Quantifying Interdependent Risks in Genomic Privacy

MATHIAS HUMBERT, CISPA, Saarland University
ERMAN AYDAY, Bilkent University
JEAN-PIERRE HUBAUX, EPFL
AMALIO TELENTI, Human Longevity Inc.

The rapid progress in human-genome sequencing is leading to a high availability of genomic data. This data is notoriously very sensitive and stable in time. It is also highly correlated among relatives. A growing number of genomes are becoming accessible online (e.g., because of leakage, or after their posting on genome-sharing websites). What are then the implications for kin genomic privacy? We formalize the problem and detail efficient reconstruction attacks based on graphical models and belief propagation. With our approach, an attacker can infer the genomes of the relatives of an individual whose genome or phenotype(s) are observed, by notably relying on Mendel's Laws, statistical relationships between the genomic variants, and between the phenotypes and the variants. We evaluate the effect of these statistical relationships on privacy with respect to the amount of observed relatives and variants. We also study how the algorithmic performance evolves when we take these various relationships into account. Furthermore, to quantify the level of genomic privacy as a result of the proposed inference attack, we discuss possible definitions of *genomic privacy* metrics, and compare their values and evolution. Genomic data reveals Mendelian disorders and the likelihood of developing severe diseases such as Alzheimer's. We also introduce the quantification of *health privacy*, specifically the measure of how well the predisposition to a disease is concealed from an attacker. We evaluate our approach on actual genomic data from a pedigree and show the threat extent by combining data gathered from a genome-sharing website and from an online social network. Finally, we show how additional knowledge of phenotypic information can improve the inference attack's success.

Categories and Subject Descriptors: C.2.0 [**Computer-Communication Networks**]: General—*Security and protection*; J.3 [**Life and Medical Sciences**]: *Biology and genetics*; K.4.1 [**Computer and Society**]: Public Policy Issues—*Privacy*

General Terms: Algorithms, Theory

Additional Key Words and Phrases: Genomic Privacy; Inference; Metrics; Kinship

ACM Reference Format:
Mathias Humbert, Erman Ayday, Jean-Pierre Hubaux, and Amalio Telenti, 2015. Quantifying Interdependent Risks in Genomic Privacy. *ACM Trans. Info. Syst. Sec.* 9, 4, Article 39 (March 2010), 30 pages.
DOI:http://dx.doi.org/10.1145/0000000.0000000

1. INTRODUCTION

With the help of rapidly developing technology, DNA sequencing is becoming less expensive. As a consequence, the research in genomics has gained speed in paving the way to personalized (genomic) medicine, and geneticists need large collections of human genomes to further increase this speed. Furthermore, individuals are using their genomes to learn about their (genetic) predispositions to diseases, their ancestries, and even their compatibilities with potential partners. This trend has also caused the launch of health-related websites and online social networks (OSNs), in which individuals share

Author's addresses: M. Humbert, CISPA, Saarland University, Saarbruecken, Germany; E. Ayday, Bilkent University, Ankara, Turkey; J.-P. Hubaux, Laboratory for Communications and Applications, EPFL, Lausanne, Switzerland; A. Telenti, Human Longevity Inc., San Diego, CA, USA.

DOI:http://dx.doi.org/10.1145/0000000.0000000

their genomic data (e.g., OpenSNP[1] or 23andMe[2]). Thus, already today, tens of thousands of genomes are available online.

Even though most of the genomes on the Internet are anonymized, it is possible to find genomes with the identifiers of their owners (e.g., OpenSNP). Furthermore, it has been shown that anonymization is not sufficient for protecting the real identities of the genome donors [Gymrek et al. 2013; Sweeney et al. 2013]. Once the owner of a genome is identified, he is faced with the risk of discrimination (e.g., by employers or insurance companies) [Ayday et al. 2015]. Some believe that they have nothing to hide about their genetic structure, hence they might decide to give full consent for the publication of their genomes on the Internet to help genomic research. However, our DNA sequences are highly correlated to our relatives' sequences. The DNA sequences between two random human beings are 99.9% similar, and this value is even higher for closely related people. Consequently, somebody revealing his genome does not only damage his own genomic privacy, but also puts his relatives' privacy at risk [Stajano et al. 2008]. Moreover, currently, a person does not need consent from his relatives to share his genome online. This is precisely where the interesting part of the story begins: *kin genomic privacy*.

A New York Times' article[3] reports the controversy about sequencing and publishing, without the permission of her family, the genome of Henrietta Lacks (who died in 1951). On the one hand, the family members think that her genome is private family information and that it should not be published without the consent of the family. On the other hand, some scientists argued that the genomes of current family members have changed so much over time (due to gene mixing during reproduction), that nothing accurate could be told about the genomes of current family members by using Henrietta Lacks' genome. As we will also show in this work, they are wrong. Minutes after Henrietta Lacks' genome was uploaded to a public website called SNPedia, researchers produced a report full of personal information about Henrietta Lacks. Later, the genome was taken offline, but it had already been downloaded by several people, hence both her and (partially) the Lacks family's genomic privacy was already lost.

Unfortunately, the Lacks, even though possibly the most publicized family facing this problem, are not the only family facing this threat. As we mentioned before, the genomes of thousands of individuals are available online. Once the identity of a genome donor is known, an attacker can learn about his relatives (or his family tree) by using an auxiliary side channel, such as an OSN, and infer significant information about the DNA sequences of the donor's relatives. We will show the feasibility of such an attack and evaluate the privacy risks by using publicly available data on the Web.

Although the researchers took Henrietta Lacks' genome offline from SNPedia, other databases continue to publish portions of her genomic data. Publishing only portions of a genome does not, however, completely hide the unpublished portions; even if a person reveals only a part of his genome, other parts can be inferred using the statistical relationships between the nucleotides in his DNA. For example, James Watson, co-discoverer of DNA, 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. However, later it was shown that the correlation (called *linkage disequilibrium* by geneticists) between one or multiple polymorphisms and ApoE could be used to predict the ApoE status [Nyholt et al. 2009]. Thus, an attacker can also use these statistical relationships (which are publicly available) to infer the DNA sequences of a donor's family members, even if the donor shares only part of his genome. It is important to note that these privacy threats not only jeopardize kin genomic privacy but, if

[1] http://opensnp.org/

[2] https://www.23andme.com/welcome/

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

not properly addressed, these issues could also hamper genomic research due to untimely fear of potential misuse of genomic information.

In this work, we quantify the genomic privacy of an individual threatened by his relatives revealing their genomes. Focusing on the most common variant in human population, single nucleotide polymorphism (SNP), and considering the statistical relationships between the SNPs on the DNA sequence, we quantify the loss in genomic privacy of individuals when one or more of their family members' genomes are (either partially or fully) revealed. To achieve this goal, first, we design a reconstruction attack based on a Bayesian network model that takes into account the statistical relationships between relatives' genomes, and genome and phenotypes. We further extend this model to a factor graph representation in order to also consider dependencies betweeen SNPs within the same genome. In order to infer the values of the unknown SNPs in linear complexity, we make use of the belief propagation algorithm, run either on a junction tree (which is a transformation of the Bayesian network that removes its loops), or on the factor graph. In the latter case, as the factor graph contains loops, the algorithm is carried out multiple times until the probability distributions converge to a stable state. Then, using various metrics, we quantify the genomic privacy of individuals and show the decrease in their level of genomic privacy caused by the published genomes of their family members. We also quantify the health privacy of the individuals by considering their (genetic) predisposition to certain serious diseases. We evaluate the proposed inference attacks and show their efficiency and accuracy by using real genomic data of a pedigree. More importantly, by using genomic and phenotypic data and pedigree information collected from a public genome-sharing website and an OSN, we show that inference attacks threaten not only the Lacks family, but also many other families.

In an earlier work, inspired from the Henrietta Lacks story, we proposed an inference attack and a technique to quantify kin genomic privacy [Humbert et al. 2013] and showed the extent of the threat. Here, we expand this work in many aspects. Different from our previous work, in this paper, we have the following contributions:

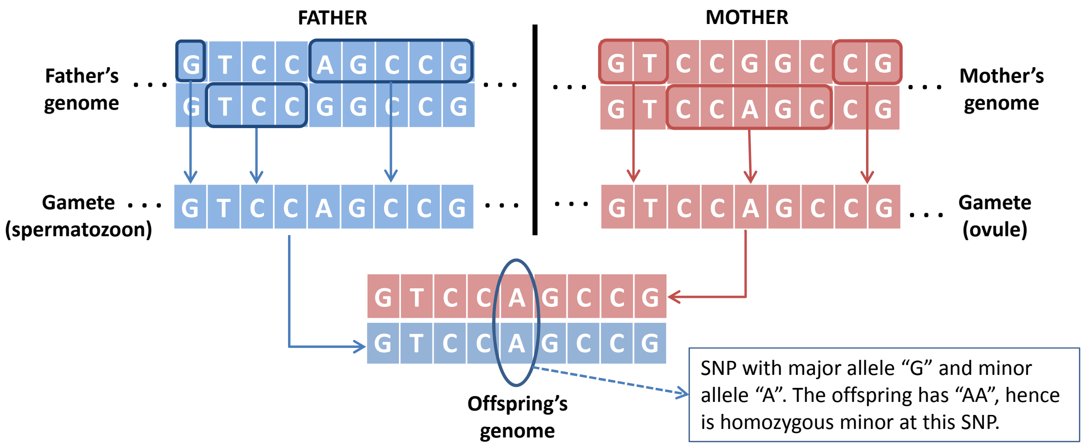
- We present a new framework for the inference attack that only considers the genomic correlations between familial members. We show that this new framework enables to perform exact inference in a single iteration of our belief propagation algorithm. We also include analytical and empirical evaluations of its computational complexity.
- We add a new layer to this new framework that enables to take additional information about relatives' phenotypes into account to improve the inference attack.
- We update the results of the inference attack by conducting several new experiments under various new settings.
- We thoroughly evaluate the relation between various metrics, also with respect to the success rate, and draw conclusions about the most appropriate metric in different settings.
- We carry out new experiments by making use of phenotypic information disclosed by OpenSNP users in combination with their genomic data.
- We include a performance evaluation, and a discussion about the potential improvements of the proposed inference attacks.

2. BACKGROUND

In this section, we briefly introduce the relevant genetic principles, as well as some important tools for modeling data dependencies and running inference efficiently.

2.1. Genomics 101

DNA is a double-helix structure that consists of two complementary polymer chains. Genetic information is encoded on the DNA as a sequence of nucleotides (A,T,G,C) and a human DNA includes around 3 billion nucleotide pairs. With the decreasing cost of DNA sequencing, genomic data is currently being used mainly in the following two ar-



SNP with major allele "G" and minor allele "A". The offspring has "AA", hence is homozygous minor at this SNP.

Fig. 1. Reproduction and single nucleotide polymorphism (SNP). 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 SNP circled on the offspring's genome is homozygous-minor for the offspring but heterozygous for the parents.

eas: (i) clinical diagnostics, for personalized genomic medicine and genetic research (e.g., genome-wide association studies[4]), and (ii) direct-to-consumer genomics, for genetic risk estimation of various diseases or for recreational activities such as ancestry search. In the following, we briefly introduce some concepts, which we use throughout this paper, about the human genome and reproduction.

2.1.1. Single Nucleotide Polymorphism. As already mentioned, human beings have 99.9% of their DNA in common. Thus, there is no need to focus on the whole DNA but rather on the variants. 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 (as illustrated in Fig. 1). There are approximately 50 million SNP positions in human population.[5] Recent discoveries show that the susceptibility of an individual to several diseases can be computed from his SNPs [Johnson and O'Donnell 2009]. For example, it has been reported that two particular SNPs (rs7412 and rs429358) on the Apolipoprotein E (ApoE) gene indicate an (increased) risk for Alzheimer's disease. SNPs carry privacy-sensitive information about individuals' health, hence we will quantify health privacy focusing on individuals' published (or inferred) SNPs and the diseases they reveal.

Two different nucleotides (called alleles) can usually be observed at a given SNP position: (i) the major allele is the most frequently observed nucleotide, and (ii) the minor allele is the rare nucleotide.[6] For each SNP position, we represent the major allele as B, and the minor allele as b (where both B and b are in $\{A, T, G, C\}$).

Furthermore, each SNP position contains two nucleotides (one inherited from the mother and one from the father, as we will discuss next). Thus, the content of a SNP position can be in one of the following states: (i) BB (*homozygous-major* genotype), if an individual receives the same 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. For simplicity of presentation, in the rest of the paper, we encode BB with 0, Bb with 1, and bb with 2. Finally, each SNP g_i is assigned a minor allele frequency (MAF), p^i_{maf}, which

[4] Examination of many genetic variants in different individuals to determine if any variant is associated with a trait.

[5] http://www.ncbi.nlm.nih.gov/projects/SNP/

[6] The two alleles for the SNP position highlighted in Fig. 1 are G and A.

Table I. Mendelian inheritance probabilities $\mathcal{F}_R(\mathbf{X}_M^i, \mathbf{X}_F^i, \mathbf{X}_C^i)$ for a SNP g_i, given the genotypes of the parents. The probabilities of the child's genotype is represented in parentheses. Each table entry represents $(P(\mathbf{X}_C^i = 0|\mathbf{X}_M^i, \mathbf{X}_F^i), P(\mathbf{X}_C^i = 1|\mathbf{X}_M^i, \mathbf{X}_F^i), P(\mathbf{X}_C^i = 2|\mathbf{X}_M^i, \mathbf{X}_F^i))$.

		Father		
		$\mathbf{X}_F^i$ = 0	$\mathbf{X}_F^i$ = 1	$\mathbf{X}_F^i$ = 2
Mother	$\mathbf{X}_M^i$=0	(1,0,0)	(0.5,0.5,0)	(0,1,0)
	$\mathbf{X}_M^i$=1	(0.5,0.5,0)	(0.25,0.5,0.25)	(0,0.5,0.5)
	$\mathbf{X}_M^i$=2	(0,1,0)	(0,0.5,0.5)	(0,0,1)

represents the frequency at which the minor allele b of the corresponding SNP occurs in a given population (typically, $0 < p_{\text{maf}}^i < 0.5$).

2.1.2. Reproduction. Mendel's First Law states that alleles are passed independently from parents to children for different meioses (the process of cell division necessary for reproduction). For each SNP position, a child inherits one allele from his mother and one from his father, as shown in Fig. 1. Each allele of a parent is passed on to a child with equal probability of 0.5. Let $\mathcal{F}_R(\mathbf{X}_M^i, \mathbf{X}_F^i, \mathbf{X}_C^i)$ be the function modeling the Mendelian inheritance for a SNP g_i, where M, F, and C represent mother, father, and child, respectively. We illustrate the Mendelian inheritance probabilities in Table I.

Based on $\mathcal{F}_R(\mathbf{X}_M^i, \mathbf{X}_F^i, \mathbf{X}_C^i)$, we can say that, given both parents' genomes, a child's genome is conditionally independent of all other ancestors' genomes.

