \sum_{j \in \mathbf{P}^i_{\mathbf{s}}} \tilde{p}^i_j(\mathbf{y}^{*D}) - \text{Pri}(i, \mathbf{P}^i_{\mathbf{s}}) \}. \qquad (14)$$

We want then to hide some SNPs j in $\tilde{\mathbf{X}}^D$ (i.e. where $y^{*D}_j = 1$) in order that the constraint of relative k is satisfied again. For all the SNPs in $\tilde{\mathbf{X}}^D$, we compute a global privacy weight δ^k_j for SNP j of k that includes the privacy loss induced by SNP j on the SNPs $l \in \mathbf{L}$ in LD with j. We compute this global privacy weight at SNP j for individual k as

$$\delta^k_j = \tilde{p}^k_j + \sum_{l \in \mathbf{L}} \tilde{p}^k_l$$
$$= s^k_j(\tilde{E}^k_j(\mathbf{y}^D{=}\mathbf{0}) - \tilde{E}^k_j(\mathbf{y}^{*D})) + \sum_{l \in \mathbf{L}} s^k_l(\tilde{E}^k_l(\mathbf{y}^D{=}\mathbf{0}) - \tilde{E}^k_l(\mathbf{y}^{*D})). \qquad (15)$$

Then, we compute the ratios of each SNP j (in $\tilde{\mathbf{X}}^D$) for individual k as $\bar{r}^k_j = \delta^k_j / u_j$. The SNPs with the highest ratios represent those where LD correlations cause the highest decrease in the genomic privacy of family member k and/or provide low utility to the optimal solution $\mathbf{y}^{*D}$ computed in Subsection 3.2. Thus, such SNPs should be removed first from the set $\tilde{\mathbf{X}}^D$ in order to meet the privacy constraint of individual k again, while causing the smallest decrease in utility.

We iteratively remove such SNPs (starting from the one with the highest ratio) from the set $\tilde{\mathbf{X}}^D$ and, after each removal, we input the new solution to the quantification box to see whether the privacy constraint is met for the family member k. We repeat this until all the privacy constraints are again satisfied for all family members in $\mathbf{F}$. Finally, the SNPs left in set $\tilde{\mathbf{X}}^D$ after the final iteration are publicly shared. This case is illustrated in state 3 of Fig. 4.

In the case where considering LD correlations in the privacy quantification actually decreases privacy losses, the privacy constraints are still met and can even allow for potential new SNPs to be included in $\tilde{\mathbf{X}}^D$. In this case, we select the genomic-privacy constraint where the remaining room between the genomic-privacy constraint and the newly computed privacy loss is the smallest, i.e. we select the constraint of the family member k, where

$$k = \arg\min_{i \in \mathbf{F}} \{\mathrm{Pri}(i, \mathbf{P}_{\mathbf{s}}^{i}) - \sum_{j \in \mathbf{P}_{\mathbf{s}}^{i}} \tilde{p}_{j}^{i}(\mathbf{y}^{*D})\}. \quad (16)$$

For all SNPs *not* in $\tilde{\mathbf{X}}^{D}$ (i.e., where $y_j^{*D} = 0$), we compute the privacy decrease led by LD for k compared to the privacy level computed without LD. We compute this privacy difference at a SNP j for individual k as

$$\Delta_j^k = E_j^k(y_j^{*D} = 0) - \tilde{E}_j^k(\mathbf{y}^{*D}), \quad (17)$$

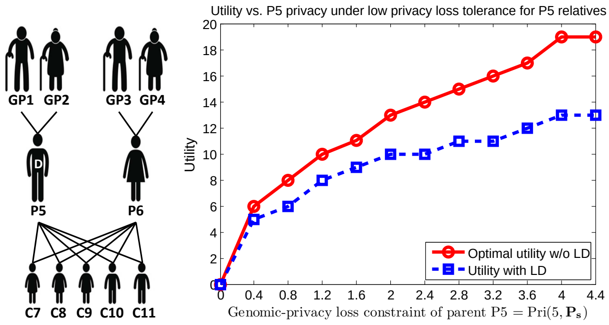
where $E_j^k(y_j^{*D} = 0)$ is the privacy value at SNP j for individual k after the linear optimization (without considering LD), and $\tilde{E}_j^k(\mathbf{y}^{*D})$ is the privacy quantified using LD. Then, we compute the ratios of each SNP j (in $\tilde{\mathbf{X}}^{D}$) for individual k as $\bar{r}_j^k = (u_j \Delta_j^k)/s_j^k$. The SNPs with highest ratios represent those where LD correlations cause highest decrease in the genomic privacy of family member k, and/or provide high utility. Thus, these SNPs are the first ones that should be included in $\tilde{\mathbf{X}}^{D}$, in order to have the smallest difference in privacy loss, thus still meeting k's privacy constraint and providing maximal utility increase.

