Published in final edited form as:

Proc Asia Pac Bioinform Conf. 2023 April ; 2023: .

Tuning Privacy-Utility Tradeoff in Genomic Studies Using Selective SNP Hiding

Nour Almadhoun Alserr[1], **Gulce Kale**[3], **Onur Mutlu**[1,3], **Oznur Tastan**[4], **Erman Ayday**[2,3]

[1]Department of Information Technology and Electrical Engineering, ETH Zurich, Zurich 8006, Switzerland

[2]Department of Electrical Engineering and Computer Science, Case Western Reserve University, Cleveland, OH 44106, USA

[3]Computer Engineering Department, Bilkent University, Ankara 06800, Turkey

[4]Computer Science and Engineering, Sabanci University, Istanbul 34956, Turkey

Abstract

Researchers need a rich trove of genomic datasets that they can leverage to gain a better understanding of the genetic basis of the human genome and identify associations between phenol-types and specific parts of DNA. However, sharing genomic datasets that include sensitive genetic or medical information of individuals can lead to serious privacy-related consequences if data lands in the wrong hands. Restricting access to genomic datasets is one solution, but this greatly reduces their usefulness for research purposes. To allow sharing of genomic datasets while addressing these privacy concerns, several studies propose privacy-preserving mechanisms for data sharing. Differential privacy is one of such mechanisms that formalize rigorous mathematical foundations to provide privacy guarantees while sharing aggregated statistical information about a dataset. Nevertheless, it has been shown that the original privacy guarantees of DP-based solutions degrade when there are dependent tuples in the dataset, which is a common scenario for genomic datasets (due to the existence of family members).

In this work, we introduce a new mechanism to mitigate the vulnerabilities of the inference attacks on differentially private query results from genomic datasets including dependent tuples. We propose a utility-maximizing and privacy-preserving approach for sharing statistics by hiding selective SNPs of the family members as they participate in a genomic dataset. By evaluating our mechanism on a real-world genomic dataset, we empirically demonstrate that our proposed mechanism can achieve up to 40% better privacy than state-of-the-art DP-based solutions, while near-optimally minimizing utility loss.

Keywords

Genomic dataset; Differential privacy; Dependent tuples

nalserr@ethz.ch, or nour.madhoun@gmail.com.

1. Introduction

As technologies improve the cost and scale of sequencing, it has become possible to sequence genomes from large cohorts of patients. Today, researchers have access to large genomic datasets, whereby they can study associations between variants and complex traits. However, as shown by earlier studies, the public availability of genomic data - even in anonymized form - raises serious privacy concerns [1]. Hence, many institutions (i.e., data owners who collect genomic data), rather than publicly releasing their genomic datasets, provide limited access to these datasets through queries. Such queries typically seek to extract statistical information about the dataset (referred to as a "*statistical dataset*"). They are formed and submitted by the researchers, computed at the data owner institution, and only the final results are shared with the querying researchers. One prominent example of such an approach is the access to the results of *genome-wide association studies* (GWAS) [2].

Although this approach provides stronger privacy protection for the dataset participants, previous work has shown that such statistical genomic datasets are prone to *membership* and *attribute inference attacks* [3]. An adversary, using the results of the queries, the genotype of a target, and the publicly available *minor allele frequencies* (MAFs) of the *single nucleotide polymorphisms* (SNPs) used in the study, can infer the membership of the target to the corresponding dataset (or to the case group of the corresponding GWAS) [4]. This attack is considered serious because in most cases, dataset participants are associated with known sensitive information (e.g., cancer predisposition).

Differential privacy (DP) [5] is one of the privacy protection concepts that has received widespread popularity for sharing aggregate statistics from human genomic datasets due to its theoretical guarantees [2]. Such that, even if there is *only* one different tuple in two datasets (called *neighboring datasets*), it is hard to differentiate between the query results of these two datasets. The probability of distinguishing the results of the neighboring datasets is controlled by a parameter called *privacy budget* ϵ. However, DP has a known drawback as it makes no assumption about the *correlation* between dataset tuples. This may degrade the privacy guarantees of DP and give the adversary a stronger ability to extract more sensitive information if the dataset includes dependent tuples, which is a common situation for genomic datasets as genomes of family members are correlated. Previous work show how dependency between dataset tuples may reduce the privacy guarantees of DP [6–8] and propose general mechanisms to tackle this problem. Recently, [9, 10] analyze and show the privacy risk due to the inference attacks on differentially-private query results by exploiting the dependency between tuples in a genomic dataset. To mitigate this privacy risk, [9] formalize the notion of ϵ-DP for genomic datasets with dependent tuples to avoid the inference of sensitive information by any adversary with prior knowledge about the tuples correlation.

However, to provide privacy guarantees for the dependent tuples in genomic datasets, existing DP-based solutions suggest changing the value of the privacy parameter ϵ (i.e., adding more noise to the released statistics based on the number of dependent tuples and the strength of relationship between them). Such higher noise amounts significantly degrade the utility of the shared GWAS statistics, especially when the query results also include data from independent tuples in the dataset. On the other hand, medical research necessitates highly accurate information for high-quality and effective research outcomes. Therefore, it is also crucial to develop utility-preserving countermeasures for this privacy risk.

In this work, we propose a novel privacy-preserving and utility-preserving mechanism for sharing statistics from genomic datasets to attain privacy guarantees while taking into consideration the dependency between tuples. As discussed, the main reason for the aforementioned privacy risk is the existence of dependent tuples in the genomic datasets due to familial relationships. Therefore, our **goal** is to reduce the level of such dependency without significantly weakening the utility by adding more noise to the released statistics as proposed in the state-of-the-art mechanisms [9, 10, 6–8]. To achieve this, inspired by our previous work [11], we propose an optimization-based countermeasure to *selectively* hide genomic data of dataset participants to distort the dependencies (familial relations) among them without significantly degrading dataset query responses, thus, the utility.

The **key idea** of our proposed "*selective hiding*" mechanism is to hide some selected SNPs of family members (as they join the genomic dataset during the dataset collecting step) to 1) reduce the kinship relationship between them, and 2) keep the utility of the shared GWAS statistics high. We consider a *static* genomic dataset, which has no updates, and hence the released private statistics depend on the one-time updated GWAS dataset. By doing so, the constructed GWAS dataset includes only the obfuscated genomes of the dependent tuples. Thus, in case of a data breach, familial relationships between the GWAS participants are also protected. Also, the proposed method selectively hides *only* the dependent tuples, keeping the genomes of independent tuples intact (which improves utility).

We assume that a potential adversary can use side-channel leaks as public records, social media sites, or own prior knowledge about the family members who participated in a genomic study for performing a targeted attack (i.e., aimed at a specific, small group of people). The adversary uses this information along with the published GWAS statistics in order to infer sensitive attributes about the dataset participants. Even if the adversary is not sure about the participation of the family members in the dataset or the dependencies between the dataset participants (i.e., kin relationships), the adversary can infer the kinship coefficient between dataset participants by issuing several well-designed queries to the dataset. We evaluate the proposed algorithm against such an adversary by using real-life genomic datasets. Our proposed DP-based mechanism can prevent the adversary from utilizing the dependencies among the dataset tuples to infer more sensitive attributes about dataset participants. In other words, we are aiming at achieving the privacy and utility guarantees of the standard DP assuming all the participants of the dataset are independent, without increasing the added noise to the query results. Hence, using our mechanism allows releasing any differentially private genomic query over GWAS. These queries include but are not limited to 1) *count* or *cohort discovery*: to query how many participants in the dataset satisfy given criteria, 2) χ^2 and p-values association tests: compute χ^2 and p-value statistics for an SNP, or 3) *minor allele frequency* (MAF): to compute the frequency of which the rare nucleotide occurs at a particular SNP. Considering our adversarial scenario (discussed in detail in Section 3), our results show that the proposed approach can near-achieve both the privacy and utility guarantees of standard DP (i.e., under independent tuples assumption) compared to existing work. As a result of our proposed countermeasure, dataset owners will share data realizing that the privacy of the dataset participants, including families, will be protected. Also, families will be more open to donating their data to medical datasets for research knowing their privacy is uncompromised. Finally, researchers will know that they receive high-utility information from medical datasets.

The rest of this paper is organized as follows. Section 2 presents related prior works on genomics privacy, DP mechanisms under dependent tuples, and our contributions. Section 3 explores our privacy threat model, followed by Section 4, which explains our approach. In Section 5 we evaluate our proposed strategy and compare it to the state-of-art mechanisms. Section 6 presents conclusions and highlights future research directions that are pointed by this paper.