2.1.3. Linkage Disequilibrium. As we discussed before, DNA sequences are highly correlated between close relatives, but there also exist correlations between different SNPs in the DNA. Linkage disequilibrium (LD) [Falconer and Mackay 1996] defines a correlation that appears between any pair of SNP in the whole genome due to the population's genetic history. Because of LD, the content of a SNP can be inferred from the contents of other SNPs.

For example, assume that g_i and g_j are in LD with each other. Let (A_1, A_2) and (B_1, B_2) be the potential alleles for SNP g_i and g_j, respectively. Further, let (p_1, p_2) and (q_1, q_2) be the allele probabilities of (A_1, A_2) and (B_1, B_2), respectively, provided by population statistics. That is, the probability that an individual in a given population will have allele A_1 at SNP g_i is p_1, and so on. If there were no LD (i.e., if g_i and g_j were independent), the probability that an individual would have both A_1 and B_1 at g_i and g_j would be p_1q_1. However, due to correlations between g_i and g_j, this probability is in reality equal to $p_1q_1 + D$, where D represents the discrepancy between the probability computed under independence assumption between the two SNPs and the probability in a given population. In Table II, we illustrate this LD relationship for all possible combinations of (A_1, A_2) and (B_1, B_2). We note that D can be either negative or positive, depending on the LD values.

Table II. Linkage disequilibrium (LD) between two SNPs g_i and g_j with potential alleles (A_1, A_2) and (B_1, B_2), respectively.

	$A_1, P(A_1) = p_1$	$A_2, P(A_2) = p_2$
B_1, $P(B_1) = q_1$	$P(A_1B_1) = p_1q_1 + D$	$P(A_2B_1) = p_2q_1 - D$
B_2, $P(B_2) = q_2$	$P(A_1B_2) = p_1q_2 - D$	$P(A_2B_2) = p_2q_2 + D$

2.2. Probabilistic Inference

In this subsection, we introduce the mathematical models and algorithms that form the basis of efficient inference methods.

2.2.1. Probabilistic Graphical Models. Probabilistic graphical models are very appropriate models to represent dependencies between random variables [Koller and Friedman 2009]. Such graph-based models can express conditional dependencies (e.g., Bayesian networks), joint dependencies (e.g., Markov random fields), or both (e.g., chain graphs). In graphical models, each node represents a random variable and arrows represent the dependencies between them. Such models are very useful to represent the factorization of the joint distribution of a large set of random variables, and then dramatically reduce the complexity of, e.g., the computation of marginal probabilities. If the graphical model contains loops or cycles,[7] it is possible to eliminate these by clustering variables into single nodes (called cliques) and build a maximum spanning tree (called junction or clique tree [Jensen and Jensen 1994]) of cliques. A more generic model that can represent both directed and undirected graphs is the factor graph. Contrary to the junction tree, it enables to find approximate solutions in situations where exact inference is computationally intractable. A factor graph is a bipartite graph with one set of vertices representing the random variables and the other set representing the (local) functions that factor the (global) joint probability function (based on the dependencies between the variables). A variable node is connected to a factor node if and only if the variable is an argument of the local function corresponding to the factor node.

2.2.2. Belief Propagation. Belief propagation [Pearl 1988] is a message-passing algorithm for performing inference on graphical models. It is also known as the sum-product algorithm [Kschischang et al. 2001]. It is typically used to compute marginal distributions of unobserved variables conditioned on observed ones. Computing marginal distributions is hard in general as it might require summing over an exponentially large number of terms. The belief propagation algorithm applies on various types of graphical models, such as Bayesian networks or Markov random fields. If the underlying graphical model contains no (directed or undirected) cycle, the belief propagation algorithm leads to exact inference, i.e. exact posterior marginal probabilities given the observed variables. If the graphical model is not a tree or polytree (not cycle-free), we can either transform it into a junction tree and then run belief propagation on it and get the exact solution, or perform *loopy* belief propagation which yields an approximate solution [Murphy et al. 1999]. The second approach is typically used when the junction tree approach is computationally intractable, and often gives good approximate results. Belief propagation is commonly used in artificial intelligence and information theory. It has demonstrated empirical success in numerous applications including LDPC codes [Pishro-Nik and Fekri 2004], reputation management [Ayday and Fekri 2012a; Ayday and Fekri 2012b], and recommender systems [Ayday et al. 2012].

As factor graphs are the most generic representation of graphical models, we will explain the generic belief propagation algorithm on them.[8] We assume that the joint distribution $g(x_1, \dots, x_n)$ factors into a product of several local functions, or *factors*, $f_a(\mathbf{x}_a)$:

$$g(x_1, \dots, x_n) = \prod_{a \in A} f_a(\mathbf{x}_a), \tag{1}$$

where A is a discrete index set, and $\mathbf{x}_a$ is a subset of $\{x_1, \dots, x_n\}$. The belief propagation algorithm simply works by passing messages between the $|A|$ factor nodes (representing the factors $f_1(\mathbf{x}_1)$ to $f_{|A|}(\mathbf{x}_{|A|})$) and the n variable nodes (representing the random variables x_1 to x_n) on the bipartite factor graph. The message $m_{a \to i}(x_i)$ from the factor node a to the variable node i can be interpreted as a statement about the relative probabili-

[7] There exists a cycle between X_1 and X_k in a graph if $X_1 = X_k$ and, for every $i = 1, \dots, k-1$, we have either a directed or undirected edge between X_i and X_{i+1} with, for at least one i, a directed edge. A loop is defined similarly except that it also allows for reverse-directed edge between X_i and X_{i+1} (i.e., directed edge between X_{i+1} and X_i). See Subsection 2.2 of [Koller and Friedman 2009] for further details.

[8] Interested readers can check [Kschischang et al. 2001] to see how it applies to other graphical models, such as Bayesian networks.

ties that i is in its different states based on the function f_a. The message $n_{i \to a}(x_i)$ from the variable node i to the factor node a can be interpreted as a statement about the relative probabilities that node i is in different states based on all the information node i has except for that based on the function f_a. The messages are updated according to the following rules [Pearl 1988; Kschischang et al. 2001]:

$$n_{i \to a}(x_i) = \frac{1}{Z} \prod_{c \in N(i) \backslash a} m_{c \to i}(x_i) \tag{2}$$

and

$$m_{a \to i}(x_i) = \sum_{\mathbf{x}_a \backslash x_i} f_a(\mathbf{x}_a) \prod_{j \in N(a) \backslash i} n_{j \to a}(x_j). \tag{3}$$

Here, $N(i) \backslash a$ denotes all the nodes that are neighbors of node i except for node a. Further, $\sum_{\mathbf{x}_a \backslash x_i}$ denotes a sum over all the variables $\mathbf{x}_a$ that are arguments of f_a except x_i. And Z is a normalization factor that is needed so that the resulting messages represent probability mass functions. At the beginning, messages are initialized as follows: $n_{i \to a}(x_i) = 1$ and $m_{a \to i}(x_i) = f_a(x_i)$. Then, at the end of the algorithm, after convergence, the (estimated) marginal distribution of x_i is given by the product of the messages received by the variable nodes:

$$P(x_i) = \frac{1}{Z} \prod_{c \in N(i)} m_{c \to i}(x_i), \tag{4}$$

where Z is such that $\sum_{x_i} P(x_i) = 1$. Note that, if the underlying graphical model is a tree, convergence can be reached after computing each message only once (for every factor and variable nodes). Otherwise, there is no guarantee of convergence to the true marginal in the general case, but there exist sufficient conditions for convergence [Mooij and Kappen 2007]. Neither is there any fixed convergence or error rates in general. We describe how many iterations of message computation for every nodes are needed in our context in Subsections 3.4 and 6.1. Finally, note that exact and approximate marginalization is NP-hard in general, but it can be solved in linear time in the number of factor nodes (or variable nodes) in our genomic setting. We refer the reader to Subsection 3.4 for more details on the computation complexity in our setting.

3. THE PROPOSED FRAMEWORK

In this section, we formalize our approach and present the different components that will allow us to quantify kin genomic privacy. Fig. 2 gives an overview of the framework.

3.1. Notations and Definitions

The SNPs of all relatives are represented by the random variable $\mathbf{X}$ that takes value in the set $\mathcal{X} = \{0, 1, 2\}^{n \times m}$, where n is the number of relatives in the targeted family and m is the number of SNPs in a single DNA sequence. Moreover, the hidden SNPs are represented by the random variable $\mathbf{X_H}$ (that takes value in the set $\mathcal{X}_H$), and the SNPs observed by the adversary by the random variable $\mathbf{X_O}$ (that takes value in the set $\mathcal{X}_O$). We define $\mathcal{R} = \{r_1, r_2, \ldots, r_n\}$ to be the set of relatives in the targeted family (whose family tree, showing the familial connections between the relatives, is denoted as $\mathcal{T}$) and $\mathcal{G} = \{g_1, g_2, \ldots, g_m\}$ to be the set of SNPs (i.e., positions on the DNA sequence). Let $\mathbf{X}^i_j$, respectively $x^i_j \in \{0, 1, 2\}$, represent the random variable representing SNP g_i of individual r_j, respectively its value. Furthermore, we let $\mathbf{x}_i = \left[x^1_i \; x^2_i \cdots \; x^m_i\right]$ represent the values of the SNPs of individual r_i, and $\mathbf{x} \in \mathcal{X}$ be the $n \times m$ matrix representing the

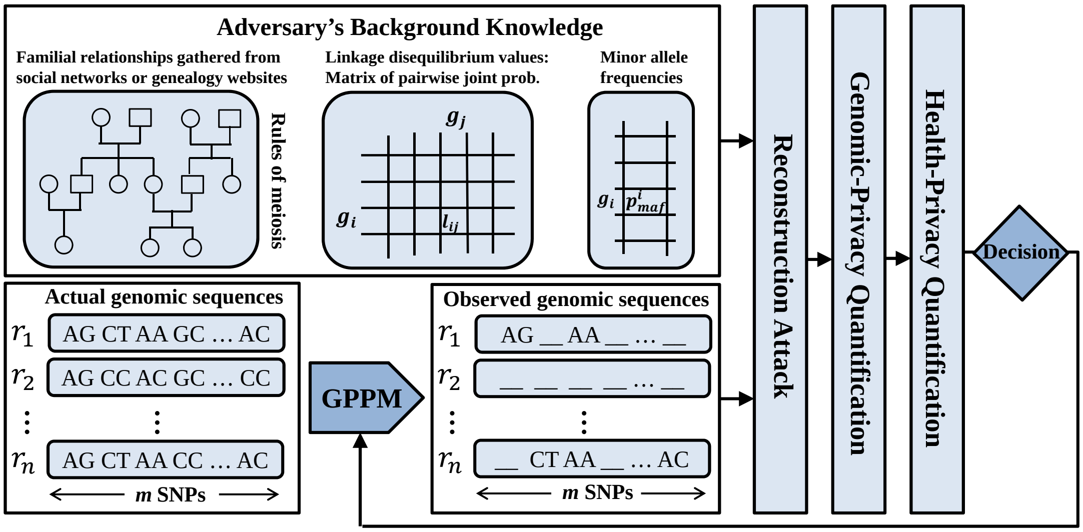


Fig. 2. Overview of the proposed framework to quantify kin genomic privacy. Each vector $\mathbf{x}_j$ ($j \in \{1, \ldots, n\}$) includes the set of SNPs for an individual in the targeted family. Furthermore, SNP g_i of relative r_j is represented by $x^i_j \in \{0, 1, 2\}$. Given its health and genomic privacy, the family should ideally decide whether to reveal less or more of their genomic data via the genomic-privacy preserving mechanism (GPPM).

values of the SNPs of all relatives:

$$\mathbf{x} = \begin{bmatrix} x^1_1 & x^2_1 & \cdots & x^m_1 \\ x^1_2 & x^2_2 & \cdots & x^m_2 \\ \vdots & \vdots & \ddots & \vdots \\ x^1_n & x^2_n & \cdots & x^m_n \end{bmatrix} \tag{5}$$

$\mathcal{F}_R(\mathbf{X}^i_M, \mathbf{X}^i_F, \mathbf{X}^i_C)$ is the function representing the Mendelian inheritance probabilities (in Table I), where M, F, C represent mother, father, and child, respectively. The $m \times m$ matrix L represents the pairwise linkage disequilibrium (LD) values between the SNPs in $\mathcal{G}$, that can be expressed by r^2; l_{ij} refers to the matrix entry at row i and column j. $l_{ij} > 0$ if i and j are in LD, and $l_{ij} = 0$ if these two SNPs are independent (i.e., there is no LD between them). The m-size vector $\mathbf{p}_{\text{maf}} = \left[p^1_{\text{maf}}\, p^2_{\text{maf}} \cdots p^m_{\text{maf}}\right]$ represents the minor allele probabilities/frequencies (MAFs) of the SNPs in $\mathcal{G}$. Finally, note that, for any $r_k \in \mathcal{R}$, $g_i \in \mathcal{G}$, and $g_j \in \mathcal{G}$, the joint probability $P(\mathbf{X}^i_k, \mathbf{X}^j_k)$ can be derived from l_{ij}, p^i_{maf}, and p^j_{maf}.

The adversary carries out a reconstruction attack to infer the value $\mathbf{x}_\mathbf{H} \in \mathcal{X}_H$ by relying on his background knowledge, $\mathcal{F}_R(\mathbf{X}^i_M, \mathbf{X}^i_F, \mathbf{X}^i_C)$, L, $\mathbf{p}_{\text{maf}}$, and on his observation $\mathbf{x}_\mathbf{O} \in \mathcal{X}_O$.[9] After carrying out this reconstruction attack, we evaluate genomic and health privacy of the family members based on the adversary's success and his certainty about the targeted SNPs and the predispositions to diseases they reveal. Finally, we discuss some ideas to preserve the individuals' genomic and health privacy.

3.2. Adversary Model

An adversary is defined by his objective(s), attack(s), and knowledge. The objective of the adversary is to compute the values of the targeted SNPs for one or more members of a targeted family by using (i) the available genomic data of one or more family members, (ii) the familial relationships between the family members, (iii) the rules of reproduction

[9] $\mathbf{x}_\mathbf{O}$ is constructed by replacing hidden SNPs in $\mathbf{x}$ by $\perp$.

(in Section 2.1.2), (iv) the minor allele frequencies (MAFs) of the nucleotides, and (v) the population LD values between the SNPs. We note that (i) and (ii) can be gathered online from genome-sharing websites and OSNs, and (iii), (iv), and (v) are publicly known information. Note that, in the future, the increasing possibility to accurately sequence, and to impute the actual haplotypes carried by an individual in each of the copies of the diploid genome will allow a more accurate inference of relatives' genotype than relying on population LD patterns only.

Various attacks can be launched, depending on the adversary's interest. The adversary might want to infer one particular SNP of a specific individual (targeted-SNP-targeted-relative attack) or one particular SNP of multiple relatives in the targeted family (targeted-SNP-multiple-relatives attack) by observing one or more other relatives' SNP at the same position. Furthermore, the adversary might also want to infer multiple SNPs of the same individual (multiple-SNP-targeted-relative attack) or multiple SNPs of multiple family members (multiple-SNP-multiple-relatives attack) by observing SNPs at various positions of different relatives. In this paper, we propose an algorithm that implements the latter attack, from which any other attacks can be carried out. We formulate this attack as a statistical inference problem.