We iteratively add SNPs in $\tilde{\mathbf{X}}^{D}$ and input the new solution to the quantification box to check whether all the privacy constraints are still met for all family members. We repeat this step until one privacy constraint is violated again, and we publicly share the last set $\tilde{\mathbf{X}}^{D}$ that satisfied all constraints. In the next section, we briefly show experimentally how close this fine-tuning algorithm is to the maximum found with exhaustive search. The thorough analytical evaluation of the discrepancy between the optimal solution and our approximation is left for future work.

4. EVALUATION

In this section, we evaluate the effectiveness of our optimization algorithm to protect individual and kin privacy. We study the balance between maximum achievable utility and the privacy of each individual in a family. The results show the total utility we can obtain for different genomic-privacy guarantees.

We make use of the CEPH/Utah Pedigree 1463 [16]. It includes the partial DNA sequences of 17 family members: 4 grandparents, 2 parents, and 11 children. In order to remain at a representative scale, we only keep 5 randomly chosen children out of 11. Fig. 5(a) presents the pedigree structure that we use in our study. We focus on 50 SNPs of chromosome 1 and assume one genomic-privacy constraint, including all the 50 SNPs for each family member. Thus, we have a total of 11 privacy constraints, which represents more constraints than other generic experiments in the optimization literature that included up to 5 or 7 constraints [20]. Considering LD strengths between $r^2 = 0.5$ (medium LD) and $r^2 = 1$ (strongest LD), each SNP is in LD with around 4.5 other SNPs, on average. We set a precision of 0.01 in our privacy weights and tolerance values, thus multiplying these real-valued elements by 10^2, and rounding them, as explained in Subsection 3.2. Parent P5 is assumed to be the donor in all scenarios presented in this section. In our evaluations, for the sake of simplicity, we assume each SNP is equally useful for the genomic research, i.e., $u_j = 1$ for all SNPs. We also assume the privacy sensitivities are equal, for all SNPs and individuals, i.e., $s_j^i = s$. Equal values of sensitivities for all SNPs would typically be the default setting, if e.g. family members do not want to bother setting their privacy sensitivities themselves. Other distributions over the utility or sensitivity values should not alter the algorithm's performances significantly. In fact, non-uniform distributions would even certainly improve its performances, because of the crucial role of orderings in the branch-and-bound method.



(a) (b)

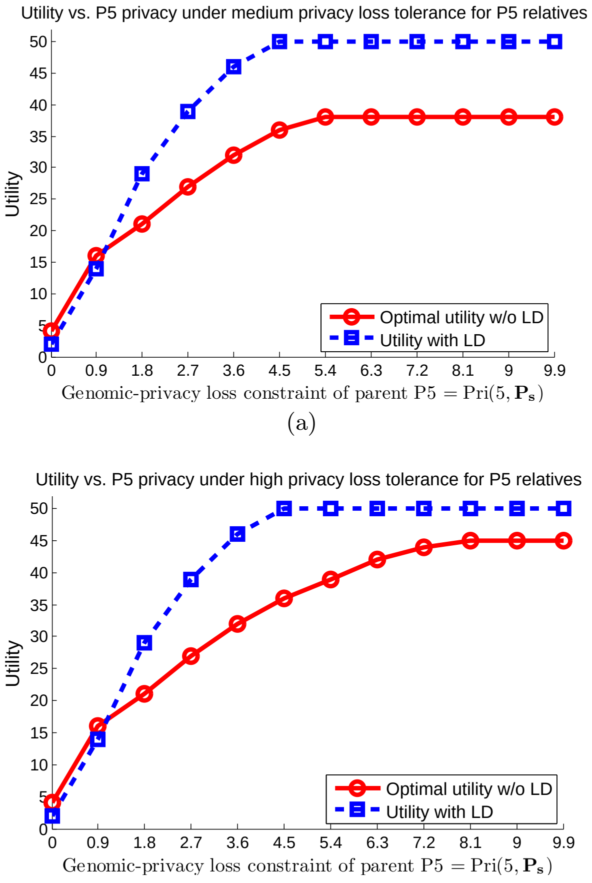
Figure 5: Evaluation of the proposed solution on a real Utah pedigree. (a) Genealogical tree, (b) Utility versus privacy under low tolerance to privacy loss for all relatives except parent P5, and varying values of privacy constraints $\mathrm{Pri}(5, \mathbf{P}_{\mathbf{s}}^{5})$ for parent P5 (x-axis). Here, $\mathbf{X_O} = \varnothing$, meaning that no relative has revealed any SNP before P5. Low tolerance is defined as 1/4 of the total privacy loss that a relative would incur if all 50 SNPs of P5 were revealed. Results are shown up to $\mathrm{Pri}(5, \mathbf{P}_{\mathbf{s}}^{5}) = 4.4$ even if P5's privacy constraint can go beyond because, from $\mathrm{Pri}(5, \mathbf{P}_{\mathbf{s}}^{5}) = 4$, the utility stops increasing (capped by other relatives' restrictive privacy constraints). The granularity of the x-axis is set to 0.4 in order to have 12 tolerance scenarios in total.