2. Related Work

In this section, we summarize the state-of-the-art published studies on genomic privacy and differential privacy in particular.

2.1. Privacy of Genomic Data

In recent years, privacy-preserving publishing of genomic data has received much attention. One of the widely-used promising privacy-preserving solutions is the DP framework. DP provides rigorous mathematical mechanisms for limiting the information leakage through adding noise to the statistics results in GWAS [2, 12, 13]. We provide all the theoretical details about DP in Section 1.2 in the Supplementary Materials. Existing works basically utilize the privacy guarantee of DP as a protective measure against inference attack scenarios (e.g., membership attack discovered by [4]) even if the attacker has access to external auxiliary information. [12, 13, 2] proposed differentially-private algorithms to release the aggregate human genomic statistical results from genomic datasets as GWAS. Using a controlled amount of noise from *Laplace distribution* [14], helps enhance the privacy of all participants in a GWAS. In these algorithms, researchers submit genomic queries e.g., cell counts, MAF, p-value and χ^2 statistics, and receive the query results in a privacy-preserving manner through DP algorithms. However, these proposed DP mechanisms assume that all the dataset tuples are independent, which may degrade the privacy guarantees when such correlations exist between the tuples in the dataset.

2.2. Differential Privacy under Dependent Tuples

The adversary can exploit auxiliary channels to get information about the tuples correlation within the genomic dataset. [15] was the first to show this DP vulnerability. Therefore, they propose the Pufferfish framework [16] as a generalization of DP to handle this threat. Following the Pufferfish, several studies [17, 18] provide perturbation mechanisms to handle the correlation between tuples for various applications. Recently, [6] show that an adversary can utilize the pairwise dependencies within a location dataset to predict the participant's location from the differentially private query results [6]. To mitigate this privacy threat, [6] propose a *Laplace* mechanism defined as *dependent differential privacy* (DDP) to tack le the pairwise correlation between any two tuples in the dataset. To improve the privacy and utility guarantees of [6], [8] present a new definition of the DDP, which can handle numeric and non-numeric queries, to address any adversary with arbitrary correlation knowledge. Moreover, [9] discuss attribute and membership inference attacks against differential privacy mechanisms, when the datasets include dependent tuples. As a countermeasure for these attacks, [9] adjust the global sensitivity of the query before applying *Laplace perturbation mechanism* (LPM) to the query results.

2.3 Contribution of This Work

DP-based solutions that aim at addressing the privacy risks due to the existence of dependent tuples in statistical datasets (including GWAS), require the addition of high noise values to the results of statistics queries. Hence, it causes a significant loss in the utility of the query responses. In this work, we propose a different approach to address the same problem. Our proposed solutions rely on one-time selective masking of genomic *loci* in a GWAS dataset to 1) decrease the estimated kinship coefficients between relatives in the dataset, 2) provide privacy against an adversary that utilizes correlations in the published statistics, and 3) provide privacy for dataset participants (e.g., against kinship inference) in case the dataset is breached. Other recent studies have attempted to propose general mechanisms to tackle kinship privacy such as [19], which target interdependent privacy in their work. Here, we compare our model (in terms of privacy and utility) with the existing similar approaches (e.g., [9] under the same goal of sharing DP-based query results from genomic datasets with dependent tuples. Our results show that the proposed scheme provides both better privacy and higher utility than the existing solutions.

3 System and Threat Models

The dataset owner maintains a statistical dataset D and responds to users' statistical queries. To provide statistical information about the dataset in a privacy-preserved way, the dataset owner computes randomized query results $A(D)$ using LPM-based DP (as in Section 1.2 in the Supplementary Materials), and sends it back to the users. The adversary in our scenario can be one of the users. The adversary can send various statistical queries to the dataset. In recent work, we discuss the vulnerability of dependent tuples in a statistical dataset due to different statistical queries (e.g., count, MAF, χ^2 statistics) and [10, 9]. Here, for simplicity, we focus on a "count query", in which the adversary forms its query asking about the sum of values of a specific SNP j among the dataset participants sharing the same demographic data, such as location or age (we assume an SNP value of 0, 1, or 2, representing the number of its minor alleles). Limiting the scope of the query to a small number of dataset participants allows the adversary to have a higher inference power about the sensitive genomic information of a target, especially if the query result is computed over the target and target's family members. The kinship data is not available in the GWAS studies due to the sensitivity of family information. However, we consider a realistic *targeted* attack scenario in which the adversary may know that a family participated in a genomic study. The adversary can infer the participation of the family members using other side-channel leaks as public records, social media sites, or personal knowledge for performing a targeted attack (aimed at a specific, small group of people). Alternatively, the adversary can also estimate the kinship coefficients between the dataset participants using the responses to its well-designed queries. We show the results of kinship coefficients estimation in Section 5 in the Supplementary Materials. With the availability of such information, considering an attribute inference attack, in which 1) the adversary does not have any prior knowledge about genotypes of individuals in the dataset, and 2) the goal of the adversary is to infer genomic data of a target individual using the released query results, we have the following assumptions:

- The adversary knows the membership information of all individuals in the dataset. The membership of an individual in a dataset means that the corresponding individual is included in the dataset.
- The adversary knows the dependencies (e.g., kinship coefficient) between the individuals in the dataset. As discussed, the adversary can obtain this information from side-channel leaks or by estimating the kinship coefficients between the dataset participants using the responses to its queries.

4 Proposed Work

Let dataset D includes N individuals and m SNPs. We assume a statistical query to the dataset is computed over q dataset participants, including a target i and other p dataset participants $(q = 1 + p)$. D_i^j represents the value of SNP j for target individual i and D_p^j represents the sum of the SNP j values for other (p) participants that are involved in the query computation. We let (δ) be the added Laplace noise with scale $2/\epsilon$. Set $\mathbf{F}(|\mathbf{F}| = f)$ includes individuals from the same family (i.e., target i and his/her family members), and set $\mathbf{U}(|\mathbf{U}| = u)$ includes the other unrelated members (non-relatives) in the dataset. Note that there may be more than one family in the dataset and the privacy risk for each family can be shown similarly. Therefore, for the sake of simplicity, we assume the dataset includes *only* one family. We show the overview of the proposed algorithm in Figure 1. We provide all the notations we use in Table 1 in Section 2 in the Supplementary Materials.

Similarly to the previous work [11], we assume family members share their data in a sequential order during the data collection process. For each new incoming family member to the dataset, we hide some selected SNPs to decrease kinship coefficients among family members and preserve their familial privacy. The main differences between this work are:

- The original selective sharing scheme in [11] considers a publicly available dataset and it aims to reduce the kinship coefficients between the dataset participants to hide the familial relationships. Here, the statistical dataset is not public. Therefore, our aim is *not* to specifically hide the relation of participants. Instead, our *goal* is to reduce the kinship coefficients so that (1) privacy vulnerability (caused by the sharing of statistics computed over dependent tuples) is minimized, and (2) the utility of the shared statistics still remains high. As a result, we exclude the outlier constraints part (details provided in the Supplementary Materials, Section 1.1) in the optimization model. We focus on satisfying the kinship constraints only. For completeness, below we describe the part of the formulation and the approach that we propose in [11] and we also use here.

- We design the proposed method to hide *overlapping* regions among the family members first and solve the optimization later. The goal is to have better privacy and higher utility.

To reduce the kinship coefficient, we hide positions based on their SNP configurations. Hence, we use a notation to denote the positions with different SNP configurations for 1) an individual, and 2) a family. For an individual i, a particular genomic position can hold a SNP configuration s_i, where s_i takes values in $\{0, 1, 2\}$. We denote the total number of positions the individual owns with SNP configuration s_i as n_{s_i} (e.g., n_0 is the number of positions with SNPs' value of 0). n_{s_i} shows how many genomic locations are 1) recessive homozygous, 2) heterozygous, and 3) dominant homozygous. For the setting of representing more than one individual, we refer to all genomic positions (with their SNP configuration) for all individuals from the family members. For example, if we have a family of three members, n_{121} indicates the number of positions in which the first individual's SNP value is 1, the second's is 2, and the third's is 1. If an individual i can hold any of the SNP values (i.e., $s_i = 0$, 1, or 2), we denote s_i with $*$.

To calculate the kinship coefficient between two individuals i and k, we use the robust kinship estimator proposed by [20]:

$$\phi_{ik} = (2n_{11} - 4(n_{02} + n_{20}) - n_{*1} + n_{1*})/4n_{1*} \tag{1}$$

when n_{1*} ¡ n_{*1}, it means that k_{th} individual has more heterozygous positions than the i_{th} member. n_{11} presents the number of genomic positions where both individuals are heterozygous. n_{20} and n_{02} indicate the number of SNPs when the individuals i and k hold homozygous dominant SNPs (e.g., $s_i = 0$) or homozygous recessive SNPs (e.g., $s_i = 2$).