3.3. Inference Attack

We formulate the reconstruction attack (on determining the values of the targeted SNPs) as finding the marginal probability distributions of the random variable $\mathbf{x_H}$ representing the hidden SNPs, given the observed values $\mathbf{x_O}$, familial relationships $\mathcal{T}$, and the publicly available statistical information. We represent the marginal distribution of a SNP g_i for an individual r_j as $P(\mathbf{X}_j^i = x_j^i | \mathbf{X_O} = \mathbf{x_O})$.

These marginal probability distributions could traditionally be extracted from $P(\mathbf{X_H} = \mathbf{x_H} | \mathbf{X_O} = \mathbf{x_O}, \mathcal{F}_R(\mathbf{X}_M^i, \mathbf{X}_F^i, \mathbf{X}_C^i), L, \mathcal{T}, \mathbf{p}_{\text{maf}})$, which is the joint probability distribution function of of the hidden SNPs, given the available side information and the observed SNPs. Then, clearly, each marginal probability distribution could be obtained as follows:

$$P(\mathbf{X}_j^i = x_j^i | \mathbf{X_O} = \mathbf{x_O}) = \sum_{\mathbf{x_{H'}} \in \mathcal{X}_H \setminus \mathcal{X}_j^i} P(\mathbf{X_{H'}} = \mathbf{x_{H'}}, \mathbf{X}_j^i = x_j^i | \mathbf{X_O} = \mathbf{x_O}, \mathcal{F}_R, L, \mathcal{T}, \mathbf{p}_{\text{maf}}), \quad (6)$$

where $\mathbf{X_{H'}}$ is the random variable representing all hidden SNPs except SNP g_i of relative r_j. However, the number of terms in (6) grows exponentially with the number of variables, making the computation infeasible considering the scale of the human genome (which includes tens of million of SNPs). In the worst case, the computation of the marginal probabilities has a complexity of $\mathrm{O}(3^{nm})$. Thus, we propose to factorize the joint probability distribution function into products of simpler local functions, each of which depends on a subset of variables. These local functions represent the dependences (due to LD and reproduction) between the different SNPs in x. Then, by running the belief propagation algorithm on graphical models, we can compute the marginal probability distributions in linear complexity (with respect to both n and m).

We present first the inference attack that takes only the familial correlations into account, which enables to efficiently perform exact inference, and then present the model where both familial and LD correlations are considered. The former attack is typically sufficient if the adversary has access to the full set of SNPs of interest of the target's relatives, whereas the latter can improve the attack's accuracy if the adversary does not observe all SNPs of interest in the genomes of the target's family members. For the second inference attack, due to the number and type of correlations, and the subsequent complexity of performing exact inference, we make use of loopy belief propagation that provides an approximate solution.

3.3.1. Inference Attack Without LD correlations. Under the assumption that there is no LD correlation between SNPs, the random variables $\mathbf{X}^i$'s representing a column of matrix x

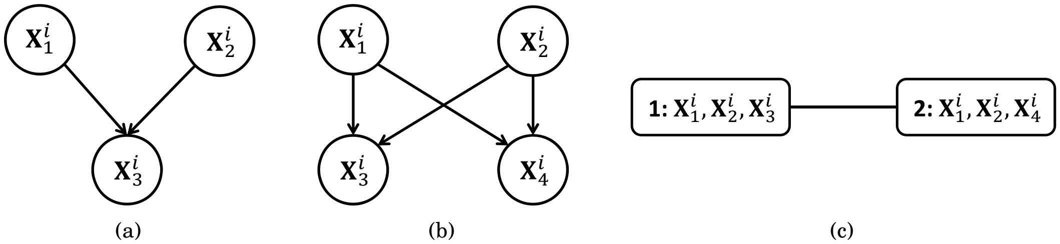


Fig. 3. Graphical models representing familial dependencies. (a) Bayesian network representing a trio (mother, father and child), (b) Bayesian network with two parents and two siblings, (c) junction tree (made of two cliques) corresponding to the Bayesian network in (b).

are independent between each other, i.e. $\mathbf{X}^i \perp \mathbf{X}^j, \forall g_i, g_j \in \mathcal{G}, g_i \neq g_j$. We can then express the marginal distribution of $\mathbf{X}_j^i$ in 6 as

$$P(\mathbf{X}_j^i = x_j^i | \mathbf{X}_{\mathbf{O}}^i = \mathbf{x}_{\mathbf{O}}^i) = \sum_{\mathbf{x}_{\mathbf{H}'}^i \in \mathcal{X}_{\mathbf{H}'}^i \setminus \mathcal{X}_j^i} P(\mathbf{X}_{\mathbf{H}'}^i = \mathbf{x}_{\mathbf{H}'}^i, \mathbf{X}_j^i = x_j^i | \mathbf{X}_{\mathbf{O}}^i = \mathbf{x}_{\mathbf{O}}^i, \mathcal{F}_R, \mathcal{T}, \mathbf{p}_{\text{maf}}), \quad (7)$$

where the set $\mathcal{X}_{\mathbf{H}'}^i$ is of maximal size 3^{n-1}, which can still be computationally intractable if we deal with a large family. However, contrary to the general case, we can here compute the exact marginal distributions in linear time by modeling the various dependencies with a Bayesian network framework and applying the junction tree algorithm on it. In general, due to Mendelian inheritance laws, the joint distribution $P(\mathbf{X}^i)$ can be factored as follows:

$$P(\mathbf{X}^i) = \prod_{r_j \in \textit{founders}} P(\mathbf{X}_j^i) \prod_{r_k \in \mathcal{R} \setminus \textit{founders}} P(\mathbf{X}_k^i | \mathbf{X}_{m(k)}^i, \mathbf{X}_{f(k)}^i), \quad (8)$$

where the *founders* are the relatives who have no ancestor in the family tree $\mathcal{T}$, and $m(k)$, $f(k)$ are the indices of the mother, respectively the father, of r_k. $P(\mathbf{X}_j^i)$ is given by the minor allele frequencies $\mathbf{p}_{\text{maf}}$, and $P(\mathbf{X}_k^i | \mathbf{X}_{m(k)}^i, \mathbf{X}_{f(k)}^i)$ by the Mendelian inheritance probabilities $\mathcal{F}_R(\mathbf{X}_M^i, \mathbf{X}_F^i, \mathbf{X}_C^i)$ in Table I. Fig. 3 shows an example of a trio (mother, father and child), which is also the main basic building block of our Bayesian-network representation of familial genetic dependencies. In this example, the joint distribution in (8) can be factored as $P(\mathbf{X}^i) = P(\mathbf{X}_1^i)P(\mathbf{X}_2^i)P(\mathbf{X}_3^i | \mathbf{X}_1^i, \mathbf{X}_2^i)$. As mentioned in Section 2.2, we can efficiently compute the exact marginal distributions on polytrees by using belief propagation. However, as soon as sibling relationships appear in the family tree $\mathcal{T}$, the underlying Bayesian network is not a polytree anymore[10] and the belief propagation does not necessarily converge to the exact marginal probabilities. In this case, in order to perform exact inference, we first need to transform the Bayesian network into a junction tree. Fig. 3(b) and 3(c) show a simple example of a Bayesian network with undirected cycles and its corresponding junction tree.

The procedure to construct the junction tree is as follows. First, we have to transform the directed graph into an undirected one, and *moralize* it, i.e. connect all unconnected parents (nodes that have outgoing edges connecting the same node in the directed graph). Second, we triangulate the resulting undirected graph, meaning that we remove all cycles containing four nodes or more by connecting some of these nodes together. More precisely, for any given cycle in the undirected graph, this step creates an edge between any two non-successive nodes in the cycle. This step is not needed in our genetic case because all cycles are already of length 3. Third, we remove cycles by clustering nodes belonging to

[10] Its underlying undirected graph is not a tree (it contains a loop made of the siblings and their parents).

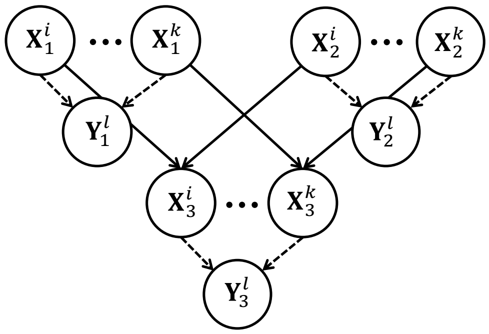


Fig. 4. Bayesian network representing a trio (mother, father and child), and two SNPs g_i and g_k influencing a disease l.

the same cycle into cliques. In this process, it is important to build cliques with the smallest number of variables[11] to minimize the inference computational burden. In our case, all cliques will be of size 3 (representing mother-father-child). Then, all cliques sharing the same variables are still connected by edges, which usually yields a loopy graph. In order to remove these cycles, we form a maximum spanning tree of cliques and ensure that if a variable is in two cliques then it is in every clique along the path connecting the two cliques. If this property holds, local propagation of information will lead to global consistency. Finally, we apply the belief propagation algorithm on the resulting junction tree, first passing messages[12] upward, from the leaves to the root, and then downward, from the root to the leaves, which eventually provides the marginal probabilities of all cliques. If we are interested in the marginal probability of a given variable in a clique, we simply sum all other variables in the clique out.

3.3.2. Inference Attack With Phenotypic Information. It could also happen that the adversary gets access to phenotypic data, such as physical traits or diseases. Such data can be found online, on health-related social networks (such as PatientsLikeMe or OpenSNP) or traditional online social networks. We show here how the Bayesian network framework can be easily expanded to take this type of information into account in our inference attack.

Fig. 4 illustrates how phenotypic nodes can be included in the Bayesian network representing a single SNP in Fig. 3(a). This updated Bayesian network shows two SNPs, g_i and g_k of a trio, and a single phenotype l. Hence, here it is assumed that two SNPs influence directly the phenotype, but there could be from one to many depending on the phenotype. The new layer of phenotypic information adds a number of nodes in the Bayesian network equal to n times the total number of phenotypic traits/diseases. Assuming a single phenotype is observed, influenced by two SNPs, the general joint distribution presented in (8) is updated as follows:

$$P(\mathbf{X}^i, \mathbf{X}^k, \mathbf{Y}^l) = \prod_{r_p \in \, founders} P(\mathbf{X}_p^i) P(\mathbf{X}_p^k) \prod_{r_c \in \mathcal{R} \setminus founders} P(\mathbf{X}_c^i | \mathbf{X}_{m(c)}^i, \mathbf{X}_{f(c)}^i) P(\mathbf{X}_c^k | \mathbf{X}_{m(c)}^k, \mathbf{X}_{f(c)}^k) \\ \times \prod_{r_j \in \mathcal{R}} P(\mathbf{Y}_j^l | \mathbf{X}_j^i, \mathbf{X}_j^k) P(\mathbf{Y}_j^l | \mathbf{X}_j^i, \mathbf{X}_j^k). \quad (9)$$

[11] Note that the size of the largest clique is called the treewidth and determines the complexity of the algorithm (which is exponential in the treewidth).

[12] The messages are constructed similarly to rule (3) depicted in Subsection 2.2.2.

The resulting Bayesian network is not a polytree if it includes sibling relationships or phenotypes influenced by more than a SNP. In this case, as explained in Subsection 3.3.1, we have to first transform the Bayesian network into a junction tree. The process is the same as in the case without phenotypic data. After the moralization step (where graphical parents are connected together), all cycles are also of length three, including those induced by the phenotype nodes. We use this framework with OpenSNP data in Subsection 5.2.

3.3.3. Inference Attack With LD Correlations. Once we take into account correlations within the same genomic sequence, the Bayesian network representation does not fit well as it cannot represent undirected dependencies, such as the pairwise joint probabilities given by LD. Also, constructing a junction tree in a Bayesian containing many cycles because of new nodes representing LD correlations would be probably untractable. A factor graph model is better suited as it can take both conditional and joint local probabilities into account. It is a bipartite graph containing two sets of nodes (corresponding to variables and factors) and edges connecting these two sets. Following [Kschischang et al. 2001], we form a factor graph by setting a variable node for each SNP x^i_j for each random variable $\mathbf{X}^i_j$ ($g_i \in \mathcal{G}$ and $r_j \in \mathcal{R}$). We use two types of factor nodes: (i) *familial factor node*, representing the familial relationships and reproduction, and (ii) *LD factor node*, representing the LD relationships between the SNPs. Note that our factor graph will contain loops because of LD nodes and sibling relationships (if any). We summarize the connections between the variable and factor nodes below (Fig. 5):

- Each variable node x^i_j has its familial factor node f^i_j to which it is connected. Furthermore, x^i_k ($k \neq j$) is also connected to f^i_j if k is the mother or father of j (in $\mathcal{T}$). Thus, the maximum degree of a familial factor node is 3.
- Variable nodes x^j_i and x^m_i are connected to a LD factor node $h^{j,m}_i$ if SNP g_j is in LD with SNP g_m. Since the LD relationships are pairwise between the SNPs, the degree of a LD factor node is always 2.

Given the conditional dependences given by reproduction and LD, the global distribution $P(\mathbf{X_H} = \mathbf{x_H} | \mathbf{X_O} = \mathbf{x_O}, \mathcal{F}_R(\mathbf{X}^i_M, \mathbf{X}^i_F, \mathbf{X}^i_C), L, \mathcal{T}, \mathbf{p}_{\text{maf}})$ can be factorized into products of several local functions, each having a subset of variables from $\mathbf{x}$ as arguments:

$$P(\mathbf{X_H} = \mathbf{x_H} | \mathbf{X_O} = \mathbf{x_O}, \mathcal{F}_R(\mathbf{X}^i_M, \mathbf{X}^i_F, \mathbf{X}^i_C), L, \mathcal{T}, \mathbf{p}_{\text{maf}}) = \frac{1}{Z}\Big[\prod_{g_i\in\mathcal{G}}\prod_{r_j\in\mathcal{R}} f^i_j(x^i_j, x^i_{m(j)}, x^i_{f(j)}, \mathcal{F}_R(\mathbf{X}^i_M,\mathbf{X}^i_F,\mathbf{X}^i_C), \mathbf{p}_{\text{maf}})\Big] \times \Big[\prod_{r_i\in\mathcal{R}} \prod_{\substack{(j,m)\text{ s.t.}\\ l_{jm}\neq 0}} h^{j,m}_i(x^j_i, x^m_i, l_{jm})\Big], \tag{10}$$

where Z is the normalization constant, and $x^i_{m(j)}$, respectively $x^i_{f(j)}$, are the SNPs g_i of the mother, respectively father, of r_i (if they exist in $\mathcal{T}$).