4.1 No Previous Disclosure by the Family

As most people have not publicly revealed their genome for the moment, we first analyze the case where no family member has shared any of his SNPs before the donor makes his decision. In other words, we assume that, initially, $\mathbf{X_O} = \varnothing$. We analyze the tension between utility and privacy for different values of parent P5's privacy constraint. Fig. 5(b) shows the increase in the utility caused by the higher privacy loss tolerance of P5. Because a low tolerance to privacy loss is assumed for all the other relatives in the family in this case, the utility (computed without LD) cannot go beyond 19, even if P5's constraint increases beyond 4. We also notice that, once the LD is included into the privacy quantification, the utility decreases, reaching a maximum value of 13 instead of 19. This is due to the fact that LD increases the privacy loss incurred when P5 reveals his SNPs, thus reducing the total number of SNPs parent P5 can reveal without violating the family's privacy constraints.

4.2 Previous Disclosure by Part of the Family

We want to mimic the situation that some of the family members have already revealed some of their SNPs. We



(a)

(b)

Figure 6: Utility versus privacy under (a) medium, (b) high tolerance to privacy loss for all relatives except parent P5, and varying values of privacy constraints for parent P5 (x-axis). Medium, respectively high, tolerance is defined as around half, respectively 3/4, of the total privacy loss that a relative would incur if all 50 SNPs of P5 were revealed. The x-axis represents the privacy loss constraint of P5, that has been split into 12 different cases, from 0 privacy loss (strongest constraint) to 9.9 privacy loss (i.e., around 0.2 privacy loss per SNP, which is a weak constraint).

simulate this by randomly selecting (with probability 0.5) some of the family members (except P5, who is the donor) who reveal a subset of their SNPs. Then, for the members who are selected to reveal their SNPs, we select, uniformly at random, some of their 50 SNPs to reveal. In the scenario we focus on, this leads to the following SNPs being revealed before the donor's decision: 8 (different) SNPs revealed by both GP1 and GP2; 35 SNPs revealed by GP3; 42 revealed by GP4; 0 by P6; 0 SNP revealed by C7, C8, C9, C10; and 30 by C11.

We analyze the relation between utility and privacy for different genomic-privacy constraint values, for each of the eleven individuals, $\mathrm{Pri}(i, \mathbf{P}^i_{\mathbf{s}})$. Fig. 6(a) and 6(b) illustrate the utility gain with respect to different privacy loss tolerance levels for the donor (P5). The two figures differ essentially in terms of the genomic-privacy constraints of the rest of the family members. In Fig. 6(a), the tolerance is medium; more precisely, the privacy constraint for each individual in the pedigree (except P5) is set to half of the maximum privacy loss that would be incurred by that individual if the donor revealed all his SNPs. In Fig. 6(a), the tolerance is higher, set to 3/4 of the maximum privacy loss.

We first focus on the utility computed using our branch-and-bound algorithm (case without LD). In Fig. 6(a), we observe that the utility does not increase beyond 38 when we increase the genomic-privacy loss constraint of the donor more than 5.4. From this point, the increased privacy tolerance of the donor does not enable him to reveal more SNPs, because he is constrained by the rest of the family's privacy requirements. In Fig. 6(b), we note that the utility keeps increasing with the privacy loss constraint of P5 because the other family members are more tolerant regarding their own privacy loss.

Looking at the utility induced once we include the LD correlations in the privacy quantification, we notice some increase in the utility. In other words, including LD enables the donor to reveal more SNPs than without LD. Utility in both curves reaches 50 SNPs after a 4.5 privacy loss constraint for the donor. This can be explained by the fact that, when LD is considered, we use Equation (7) (privacy loss with LD) instead of Equation (5) (privacy loss without LD) to compute the privacy weights for each SNP in each constraint. And the privacy loss in Equation (7) is actually smaller than in Equation (5) in this scenario, essentially because LD already decreases significantly the relatives' privacy before the donor reveals any of his own SNPs. This is very visible in Fig. 7(a) and 7(b). In Fig. 7(a), we show the privacy levels for any family member when LD is not included in the privacy quantification. Fig. 7(b) shows the privacy levels when LD correlations are also used in the privacy quantification.

