Privacy-Preserving Statistical Analysis of Genomic Data using Compressive Mechanism with Haar Wavelet Transform*

Akito Yamamoto † Tetsuo Shibuya

Human Genome Center, the Institute of Medical Science, the University of Tokyo, Tokyo, Japan

Abstract

To promote the use of personal genome information in medicine, it is important to analyze the relationship between diseases and the human genomes. Therefore, statistical analysis using genomic data is often conducted, but there is a privacy concern with respect to releasing the statistics as they are. Existing methods to address this problem using the concept of differential privacy cannot provide accurate outputs under strong privacy guarantees, making them less practical. In this study, for the first time we investigate the application of a compressive mechanism to genomic statistical data and propose two approaches. The first is to apply the normal compressive mechanism to the statistics vector along with an algorithm to determine the number of nonzero entries in a sparse representation. The second is to alter the mechanism based on the data, aiming to release significant SNPs with a high probability. In this algorithm, we apply the compressive mechanism with the input as a sparse vector for significant data and the Laplace mechanism for non-significant data. Using the Haar transform for the wavelet matrix in the compressive mechanism is advantageous to determine the number of nonzero elements and the amount of noise. In addition, we theoretically prove that our proposed method achieves ϵ-differential privacy. We evaluated our methods in terms of accuracy, rank error, and run time compared to the Laplace and exponential mechanisms. The results show that our second method in particular can guarantee both high privacy assurance as well as utility. The Python implementation of our experiments is available at `https://github.com/ay0408/CompLaplace`.

1 Introduction

In recent years, the availability of the information on human genome has increased, and more opportunities to utilize these data for personalized medicine are expected [20]. Therefore, it is essential to analyze the relationship between diseases and genomic information, and investigations, such as genome-wide association studies (GWAS) are often conducted. However, these tests contain sensitive personal information, and there is a risk of identifying an individual [33]. In fact, several studies [21, 30, 34] have been conducted on methods to identify individuals by using large-scale genomic data. As a result, the NIH has stopped releasing aggregate GWAS data [40], making it difficult to freely utilize these data.

Therefore, it is desirable to develop techniques for secure genomic analysis and methods for the private publication of statistical data. Various studies have been conducted on protecting data privacy using cryptography, secret sharing, and multiparty computation [6, 7, 11, 41], but they need additional protections of calculation results. Thus, in this study, we focus on the concept of differential privacy [15], a strategy that is being considered for protecting and utilizing genomic data [5, 17, 32, 38]. The concept of differential privacy is based on the fact that when some degree of perturbation is introduced in the original dataset, it becomes almost impossible to determine whether the dataset contains information pertaining to a specific individual. This concept might enable privacy protection even if noisy data are released to the public.

However, existing methods [5, 17, 36, 38] that employ the strategy of differential privacy have the limitation that there is a lot of differences between noisy data and the original data. These methods require a large budget ϵ for ensuring the utility of the data while achieving high privacy. The exponential mechanism using the shortest Hamming distance scores [23] provides relatively high accuracy, but it requires the calculation of the score for each statistic every time, and few efficient algorithms exist for this purpose. To circumvent

*Presented at Privacy and Security Workshop at RECOMB'22.

†Corresponding Author. Email: a-ymmt@ims.u-tokyo.ac.jp

this problem, methods [22, 31] employing probabilistic modeling have been proposed, but there is still room for improvement in terms of accuracy.

In this study, we propose new methods providing differentially privacy with a high utility to release genomic data. The goal of this study is to privately release the top K significant single nucleotide polymorphisms (SNPs). First, we consider subjecting genomic data to a compressive mechanism [25], which is based on compressed sensing and is suitable for vector data with a sparse representation. It has been studied for publication of histograms [2] and for use in network systems [12]; however, to the best of our knowledge, this study is the first to apply this method to genomic statistical data. We also present an algorithm to determine the number of nonzero entries in a sparse vector obtained using compressed sensing to make the compressive mechanism more reliable. However, because statistical data do not always have a sparse distribution, we consider the input data as a sparse vector and proposed another mechanism that combines the compressive mechanism and Laplace mechanism [16]. This mechanism is aimed at enabling the release of significant SNPs with a high probability by distinguishing between significant and non-significant data. In our methods, we use the Haar wavelet transform [18] as an orthonormal basis in the compressive mechanism as it is easy to determine the number of nonzero entries because the inverse matrix matches the transpose matrix, and as the *sensitivity* can be analyzed easily to calculate the amount of noise. We also theoretically guarantee that our methods achieve ϵ-differential privacy.