Next, we introduce the messages between the factor and the variable nodes to compute the marginal probability distributions using belief propagation. We denote the messages from the variable nodes to the factor nodes as μ. We also denote the messages from familial factor nodes to variable nodes as λ, and from LD factor nodes to variable nodes as β. Let $X^{(\nu)} = \{{x^i_j}^{(\nu)} : r_j \in \mathcal{R}, g_i \in \mathcal{G}\}$ be the collection of variables representing the values of the variable nodes at the iteration ν of the algorithm. The message $\mu^{(\nu)}_{i\to k}({x^i_j}^{(\nu)})$ denotes the probability of ${x^i_j}^{(\nu)} = \ell$ ($\ell \in \{0,1,2\}$), at the ν^{th} iteration. Furthermore, $\lambda^{(\nu)}_{k\to i}({x^i_j}^{(\nu)})$ denotes the probability that ${x^i_j}^{(\nu)} = \ell$, for $\ell \in \{0,1,2\}$, at the ν^{th} iteration given $x^i_{m(j)}$, $x^i_{f(j)}$,

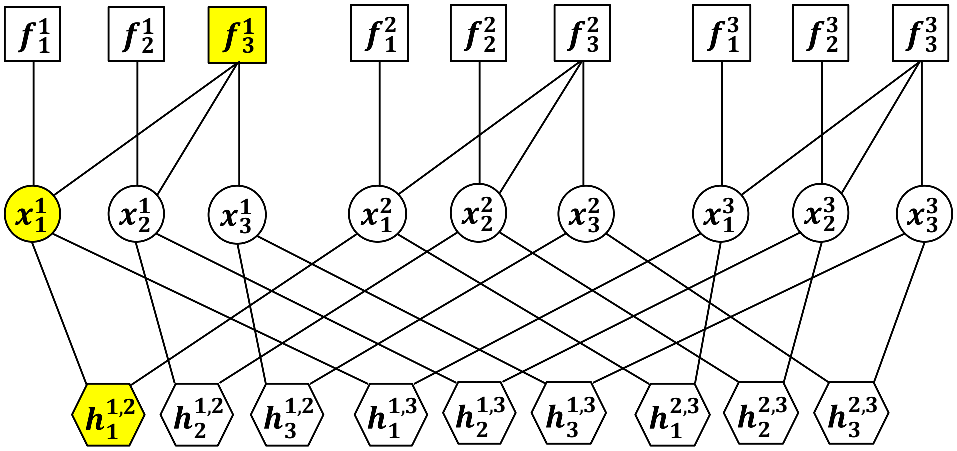


Fig. 5. The factor graph representation of a trio (mother, father, child) and 3 SNPs per family member. The square, circle, and hexagonal nodes represent the familial factor nodes, variable nodes, and LD factor nodes, respectively. The message passing described in the main text is between the nodes x_1^1, f_3^1, and $h_1^{1,2}$ highlighted in the graph.

$\mathcal{F}_R(\mathbf{X}_M^i, \mathbf{X}_F^i, \mathbf{X}_C^i)$, and $\mathbf{p}_{\text{maf}}$. Finally, $\beta_{k\to i}^{(\nu)}(x_j^{i(\nu)})$ denotes the probability that $x_j^{i(\nu)} = \ell$, for $\ell \in \{0,1,2\}$, at the ν^{th} iteration given the LD relationships between the SNPs.

For the clarity of presentation, we choose a simple family tree consisting of a trio (i.e., mother, father, and child) and 3 SNPs (i.e., $|\mathcal{R}| = 3$ and $|\mathcal{G}| = 3$). In Fig. 5, we show how the trio and the SNPs are represented on a factor graph, where r_1 represents the mother, r_2 represents the father, and r_3 represents the child. Furthermore, the 3 SNPs are g_1, g_2, and g_3. We describe the message exchange between the variable node representing the first SNP of the mother (x_1^1), the familial factor node of the child (f_3^1), and the LD factor node $h_1^{1,2}$. The belief propagation algorithm iteratively exchanges messages between the factor and the variable nodes in Fig. 5, updating the beliefs on the values (in $\mathbf{x_H}$) of the targeted SNPs at each iteration, until convergence. We denote the variable and factor nodes x_1^1, f_3^1, and $h_1^{1,2}$ with the letters i, k, and z, respectively.

The variable nodes generate their messages (μ) and send them to their neighbors. Variable node i forms $\mu_{i\to k}^{(\nu)}(x_1^{1(\nu)})$ by multiplying all information it receives from its neighbors excluding the familial factor node k.[13] Hence, the message from variable node i to the familial factor node k at the ν^{th} iteration is given by

$$\mu_{i\to k}^{(\nu)}(x_1^{1(\nu)}) = \frac{1}{Z} \times \prod_{w\in(\sim k)} \lambda_{w\to i}^{(\nu-1)}(x_1^{1(\nu-1)}) \times \prod_{y\in\{z,h_1^{1,3}\}} \beta_{y\to i}^{(\nu-1)}(x_1^{1(\nu-1)}), \tag{11}$$

where Z is a normalization constant, and the notation $(\sim k)$ means all familial factor node neighbors of the variable node i, except k. This computation is repeated for every neighbor of each variable node. It is important to note that the message in (11) is valid if the value of x_1^1 is unobserved to the adversary. However, the value of x_1^1 can also be observed by the adversary. In this case, if $x_1^1 = \rho$ ($\rho \in \{0,1,2\}$), then $\mu_{i\to k}^{(\nu)}(x_1^{1(\nu)} = \rho) = 1$ and $\mu_{i\to k}^{(\nu)}(x_1^{1(\nu)}) = 0$ for other potential values of x_1^1 (regardless of the values of the messages received by the variable node i from its neighbors).

Next, the factor nodes generate their messages. The message from the familial factor node k to the variable node i at the ν^{th} iteration is formed using the principles of belief

[13] The message $\mu_{i\to z}^{(\nu)}(x_1^{1(\nu)})$ from the variable node i to the LD factor node z is constructed similarly.

propagation as

$$\lambda^{(\nu)}_{k\to i}({x_1^1}^{(\nu)}) = \sum_{\{x_2^1,x_3^1\}} f_3^1(x_1^1, x^1_{m(1)}, x^1_{f(1)}, \mathcal{F}_R(\mathbf{X}^i_M, \mathbf{X}^i_F, \mathbf{X}^i_C), \mathbf{p}_{\mathrm{maf}}) \prod_{y\in\{x_1^2,x_1^3\}} \mu^{(\nu)}_{y\to k}({x_1^1}^{(\nu)}). \quad \textbf{(12)}$$

Note that $f_3^1(x_1^1, x^1_{m(1)}, x^1_{f(1)}, \mathcal{F}_R(\mathbf{X}^i_M, \mathbf{X}^i_F, \mathbf{X}^i_C), \mathbf{p}_{\mathrm{maf}}) \propto P(x_1^1|x^1_{m(1)}, x^1_{f(1)}, \mathcal{F}_R(\mathbf{X}^i_M, \mathbf{X}^i_F, \mathbf{X}^i_C))$, and this probability is computed using Table I. Furthermore, if the degree of the familial factor node is 1 for a particular SNP, then the local function corresponding to the familial factor node only depends on the MAF of the corresponding SNP. For example, the degree of f_1^1 (in Fig. 5(c)) is 1, hence $f_1^1(x_1^1, x^1_{m(1)}, x^1_{f(1)}, \mathcal{F}_R(\mathbf{X}^i_M, \mathbf{X}^i_F, \mathbf{X}^i_C), \mathbf{p}_{\mathrm{maf}}) \propto P(x_1^1|p^1_{\mathrm{maf}})$. The above computation must be performed for every neighbor of each familial factor node.

Similarly, the message from the LD factor node z to the variable node i at the ν^{th} iteration is formed as

$$\beta^{(\nu)}_{z\to i}({x_1^1}^{(\nu)}) = \sum_{x_1^2} g_1^{1,2}(x_1^1, x_1^2, l_{12}) \prod_{y\in\{x_1^2\}} \mu^{(\nu)}_{y\to z}({x_1^1}^{(\nu)}). \quad \textbf{(13)}$$

As before, this computation is performed for every neighbor of each LD factor node. We further note that $h_1^{1,2}(x_1^1, x_1^2, l_{1,2}) \propto P(x_1^1, x_1^2)$, which is derived from $l_{1,2}$, p^1_{maf}, and p^2_{maf}. The algorithm proceeds to the next iteration in the same way as the ν^{th} iteration.

The algorithm starts at the variable nodes. Thus, at the first iteration of the algorithm (i.e., $\nu = 1$), the variable node i sends messages to its neighboring factor nodes based on the following rules: (i) If the value of x_1^1 is hidden from the adversary, $\mu^{(1)}_{i\to k}({x_1^1}^{(1)}) = 1$ for all potential values of x_1^1 and, (ii) if the value of x_1^1 is observed by the adversary and $x_1^1 = \rho$ ($\rho \in \{0, 1, 2\}$), $\mu^{(1)}_{i\to k}({x_1^1}^{(1)} = \rho) = 1$ and $\mu^{(1)}_{i\to k}({x_1^1}^{(1)}) = 0$ for other potential values of x_1^1. The iterations stop when all variable nodes have converged to stable distributions. The marginal probability of each variable in $\mathcal{X}_H$ is given by multiplying all the incoming messages at each variable node representing an unobserved SNP, as in (4). Note that the factor graph could also embed phenotypic information by adding one factor node and one variable node per phenotype and individual. We do not present it here for the sake of clarity and conciseness.

3.4. Computational Complexity

The computational complexity of the inference without LD correlations is linear in the number of nodes n (i.e., number of family members) in the original Bayesian network, the number of SNPs m, and exponential in the treewidth, i.e., the maximum number of variables in cliques. In our case, the treewidth is 2, which is negligible compared to n and m. We can thus state that the computational complexity is $\mathrm{O}(nm)$. Note that, in general, finding an optimal triangulation ordering to construct the junction tree is NP-hard, but, in our case, all the cycles are already of size 3 after the moralization step, thus there is no need to triangulate the graph. The same analysis applies for the inference with phenotypic information. Therefore, the computational complexity increases linearly with the number of phenotypes times the number of family members sharing these phenotypes.

The computational complexity of the inference with LD correlations is proportional to the number of factor nodes. In our setting, there are nm familial factor nodes and a maximum of $nm(m-1)/2$ LD factor nodes. Hence, the worst-case computational complexity per iteration is $\mathrm{O}(nm^2)$. However, as each SNP is in LD with a limited number of other SNPs, the matrix L is sparse and the number of LD factor nodes grows with m rather than with $m(m-1)/2$, especially if we focus on SNPs in strong LD only. Thus, the average computational complexity per iteration is $\mathrm{O}(nm)$. Based on our experiments, we can state that the number of iterations before convergence is a small constant, between 7 and 15. Note finally that this complexity can be further reduced by using similar techniques developed for message-passing decoding of LDPC codes (e.g., working in log-domain [Chen et al. 2002]). We implement the proposed attack and evaluate its performance in practice in Subsection 6.1.

3.5. Privacy Metrics

A crucial step towards protecting kin genomic privacy is to quantify the privacy loss induced by the release of genomic information. Through the inference attack, the adversary infers the targeted SNPs belonging to the members of a targeted family by using his background knowledge and observed genomic data (of the family members). The inferred information can be expressed as the posterior distribution $P(\mathbf{X_H} = \mathbf{x_H} | \mathbf{X_O} = \mathbf{x_O}, \mathcal{F}_R, L, \mathcal{T}, \mathbf{p}_{\text{maf}})$. Moreover, each posterior marginal probability distribution is represented as $P(\mathbf{X}^i_j = \hat{x}^i_j | \mathbf{X_O} = \mathbf{x_O})$, $\forall r_j \in \mathcal{R}, g_i \in \mathcal{G}$.[14] We propose to quantify kin genomic privacy by measuring the expected estimation error (incorrectness) and the uncertainty of the adversary.[15]

Correctness was already proposed in the context of location privacy [Shokri et al. 2011]. In our scenario, correctness quantifies the adversary's success in inferring the targeted SNPs. That is, it quantifies the expected distance between the adversary's estimate on the value of a SNP, $\hat{x}^i_j$ and the true value of the corresponding SNP, x^i_j. This distance can be expressed as the expected estimation error as follows:

$$E^i_j = \sum_{\hat{x}^i_j \in \{0,1,2\}} P(\mathbf{X}^i_j = \hat{x}^i_j | \mathbf{X_O} = \mathbf{x_O}) \left\| x^i_j - \hat{x}^i_j \right\|. \quad (14)$$

Note that $\|.\|$ can be any norm, such as the L_1 or L_2 (Euclidean) norms. We select the L_1 norm in our evaluation as it is the most intuitive and most representative of the discrepancy we want to measure. If we rely on the Hamming distance[16] instead, the expected estimation error becomes equal to $1 - P(\hat{x}^i_j = x^i_j)$, i.e. one minus the probability of success (or success rate). We discuss this further in Subsection 4.2.

Privacy can also be represented as the adversary's *uncertainty* [Diaz et al. 2003; Serjantov and Danezis 2003], that is the ambiguity of $P(\mathbf{X}^i_j = \hat{x}^i_j | \mathbf{X_O} = \mathbf{x_O})$. This uncertainty is generally considered to be maximum if the posterior distribution is uniform. This definition of uncertainty can be quantified as the (normalized) entropy of $P(\mathbf{X}^i_j = \hat{x}^i_j | \mathbf{X_O} = \mathbf{x_O})$ as follows:

$$H^i_j = \frac{-\sum_{\hat{x}^i_j \in \{0,1,2\}} P(\mathbf{X}^i_j = \hat{x}^i_j | \mathbf{X_O} = \mathbf{x_O}) \log P(\mathbf{X}^i_j = \hat{x}^i_j | \mathbf{X_O} = \mathbf{x_O})}{\log(3)} := \frac{H(\mathbf{X}^i_j | \mathbf{X_O})}{\log(3)}. \quad (15)$$

The higher the entropy is, the higher is the uncertainty.

Finally, we propose another entropy-based metrics that quantifies the mutual dependence between the hidden genomic data that the adversary is trying to reconstruct, and the observed data. This is quantified by mutual information $I(\mathbf{X}^i_j; \mathbf{X_O}) = H(\mathbf{X}^i_j) - H(\mathbf{X}^i_j | \mathbf{X_O})$ [Agrawal and Aggarwal 2001]. As privacy decreases with mutual information, we propose the following (normalized) privacy metrics:

$$I^i_j = 1 - \frac{H(\mathbf{X}^i_j) - H(\mathbf{X}^i_j | \mathbf{X_O})}{H(\mathbf{X}^i_j)} = \frac{H(\mathbf{X}^i_j | \mathbf{X_O})}{H(\mathbf{X}^i_j)}. \quad (16)$$

We can then evaluate the genomic privacy of an individual r_j by computing the average of the per-SNP values over all SNPs $g_i \in \mathcal{G}$, for any of the three aforementioned metrics.

[14] We use here $\hat{x}^i_j$ to refer to the estimate of x^i_j.

[15] These metrics are not specific to the proposed inference attack; they can be used to quantify genomic privacy in general.

[16] $\left\| x^i_j - \hat{x}^i_j \right\| = 0$ if $\hat{x}^i_j = x^i_j$ and $\left\| x^i_j - \hat{x}^i_j \right\| = 1$ otherwise.