Our solutions find the appropriate positions to hide based on the SNP configuration. We define a variable, x_{s_i}, to denote the number of a particular SNP configuration we need to hide from the most recent entrants (i.e., last arrived family member). Using Equation 2, one can easily calculate x_{11}; the number of heterozygous genomic positions to be removed in order to decrease the kinship coefficient down to a preset ϕ' value between two individuals, as:

$$x_{11} = \frac{2n_{11} - 4(n_{02} + n_{20}) - n_{1*} + \left(1 - 4\phi_{ik}^{'}\right)n_{*1}}{2\left(1 - 2\phi_{ik}^{'}\right)} \tag{2}$$

To have a kinship coefficient lower than a preset Φ, Equation (2) can be cast as an integer programming problem as follows:

$$\begin{aligned} &\min \quad x_{11} \\ &\text{subject to} \\ &2n_{11} - 4(n_{02} + n_{20}) - n_{1*} + (1 - 4\Phi)n_{*1} \le (2 - 4\Phi)x_{11} \\ &x_{11} \le n_{11} \end{aligned} \tag{3}$$

$$\begin{array}{l} \min \;\; x_{101} + x_{111} + x_{121} + x_{110} + x_{112} \\ \text{s.t.} \\ 2n_{11*} - \left[4\left(n_{02*} + n_{20*}\right)\right] - n_{1**} + \left[(1 - 4\Phi)n_{*1*}\right] \le \left[(2 - 4\Phi)x_{11*}\right] - x_{101} - x_{121} \\ 2n_{1*1} - \left[4\left(n_{2*0} + n_{0*2}\right)\right] - n_{1**} + \left[(1 - 4\Phi)n_{**1}\right] \le \left[(2 - 4\Phi)x_{1*1}\right] - x_{110} - x_{112} \\ 2n_{*11} - \left[4\left(n_{*02} + n_{*20}\right)\right] - n_{**1} + \left[(1 - 4\Phi)n_{*1*}\right] \le \left[(1 - 4\Phi)x_{11*}\right] + 2x_{111} - x_{1*1} \\ x_{11*} = x_{111} + x_{110} + x_{112} \\ x_{1*1} = x_{111} + x_{101} + x_{121} \\ x_{101}, x_{111}, x_{121}, x_{110}, x_{112} \in Z_{\ge 0}\,. \end{array} \qquad (4)$$

We show how to select the kinship coefficient threshold *Φ* in Section 7 in the Supplementary Materials. Equation 4 shows the extended optimization model (in Equation 3) for a three members family.The objective function in the mixed integer programming model (shown in Equation 4) minimizes the number of SNP positions we need to hide, subject to kinship constraints derived using the kinship formula in [20]. For larger families with more than two members, the optimization model considers all the pairwise kinship coefficients among the related members. We use CPLEX (IBM Inc.) to solve the mixed-integer programming problem [21]. In our model, the number of constraints increases exponentially with the augmentation of family size, thus, becoming more difficult. We show an estimate of the time and memory requirements of the optimization model considering the different number of i) family members and, 2) SNPs, in Section 8 in the Supplementary Materials. The optimization model is run regularly when a new family member arrives at the dataset. First, we consider the overlapping SNP positions among the family members in the dataset. Once the number of positions and their configurations is determined by the optimization procedure, we select these positions from the overlapping region among the first-degree relatives e.g., the overlapping region between the mother-son and the father-son. If the number of required SNPs to hide is larger than the number of SNPs in the overlapping region (i.e., not enough SNPs exist in the overlapping region), we run the model to randomly remove the rest of the SNPs (i.e., outside the overlapping region) from the family members. Since the dataset is not public, we assume that the dataset owner knows the previously removed SNPs from the former arrivals. If not, alternatively, after completing the data collection, the dataset owner can 1) identify the families, and 2) process the genomes one by one to apply the selective hiding process, before sharing any statistical query from the dataset.

Hiding the overlapping SNPs among the family members allows to (1) preserve higher utility guarantees: it reduces the kinship estimation between multiple family members by hiding less number of SNPs, and (2) preserve higher privacy guarantees: it hides multiple SNPs for an SNP position to confuse a potential adversary trying to know sensitive information from the query results. Figure 2 shows how to hide from the new SNP set by choosing the SNPs overlapped with the previously hidden set. Note that the adversary (who sends statistical queries to the dataset) cannot observe the hidden SNPs as the dataset is not published.

In the following, we provide a toy example describing Figure 2 and how the proposed selective hiding process work for the individuals in the *Manual Corpas* family tree (described in detail in Section 5.1 and the family tree is shown in Figure 3).

1. Manual Corpas arrives at the dataset (or his genome is processed the first). No SNPs are hidden from his genome.

2. When the father arrives (or the father's genome is processed), we first calculate the number of required SNPs to be hidden from the father using the optimization model with the aim of reducing the kinship between the son and the father. Then, we find the available SNPs that we can hide from the father.

3. When the mother arrives, we also calculate the number of required SNPs to be hidden using the optimization model. We locate the available SNPs to be hidden from the mother in order to decrease the kinship coefficient between the son and the mother.

4. We determine the SNPs that lie in the overlapping region between the available SNPs that can be hidden from the father and the available SNPs that can be hidden from the mother since these SNPs are proven to contribute the most to the kinship coefficient between the son, father, and mother.

5. We start hiding from the father. We pick the required SNPs from the overlapping region, and the rest of the SNPs are selected randomly.

6. Then, we hide the mother's SNPs that lie in the overlapping region. After removing these SNPs, the mother and the son's kinship coefficient is then decreased by one familial degree compared to their original value. No need to hide extra SNPs from the mother. (This step shows how the heuristic approach minimizes the random selection).

7. If the aunt arrives later, we run the optimization model for four people in such a way that kinship coefficients between both aunt-mother and aunt-son decrease while preserving the decreased kinship coefficients as in the previous steps.

After repeating this selective hiding process for each dataset participant, sequentially, all (required) records in the dataset become obfuscated and the dataset can now accept statistical queries. We consider here the count query by the users (or the adversary). Following the attack scenario proposed by [9], to limit the number of dataset members included in the query results, the adversary sends its query specified by some demographic properties (e.g., age, address). Dataset owner computes the result of the query on the dataset with missing SNPs (missing SNPs of some dataset participants are due to the proposed selective hiding algorithm). Dataset owner reports (1) the query result (sum of all SNP values for the dataset participants that are considered in the query computation), and (2) the number of dataset participants that are used to compute the query results (q). Note that if a dataset participant is involved in the query computation, but its corresponding SNP has been hidden (due to the proposed selective hiding algorithm), that participant still contributes to the number of dataset participants q, which are used to compute the query result (i.e., from the adversary's point of view, the query is still computed over q individuals). In a response to a count query for an SNP j, the dataset owner computes a noisy query result $\widetilde{D^j_{pi}}$, by adding Laplace noise with parameter $2/\epsilon$. The query result includes the sum of the SNP j values for a target $i\left(D^j_i\right)$ and other p participants included in the query results $\left(D^j_p\right)$. We assume that the adversary has access to 1) auxiliary information about the membership of each participant including the target i, and 2) familial relationship R between the target and other individuals in the dataset (that is computed over the obfuscated dataset with the hidden SNPs and released as metadata by the dataset owner). After receiving the noisy query result $\widetilde{D_{pi}^{j}}$, the adversary can use the coin change algorithm [22] to obtain all possible partitions of total count (for SNP values) as a combination of the set {0, 1, 2}, where each partition should only include $\leq (p + 1)$ individuals. Next, for each valid partition, the adversary validates all the unique permutations using Mendel's law. Once validated, the adversary computes the probability of each permutation from Mendel's law by considering potential values of SNP j (0, 1, and 2) for the target i. Hence, the adversary can infer the value of D_i^j for target i using the SNP values of dependent people related to the target that is used to compute the query result, as shown in [9]. We show an example of how to perform an attribute inference attack using the count query results in Section 4 in the Supplementary Materials. To evaluate the privacy and utility performance of our proposed selective hiding algorithm, we use the correctness and utility loss metrics (explained in detail in Section 5.3) over a real-world genomic dataset to show the robustness of our mechanism. We next discuss our evaluation in detail.