In our experiments, we compare the results obtained using our two methods with those obtained using the Laplace mechanism [16] and the exponential mechanism [27], the key mechanisms for ensuring differential privacy. First, we evaluate the accuracy and rank error of the top K outputs, and show that our methods have high utility. We also measure the run time while varying the size of the datasets. In addition, we subject real data [24] to our methods to verify their usefulness.

In Section 2, we describe the basic and preliminary assumptions. In Section 3, we present ϵ-differentially private methods using the compressive mechanism with the Haar wavelet transform to release the top K significant SNPs based on genomic statistics data. In Section 4, we evaluate their utility based on simulations and real data. In Section 5, we summarize our study and directions for future work.

2 Preliminaries

2.1 Differential Privacy

Differential privacy [15] is a framework developed in the field of cryptography to protect personal data; currently, it is being extensively considered for releasing genomic data [17, 32, 38]. It is also being extensively used in the field of deep learning [1] and healthcare [3]. The concept of differential privacy is based on the idea that it is almost impossible to distinguish between two *neighboring* datasets that differ with respect to only one element. This element is determined based on the data to be protected, which often represents a single individual or one family in the case of genomic data. The privacy level in differential privacy is evaluated by the parameter $\epsilon > 0$, and the following is the definition of ϵ-differential privacy.

Definition 1. (ϵ-*Differential Privacy*)
A randomized mechanism M is ϵ-differentially private if, for all neighboring datasets D and D' and any $S \subset$range(M),

$$\Pr[M(D) \in S] \leq e^{\epsilon} \cdot \Pr[M(D') \in S].$$

The smaller the value of ϵ, the stronger the privacy guarantee achieved, but the utility of such data is low. In general, the value of ϵ is set in the range of 0.01 to 10 [19], but when a higher privacy is required such as in the case of genomic data, it is desirable to adopt a value less than 1.

2.1.1 Laplace Mechanism

The simplest mechanism to achieve ϵ-differential privacy is the Laplace mechanism [16], which introduces perturbations according to the *sensitivity* of the function to obtain the output. The definition of the *sensitivity* is as follows, and in this study, we view a statistic as a function.

Definition 2. (*Sensitivity for the Laplace mechanism*)
The sensitivity of a function $f : \mathcal{D}^M \to \mathbb{R}^d$ *is*

$$\Delta f = \max_{D,D'} \|f(D) - f(D')\|_1,$$

where $D, D' \in \mathcal{D}^M$ *are neighboring datasets.*

For a statistic $f(D)$ obtained from the original dataset D, release of $f(D)+b$ satisfies ϵ-differential privacy when b is random noise derived from a Laplace distribution with mean 0 and scale $\frac{\Delta f}{\epsilon}$ [16].

2.1.2 Exponential Mechanism

The exponential mechanism [27] is a technique for providing the desired data with a high probability and has often been used to release statistically significant data [32, 38]. The desirability of the output is indicated by a score function, and we consider an exponential distribution using the *sensitivity* of the function. As a score function, the χ^2-statistics, p-values, and shortest Hamming distance (SHD) score [23] are commonly used. The SHD score provides higher accuracy than statistics [32, 38], but for most tests, efficient methods to obtain the score has not been proposed and is not yet generally available. Therefore, in our experiments, we compare our proposed methods with the exponential mechanism using χ^2-statistics to confirm their usefulness. The *sensitivity* for the exponential mechanism is defined as follows.

Definition 3. (*Sensitivity for the exponential mechanism*)
The sensitivity of a score function $u : \mathcal{D}^M \times \{1, 2, \ldots, M\} \to \mathbb{R}$ *is*

$$\Delta u = \max_{r} \max_{D,D'} \|u(D, r) - u(D', r)\|_1,$$

where $r \in \{1, 2, \ldots, M\}$ *and* $D, D' \in \mathcal{D}^M$ *are neighboring datasets.*

Following the above definition, we choose mechanism $\mathcal{M}_u^\epsilon$, which has the distribution:

$$\mathcal{M}_u^\epsilon = \frac{\exp\left(\frac{\epsilon u(D,r)}{2\Delta u}\right)}{\sum_{s\in\{1,\ldots,M\}} \exp\left(\frac{\epsilon u(D,s)}{2\Delta u}\right)}.$$

Then, releasing $\mathcal{M}_u^\epsilon$ satisfies the definition of ϵ-differential privacy.