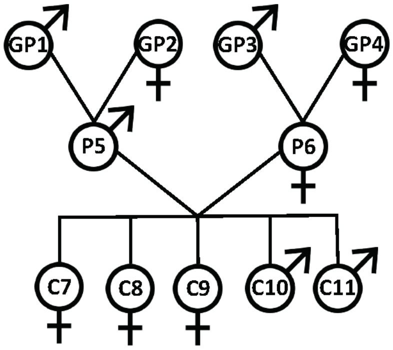


Fig. 6. Family tree of *CEPH/Utah Pedigree 1463* consisting of the 11 family members that were considered. The symbols ♂ and ♀ represent the male and female family members, respectively.

For instance, the privacy level of individual r_j using the metrics in (14) is:

$$E_j = \frac{1}{m}\sum_{i=1}^{m}\sum_{\hat{x}_j^i\in\{0,1,2\}} P(\mathbf{X}_j^i = \hat{x}_j^i|\mathbf{X_O} = \mathbf{x_O})\,\|x_j^i - \hat{x}_j^i\|. \tag{17}$$

Similar aggregated expressions can be derived for the other metrics. Finally, we also evaluate the global privacy level of a whole family in Section 4.3. In such case, we average over all SNPs $g_i \in \mathcal{G}$ and family members $r_j \in \mathcal{R}$, which gives, for the expected estimation error:

$$E = \frac{1}{nm}\sum_{j=1}^{n}\sum_{i=1}^{m}\sum_{\hat{x}_j^i\in\{0,1,2\}} P(\mathbf{X}_j^i = \hat{x}_j^i|\mathbf{X_O} = \mathbf{x_O})\,\|x_j^i - \hat{x}_j^i\|. \tag{18}$$

If we are interested in a more tangible privacy, we can also convert the per-SNP genomic-privacy metrics into health-privacy metrics. To quantify an individual's health privacy, we focus on his predisposition to different diseases. Let $\mathcal{S}_d$ 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 r_i regarding the disease d can be defined as follows:

$$D_i^d = \frac{1}{\sum_{k:g_k\in\mathcal{S}_d} c_k}\sum_{k\in\mathbf{S_d}} c_k G_i^k, \tag{19}$$

where G_i^k is the genomic privacy of a SNP g_k for individual r_i, computed using (14), (15), or (16), and c_k is the contribution of SNP g_k to disease d.[17] Other health-privacy metrics based on non-linear combinations of genotypes or combinations of alleles will be defined in future work. Note that health-privacy metrics are valid at a given time, and cannot be used to evaluate future privacy provision, as genome research can change knowledge on the contribution of SNPs to diseases.

4. EVALUATION

In this section, we first evaluate the performance of the proposed inference attack, then compare the entropy-based metrics with respect to the expected estimation error, and finally evaluate the accuracy of the inference attack with and without considering the linkage disequilibrium (LD) between SNPs.

For this evaluation, we use the *CEPH/Utah Pedigree 1463* that contains the partial DNA sequences of 17 family members (4 grandparents, 2 parents, and 11 children) [Dr-

[17] These contributions are determined as a result of medical studies. Some SNPs might increase (or decrease) the risk for a disease more than others.

manac et al. 2010]. We note in Fig. 6 that we only use 5 (out of 11) children for our evaluation because (i) 11 is much above the average number of children per family, and (ii) we observe that the strength of adversary's inference does not increase further (due to the children's revealed genomes) when more than 5 children's genomes are revealed. As the SNPs related to important diseases, like Alzheimer's, are not included in this dataset, we quantify health privacy in Section 5 by using the data collected from a genome-sharing website.

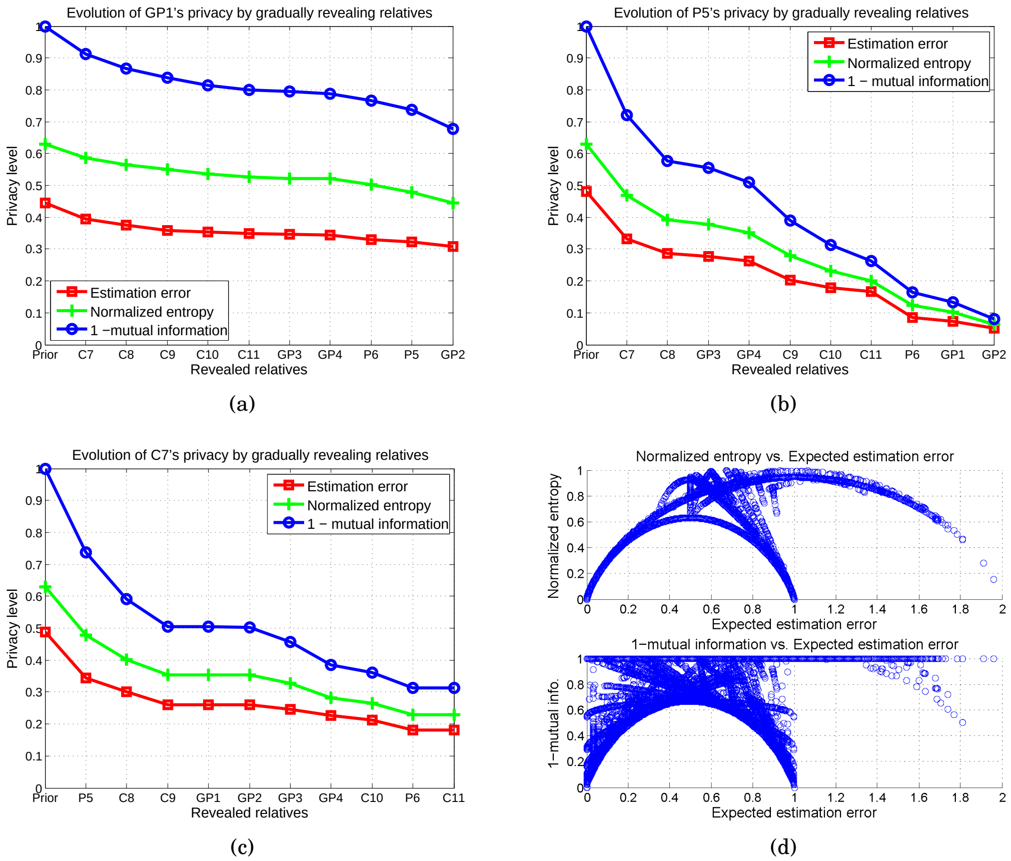
To quantify the genomic privacy of the individuals in the CEPH family, we focus on their SNPs on chromosome 1 (which is the largest chromosome). We make use of the three base metrics introduced in Section 3.5. That is, we compute the personal genomic privacy of each family member using expression (17) if we choose the expected estimation error in (14) as the base metric. We rely on the L_1 norm to measure the distance between two SNP values in (14), meaning that the distance for a single SNP can go from 0 to 2. We aggregate the per-SNP entropy-based metrics (15) and (16) by averaging them in the same way as in (17) for the estimation error. We study the relationship between these metrics in Subsection 4.2. Note that, for the inference without LD, we made use of the Matlab implementation of the junction tree algorithm provided in the Bayes Net Toolbox [Murphy et al. 2001] and, for the inference with LD, we implemented our own factor graph and loopy belief propagation algorithm in Python.

4.1. Inference Without LD Correlations

First, we assume that the adversary targets one family member and tries to infer his/her SNPs by using the published SNPs of other family members without considering the LD between the SNPs. We select an individual from the CEPH family and denote him as the target individual. We construct $\mathcal{G}$, the set of SNPs that we consider for evaluation, from all $81,899$ available SNPs on chromosome 1. Thus, the random variable $\mathbf{X_H}$ represents the hidden $81,899$ SNPs of the target individual that we want to infer. Furthermore, the random variable $\mathbf{X_O}$) represents the $81,899$ SNPs of each other family members that is observed. That is, we sequentially reveal all $81,899$ SNPs on chromosome 1 of all family members (excluding the target individual).The exact sequence of the family members (whose SNPs are revealed) is indicated on the figure of each evaluation. Note that we changed the order compared to the conference paper [Humbert et al. 2013] in order to convey new and complementary messages. In this endeavor, we also included Table III

In Fig. 7 we show the evolution of the genomic privacy of three target individuals from the CEPH family (in Fig. 6): (i) grandparent (GP1), (ii) parent (P5), and (iii) child (C7). We note that all entropy-based metrics for each target individual start from the same values. This is logical as these do not depend on the actual SNP values but only on the minor allele frequencies given by population statistics. We also observe that the parent's genomic privacy decreases to a lower level than the child's genomic privacy, which itself degrades more than the grandparent's (e.g., the adversary's error for the grandparent's genome does not go below 0.3). Compared to the graphs in [Humbert et al. 2013], the observation of GP3, GP4 and P6's genomes has an impact on GP1 and P5's privacy. This is due to the fact that here we reveal the children's genomes first, which creates a conditional probabilistic dependence between the genomes on the P5 and P6 sides of the pedigree tree.

We observe in Fig. 7(a) that the grandparent's genomic privacy is mostly affected by the SNPs of the first revealed children (C7, C8), and also by those of his spouse (GP2) and his child (P5). Table III also shows that the observation of only P5 already decreases the genomic privacy of GP1 a lot, and the observation of both P5 and GP2 decreases it to its minimal value. Hence, in some scenarios, it is not necessary to observe many relatives to threaten an individual's genomic privacy. We also observe (in Fig. 7(b)) that, by revealing all family members' SNPs (expect P5), the adversary can almost reach an estimation error of 0 about P5's genome. The target parent's genomic privacy significantly decreases ones essentially with the observation of his children's and spouse's SNPs. GP1 and GP2



(a)

(b)

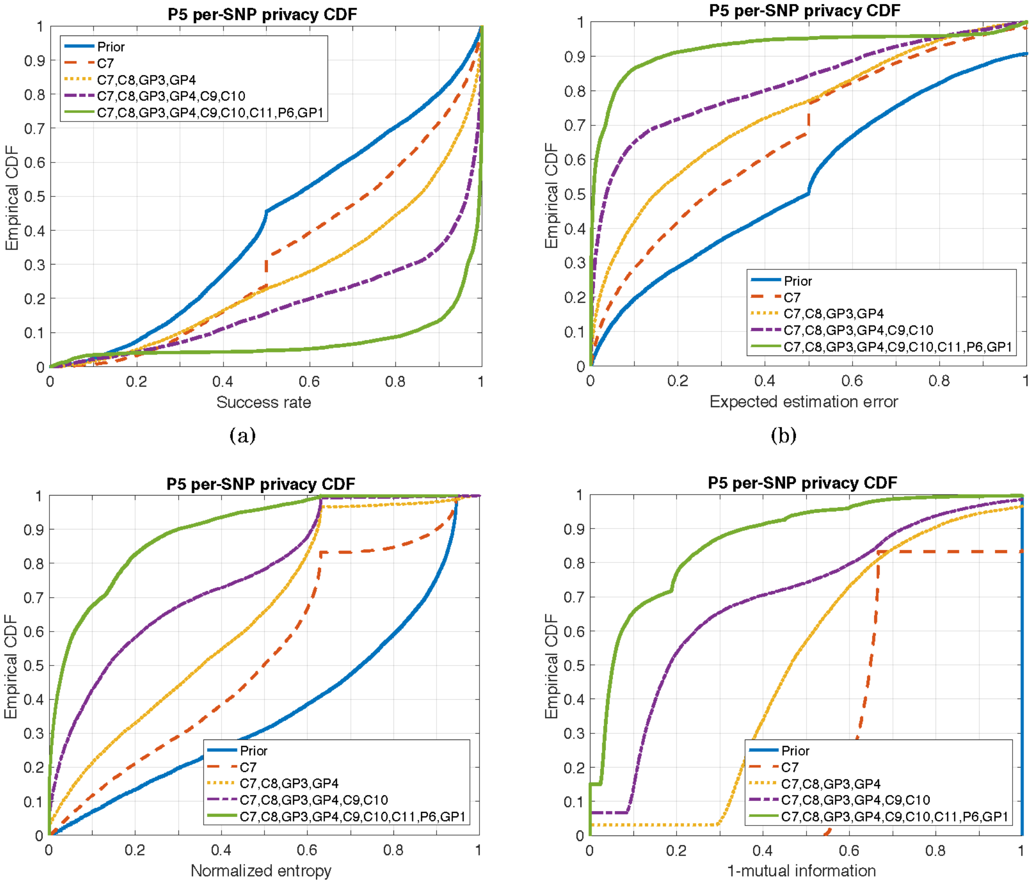
(c)

(d)

Fig. 7. Metrics for measuring personal genomic privacy. Evolution of the average genomic privacy measured with our three base metrics defined in Subsection 3.5 for the (a) grandparent (GP1), (b) parent (P5), and (c) child (C7) by gradually revealing other relatives' genomes. We reveal all the 81,899 SNPs on chromosome 1 of other family members while inferring the 81,899 SNPs of the targeted individual (GP1, P5 or C7), per-SNP metrics being aggregated as in (17). The x-axis represents the cumulative disclosure sequence. The order of disclosure has been chosen such that the results provide new insights on how relatives affect personal genomic privacy compared to previous work. We note that $x = 0$ represents the prior distribution, when no genomic data is observed by the adversary. (d) Per-SNP comparison of the two entropy-based metrics w.r.t to the expected estimation error, with data points taken from the same scenario as Fig. (c). Each point in the two plots represents the expected estimation error (x-axis) and the normalized entropy (y-axis, top) or 1-mutual information (y-axis, bottom) at a single SNP of child C7 for different amount of observed kin genomic information (from 0 to 10 relatives, as for Fig. (c)). The closer to the x=y line the points are, the closer two metrics are.

do not have so much influence, also because of the fact that they are observed in the end. Table III shows that, if we observe only GP1 and GP2, we can reduce the genomic privacy of P5 by 50%, which is more than with the observation of two children (40%), or one child and the spouse (35%).

We observe in Fig. 7(c) that C7's genomic privacy decreases already significantly with the observation of one parent (P5) and two siblings (C8 and C9). We also notice that, once P5 is known, the disclosure of GP1 and GP2's genomes has no impact on C7's privacy. In the same way, we observe that once both parents' genomes are revealed, the knowledge of an additional child's genome does not help the attacker. Indeed, as each new offspring is created independently of another (except in the case of twins), each sibling's genomic inheritance is independent of the others given his/her parents' genetic background. This is confirmed by Table III, where we see that the observation of C8 in addition of P5 and



(a) (b)

(c) (d)

Fig. 8. Empirical cumulative distribution function (CDF) of (a) the success rate, and our three base metrics: (b) expected estimation error, (c) normalized entropy, and (d) 1- (normalized) mutual information. We plot here the CDF of the per-SNP privacy levels of parent P5. We selected 5 out of the 11 disclosure scenarios of Fig. 7(b), specifically, (i) no disclosure ("prior"), disclosure (ii) of C7 only, (iii) of C7, C8, GP3, GP4, (iv) of C7, C8, C9, C10, GP3, GP4, and (v) of C7, C8, C9, C10, P6, GP1, GP3, GP4.

P6 does not change C7 privacy. Like for the other cases, Table III tells us that we can infer a lot of genomic information by knowing only a few relatives' genomes. For instance, P5's observation already reduces the privacy by 30%. Moreover, the observation of the two parents provides the minimal privacy level that C7 can expect in this scenario.