First, we notice that in both figures, it is P5's privacy level that decreases the most, as he is the one who actually reveals new SNPs in the process. Other relatives' privacy is only damaged due to familial correlations. At the origin of the x-axis (i.e., on the y-axis), we see the privacy levels before the donor makes a decision, i.e., before the optimization algorithm. We notice that, here again, privacy without LD is much higher than privacy once LD is used to infer the SNPs. This is because some relatives have already revealed part of their genomic data. This is the reason why, once P5 reveals his own SNPs, the privacy loss is much smaller in Fig. 7(b) than in Fig. 7(a). As a consequence, the donor (P5) can reveal more SNPs while still meeting his family's privacy constraints, thus leading to the utility increase displayed in Fig. 6(a) and 6(b). We conclude that the values of the privacy-loss constraints have to be carefully determined by the family members or the genetic counsellors, based on family members' privacy expectations and on whether LD is included or not in the initial inference and privacy quantification. In our case, in order to make use of the linear optimization framework, we defined the privacy loss constraints based on the privacy levels computed without LD.

Finally, we compared the optimal solutions computed with exhaustive search over a subset of 10 SNPs whose privacy weights were computed with LD, with the solutions derived from our optimization algorithm presented in Fig. 4. In the various scenarios we tested, the exhaustive search method

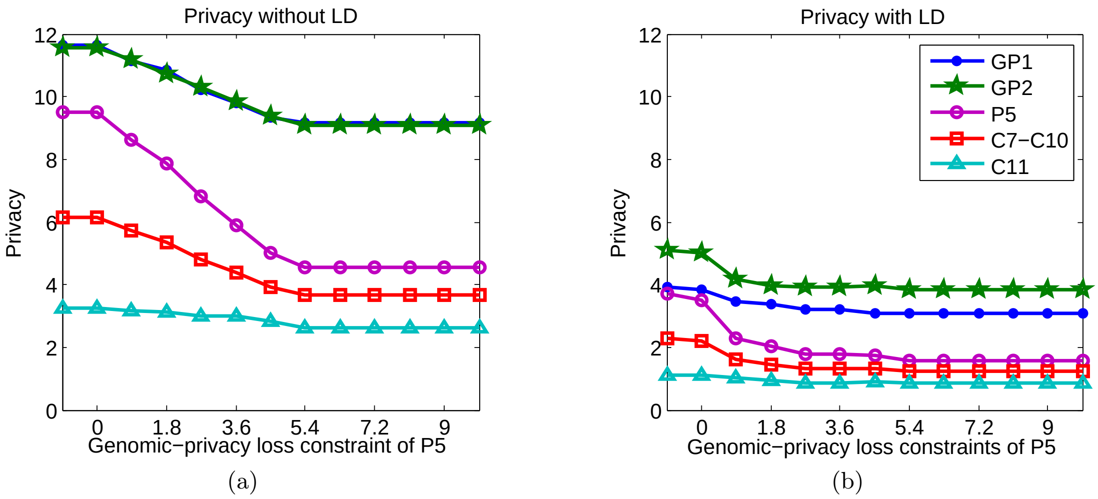


Figure 7: Genomic privacy of all family members given the genomic-privacy constraint of P5, under the same setting as in Fig. 6(a) i.e. under medium privacy loss constraints for P5 relatives: (a) privacy computed without LD, and (b) privacy computed with LD, before the fine-tuning phase. We do not show the privacy levels of GP3, GP4 and P6 as these remain constant. Note the large discrepancy in absolute privacy values and privacy losses between Fig. 7(a) and 7(b). Also notice that GP1 privacy curve is hidden by GP2 privacy curve in Fig. 7(a) (they have same privacy levels w/o LD).

could never find higher utility values than our fine-tuning algorithm. In all scenarios, our fine-tuning algorithm reached the maximum utility. Thus, even though we do not have any formal demonstration that the fine-tuning step is optimal, we are confident that it provides in general a very good approximation of the optimum.

4.3 Computational Complexity

As expected, the highest computation time was on average induced by the branch-and-bound algorithm (Subsec. 3.2), due to the high complexity of the multidimensional knapsack problem. The non-linear extension (Subsec. 3.3) is by design very efficient as it relies on previous optimal computations, and it updates a minimal set of decision variables, trading-off exact optimality for computational efficiency. This last part only requires to quantify privacy levels twice at the beginning (in the quantification box), to get the $\tilde{E}^i_j(\mathbf{y}^D = \mathbf{0})$'s and $\tilde{E}^i_j(\mathbf{y}^{*D})$'s, and then quantify once per update on a decision variable y^{*D}_j.[12]

As discussed in Appendix B, the multidimensional knapsack problem (with at least two constraints) is NP-complete and admits no fully polynomial-time approximation scheme. From our experiments, we notice that the complexity of the branch-and-bound algorithm highly differs for different settings, e.g., different privacy-loss tolerance values or privacy weights. With 50 SNPs, the vast majority of the solutions were found in less than one second. However, the algorithm did not scale well for more than 50 decision variables. The positive side is that this whole process has to be undertaken only once by the donor and can be run offline. Furthermore, we considered one privacy constraint for each family member, thus eleven constraints in total. In practice, some relatives would certainly not care much about their genomic privacy, and thus some constraints could be relaxed, enabling us to consider more SNPs in the optimization problem. Also, an advantage of the branch-and-bound algorithm is that it can be run in parallel and distributed using a computer cluster. The algorithm's running time then scales linearly with the number of machines and cores [9]. Another way to reduce the complexity is to cluster subsets of SNPs together (based on the diseases they are associated with, or based on the LD correlations between them), thus trading-off the granularity of the obfuscation mechanism for computational efficiency. Note that our optimization problem can easily be adapted to deal with clusters of SNPs: we can simply define the privacy weight of one cluster as the sum of the privacy losses over the SNPs in this cluster. Finally, instead of using an exact optimization method, heuristic approaches [20] could be used to approximate the optimal solution of the optimization problem and improve computation efficiency. We intend to further study the efficiency of these approaches in future work.