5 Evaluation

5.1 Dataset Description

For the evaluation, our dataset D contains partial DNA sequences from two sources:

- 1000 Genomes phase 3 data [23]
- Manuel Corpas Family Pedigree [24]

1000 Genomes Phase 3 data—We use data from 1000 Genomes Phase 3 [23], to obtain data for the unrelated individuals from the same or different population of the target and his family members. We extract the genotypes from chromosome 22 for 176 participants from the European population using the Beagle genetic analysis package [25] to convert the values of genotypes to 0, 1, or 2 according to the number of minor alleles for each SNP.

Manuel Corpas (MC) Family Pedigree—Manuel Corpas [24] released his and his family members' genomes for research purposes. The dataset contains the DNA sequences in variant call format (VCF) for the father, mother, son (Manuel Corpas), daughter, and aunt. The family tree of the individuals in this dataset is illustrated in Figure 3. We choose the son to be the target and we use the genomic records of his first and second-degree family members (father, mother, and aunt).

We extract the common SNPs from all MC family members and 1000 Genomes members for the evaluation of the proposed algorithm. Finally, we combine the family genomic data with the unrelated individuals.

5.2 Evaluation Settings

To evaluate the proposed countermeasure against the attribute inference attack, we defined a *case-control dataset* D. D includes N individuals ($N = 180$) from the European population from the 1000 Genomes project dataset and MC family, in which $\left(\frac{N}{2} = 90\right)$ are cases and $\left(\frac{N}{2} = 90\right)$ controls. As discussed in Section 3, the adversary aims to infer m SNPs for a target i using the results of queries over dataset D. Here, we assume that the adversary knows 1) the true number of participated individuals (i.e., true number of SNPs) in the query result, and 2) the kinship coefficients of the dataset participants (e.g., from the metadata of the dataset). Note that kinship coefficients shared by the dataset are computed after the proposed selective sharing algorithm (reflecting the actual kinship coefficients in the final dataset), and hence they are obfuscated to provide robustness.

5.3 Evaluation Metrics

To evaluate the performance of the proposed algorithm against attribute inference attacks, we use the correctness metric. Utilizing the notion of the expected *estimation error*, the *correctness* of the adversary quantifies the difference distance (*Dist*) between 1) D_i^j, which is the true value of SNP j for the target individual i, and 2) $\widetilde{D}_i^j$, which is the inferred value of SNP j for the target individual i by the adversary. We can compute the probabilities for D_i^j using the Mendelian inheritance probabilities for an SNP i given all the potential SNP values (i.e., 0, 1, or 2) for D_i^j. We compute the correctness for all m targeted SNPs of the target i as follows:

$$C = 1 - \sum_{j=1}^{m} P\left(D_i^j \middle| \widetilde{D}_{pi}^j\right) \left| Dist\left(D_i^j, \widetilde{D}_i^j\right) \right|, \tag{5}$$

To quantify the *utility loss* (in terms of the quality or accuracy of the shared query responses) due to the proposed mechanism, we calculate the average change in the actual query result D_{pi}^j and the noisy query result $\widetilde{D}_{pi}^j$ considering all m targeted SNPs as follows:

$$U = \frac{1}{m} \sum_{j=1}^{m} \left| Dist\left(D_{pi}^j, \widetilde{D}_{pi}^j\right) \right|, \tag{6}$$

5.4 Experimental Results

In an inference attack, we assume the differentially private query results are computed by accounting for (1) target i and multiple first and second-degree family members in **F**; and (2) target i, multiple family members in **F**, and multiple other unrelated members (non-relatives) in **U**. We evaluate the performance of the attack under two assumptions:

- Independent assumption (w/o dep): the adversary assumes that there is no correlation between the participants in D.
- Dependent assumption (w/ dep): the adversary utilizes the familial relationships between the participants in D to perform the genome reconstruction for target i.

We also compare the proposed algorithm with the one proposed in [9], which aims to adjust the *privacy parameter* of DP to provide privacy guarantees for the dependent tuples in the dataset. According to [9, 10], if all the tuples in the dataset are independent, then the noisy query output achieves DP with the same privacy budget ϵ. However, if the dataset includes dependent tuples, one needs to augment the scale of Laplace noise using a smaller ϵ value (or a larger query sensitivity) to achieve DP. Using the notion of the “leaked information” ratio for different privacy budgets ϵ, [9] adjust the global sensitivity of the query to mitigate the information leaks resulting from the attribute inference attack. In the following, we (1) compare the dependent (referred to as “no hiding w/ dep” in the figure) and independent assumptions (referred to as “no hiding w/o dep” in the figure) to show the vulnerability due to independent assumption, (2) show the performance of our proposed mitigation algorithm (by hiding selective SNPs from the family members) against an adversary that uses the dependencies in its attack (referred to as “selective hiding” in the figure), (3) hide random SNPs (without using any optimization) from the family members rather than selective hiding, to show the benefits of selective hiding (referred to as “random hiding” in the figure), and (4) compare the proposed mitigation algorithm with the one in [9] to assess the proposed algorithm (referred to as “dependent sensitivity” in the figure).

Privacy Performance—In Figure 4, we evaluate the effect of different values of the privacy budget, ϵ, on the adversary’s correctness in inferring the targeted m SNPs. We also analyze the robustness of our proposed mechanism to the inference attack and compare it with the most similar existing work [9]. Here the query results include the statistics from the family members only. We start including 1 first-degree family member with the target i. First, we include the mother to the query results as in Figure 4(a), then we include the father of the target as in Figure 4(b)). Third, we include both the father and the mother in the query results, as in Figure 4(c). Last, we consider a second-degree family member (aunt of target i) in the query results along with the father and the mother of the target (Figure 4(d)).

Using the results of count queries over the case-control dataset D, we make the following key observations: (1) The correctness of the adversary with the knowledge of the data dependency is up to 50% more compared to the case in which the adversary does not consider the data dependency in the query results (Figure 4). (2) In accordance with the results of [10], the most accurate inference of the adversary is achieved when the query computation includes target i along with his father and mother (Figure 4(c)). Including a second-degree family member as in (Figure 4(d)) can enlarge the range of possible SNP values for the target, and hence make it more difficult to accurately infer the correct SNP value with a high probability. (3) Proposed selective hiding mechanism achieves better privacy for various privacy budgets, compared to the random hiding for different family members included in the query results, as illustrated in Figure 4.

Figure 5 shows the effect of different values of ϵ on adversary’s success in terms of its correctness in inferring m SNPs of the target i. We increase the number of non-relatives (from 5 to 20) that are included in the query computation along with first-degree family members of the victim. From these experimental results, we make the following observations:

1. In accordance with our previous observations in Figure 4, the probability of inferring the true value of the targeted m SNPs slightly increases (mostly 2%

−20%) depending on the knowledge of the adversary about the dependency between tuples, as the value of the privacy budget, ϵ, increases from 0.1 to 5. Hence, even when including a different number of non-relatives in the query results (e.g., the size of **U** changes from 5 to 20), there is a significant increase in the correctness of the adversary if the adversary has the knowledge of the data dependency, as shown Figure 5. However, in Figure 5, we observe that the difference between the correctness of the inferred SNPs with and without the knowledge of the data dependency is about 3 times less than when the query results include data for only family members of the target i (Figure 4).

2. Applying our proposed countermeasure by selectively hiding the family members' SNP values is superior to the dependent sensitivity mechanism in terms of correctness metric. Compared to the optimal DP privacy guarantees, in which we consider all the tuples to be independent ((No hiding w/o dep) in Figure 5), our proposed mechanism achieves (~5%) less privacy, while dependent sensitivity mechanism achieves (~15%) less privacy guarantees under the same privacy budget, ϵ.

3. Randomly hiding the SNPs of the family members results in achieving less privacy guarantees, even if we compare it with the correctness results of the attribute inference attack, where no hiding method is applied (e.g., no hiding w/dep in (Figure 5(a) and (b) for privacy budget, $\epsilon > 2.5$).

Next, Figure 6 shows the effect of different values of the privacy budget, ϵ, used in DP, on the correctness of the adversary, when we apply the selective hiding mechanism for family SNPs, considering a different number of family members to be included in the query results. The results illustrate the association between the privacy budget, ϵ, and the correctness of the adversary for inferring the *actual* values of the targeted m SNPs. The probability of inferring the correct values increases significantly (by 30%) as the budget privacy, ϵ, increases from 0.1 to 5, as shown in Figure 6. This is expected as the more ϵ values we use in the LPM-based DP, the less the added noise, and hence increasing the success of the inference attack. The choice of the parameter ϵ is difficult: values reported in the literature vary from as little as 0.01 to as much as 7 according to [26] or up to 100 according to [12], based on the required levels of privacy and utility for the given use case.