Instead of averaging the privacy levels over the whole set of SNPs in $\mathcal{G}$ following (17), Fig. 8 depict the cumulative distribution function (CDF) of the per-SNP privacy levels under five different settings of Fig. 7(b). In addition to the three base metrics, we plot also the success rate, i.e., $P(\hat{x}_j^i = x_j^i)$ (Fig. 8(a)). We first note that the success rate and the expected estimation error CDFs are ery symmetric around the diagonal. We also observe when no relative is observed, around half of the SNPs have a success rate greater than 0.5, whereas, once C7's SNPs are observed, half of the SNPs have a success rate higher than 0.7. Moreover, under no observation, only 20% of the SNPs can be guessed with success higher or equal to 0.9, whereas this percentage goes up to 65% when six of P5's relatives are observed and 87% when nine of P5's relatives are revealed. We also show the percentage of SNPs with success higher than or equal to 0.9 for different scenarios

Table III. Absolute and relative levels of genomic privacy of the grandparent (GP1), parent (P5), and child (C7) given the observation of 0 to 3 of their relatives. We use here the expected estimation error E_j to measure the genomic privacy of GP1, P5 and C7 (first two rows for each individual) but also the success rate (third row, denoted with $*$). Here we represent the percentage of SNPs for which the success rate is higher than 0.9, i.e. $P(x_j^i = \hat{x}_j^i) > 0.9$.

H\O		∅	P5	P5, GP2	C7, GP2	C7, C8, GP2
GP1	E_j	0.446	0.322	0.309	0.404	0.385
		100%	72%	69%	91%	86%
	$*$	20%	28%	29%	23%	23%
H\O		∅	GP1,GP2	C7,C8	C7,P6	GP1,GP2,C7
P5	E_j	0.48	0.242	0.286	0.312	0.203
		100%	50%	60%	65%	42%
	$*$	20%	57%	38%	29%	57%
H\O		∅	P5	P5, C8	P5, P6	P5, P6, C8
C7	E_j	0.489	0.344	0.301	0.182	0.182
		100%	70%	62%	37%	37%
	$*$	20%	28%	40%	64%	64%

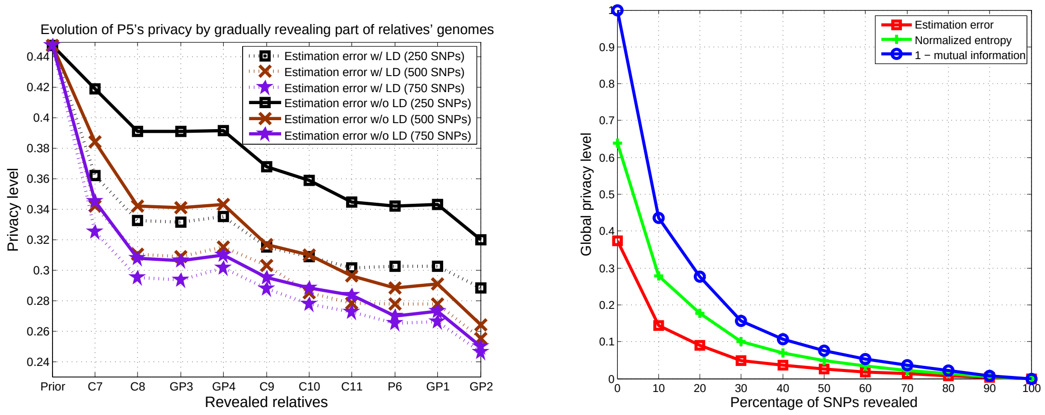
in Table III. We notice that, e.g., by observing only the two parents of P5 (GP1 and GP2), the percentage of SNPs inferred with 0.9 success increases already to 57%.

4.2. Metrics Comparison

First of all, we consider that the expected estimation error is the best metric, in the sense that it measures the distortion between the adversary's inferred SNPs' values and their actual values. Wagner states that the success rate is more intuitive in [Wagner 2015], but it is merely the opposite of the expected estimation error. As mentioned in Subsection 3.5, if we use the Hamming distance between x_j^i and x_j^i, the expected estimation error is simply one minus the success rate. Even with the L_1 norm, which we rely on because it is more accurate and appropriate for our quantification aim, we notice by comparing Fig. 8(a) and 8(b) are really symmetric and opposite. This leads us to conclude that the expected estimation error is as intuitive as the success rate and that it is the best metric for privacy measurement as it is increasing monotically with privacy, whereas the success rate is decreasing with privacy.

Despite being certainly the most appropriate metric to measure genomic privacy, the expected estimation error has a non-negligible drawback in requiring the knowledge of the ground truth of the SNP values. As we show in Section 5, this knowledge is not always available. In such case, entropy-based metrics, which measure the uncertainty rather than the error of the adversary, are certainly best alternatives. In Fig. 7(d), we compare both our entropy-based metrics with the estimation error, point by point, over all 81899 SNPs of chromosome 1 and for all values aggregated in Fig. 7(c) to measure C7's privacy evolution.

Apart from the fact that normalized entropy slightly overestimates the expected estimation error, it is growing quite similarly than the estimation error, especially in the estimation error range $[0, 0.5]$, where the majority of the points are located. We also notice that the third metric, 1- (normalized) mutual information, is worse than the normalized entropy in approximating the estimation error. This is corroborated by Fig. 8 that shows that the normalized entropy empirical CDFs are closer to those of the estimation error than the empirical CDFs of the mutual information-based metric. This motivates us to rely on the normalized entropy to measure genomic and health privacy in Section 5, when we do not know the ground truth. Note that Wagner does not directly compare the privacy metrics on a single graph, contrary to us in Fig. 7(d), neither does she make use of concrete inference attacks to evaluate her metrics [Wagner 2015].



(a) (b)

Fig. 9. Evaluation of the impact of LD correlations on genomic privacy. (a) Evolution of parent P5's privacy with and without considering LD. For each family member, we reveal 250, 500, or 750 randomly picked SNPs (among the 1000 SNPs in $\mathcal{G}$), following the same order of familial disclosure as in Figure 7(b). Privacy level in measured using the expected estimation as base metrics, with per-SNP privacy values being aggregated over the 1000 SNPs in $\mathcal{G}$ following expression (17). Note that $x = 0$ represents the prior distribution, when no genomic data is revealed. (b) Evolution of the global privacy of a family by gradually revealing 10% of its SNPs. Global privacy level measured as in expression (18) (same averaging method used for the other two base metrics shown in the figure).

4.3. Inference With LD Correlations

Next, we include the LD relationships and observe the change in the inference power of the adversary using the LD values. We construct $\mathcal{G}$ from 1000 SNPs on chromosome 1. Among these 1000 SNPs, each SNP is in LD with 13 other SNPs on average. Furthermore, the strength of the LD (r^2 value in Section 2.1.3) uniformly varies between 0.5 and 1 (where $r^2 = 1$ represents the strongest LD relationship, as discussed before). As before, we define a target individual from the CEPH family, construct the set $\mathbf{X_H}$ from his/her SNPs, and sequentially reveal other family members' SNPs to observe the decrease in the genomic privacy of the target individual. We observe that individuals sometimes do not always reveal all their genome, or disclose different parts of their genomes (e.g., different sets of SNPs). Thus, we assume that for each family member (except for the target individual), the adversary does not observe the full set of 1000 SNPs of the individuals, but only a fraction of them. We instead assume that people reveal 25%, 50% or 75% of their genomic data, and that they reveal different subsets of their SNPs. Fig. 9(a), shows the evolution of genomic privacy (measured by the expected estimation error) of parent P5 with and without making use of LD correlations. First of all, we observe that LD clearly improves the inference attack, thus decreases genomic privacy compared to the case when LD is not used. We also note that the smaller is the percentage of observed SNPs, the higher is the effect of LD correlations on P5's privacy. This is due to the fact that LD correlations help fill the missing SNPs. We also observe that the more relatives reveal their SNPs, the smaller is the gap between the privacy with and without LD.

Finally, we also evaluate the global inference power of the adversary when inferring multiple SNPs among all family members, given a subset of SNPs belonging to some family members, and also considering the LD correlations between SNPs. That is, we evaluate the inference power of the adversary for different fractions of observed data for the family members. Using a set of 100 SNPs for every family member, we construct $\mathbf{X_H}$ from $(\kappa \times 100 \times n)$ SNPs, randomly selected from all family members, where n is the number of family members in the family tree ($n = 11$ for this scenario), and $\kappa \in \{0, 0.1, \ldots, 0.9, 1\}$. We assume that the SNPs that are not in $\mathbf{X_H}$ are observed by the adversary (i.e., in $\mathbf{X_O}$), and we evaluate the inference power of the adversary for the SNPs represented by $\mathbf{X_H}$, for different values of κ. In Fig. 9(b), we observe a very fast decrease in the global genomic privacy (privacy of all family members), showing that the observation of a small portion of the family's SNPs can have a huge impact on genomic privacy. For instance, the estimation error is decreased by around 3 by observing only the first 10% of the SNPs.

5. EXPLOITING GENOME-SHARING WEBSITES

We present here two concrete attacks that can be carried out using existing genome-sharing websites and online social networks.

5.1. Cross-Website Attack with Online Social Networks

In order to show that the proposed inference attack threatens not only the Lacks family, but potentially *all* families, we collected publicly available data from a genome-sharing website and familial relationships from an OSN, and evaluated the decrease in genomic and health privacy of people due to the observation of their relatives' genomic data.

We gathered individuals' genomic data from OpenSNP, a website on which people can publicly share sets of SNPs. Then, we identified the owners of some gathered genomic profiles by using their names and sometimes profile pictures. Among these identified individuals, we managed to find family relationships of 6 of them (who publicly reveal the names of some of their relatives) on other Web resources such as Facebook.[18] We expect this number to increase in the future, as more health-related OSNs (which let people share their genomic profiles, such as 23andMe) emerge. Furthermore, we anticipate that the current widely used health-related OSNs (e.g., PatientsLikeMe[19]) will let users upload and share their genomic data. Note that at the time of this study the number of OpenSNP users were around 500. Today, this number is 2297, which also shows the rapid increase in the number of users who are susceptible for such attacks. For each of the 6 OpenSNP users sharing their SNPs on OpenSNP, we could retrieve several of their relatives publicly exposed on their OSN profiles. Out of these 6 families, we could identify in total 29 relatives whose genomic privacy was indirectly threatened by the OpenSNP users sharing their family ties (with real identities) on online social networks.

We focus on 2 individuals I_1 and I_2 out of these 6 identified OpenSNP individuals and evaluate their impact on the genomic and health privacy of their family members. We observed that both I_1 and I_2 publicly disclosed around 1 million of their SNPs. Furthermore, we identified the names of (i) 1 mother, 2 sons, 2 daughters, 1 grandchild, 1 aunt, 2 nieces, and 1 nephew of I_1, and (ii) 1 sibling, 1 aunt, 1 uncle, and 6 cousins of I_2 on Facebook. We compute the genomic and health privacy of these target individuals using the (normalized) entropy in (15) as the base metric, and average over all targeted SNPs for each individual. We cannot use the expected estimation error in (14) here as we do not have the ground truth for the genomes of the target individuals. Thus, privacy is quantified as the uncertainty of the adversary in this section.

To quantify the genomic privacy of the target individuals (i.e., family members of I_1 and I_2), we first construct $\mathcal{G}$ from all SNPs on chromosome 1 (from the observed genomes of I_1 and I_2). The set of observed SNPs includes the observed SNPs of I_1 (respectively I_2) for the inference of family members of I_1 (respectively I_2). The set of targeted SNPs includes 77k SNPs for I_1's family and 79k for I_2's family (from $\mathcal{G}$) for each evaluation. In Fig. 10, we show the decrease in the genomic privacy for different family members of I_1 (aunt, niece/nephew, grandchild, mother, child) and I_2 (cousin, aunt/uncle, sibling) as a result of our proposed inference attack, first without considering the LD dependencies (similarly to previous section). We observe that as expected, the decrease in the genomic privacy of close family members is significantly higher than that of more distant family members.

[18] According to [Gundecha et al. 2011], around 12% of Facebook users publicly share at least one family member on their profiles.

[19] http://www.patientslikeme.com/

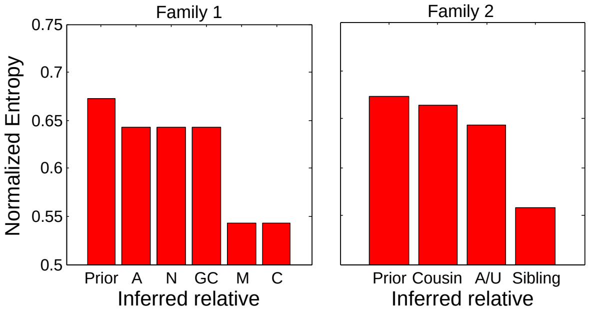


Fig. 10. Attacker's uncertainty about all SNP values on chromosome 1 for two different families, without using LD. A stands for aunt, N for niece/nephew, GC for grandchild, M for mother, C for child, U for uncle. Same notations are used in Fig. 11 and 12.

.

However, as we have seen in Section 4, the observation of one (or more) additional family member(s) has often much more impact on the target's privacy than the observation of only one relative.

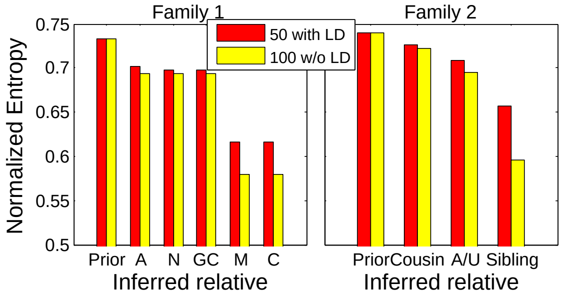


Fig. 11. Attacker's uncertainty about values of 100 SNPs on chromosome 1 for two families, by observing (i) all 100 SNPs of the relative that reveals his/her genome, and (ii) only 50 SNPs but using LD.

In Fig. 11, we display the decrease of genomic privacy with respect to 100 SNPs of chromosome 1.[20] We first show the different privacy levels by using all 100 SNPs of the observed relative (i.e., I_1 or I_2), and then show the same by using only 50 SNPs of the observed relative and LD values. We note that the use of LD decreases privacy slightly more for the first family than for the second family. This is because we randomly picked 50 different SNPs for both families, and those picked in the second family had weaker LD relationships with other SNPs. We finally observe that the difference between the two observation cases (50 SNPs with LD and 100 SNPs without LD) is higher for close relatives (mother, child, or sibling) than for others.