5. RELATED WORK

Stajano *et al.* [42] were among the first to raise the issue of kin privacy in genomics, and suggest to discuss questions such as: Should you be allowed to disclose your genome if other relatives do not want to? Our work notably aims to address this concern. Cassa *et al.* [10] provide a framework for measuring the interdependent privacy risks between two siblings. They show that the inference error is substantially reduced when the sibling's SNPs are known, compared to when only the population frequencies are used. Humbert *et al.* [27] generalized this evaluation of kin genomic privacy risks by considering any kind of observation from family members, LD relationships between SNPs, and well-defined privacy metrics. We build upon this work to propose privacy protection mechanisms that meet all family members'

[12] Note that the computational complexity of one quantification step is $\mathcal{O}(nm)$ [27].

privacy requirements while maximizing utility.

Homer *et al.* [26] prove that de-identification is an ineffective way to protect the privacy of genomic data, which is also supported by other works [23, 28, 36, 45, 48]. Most recently, Gymrek *et al.* [25] show how they identified DNAs of several individuals and families who participated in scientific studies. Building upon [26], Sankararaman *et al.* [39] provide quantitative guidelines for researchers willing to make a certain number of SNPs publicly available in GWAS, without revealing the presence of a single individual within a study group. Fienberg *et al.* [19] propose using differential privacy to protect the identities of participants in scientific study. In the same vein, Johnson and Shmatikov [30] propose privacy-preserving algorithms for computing various statistics related to the SNPs, while guaranteeing differential privacy. However, differential privacy reduces the accuracy of research results and it is aimed to be applied on aggregate results. Our work focuses on protecting individual genomes' privacy.

Some pieces of work also focus on protecting the privacy of genomic data and on preserving utility in medical tests such as (i) searching of a particular pattern in the DNA sequence [7, 44], (ii) comparing the similarity of DNA sequences [6, 8, 13, 14, 29, 34], and (iii) performing statistical analysis on several DNA sequences [32]. Furthermore, Ayday *et al.* propose privacy-preserving schemes for medical tests and personalized medicine methods that use patients' genomic data [5]. For privacy-preserving clinical genomics, a group of researchers proposes to outsource some costly computations to a public cloud or semi-trusted service provider [11, 46]. All aforementioned works make use of cryptographic protocols to protect the privacy of genomic data. In this paper, we propose a non-cryptographic approach to protect genomic privacy.

Finally, Calmon and Fawaz propose an inference framework to evaluate privacy risks under utility constraints in a generic settings [17]. Their goal is to minimize information leakage subject to certain utility constraints. They show that their optimization problem can be cast as a modified rate-distortion problem. They eventually compare their framework with differential privacy.

6. DISCUSSION

There is little doubt that the momentum in genome sequencing will bring new challenges to data security and privacy. In this work, we convey the importance of building mechanisms for preserving genomic privacy. Such privacy goes beyond the protection of genome information of the individual to consider the interests of family members. Relatives might be unwilling to allow predictions of their SNPs based on leakage of information from one or several individuals of the kin. The approach presented here searches for balance between accuracy (utility) of genomic data and privacy by relying on graphical models and optimization.

Our solution has also some limitations. The approach requires input that could be difficult to obtain in practice. However, default privacy preferences could be set by the system (e.g., considering the SNPs revealing privacy-sensitive diseases), letting individuals provide personal input about their privacy sensitivities if they wish. Another possible approach is to let genetic counselors help relatives in their endeavors by proposing different genomic-privacy profiles. For example, some profiles could be more or less restrictive in general terms, or forbid access to information related to specific diseases (e.g., Mendelian diseases, or predispositions to dementia). The details are out of the scope of this work. We do not claim to solve the whole genomic-privacy problem and to thwart all possible attacks. Our solution protects against the best known attribute-inference attack, given the current data model. Following [27], we did not make use of genetic imputation via IBD as we do not assume the haplotypes to be accessible to the adversary.[13] Availability of this information or phenotypes could further improve the inference attack, but we leave its evaluation for future work.

Acknowledgements

We would like to thank Reza Shokri, Jean Louis Raisaro, and Zhicong Huang for their valuable comments on the submitted manuscript, as well as the anonymous reviewers for their helpful feedback.