Finally, we explore the robustness of the selective hiding mechanism for a different number of related and unrelated people in the query results, without applying differential privacy. Figure 7 shows the relationship between the number of family members (as in Figure 7(a)) or the number of non-relatives (as in Figure 7(b)) in the query results and the probability of inferring the true SNPs value by the adversary when we apply selective hiding mechanism. The results show that increasing the number of family members or unrelated individuals included in the query result, using selective hiding mechanism slightly decreases the correctness of the adversary, thus improving privacy.

Utility Performance—Publishing statistics of genomic datasets results in utility gain for society as a whole. However, publishing these statistics could also result in privacy loss for the participants of the dataset, especially if the dataset includes correlated tuples.

Hence, the goal of our proposed mechanism is to ensure that the privacy loss is restricted to an acceptable level, without causing a high loss in the potential utility gain, when compared with the case of publishing the original statistical results. Using the utility loss metric introduced in Section 5.3, in the following we compare our proposed mechanism (referred to as “selective hiding” in the figure) with the existing dependent sensitivity countermeasure proposed in [9] (referred to as “dependent sensitivity” in the figure) and random hiding mechanism (referred to as “random hiding” in the figure) in terms of utility, using a MAF query over a dataset D with m=100 SNPs. Figure 8 and Figure 9 show the utility loss caused by hiding selective SNPs from the family members participating in the dataset D and then adding noise to achieve ϵ-DP by considering the dependence between tuples. As in Section 5.4, we consider the query results to include the statistics from the family members only (Figure 8). Then, we calculate the utility performance of the three mechanisms considering query results with different numbers of unrelated individuals (Figure 9). The results show that with smaller ϵ values, utility loss caused by the three mechanisms decreases. As previously discussed, the main idea of the dependent sensitivity mechanism [9] is augmenting the Laplace noise by decreasing the privacy budget, ϵ, value to achieve DP for any dataset with dependent tuples. Our proposed mechanism adds a significantly smaller amount of noise, when $\epsilon \leq 1$, and hence provides better utility. For example, when $\epsilon = 0.5$, and the query results include 5 unrelated individuals along with the family members (Figure 9(a)), the amount of utility loss caused by our mechanism is 33% of utility loss caused by the dependent sensitivity.

6 Conclusion

Developing new privacy-preserving techniques that facilitate sharing the outcomes of human genomic studies is necessary. The main goal of such techniques is to preserve the privacy of dataset donors without undermining the utility of the dataset, and hence the research outcomes. Differential privacy-based data perturbation techniques have known privacy limitations while sharing statistics from a genomic dataset that contains dependent tuples. In this paper, we propose a “selective hiding” mechanism to mitigate the privacy risks caused by the correlations between the dataset tuples. We assume a strong adversary who can send one query about one SNP, then the dataset owner can choose the appropriate privacy budget ϵ to release a noisy query result according to i) the required level of privacy and utility of the released data, and ii) the sensitive nature of the genomic dataset. We evaluate our perturbation mechanism over real-world genomic datasets and proved that it can achieve high privacy guarantees while minimizing utility loss. Our results show that the proposed scheme achieves both significantly better privacy and utility than the existing DP-based mechanisms. However, as a limitation of any DP scheme, it is known that sending multiple queries per the same dataset may degrade the privacy guarantees of DP [27–30]. It may be possible for us to consider this setting in our future research directions.

Supplementary Material

Refer to Web version on PubMed Central for supplementary material.

References

1. Humbert M, Huguenin K, Hugonot J, Ayday E, and Hubaux J-P, “De-anonymizing genomic databases using phenotypic traits,” Proceedings on Privacy Enhancing Technologies, vol. 2015, no. 2, pp. 99–114, 2015.
2. Uhlerop C, Slavković A, and Fienberg SE, “Privacy-preserving data sharing for genome-wide association studies,” The Journal of privacy and confidentiality, vol. 5, no. 1, p. 137, 2013. [PubMed: 26525346]
3. Backes M, Berrang P, Humbert M, and Manoharan P, “Membership privacy in microrna-based studies,” in Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security, pp. 319–330, 2016.
4. Homer N, Szelinger S, Redman M, Duggan D, Tembe W, Muehling J, Pearson JV, Stephan DA, Nelson SF, and Craig DW, “Resolving individuals contributing trace amounts of dna to highly complex mixtures using high-density snp genotyping microarrays,” PLoS genetics, vol. 4, no. 8, p. e1000167, 2008. [PubMed: 18769715]
5. Dwork C, “Differential privacy: A survey of results,” in International Conference on Theory and Applications of Models of Computation, pp. 1–19, Springer, 2008.
6. Liu C, Chakraborty S, and Mittal P, “Dependence makes you vulnberable: Differential privacy under dependent tuples.,” in NDSS, vol. 16, pp. 21–24, 2016.
7. Song S, Wang Y, and Chaudhuri K, “Pufferfish privacy mechanisms for correlated data,” in Proceedings of the 2017 ACM International Conference on Management of Data, pp. 1291–1306, ACM, 2017.
8. Zhao J, Zhang J, and Poor HV, “Dependent differential privacy for correlated data,” in 2017 IEEE Globecom Workshops (GC Wkshps), pp. 1–7, IEEE, 2017.
9. Almadhoun N, Ayday E, and Ulusoy Ö, “Differential privacy under dependent tuples—the case of genomic privacy,” Bioinformatics, vol. 36, no. 6, pp. 1696–1703, 2020. [PubMed: 31702787]
10. Almadhoun N, Ayday E, and Ulusoy Ö, “Inference attacks against differentially private query results from genomic datasets including dependent tuples,” Bioinformatics, vol. 36, no. Supplement_1, pp. i136–i145, 2020. [PubMed: 32657411]
11. Kale G, Ayday E, and Tastan O, “A utility maximizing and privacy preserving approach for protecting kinship in genomic databases,” Bioinformatics, vol. 34, no. 2, pp. 181–189, 2018. [PubMed: 28968635]
12. Yu F, Fienberg SE, Slavković AB, and Uhler C, “Scalable privacy-preserving data sharing methodology for genome-wide association studies,” Journal of biomedical informatics, vol. 50, pp. 133–141, 2014. [PubMed: 24509073]
13. Johnson A and Shmatikov V, “Privacy-preserving data exploration in genome-wide association studies,” in Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining, pp. 1079–1087, ACM, 2013.
14. Nissim K, Raskhodnikova S, and Smith A, “Smooth sensitivity and sampling in private data analysis,” in Proceedings of the thirty-ninth annual ACM symposium on Theory of computing, pp. 75–84, ACM, 2007.
15. Kifer D and Machanavajjhala A, “No free lunch in data privacy,” in Proceedings of the 2011 ACM SIGMOD International Conference on Management of data, pp. 193–204, ACM, 2011.
16. Kifer D and Machanavajjhala A, “A rigorous and customizable framework for privacy,” in Proceedings of the 31st ACM SIGMOD-SIGACT-SIGAI symposium on Principles of Database Systems, pp. 77–88, ACM, 2012.
17. He X, Machanavajjhala A, and Ding B, “Blowfish privacy: Tuning privacy-utility trade-offs using policies,” in Proceedings of the 2014 ACM SIGMOD international conference on Management of data, pp. 1447–1458, ACM, 2014.
18. Yang B, Sato I, and Nakagawa H, “Bayesian differential privacy on correlated data,” in Proceedings of the 2015 ACM SIGMOD international conference on Management of Data, pp. 747–762, ACM, 2015

19. Humbert M, Ayday E, Hubaux J-P, and Telenti A, “Addressing the concerns of the lacks family: quantification of kin genomic privacy,” in Proceedings of the 2013 ACM SIGSAC conference on Computer & communications security, pp. 1141–1152, ACM, 2013.

20. Manichaikul A, Mychaleckyj JC, Rich SS, Daly K, Sale M, and Chen W-M, “Robust relationship inference in genome-wide association studies,” Bioinformatics, vol. 26, no. 22, pp. 2867–2873, 2010. [PubMed: 20926424]

21. Cplex II, “V12. 1: User’s manual for cplex,” International Business Machines Corporation, vol. 46, no. 53, p. 157,2009.