We also evaluate the health privacy of the family members of I_1 and I_2 considering their predispositions to various diseases. We first noticed that almost all important SNPs for privacy-sensitive diseases affected by genomic factors, like Alzheimer's, ischemic heart disease, or macular degeneration, were revealed by I_1 and I_2. Due to lack of space, we focus on Alzheimer's as it is one of the most important diseases that are mainly attributable to genetic factors. Having two ApoE4 alleles (SNP rs7412 being equal to CC and rs429358 equal to CC too) dramatically increases an individual's probability of having Alzheimer's by the age of 80. Thus, the contents of these two SNPs carry privacy-sensitive information for individuals. We use the metrics in (19) to quantify the health privacy of family

[20] We consider only 100 SNPs here for the same reason as in Section 4.

members for Alzheimer's disease. We assign equal weights to both associated SNPs (as their combination determines the predisposition to Alzheimer's disease). In Fig. 12, we show the attacker's uncertainty about the predisposition to Alzheimer's disease for the family members of I_1 and I_2. We notice a decrease of around 0.2 (from 0.5 to 0.3) in uncertainty between close relatives. Clearly, the knowledge of the SNPs of more relatives would further worsen the situation.

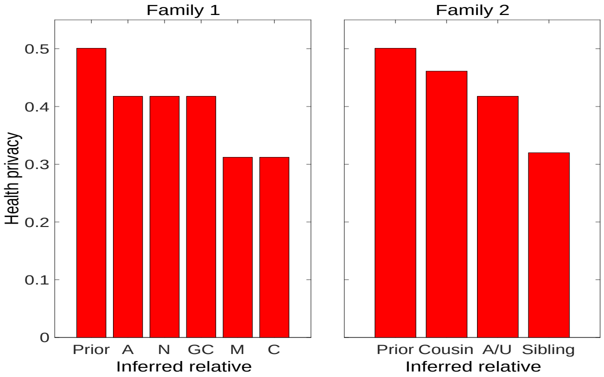


Fig. 12. Health privacy regarding Alzheimer's disease for 2 families, quantified using D_i^d as defined in 19, with normalized entropy as base metric, i.e. $G_i^k = H_i^k$.

5.2. Inference Attack with Phenotypic Information

We also rely on publicly available data shared by OpenSNP users to evaluate the impact of having additional phenotypic information on genomic privacy. In particular, we noticed that tens of OpenSNP users share both their SNPs and a specific phenotype: "Do you have a parent who was diagnosed with Alzheimer's disease?". Among those, 11 users answered that either their mother or father was indeed diagnosed with this disease. Hence, we build a Bayesian network of a trio (child and two parents), two SNPs $X_\star^1$ and $X_\star^2$ per person representing the APOE gene (rs7412 and rs429358), and a phenotypic node Y_i^{AD} of one of the parents r_i representing his/her Alzheimer's disease status (set as an evidence, as his child – OpenSNP user – reported his/her status), connected to both APOE SNPs $X_\star^1$ and $X_\star^2$. We derive the conditional probability table $P(Y_i^{\text{AD}}|X_i^1, X_i^2)$ from the risks presented in the 23andMe technical report on the APOE variants.[21]

Now we evaluate how this evidence changes the inference of the APOE SNPs of the child (i.e., of the OpenSNP user). In this case, we can rely on the expected estimation error, for 7 of the 11 OpenSNP users who also publicly disclose both their APOE SNPs. Note that among those 7 individuals, all have their rs7412 SNP equal to CC, and 6 have their rs429358 SNP equal to TT. These values are the most common variants, leading to normal risk for Alzheimer's disease. However, having one or two C at rs429358 in combination with a C at rs7412 substantially increases the risk of getting Alzheimer's by 85. Only one out of the 7 OpenSNP users takes CT at rs429358, leading to an increased risk. As Alzheimer's disease is linked to the C allele at both SNPs, knowing the Azheimer's status of the users' parents increases the posterior probabilities of these users carrying the C allele. For rs7412, knowing the phenotype leads to a decrease of privacy (estimation error) from 0.15 to 0.13 for all 7 users (sharing all the CC value at this SNP). However, for rs429358, observing the phenotype increases the genomic privacy from 0.3 to 0.47 for the 6 users who have non-risky SNP values (TT). This is due to the fact that observing

[21] It can be found here: https://www.23andme.com/en-ca/health/i_alzheimers/techreport/

that parents have been diagnosed Alzheimer's disease misleads the adversary who believes that it is more likely a posteriori that the OpenSNP user carries at least one risky allele (i.e., SNP being either CT or CC). On the contrary, for the single OpenSNP user taking SNP value CT at rs429358, the genomic privacy of this SNP after observing the phenotype value of the parent decreases from 0.75 to 0.63.

Note that the relationship between the APOE SNPs and Alzheimer's disease is highly probabilistic and it could also well be that the parent who was diagnosed had normal alleles at these SNPs. If the observed phenotype is more deterministically linked to the genotype, such as blood type, the observation of such phenotype will surely help improve the inference on the genotype. We take SNP rs7853989 as an example. If there is at least one minor allele C at this SNP, the blood type of his owner contains most likely a B (thus is either B or AB). By collecting data of OpenSNP users publicly sharing this SNP and their blood types, we could compute the expected estimation error prior and posterior to the observation of their blood types. The prior error was equal to 0.76 for all of those having a B in their blood type, and the posterior error (i.e., genomic privacy) was equal to 0.1. For those not having a B in their blood type, the prior error was equal to 0.28 (because the SNP then takes the two major alleles GG, thus is easier to infer only with the allele frequencies), and the posterior error became 0 (because oberving the phenotype tells us that it is impossible that the user carries the C allele).

6. DISCUSSION

In this section, we study the performance of the proposed attack, and discuss potential improvements of the investigated attack.

6.1. Performance

We implemented the proposed attack and evaluated its real-time computational performance for both the inference without and with LD correlations. All experiments were carried out on machines with Intel Xeon processors E3-1270 v3 of 3.5 Ghz and 32GB of RAM. For case without LD, the average time to run the junction algorithm is 2023 seconds ≈ 34 minutes for one observation scenario and the inference of all family members' targeted SNPs. The average time is computed over all scenarios plotted in Fig. 7. The standard deviation is equal to 117 seconds. As we were dealing with around 82,000 SNPs, we can derive that the inference time for one SNP is equal to around 0.025 second. Note that what takes most of the computational times here is the belief propagation step and not the construction of the junction tree that is quite straightforward with a family tree. In this scenario, we can easily parallelize the inference algorithm as SNPs are considered to be independent.

The inference with LD correlations is more computationally expensive: 3210 seconds ≈ 53 minutes on average (with a standard deviation equal to 144 seconds) for one observation scenario and the inference of all family members' targeted SNPs. The average time is computed over all scenarios plotted in Fig. 9(a). As we are in this case inferring 1,000 SNP/family member, the inference time per SNP is equal to around 3.2 seconds. This is approximately two orders of magnitude more computationally expensive than the scenario without LD correlations. This overhead can be explained by two factors: the number of iterations, and the number of LD factor nodes. First, as mentioned in 3.4, we have to run 7 to 15 iterations before reaching a stable state of posterior distributions. Second, we derive that the asymptotic complexity is equal to $\mathrm{O}(nm)$, but the constant number of factor nodes per SNP is equal to 13 in our practical case, which explains the second order of magnitude. Note also that, as we already mentioned, the junction tree algorithm was implemented by using the Bayes Net Toolbox [Murphy et al. 2001], which is certainly more optimized than the algorithm we implemented for the case with LD correlations.

6.2. Potential Improvements

One thing that we do not consider in the proposed inference attack is genetic imputation via identity by descent (IBD) [Burdick et al. 2006; Li et al. 2009], which can make the inference more powerful. IBD is a case in which a DNA segment (of around hundreds of thousands base pairs) is directly transferred from the ancestors to the descendants (e.g., from the grandfather to the father, and then from the father to the son). For instance, consider two relatives grandparent (GP) and child (C), both of whom share some of their SNPs in public platforms (e.g., OpenSNP). Assume GP and C both release all SNPs in a 1Mb (megabase) region X. Additionally GP releases a SNP at locus L which is about 100kb (kilobase) away from X, but C does not. Using the proposed algorithm (in Section 3), knowledge of SNPs in region X reveals nothing about L since linkage disequilibrium is typically not observed at distances of 100Kb (which is roughly 30 SNPs away from region X). However, suppose GP and C have an IBD relationship in the region X. Then, with probability close to 1, this shared segment extends to region L as well (IBD segments are typically tens of megabases). This means the adversary can impute one of C's alleles at L with near-certainty. Note that IBD could be integrated to the proposed algorithm in an ad-hoc manner. That is, the IBD occurrence in the observed genomes (in $\mathbf{X_O}$) can be first determined and an *initial inference* can be made only based on the IBD. Then, the inference method discussed above can be applied on top of this initial inference. We note that in this work, we did not observe any occurrence of IBD in the dataset we used to evaluate the proposed method (in Section 4).

Furthermore, in the proposed framework, we considered pairwise correlations (LD) between the SNPs, because, to the best of our knowledge, public LD data is always provided pairwise. However, higher-order correlations between the SNPs can make the inference more powerful. Such higher-order correlations can be obtained by analyzing a large genome dataset (of a particular population) and used in the proposed inference attack. Note that when such higher-order correlations are considered, the degree of each LD factor node in the proposed framework will also increase and the messages from LD factor nodes will be modified accordingly.

7. RELATED WORK

Stajano *et al.* [Stajano et al. 2008] were among the first to raise the issue of kin privacy in genomics. Cassa *et al.* [Cassa et al. 2008] provide a framework for measuring the risks to siblings of someone who reveals his SNPs. 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. We push this work further, by considering any kind of family members, and LD relationship between SNPs, by proposing and evaluating different privacy metrics, and by presenting a real attack scenario using publicly available data. Our generic framework considers any observation of a family's genomic data, and the adversary's background knowledge.

Several algorithms for inference on graphical models have been proposed in the context of pedigree analysis. Exact inference techniques on Bayesian networks are used in order to map disease genes and construct genetic maps [Fishelson and Geiger 2002; Lauritzen and Sheehan 2003; Jordan 2004]. Monte Carlo methods (Gibbs sampling) were also proved to be efficient for genetic analyses in the case of complex pedigrees [Jensen et al. 1995; Thomas et al. 2000; Sheehan 2000]. All these methods aim to infer specific genotypes given phenotypes (like diseases). Another paper relies on Gibbs sampling in order to infer haplotypes (used in association studies) from genotype data [Kirkpatrick et al. 2010]. Genotype imputation [Li et al. 2009] is another technique used by geneticists to complete missing SNPs based upon given genotyped data. A similar approach has recently been used to infer high-density genotypes in pedigrees, by relying notably on low-resolution genotypes and identity-by-descent regions of the genome [Burdick et al. 2006]. Neither these contributions address privacy, nor have they been applied to large pedigrees (such as our Utah family).

We also briefly summarize the research on the privacy of genomic data in the following. Homer *et al.* [Homer et al. 2008] prove that de-identification is an ineffective way to protect the privacy of genomic data, which is also supported by other works [Wang et al. 2009; Gitschier 2009; Zhou et al. 2011]. Most recently, Gymrek *et al.* [Gymrek et al. 2013] show how they identified DNAs of several individuals who participated in scientific studies. Fienberg *et al.* [Fienberg et al. 2011] propose using differential privacy to protect the identities of scientific study participants releasing statistics such as minor allele frequencies, p-values, and the top-k most relevant SNPs for a particular phonotype. Yu *et al.* [Yu et al. 2014] extended this work to compute differentially private statistics for arbitrary number of cases and controls. Johnson and Shmatikov propose an exponential mechanism called a distance-score mechanism to add noise to the output [Johnson and Shmatikov 2013]. Three papers related to differential privacy have been published in the framework of the iDASH genomic privacy workshop 2014 [Jiang et al. 2014]. In order to retain data utility, Wang *et al.* propose an algorithm that splits raw genome sequences into blocks before adding Laplace noise to them [Wang et al. 2014]. Yu and Ji adapt the methods of [Yu et al. 2014] and show new results about the Hamming distance score, notably its sensitivity [Yu and Ji 2014]. However, a major drawback of these approaches is that they reduce the accuracy of the research results. Fredrikson *et al.* have recently confirmed this finding in their study of privacy in pharmacogenetics [Fredrikson et al. 2013]. They show that given the model and some demographic information and drug dosage about a patient, an attacker can predict the patient's genetic markers. They also show that differentially private mechanisms can only improve genomic privacy at the cost of increased risk of stroke, bleeding events, and mortality.

Some pieces of work also focus on protecting the privacy of genomic data and on preserving utility in medical tests such as (i) search of a particular pattern in the DNA sequence [Troncoso-Pastoriza et al. 2007; Blanton and Aliasgari 2010], (ii) comparing the similarity of DNA sequences [Jha et al. 2008; Bruekers et al. 2008; Baldi et al. 2011], (iii) performing statistical analysis on several DNA sequences [Kantarcioglu et al. 2008; Xie et al. 2014], and (iv) using genomic data in clinical settings for healthcare [Ayday et al. 2013b; Danezis and De Cristofaro 2014; Djatmiko et al. 2014]. Furthermore, Ayday *et al.* propose privacy-preserving schemes for medical tests and personalized medicine methods that use patients' genomic data [Ayday et al. 2013c]. For privacy-preserving clinical genomics, a group of researchers proposes to outsource some costly computations to a public cloud or semi-trusted service provider [Wang et al. 2009; Chen et al. 2012]. Ayday *et al.* propose techniques for privacy-preserving management of raw genomes [Ayday et al. 2013a]. Karvelas *et al.* present a flexible framework based on oblivious RAM that allows for the private processing of whole-genome sequences, that supports any query and that also hides the access patterns [Karvelas et al. 2014]. Other similar privacy-preserving mechanisms for GWAS based on homomorphic encryption [Lu et al. 2015; Kim and Lauter 2015; Zhang et al. 2015b] or secure multi-party computation [Constable et al. 2015; Zhang et al. 2015a] have recently been proposed in the context of the iDASH challenge 2015.

In contrast with these contributions, in this paper, we propose an original and efficient inference attack in order to reconstruct genomic data of individuals given observed genomic and phenotypic data of their family members and special characteristics of genomic data. Furthermore, we quantify the genomic and health privacy of individuals as a result of this attack using different metrics, and show the real threat by using the data collected from genome-sharing website and OSNs.

8. CONCLUSION

In this article, we have proposed and studied a novel reconstruction attack for inferring the genomic data of individuals from the observed genomes and phenotypes of their relatives. We have studied its computational complexity both theoretically and practically, have compared several metrics to quantify genomic and health privacy, and have carried out a real-world cross-website attack by notably making use of a popular online social network. From our performance evaluation, we notice a trade-off between time efficiency and inference power. If the attacker is interested only in a subset of targeted SNPs or if he cannot observe the full set of SNPs of the target's relatives, he could use the inference method that includes LD correlations without having to incur too much computational cost. From a policy maker's viewpoint, the inference method without LD correlations gives essentially an upperbound on the actual level of genomic privacy of the family members.

REFERENCES

Dakshi Agrawal and Charu C Aggarwal. 2001. On the design and quantification of privacy preserving data mining algorithms. In *Proceedings of the twentieth ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems*. ACM, 247–255.

Erman Ayday, Emiliano De Cristofaro, Jean-Pierre Hubaux, and Gene Tsudik. 2015. Whole Genome Sequencing: Revolutionary Medicine or Privacy Nightmare? *Computer* 2 (2015), 58–66.