7. REFERENCES

[1] `http://www.nytimes.com/2013/03/24/opinion/sunday/the-immortal-life-of-henrietta-lacks-the-sequel.html?pagewanted=all`.

[2] `http://www.nytimes.com/2013/08/08/science/after-decades-of-research-henrietta-lacks-family-is-asked-for-consent.html?pagewanted=all`.

[3] `http://www.ncbi.nlm.nih.gov/projects/SNP/`. Visited on 6-Feb-2013.

[4] E. Ayday, J. L. Raisaro, and J. P. Hubaux. Protecting and evaluating genomic privacy in medical tests and personalized medicine. *WPES '13: Proceedings of ACM Workshop on Privacy in the Electronic Society*, 2013.

[5] E. Ayday, J. L. Raisaro, P. J. McLaren, J. Fellay, and J. P. Hubaux. Privacy-preserving computation of disease risk by using genomic, clinical, and environmental data. *HealthTech*, 2013.

[6] P. Baldi, R. Baronio, E. De Cristofaro, P. Gasti, and G. Tsudik. Countering GATTACA: Efficient and secure testing of fully-sequenced human genomes. *CCS*, 2011.

[7] M. Blanton and M. Aliasgari. Secure outsourcing of DNA searching via finite automata. *DBSec*, 2010.

[8] F. Bruekers, S. Katzenbeisser, K. Kursawe, and P. Tuyls. Privacy-preserving matching of DNA profiles. Technical report, 2008.

[9] M. Budiu, D. Delling, and R. F. Werneck. Dryadopt: Branch-and-bound on distributed data-parallel execution engines. In *Parallel & Distributed Processing Symposium (IPDPS), 2011 IEEE International*, pages 1278–1289. IEEE, 2011.

[10] C. A. Cassa, B. Schmidt, I. S. Kohane, and K. D. Mandl. My sister's keeper?: genomic research and the identifiability of siblings. *BMC Medical Genomics*, 1(1):32, 2008.

[11] Y. Chen, B. Peng, X. Wang, and H. Tang. Large-scale privacy-preserving mapping of human genomic sequences on hybrid clouds. *NDSS*, 2012.

[13] Note that, e.g., haplotypes are not available on OpenSNP.

[12] G. Danezis and E. De Cristofaro. Simpler protocols for privacy-preserving disease susceptibility testing. In *GenoPri*, 2014.

[13] E. De Cristofaro, S. Faber, P. Gasti, and G. Tsudik. Genodroid: Are privacy-preserving genomic tests ready for prime time? *Proceedings of the ACM workshop on Privacy in the electronic society - WPES*, pages 97–108, 2012.

[14] E. De Cristofaro, S. Faber, and G. Tsudik. Secure genomic testing with size-and position-hiding private substring matching. In *Proceedings of the 12th ACM Workshop on privacy in the electronic society*, pages 107–118. ACM, 2013.

[15] C. Diaz, S. Seys, J. Claessens, and B. Preneel. Towards measuring anonymity. In *Privacy Enhancing Technologies*, pages 54–68. Springer, 2003.

[16] R. Drmanac, A. B. Sparks, M. J. Callow, A. L. Halpern, N. L. Burns, B. G. Kermani, P. Carnevali, I. Nazarenko, G. B. Nilsen, G. Yeung, et al. Human genome sequencing using unchained base reads on self-assembling dna nanoarrays. *Science*, 327(5961):78–81, 2010.

[17] F. du Pin Calmon and N. Fawaz. Privacy against statistical inference. In *Communication, Control, and Computing (Allerton), 2012 50th Annual Allerton Conference on*, pages 1401–1408. IEEE, 2012.

[18] Y. Erlich and A. Narayanan. Routes for breaching and protecting genetic privacy. abs/1310.3197v1, 2013.

[19] S. E. Fienberg, A. Slavkovic, and C. Uhler. Privacy preserving GWAS data sharing. *Proceedings of the IEEE 11th International Conference on Data Mining Workshops (ICDMW)*, Dec. 2011.

[20] A. Fréville. The multidimensional 0–1 knapsack problem: An overview. *European Journal of Operational Research*, 155(1):1–21, 2004.

[21] B. Gavish and H. Pirkul. Efficient algorithms for solving multiconstraint zero-one knapsack problems to optimality. *Mathematical Programming*, 31(1):78–105, 1985.

[22] P. Gilmore and R. Gomory. The theory and computation of knapsack functions. *Operations Research*, 14(6):1045–1074, 1966.

[23] J. Gitschier. Inferential genotyping of Y chromosomes in Latter-Day Saints founders and comparison to Utah samples in the HapMap project. *Am. J. Hum. Genet.*, 84:251–258, 2009.

[24] F. Glover. A multiphase-dual algorithm for the zero-one integer programming problem. *Operations Research*, 13(6):879–919, 1965.