22. D’Errico J, “Partitions of an integer,” MATLAB central file exchange, vol. 12009, 2018.

23. G. P. Consortium et al. , “A global reference for human genetic variation,” Nature, vol. 526, no. 7571, p. 68, 2015. [PubMed: 26432245]

24. Corpas M, “Crowdsourcing the corpasome,” Source code for biology and medicine, vol. 8, no. 1, p. 13, 2013. [PubMed: 23799911]

25. Browning BL, Zhou Y, and Browning SR, “A one-penny imputed genome from next-generation reference panels,” The American Journal of Human Genetics, vol. 103, no. 3, pp. 338–348, 2018. [PubMed: 30100085]

26. Hsu J, Gaboardi M, Haeberlen A, Khanna S, Narayan A, Pierce BC, and Roth A, “Differential privacy: An economic method for choosing epsilon,” in 2014 IEEE 27th Computer Security Foundations Symposium, pp. 398–410, IEEE, 2014.

27. Kairouz P, Oh S, and Viswanath P, “The composition theorem for differential privacy,” in International conference on machine learning, pp. 1376–1385, PMLR, 2015.

28. Wang Y-X, Lei J, and Fienberg SE, “On-average kl-privacy and its equivalence to generalization for maxentropy mechanisms,” in International Conference on Privacy in Statistical Databases, pp. 121–134, Springer, 2016.

29. Mironov I, “Rényi differential privacy,” in 2017 IEEE 30th computer security foundations symposium (CSF), pp. 263–275, IEEE, 2017.

30. Chaudhuri K, Imola J, and Machanavajjhala A, “Capacity bounded differential privacy,” Advances in Neural Information Processing Systems, vol. 32, 2019.

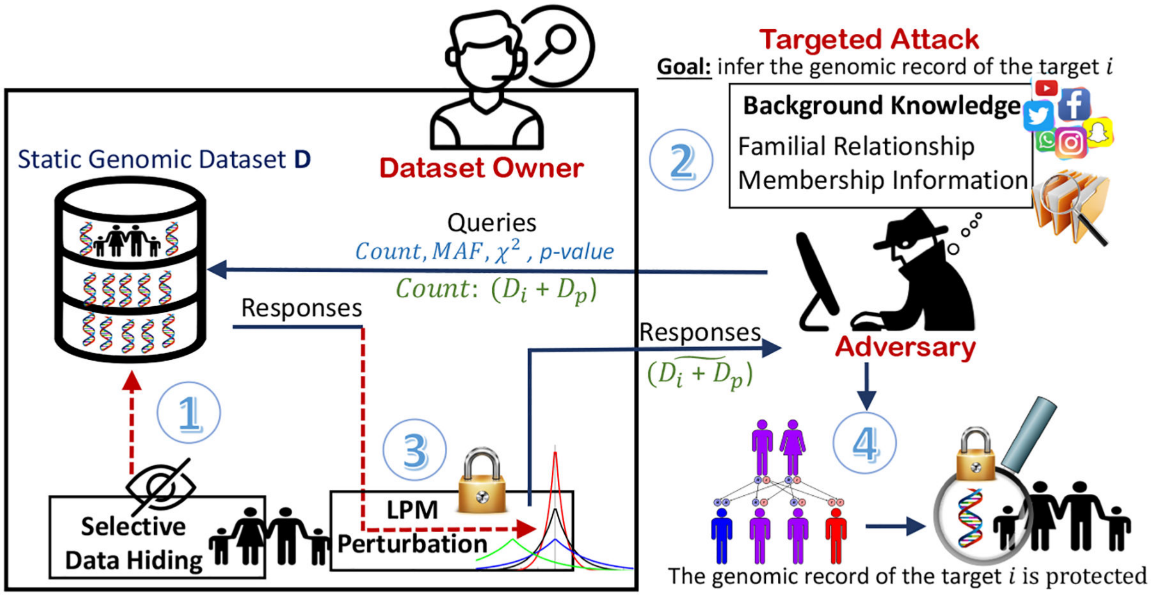

Fig. 1:

Our proposed model. (1) The dataset owner selectively hides SNPs from the family members included in the dataset during data collection. We consider a static genomic dataset, which is a one-time updated dataset. (2) The adversary sends the count queries to the dataset owner. (3) The dataset owner applies LPM to the query results and sends them to the adversary. (4) The adversary runs the attribute inference attack against the target i by using i) results of differentially-private count queries, ii) dependency between the target and target's family members that are in the dataset D, and iii) Mendel's law. In our threat settings, the adversary can obtain: i) the membership information of all dataset participants, and ii) the kinship coefficient between the dataset participants, from exploiting side-channel leaks.

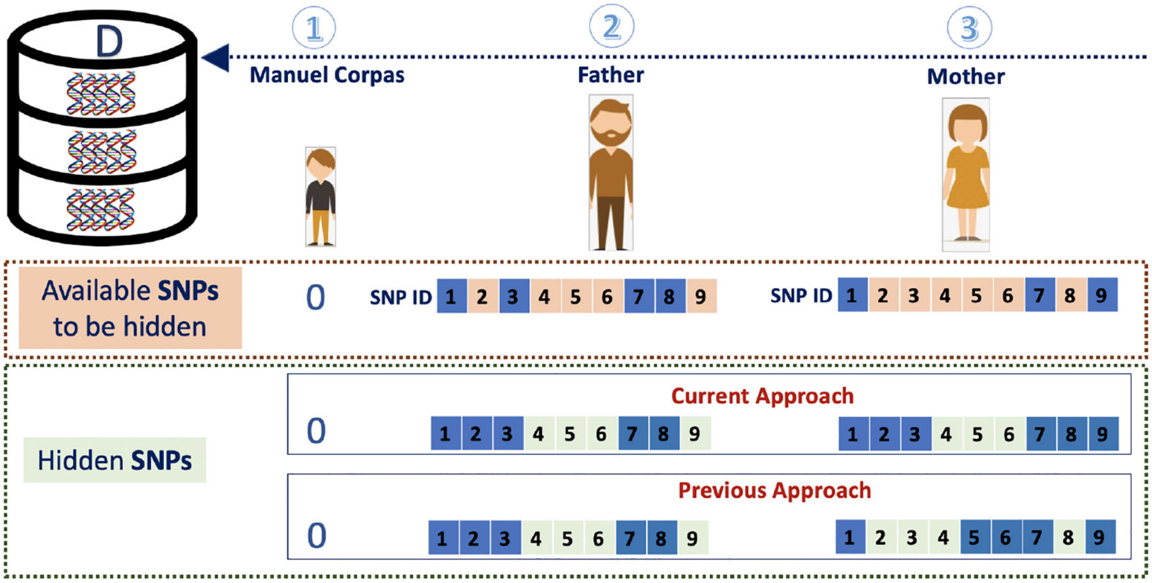

Fig. 2:

Comparison of our proposed approach and the one in [11]. The orange-colored SNPs denote the available SNP positions that can be hidden. Green-colored areas are the removed regions. In the proposed approach, we aim to hide from the region with maximal overlap.