Erman Ayday, A. Einolghozati, and Faramarz Fekri. 2012. BPRS: Belief Propagation Based Iterative Recommender System. *IEEE ISIT* (2012).

Erman Ayday and Faramarz Fekri. 2012a. Belief Propagation Based Iterative Trust and Reputation Management. *IEEE Transactions on Dependable and Secure Computing* 9, 3 (2012).

Erman Ayday and Faramarz Fekri. 2012b. BP-P2P: A Belief Propagation-Based Trust and Reputation Management for P2P Networks. *SECON* (2012).

Erman Ayday, Jean Louis Raisaro, Urs Hengartner, Adam Molyneaux, and Jean-Pierre Hubaux. 2013a. Privacy-Preserving Processing of Raw Genomic Data. *DPM 2013* (2013).

Erman Ayday, Jean Louis Raisaro, Jean-Pierre Hubaux, and Jacques Rougemont. 2013b. Protecting and Evaluating Genomic Privacy in Medical Tests and Personalized Medicine. *Proceedings of the ACM workshop on Privacy in the electronic society - WPES* (2013).

Erman Ayday, Jean Louis Raisaro, Paul J. McLaren, Jacques Fellay, and Jean-Pierre Hubaux. 2013c. Privacy-Preserving Computation of Disease Risk by Using Genomic, Clinical, and Environmental Data. *HealthTech* (2013).

Pierre Baldi, Roberta Baronio, Emiliano De Cristofaro, Paolo Gasti, and Gene Tsudik. 2011. Countering GATTACA: Efficient and secure testing of fully-sequenced human genomes. *CCS* (2011).

Marina Blanton and Mehrdad Aliasgari. 2010. Secure outsourcing of DNA searching via finite automata. *DBSec* (2010).

Fons Bruekers, Stefan Katzenbeisser, Klaus Kursawe, and Pim Tuyls. 2008. *Privacy-preserving matching of DNA profiles*. Technical Report.

Joshua T Burdick, Wei-Min Chen, Gonçalo R Abecasis, and Vivian G Cheung. 2006. In silico method for inferring genotypes in pedigrees. *Nature genetics* 38, 9 (2006), 1002–1004.

Christopher A Cassa, Brian Schmidt, Isaac S Kohane, and Kenneth D Mandl. 2008. My sister's keeper?: genomic research and the identifiability of siblings. *BMC Medical Genomics* 1, 1 (2008), 32.

Jinghu Chen, Ajay Dholakia, Evangelos Eleflhetiou, Mac P. C. Fossotier, and Xiao-Yu Hu. 2002. Near optimum reduced-complexity decoding algorithm for LDPC codes. *IEEE ISIT* (2002).

Yangyi Chen, Bo Peng, XiaoFeng Wang, and Haixu Tang. 2012. Large-Scale Privacy-Preserving Mapping of Human Genomic Sequences on Hybrid Clouds. *NDSS* (2012).

Scott D Constable, Yuzhe Tang, Shuang Wang, Xiaoqian Jiang, and Steve Chapin. 2015. Privacy-preserving GWAS analysis on federated genomic datasets. *BMC medical informatics and decision making* 15, Suppl 5 (2015), S2.

George Danezis and Emiliano De Cristofaro. 2014. Fast and Private Genomic Testing for Disease Susceptibility. *Proceedings of the ACM workshop on Privacy in the electronic society - WPES* (2014).

Claudia Diaz, Stefaan Seys, Joris Claessens, and Bart Preneel. 2003. Towards measuring anonymity. In *Privacy Enhancing Technologies*. Springer, 54–68.

Mentari Djatmiko, Arik Friedman, Roksana Boreli, Felix Lawrence, Brian Thorne, and Stephen Hardy. 2014. Secure Evaluation Protocol for Personalized Medicine. *Proceedings of the ACM workshop on Privacy in the electronic society - WPES* (2014).

Radoje Drmanac, Andrew B Sparks, Matthew J Callow, Aaron L Halpern, Norman L Burns, Bahram G Kermani, Paolo Carnevali, Igor Nazarenko, Geoffrey B Nilsen, George Yeung, and others. 2010. Human genome sequencing using unchained base reads on self-assembling DNA nanoarrays. *Science* 327, 5961 (2010), 78–81.

Douglas S. Falconer and Trudy F.C. Mackay. 1996. *Introduction to Quantitative Genetics (4th Edition)*. Addison Wesley Longman, Harlow, Essex, UK.

Stephen E Fienberg, Aleksandra Slavkovic, and Caroline Uhler. 2011. Privacy Preserving GWAS Data Sharing. *Proceedings of the IEEE 11th International Conference on Data Mining Workshops (ICDMW)* (Dec. 2011).

Maayan Fishelson and Dan Geiger. 2002. Exact genetic linkage computations for general pedigrees. *Bioinformatics* 18, suppl 1 (2002), S189–S198.

Matthew Fredrikson, Eric Lantz, Somesh Jha, Simon Lin, David Page, and Thomas Ristenpart. 2013. Privacy in Pharmacogenetics: An End-to-End Case Study of PersonalizedWarfarin Dosing. *Proceedings of the 23rd USENIX Security Symposium (USENIX Security 14)* (2013).

Jane Gitschier. 2009. Inferential genotyping of Y chromosomes in Latter-Day Saints founders and comparison to Utah samples in the HapMap project. *The American Journal of Human Genetics* 84, 2 (2009), 251–258.

L. A. Goodman. 1961. Snowball sampling. *The Annals of Mathematical Statistics* 32, 1 (1961).

Pritam Gundecha, Geoffrey Barbier, and Huan Liu. 2011. Exploiting vulnerability to secure user privacy on a social networking site. In *KDD*. ACM.

Melissa Gymrek, Amy L. McGuire, David Golan, Eran Halperin, , and Yaniv Erlich. 2013. Identifying Personal Genomes by Surname Inference. *Science: 339 (6117)* (Jan. 2013).

Nils Homer, Szabolcs Szelinger, Margot Redman, David Duggan, Waibhav Tembe, Jill Muehling, John V Pearson, Dietrich A Stephan, Stanley F Nelson, and David W Craig. 2008. Resolving individuals contributing trace amounts of DNA to highly complex mixtures using high-density SNP genotyping microarrays. *PLoS Genetics* 4 (Aug. 2008).

Mathias Humbert, Erman Ayday, Jean-Pierre Hubaux, and Amalio Telenti. 2013. Addressing the Concerns of the Lacks Family: Quantification of Kin Genomic Privacy. *Proceedings of the 20th ACM Conference on Computer and Communications Security - CCS* (2013).

Mathias Humbert, Erman Ayday, Jean-Pierre Hubaux, and Amalio Telenti. 2014. Reconciling Utility with Privacy in Genomics. *Proceedings of the ACM workshop on Privacy in the electronic society - WPES* (2014).

Claus Skaanning Jensen, Augustine Kong, and Uffe Kjærulff. 1995. Blocking Gibbs sampling in very large probabilistic expert systems. *International Journal of Human Computer Studies* 42, 6 (1995), 647–666.

Finn V Jensen and Frank Jensen. 1994. Optimal junction trees. In *Proceedings of the Tenth international conference on Uncertainty in artificial intelligence*. Morgan Kaufmann Publishers Inc., 360–366.

Somesh Jha, Louis Kruger, and Vitaly Shmatikov. 2008. Towards Practical Privacy for Genomic Computation. *Proceedings of the 2008 IEEE Symposium on Security and Privacy* (2008), 216–230.

Xiaoqian Jiang, Yongan Zhao, Xiaofeng Wang, Bradley Malin, Shuang Wang, Lucila Ohno-Machado, and Haixu Tang. 2014. A community assessment of privacy preserving techniques for human genomes. *BMC medical informatics and decision making* 14, Suppl 1 (2014), S1.

Aaron Johnson and Vitaly Shmatikov. 2013. Privacy-preserving data exploration in genome-wide association studies. *Proceedings of ACM international conference on Knowledge discovery and data mining* (2013).

Andrew D Johnson and Christopher J O'Donnell. 2009. An Open Access Database of Genome-wide Association Results. *BMC Medical Genetics 10:6* (2009).

Michael I Jordan. 2004. Graphical models. *Statist. Sci.* (2004), 140–155.

M. Kantarcioglu, Wei Jiang, Ying Liu, and B. Malin. 2008. A Cryptographic Approach to Securely Share and Query Genomic Sequences. *IEEE Transactions on Information Technology in Biomedicine* 12, 5 (2008), 606–617.

Nikolaos Karvelas, Andreas Peter, Stefan Katzenbeisser, Erik Tews, and Kay Hamacher. 2014. Privacy-preserving whole genome sequence processing through proxy-aided oram. In *Proceedings of the 13th Workshop on Privacy in the Electronic Society*. ACM, 1–10.

Miran Kim and Kristin Lauter. 2015. Private genome analysis through homomorphic encryption. *BMC medical informatics and decision making* 15, Suppl 5 (2015), S3.

Bonnie Kirkpatrick, Eran Halperin, and Richard M Karp. 2010. Haplotype inference in complex pedigrees. *Journal of Computational Biology* 17, 3 (2010), 269–280.

Daphne Koller and Nir Friedman. 2009. *Probabilistic graphical models: principles and techniques*. MIT press.

F. Kschischang, B. Frey, and H. A. Loeliger. 2001. Factor Graphs and the sum-product algorithm. *IEEE Transactions on Information Theory* 47 (2001).

Steffen L Lauritzen and Nuala A Sheehan. 2003. Graphical models for genetic analyses. *Statist. Sci.* (2003), 489–514.

Yun Li, Cristen Willer, Serena Sanna, and Gonçalo Abecasis. 2009. Genotype imputation. *Annual review of genomics and human genetics* 10 (2009), 387.

Wen-Jie Lu, Yoshiji Yamada, and Jun Sakuma. 2015. Privacy-preserving genome-wide association studies on cloud environment using fully homomorphic encryption. *BMC medical informatics and decision making* 15, Suppl 5 (2015), S1.

Joris M Mooij and Hilbert J Kappen. 2007. Sufficient conditions for convergence of the sum–product algorithm. *Information Theory, IEEE Transactions on* 53, 12 (2007), 4422–4437.

Kevin Murphy and others. 2001. The bayes net toolbox for matlab. *Computing science and statistics* 33, 2 (2001), 1024–1034.

Kevin P Murphy, Yair Weiss, and Michael I Jordan. 1999. Loopy belief propagation for approximate inference: An empirical study. In *Proceedings of the Fifteenth conference on Uncertainty in artificial intelligence*. Morgan Kaufmann Publishers Inc., 467–475.

Muhammad Naveed, Erman Ayday, Ellen W Clayton, Jacques Fellay, Carl A Gunter, Jean-Pierre Hubaux, Bradley A Malin, and XiaoFeng Wang. 2015. Privacy in the Genomic Era. *ACM Computing Surveys (CSUR), 2015* (2015).

Dale R Nyholt, Chang-En Yu, and Peter M Visscher. 2009. On Jim Watson's APOE status: Genetic information is hard to hide. *European Journal of Human Genetics* 17 (2009), 147–149.

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

H. Pishro-Nik and F. Fekri. 2004. On Decoding of Low-Density Parity-Check Codes on the Binary Erasure Channel. *IEEE Transactions on Information Theory* 50 (March 2004), 439–454.

Andrei Serjantov and George Danezis. 2003. Towards an information theoretic metric for anonymity. In *Privacy Enhancing Technologies*. Springer, 41–53.

Nuala A Sheehan. 2000. On the application of Markov chain Monte Carlo methods to genetic analyses on complex pedigrees. *International Statistical Review* 68, 1 (2000), 83–110.

Reza Shokri, George Theodorakopoulos, J-Y Le Boudec, and J-P Hubaux. 2011. Quantifying location privacy. In *IEEE Symposium on Security and Privacy*.

Frank Stajano, Lucia Bianchi, Pietro Liò, and Douwe Korff. 2008. Forensic genomics: kin privacy, driftnets and other open questions. In *Proceedings of the 7th ACM workshop on Privacy in the electronic society*.

Latanya Sweeney, Akua Abu, and Julia Winn. 2013. Identifying Participants in the Personal Genome Project by Name. *Available at SSRN 2257732* (2013).

Alun Thomas, Alexander Gutin, Victor Abkevich, and Aruna Bansal. 2000. Multilocus linkage analysis by blocked Gibbs sampling. *Statistics and Computing* 10, 3 (2000), 259–269.

Juan Ramón Troncoso-Pastoriza, Stefan Katzenbeisser, and Mehmet Celik. 2007. Privacy preserving error resilient DNA searching through oblivious automata. *CCS '07: Proceedings of the 14th ACM Conference on Computer and Communications Security* (2007).

Isabel Wagner. 2015. Genomic Privacy Metrics: A Systematic Comparison. *International Workshop on Genome Privacy and Security (in conjunction with IEEE Symposium on Security and Privacy)* (2015).

Rui Wang, Yong Fuga Li, XiaoFeng Wang, Haixu Tang, and Xiaoyong Zhou. 2009. Learning your identity and disease from research papers: Information leaks in genome wide association study. *Proceedings of the 16th ACM CCS* (2009), 534–544.

Rui Wang, XiaoFeng Wang, Zhou Li, Haixu Tang, Michael K. Reiter, and Zheng Dong. 2009. Privacy-preserving genomic computation through program specialization. *Proceedings of the 16th ACM CCS* (2009), 338–347.

Shuang Wang, Noman Mohammed, and Rui Chen. 2014. Differentially private genome data dissemination through top-down specialization. *BMC medical informatics and decision making* 14, Suppl 1 (2014), S2.

Wei Xie, Murat Kantarcioglu, William S Bush, Dana Crawford, Joshua C Denny, Raymond Heatherly, and Bradley A Malin. 2014. SecureMA: protecting participant privacy in genetic association meta-analysis. *Bioinformatics* 30, 23 (2014), 133–141.

Fei Yu, Stephen E. Fienberg, Aleksandra B. Slavkovic, and Caroline Uhler. 2014. Scalable privacy-preserving data sharing methodology for genome-wide association studies. *Journal of Biomedical Informatics* 50 (2014), 133–141.

Fei Yu and Zhanglong Ji. 2014. Scalable privacy-preserving data sharing methodology for genome-wide association studies: an application to iDASH healthcare privacy protection challenge. *BMC medical informatics and decision making* 14, Suppl 1 (2014), S3.

Yihua Zhang, Marina Blanton, and Ghada Almashaqbeh. 2015a. Secure distributed genome analysis for GWAS and sequence comparison computation. *BMC medical informatics and decision making* 15, Suppl 5 (2015), S4.

Yuchen Zhang, Wenrui Dai, Xiaoqian Jiang, Hongkai Xiong, and Shuang Wang. 2015b. FORESEE: Fully Outsourced secuRe gEnome Study basEd on homomorphic Encryption. *BMC medical informatics and decision making* 15, Suppl 5 (2015), S5.

Xiaoyong Zhou, Bo Peng, Yong Fuga Li, Yangyi Chen, Haixu Tang, and XiaoFeng Wang. 2011. To release or not to release: Evaluating information leaks in aggregate human-genome data. *ESORICS* (2011).

Received February 2007; revised March 2009; accepted June 2009