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

[26] N. Homer, S. Szelinger, M. Redman, D. Duggan, and W. Tembe. Resolving individuals contributing trace amounts of DNA to highly complex mixtures using high-density SNP genotyping microarrays. *PLoS Genetics*, 4, Aug. 2008.

[27] M. Humbert, E. Ayday, J. P. Hubaux, and A. Telenti. Addressing the concerns of the Lacks Family: Quantification of kin genomic privacy. *CCS '13: Proceedings of 20th ACM Conference on Computer and Communications Security*, 2013.

[28] K. B. Jacobs, M. Yeager, S. Wacholder, D. Craig, P. Kraft, D. J. Hunter, J. Paschal, T. A. Manolio, M. Tucker, R. N. Hoover, et al. A new statistic and its power to infer membership in a genome-wide association study using genotype frequencies. *Nature Genetics*, 41(11):1253–1257, 2009.

[29] S. Jha, L. Kruger, and V. Shmatikov. Towards practical privacy for genomic computation. *Proceedings of the 2008 IEEE Symposium on Security and Privacy*, pages 216–230, 2008.

[30] A. Johnson and V. Shmatikov. Privacy-preserving data exploration in genome-wide association studies. In *Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 1079–1087. ACM, 2013.

[31] A. D. Johnson and C. J. O'Donnell. An open access database of genome-wide association results. *BMC Medical Genetics 10:6*, 2009.

[32] M. Kantarcioglu, W. Jiang, Y. Liu, and B. Malin. A cryptographic approach to securely share and query genomic sequences. *IEEE Transactions on Information Technology in Biomedicine*, 12(5):606–617, 2008.

[33] H. Kellerer, U. Pferschy, and D. Pisinger. *Knapsack problems.* Springer, 2004.

[34] F. Kerschbaum, M. Beck, and D. Schönfeld. Inference control for privacy-preserving genome matching. In *GenoPri*, 2014.

[35] F. Kschischang, B. Frey, and H. A. Loeliger. Factor graphs and the sum-product algorithm. *IEEE Transactions on Information Theory*, 47, 2001.

[36] N. Masca, P. R. Burton, and N. A. Sheehan. Participant identification in genetic association studies: improved methods and practical implications. *International Journal of Epidemiology*, 40(6):1629–1642, 2011.

[37] D. Nyholt, C. Yu, and P. Visscher. On Jim Watson's APOE status: Genetic information is hard to hide. *European Journal of Human Genetics*, 17:147–149, 2009.

[38] J. Pearl. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference.* Morgan Kaufmann Publishers, Inc., 1988.

[39] S. Sankararaman, G. Obozinski, M. I. Jordan, and E. Halperin. Genomic privacy and limits of individual detection in a pool. *Nature genetics*, 41(9):965–967, 2009.

[40] A. Serjantov and G. Danezis. Towards an information theoretic metric for anonymity. In *Privacy Enhancing Technologies*, pages 41–53. Springer, 2003.

[41] W. Shih. A branch and bound method for the multiconstraint zero-one knapsack problem. *Journal of the Operational Research Society*, pages 369–378, 1979.

[42] F. Stajano, L. Bianchi, P. Liò, and D. Korff. Forensic genomics: kin privacy, driftnets and other open questions. In *Proceedings of the 7th ACM workshop on Privacy in the electronic society*, 2008.

[43] L. Sweeney, A. Abu, and J. Winn. Identifying participants in the personal genome project by name. *Available at SSRN 2257732*, 2013.

[44] J. R. Troncoso-Pastoriza, S. Katzenbeisser, and M. Celik. Privacy preserving error resilient DNA

searching through oblivious automata. *CCS '07: Proceedings of the 14th ACM Conference on Computer and Communications Security*, 2007.

[45] 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. *Proceedings of the 16th ACM CCS*, pages 534–544, 2009.

[46] R. Wang, X. Wang, Z. Li, H. Tang, M. K. Reiter, and Z. Dong. Privacy-preserving genomic computation through program specialization. *Proceedings of the 16th ACM CCS*, pages 338–347, 2009.

[47] H. M. Weingartner and D. N. Ness. Methods for the solution of the multidimensional 0/1 knapsack problem. *Operations Research*, 15(1):83–103, 1967.

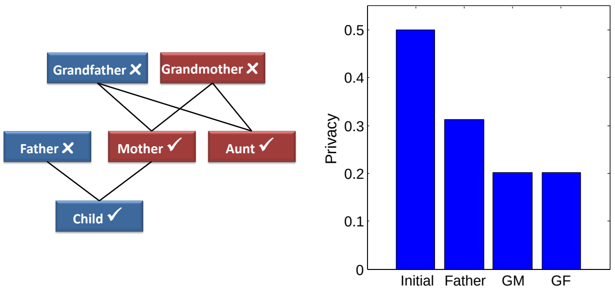
[48] X. Zhou, B. Peng, Y. F. Li, Y. Chen, H. Tang, and X. Wang. To release or not to release: Evaluating information leaks in aggregate human-genome data. *ESORICS*, 2011.