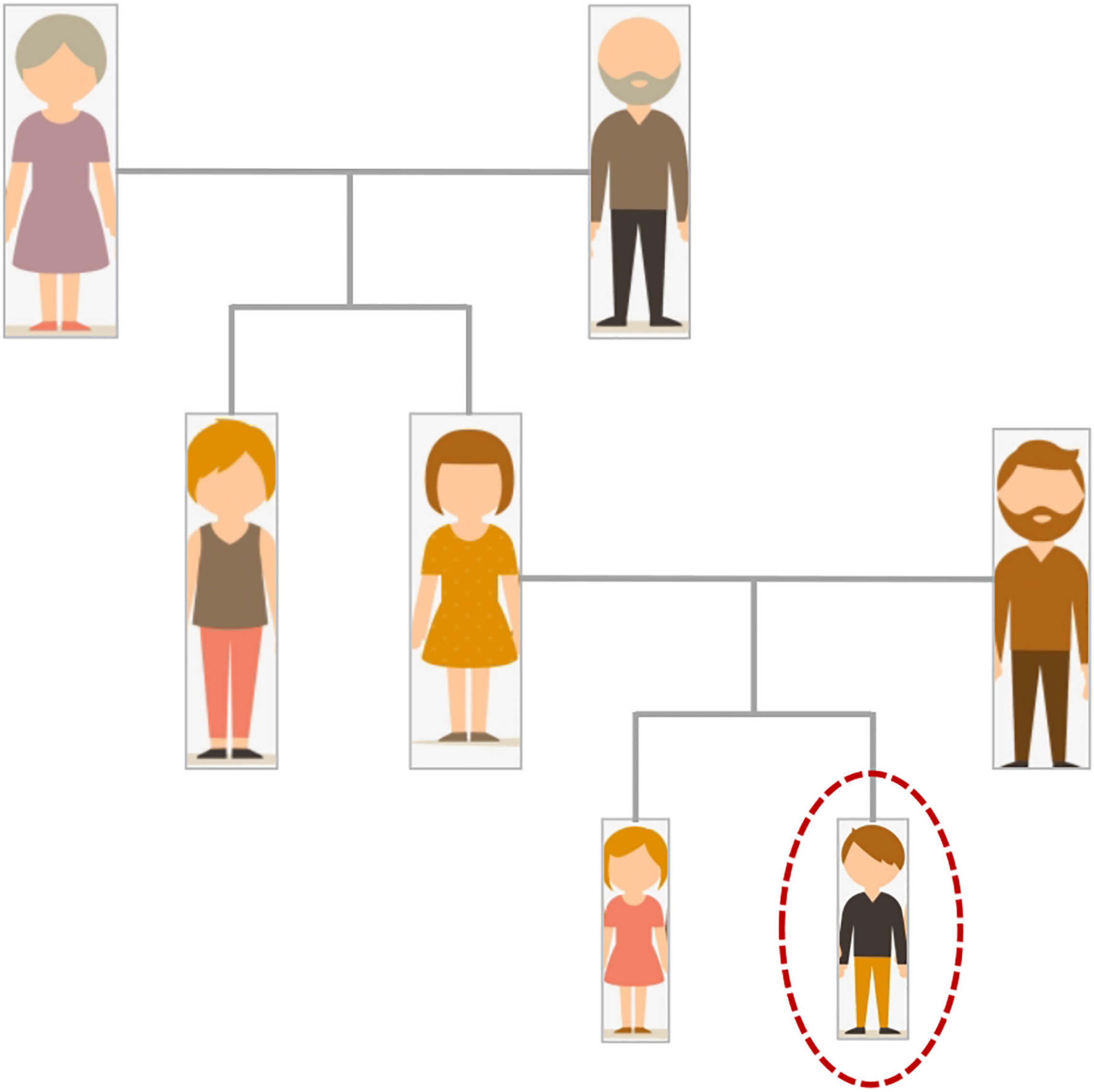

Fig. 3:
Manuel Corpas family tree.

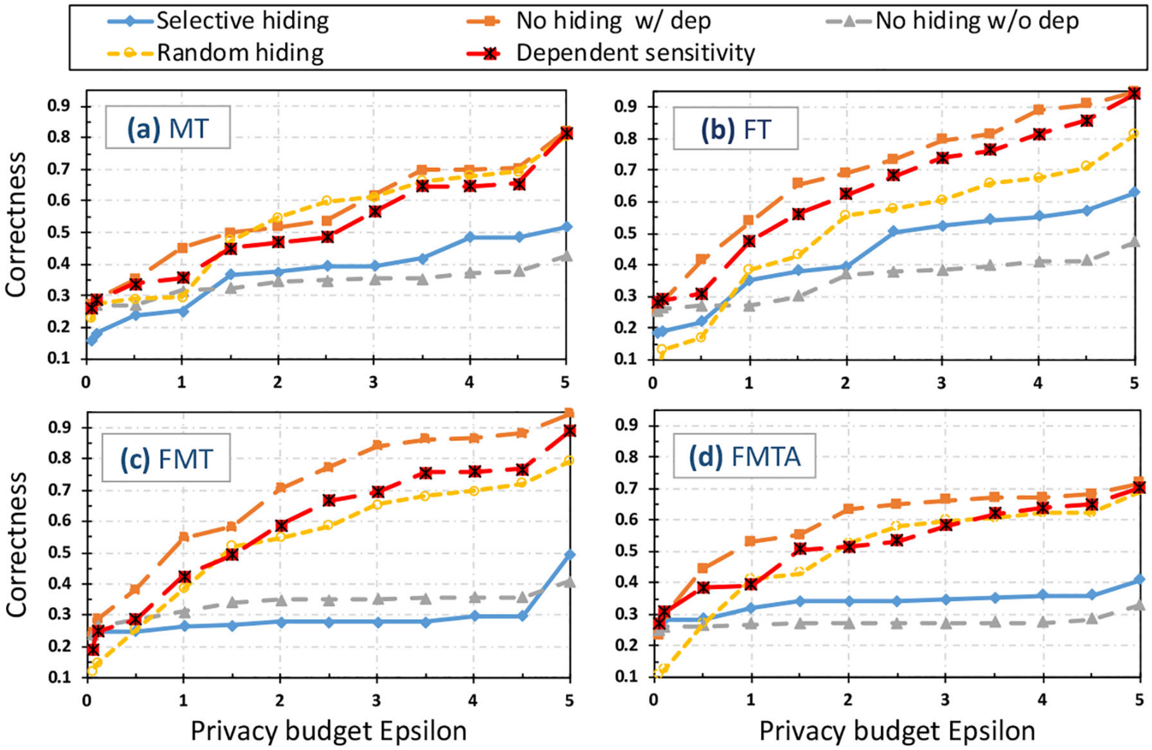

Fig. 4:

The effect of different values of the privacy budget, ϵ, on the adversary's correctness in inferring the targeted SNPs, considering a different number of family members in $\mathbf{F}(|\mathbf{F}| = f)$ included in the noisy results of count query. The query results include (a) MT: mother and target, (b) FT: father and target, (c) FMT: father, mother, and target (d) FMTA: father, mother, target, and aunt.

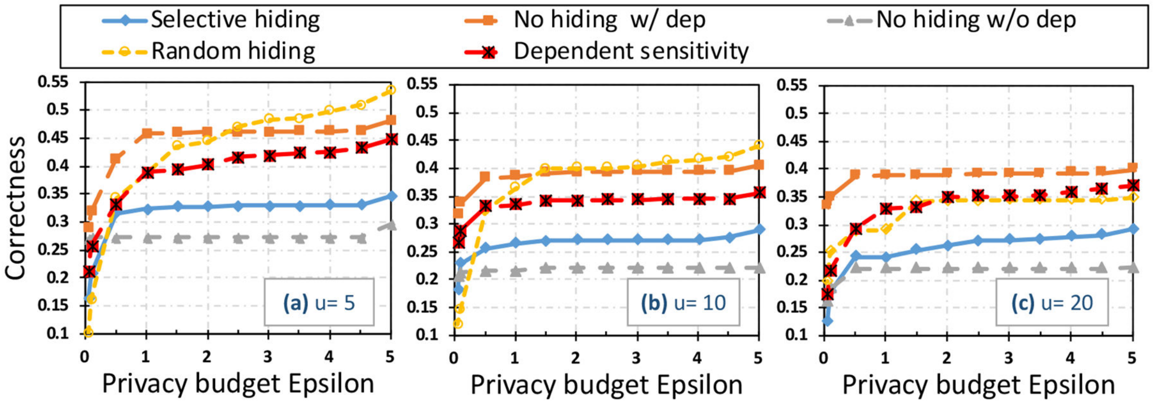

Fig. 5:

The effect of different values of the privacy budget, ϵ, on the adversary's correctness in inferring the targeted SNPs, considering 2 first-degree relatives (father and mother) with different numbers of non-relatives in $\mathbf{U}(|\mathbf{U}| = u)$ included in the noisy results of count query. The query results include 5, 10, and 20 unrelated members in (a), (b), and (c) respectively.