APPENDIX

Appendix A. SAMPLE ATTACK ON KIN GENOMIC PRIVACY

In this section, we give a concrete example of the threat discussed in Section 2.2 by using publicly available data on the Internet. We gathered individuals' genomic data from OpenSNP.org, a website on which people can publicly share sets of SNPs. Then, we identified the owners of 149 OpenSNP profiles. By relying on publicly accessible resources (e.g., genealogical websites), we could collect the family tree of 47 of these identified individuals. We also merged family trees with common ancestors to reconstruct hidden family relationships. We noticed that three of the identified individuals were associated to the same family (which is hereafter referred to as the targeted family). Furthermore, from the family tree, we obtained the names of 3 *target individuals* (only considering ancestors up to the grandparents of youngest identified individual revealing his SNPs) in the same family, as shown in Fig. 8(a). We emphasize again that these 3 target individuals did not publicly share any genomic data and that they would possibly be against such a disclosure. We compute the health privacy of the three targets closest to the observed individuals, about their predispositions to Alzheimer's disease.[14]

We used the metrics in (4) to quantify the health privacy of the target individuals.[15] We assigned equal weights to both associated SNPs (as their combination determines the predisposition to Alzheimer's disease). In Fig. 8(b), we show the attacker's uncertainty about the predisposition to Alzheimer's disease for the target individuals. We notice a decrease of 40% for the father, and of 60% for both the grandmother and the grandfather, compared to their initial privacy, without any information about the genomes of their relatives.



(a) (b)

Figure 8: Sample attack showing the threat on genomic and health privacy. (a) The family tree of the target family, in which the "check" means that the genome of the family member is publicly available in OpenSNP, and "cross" means it is not. (b) The decrease in health privacy of family members (focusing on the Alzheimer's disease), whose genomes are not publicly available.

Appendix B. THE MULTIDIMENSIONAL KNAPSACK PROBLEM

The knapsack problem is one the most well-known problems in combinatorial optimization [33]. Given a set of items, each with a weight and a value, the goal is to determine the optimal set of items so that the total weight in the sack is less than or equal to a given limit and the total value is maximized. The 0-1 knapsack problem can be formulated as an optimization problem with, for item i, decision variable $x_i \in \{0,1\}$, weight w_i, and value v_i. The optimization problem is NP-hard. However, dynamic programming and branch and bound methods can solve this problem in pseudo-polynomial time. The multidimensional 0-1 knapsack problem is a generalization of the 0-1 knapsack problem, and can be formulated as

$$\begin{aligned} \underset{\mathbf{x}}{\text{maximize}} \quad & \sum_{i=1}^{N} v_i x_i \\ \text{subject to} \quad & \sum_{i=1}^{N} w_{i,j} x_i \leq c_j, \forall j \in \{1,2,...,M\} \\ & x_i \in \{0,1\}, \forall i \in \{1,2,...,N\}, \end{aligned} \tag{18}$$

where $\mathbf{x}$ is the decision vector embedding all N decision variables, $w_{i,j}$ is the weight of decision variable x_i for the jth constraint, c_j is the capacity of constraint j, M is the number of constraints, and N is the number of decision variables. Note that $M = 1$ leads to the original 0-1 knapsack problem (with a single constraint). Solving the multidimensional knapsack problem (MKP) is NP-hard and it remains a challenge, especially when the number of constraints increases. For any fixed $M \geq 2$, the MKP admits no fully polynomial-time approximation scheme unless P = NP. There are two types of approaches for solving this problem: exact methods and heuristics [20]. Dynamic programming has been initially proposed for solving the MKP [22, 47]. However, due to its excessive memory space requirements, only problems with small values of N and capacities c_j can be solved with dynamic programming. Exact solutions can be computed more efficiently by relying on branch and bound methods [21, 24, 41]. Shih [41] makes use of linear relaxations to estimate an upperbound of the optimal value, whereas

[14] Two particular SNPs (rs7412 and rs429358) on the Apolipoprotein E (ApoE) gene indicate an (increased) risk for Alzheimer's disease.

[15] We used the (normalized) entropy in (2) on the targeted SNPs for G_k^i in (4).

Glover [24], and Gavish and Pirkul rely on surrogate relaxations [21]. Shih reports experiments with 5 constraints and 30-90 variables, whereas Gavish and Pirkul tested their algorithm with sizes up to 80 variables and 7 constraints. Heuristic approaches can also be competitive alternatives to exact methods, particularly when the number of constraints is large. These heuristics can be grouped into greedy algorithms, mathematical programming approaches and metaheuristics (that notably include genetic algorithms). More details about heuristic methods can be read in Section 4 of [20].