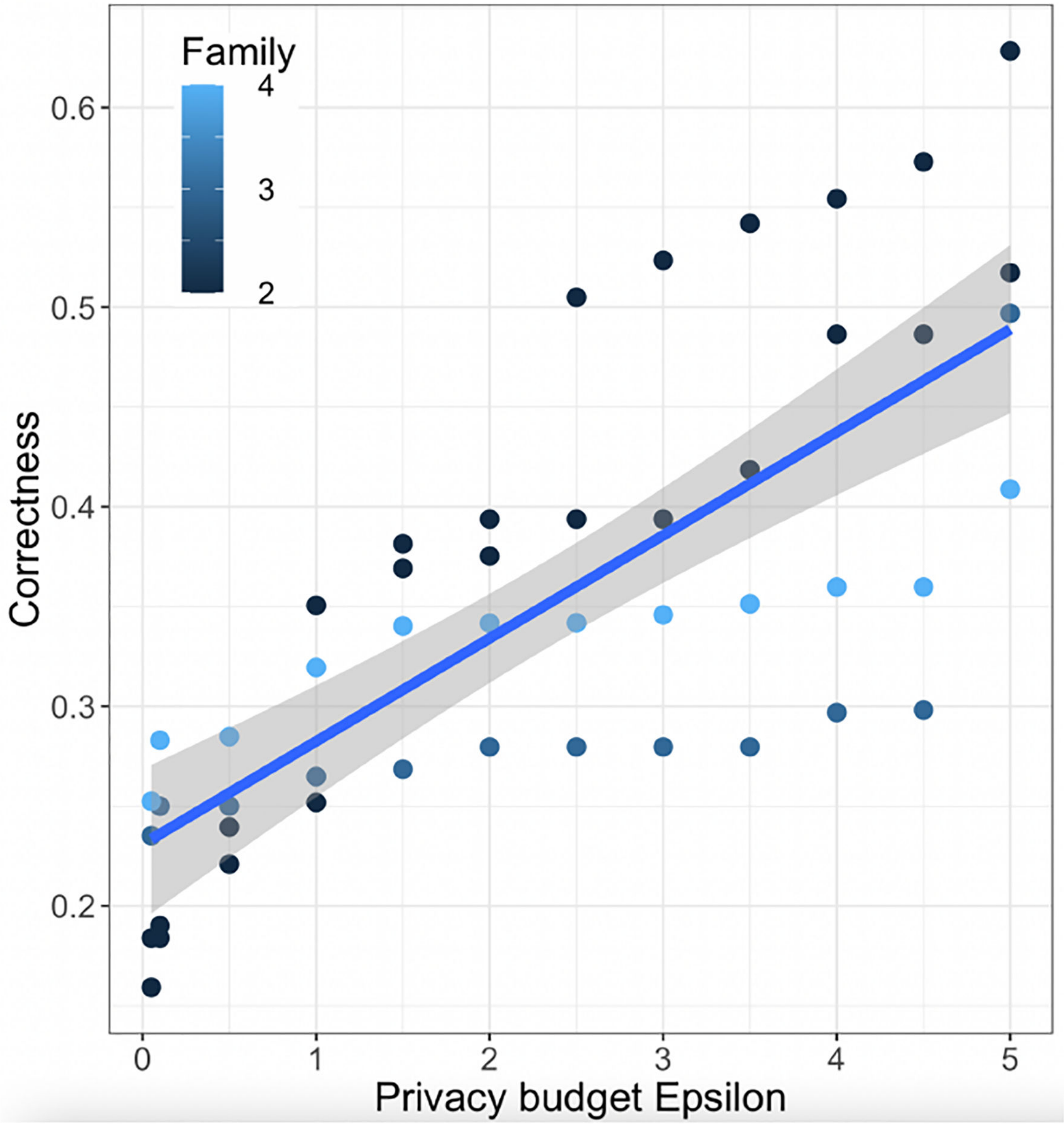

Fig. 6:

The effect of different values of the privacy budget, ϵ, on the adversary’s correctness in inferring the targeted SNPs, using a different number of family members in $\mathbf{F}(|\mathbf{F}| = f)$ included in the noisy results of count query.

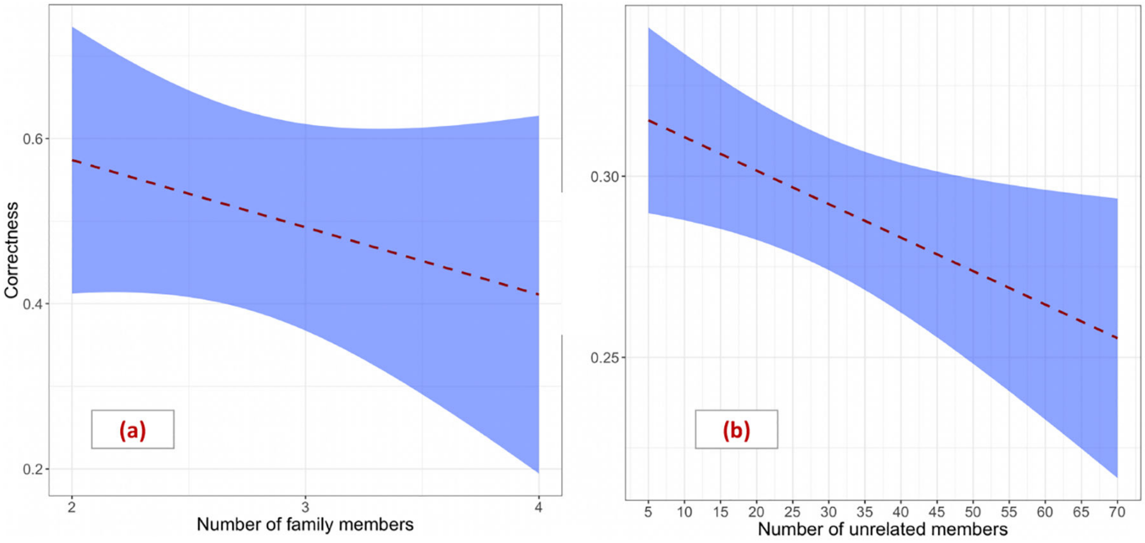

Fig. 7:

The relationship between different numbers of (a) family members in $\mathbf{F}(|\mathbf{F}| = f)$ and (b) non-relatives in $\mathbf{U}(|\mathbf{U}| = u)$ included in the noisy results of count query, and the adversary's correctness in inferring the targeted SNPs.

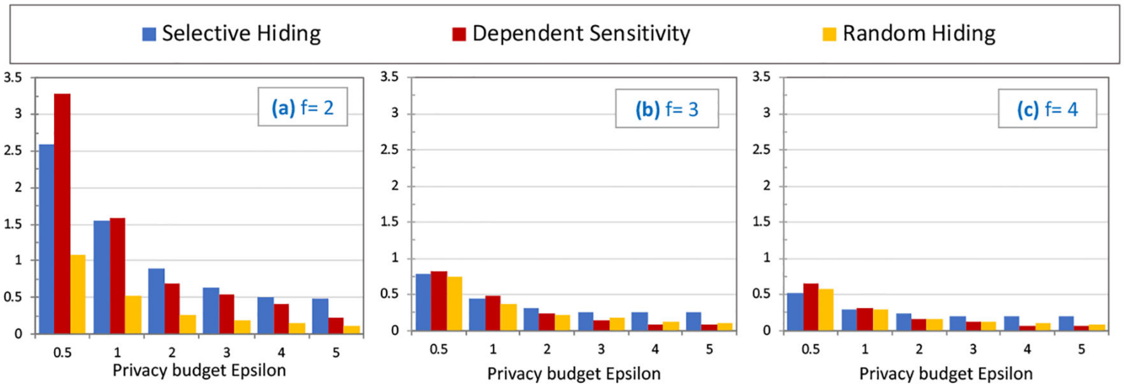

Fig. 8:

The effect of different values of the privacy budget, ϵ, on the utility loss caused by applying different mechanisms as countermeasures against the attribute inference attack, using a different number of family members in $\mathbf{F}(|\mathbf{F}| = f)$ included in the noisy results of MAF query.

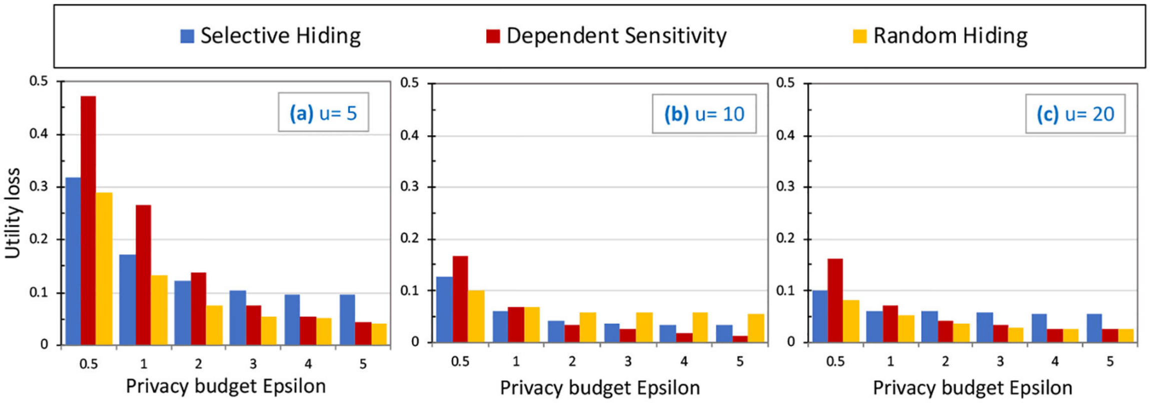

Fig. 9:

The effect of different values of the privacy budget, ϵ, on the utility loss caused by applying different mechanisms as countermeasures against the attribute inference attack, using a different number of non-relatives in $\mathbf{U}(|\mathbf{U}| = u)$ included in the noisy results of MAF query.