2.2 Compressive Mechanism

The compressive mechanism is a differential privacy method proposed by Li *et al.* [25]. This mechanism uses the technique of compressed sensing and has the advantage of requiring less noise than other mechanisms when the input has a sparse representation. Firstly, we briefly review the procedure involved in compressed sensing.

2.2.1 Compressed Sensing

Compressed sensing was introduced by Candès *et al.* [8, 9] and Donoho [14], and has been employed in various fields, such as image processing [39] and computational biology [10].

In compressed sensing, we consider an input vector $d \in \mathbb{R}^n$ with a sparse representation. Here, d can be expressed as $d = \Psi x$, where $\Psi \in \mathbb{R}^{n\times n}$ is an orthonormal basis and $x \in \mathbb{R}^n$ is a sparse vector. Note that if x has s $(< n)$ nonzero entries, then x is an s-sparse vector. The goal of compressed sensing is to recover this sparse vector x form the input vector d. The procedure is divided into the following processes: "sampling process" and "reconstruction process", and we explain each in the following.

The sampling process reduces the size of the input from n to k $(< n)$ by mapping from the input vector d to a vector $y \in \mathbb{R}^k$. In this process, we use a random matrix $\Phi \in \mathbb{R}^{k\times n}$ and calculate $y = \Phi d$. For the

compressive mechanism, the random matrix Φ is formed by sampling independent and identically distributed (i.i.d.) entries from a symmetric Bernoulli distribution. Half of the Φ entries are expected to be $\frac{1}{\sqrt{k}}$ and the other half $-\frac{1}{\sqrt{k}}$, or more precisely, the following equation holds: $\Pr\left(\Phi(i,j) = \pm\frac{1}{\sqrt{k}}\right) = \frac{1}{2}$. Here, when we set $k = \Theta\left(s\log(n/s)\right)$, it is known that for any orthonormal basis Ψ, $A = \Phi\Psi \in \mathbb{R}^{k\times n}$ satisfies the restricted isometry property (RIP), a property favorable for reconstructing x, with a very high probability [4].

The reconstruction process calculates a vector $x \in \mathbb{R}^n$ that satisfies $y = \Phi\Psi x$. Using l_1-norm minimization, such as orthogonal matching pursuit (OMP) [28] or subspace pursuit [13], we solve the following optimization problem:

$$\hat{x} = \arg\min_{x'} \|x'\|_1, \text{ subject to } y = Ax,$$

where $A = \Phi\Psi$. In this study, we adopt the Haar wavelet transform as Ψ.

In the compressive mechanism, we first calculate $y = \Phi d$ from the original data d, and then subject the vector y to the Laplace mechanism. Here, if the *sensitivity* of d is Δ_d, that of y is expressed as $\Delta_d \cdot \sqrt{k}$ [25]. Therefore, we can introduce Laplace noise with scale $\frac{\Delta_d \cdot \sqrt{k}}{\epsilon}$ to the vector y. Based on the resulting $\hat{y}$, we can estimate a sparse vector $\hat{x}$ using the reconstruction process, and by computing $\hat{d} = \Psi\hat{x}$, we can release privacy-protected data.

2.3 Haar Wavelet Transform

The Haar wavelet transform [18] is one of the simplest wavelet transforms and can be used for frequency analysis. Similar to compressed sensing, it has been employed in various fields such as image processing [29] and pattern recognition [26]. Owing to the fact that the *sensitivity* of the Haar wavelet transform is easy to analyze, it can be used along with the concept of differential privacy [35].

For the Haar wavelet transform, we use the Haar transform matrix $H \in \mathbb{R}^{n\times n}$, where n is a power of 2. This matrix H is an orthonormal basis, and the elements of H can be obtained using the following equation:

$$\begin{cases} H_{0j} = \frac{1}{\sqrt{n}} \quad \text{if } j = 0, 1, \ldots, n-1, \\ H_{ij} = \frac{1}{\sqrt{n}} \cdot \begin{cases} 2^{\frac{\lfloor \log_2 i \rfloor}{2}} & \text{if } i \neq 0 \wedge n\left(i - 2^{\lfloor \log_2 i \rfloor}\right)/2^{\lfloor \log_2 i \rfloor} \leq j < n\left(i - 2^{\lfloor \log_2 i \rfloor} + \frac{1}{2}\right)/2^{\lfloor \log_2 i \rfloor}, \\ -2^{\frac{\lfloor \log_2 i \rfloor}{2}} & \text{if } i \neq 0 \wedge n\left(i - 2^{\lfloor \log_2 i \rfloor} + \frac{1}{2}\right)/2^{\lfloor \log_2 i \rfloor} \leq j < n\left(i - 2^{\lfloor \log_2 i \rfloor} + 1\right)/2^{\lfloor \log_2 i \rfloor}, \\ 0 & \text{otherwise.} \end{cases} \end{cases}$$

In this study, we calculate the *sensitivity* of the Haar wavelet transform and combine it with the compressive mechanism to propose a new method for releasing differentially private genomic data.

3 Methods

The goal of this study is to release the top K significant SNPs based on χ^2-statistics in high-throughput analyses of genomic datasets, such as GWAS, while achieving ϵ-differential privacy. Therefore, we assume that the statistics vector g in the following algorithms consists of χ^2-statistics. We first consider subjecting the genomic data to the simple compressive mechanism proposed by Liu *et al.* [25]. In this method, we focus on determining the number of nonzero entries in the sparse vector. In addition, we combine the compressive mechanism with the Haar wavelet transform and the Laplace mechanism, which can release significant SNPs with high accuracy. We also theoretically prove that this method satisfies ϵ-differential privacy.

3.1 Compressive Mechanism

In this study, we use the value of *sensitivity* according to each statistic. Several studies [17, 36, 38, 32, 37] have shown the *sensitivity* for the χ^2-test, the Cochran-Armitage trend test, and the transmission disequilibrium test. In the future, it is expected that other statistics will also be analyzed and when we know the *sensitivity*

of statistics that we want to release in the same way as in previous studies, the statistical data vector can be subjected to the compressive mechanism. Algorithm 1 represents the detailed procedure for releasing the top K data based on the statistics. Note that we need to introduce $2K$ times more Laplace noise [5] in each statistic than in the case where we release a single statistic. In addition, the size k of the vector y is calculated by $k = \lfloor s\log(n/s) \rfloor$ for ease of reconstruction.

Algorithm 1 ϵ-differentially private algorithm for releasing the top K significant SNPs using the compressive mechanism.

Input: Genomic statistics data $g \in \mathbb{R}^m$, the *sensitivity* of the statistic Δ_g, number of nonzero entries in the sparse representation s, number of data to release K, and privacy budget ϵ.

Output: Top K significant SNPs.

1: $n = 2^{\lceil \log_2 m \rceil}$, $k = \lfloor s\log(n/s) \rfloor$
2: Create $d \in \mathbb{R}^n$, such that if $i = 0, 1, \ldots, m-1$ then $d_i = g_i$, otherwise $d_i = 0$.
3: Generate a random matrix $\Phi \in \mathbb{R}^{k \times n}$ with i.i.d. symmetric Bernoulli distribution and a normalized Haar transform matrix $\Psi \in \mathbb{R}^{n \times n}$.
4: Calculate $y = \Phi d \in \mathbb{R}^k$.
5: Add Laplace noise with mean 0 and scale $\frac{2K\Delta_g \cdot \sqrt{k}}{\epsilon}$ to each y_i, and obtain a noisy sample $\hat{y}$.
6: Reconstruct $\hat{x} \in \mathbb{R}^n$, such that $\hat{y} = \Phi\Psi\hat{x}$.
7: Choose the top K SNPs based on the first m-th element of $\hat{d} = \Psi\hat{x}$.

In Algorithm 1, we assume that the number of nonzero entries of the sparse vector x is known. However, because the actual statistic vector may not have a sparse representation, it is necessary to establish an appropriate s. Therefore, we also propose an algorithm to determine the value s. In this algorithm, we first find x^* that satisfies $d = \Psi x^*$. Here, Ψ is the normalized Haar transform matrix, and $\Psi^{\mathrm{T}} = \Psi^{-1}$. Hence, x^* can be calculated as $x^* = \Psi^{\mathrm{T}} d$. Then, we describe s as the number of elements of x^* that are greater than the threshold η. The above procedure is shown in Algorithm 2.

Algorithm 2 Algorithm for determining the number of nonzero entries in a sparse representation of $g \in \mathbb{R}^m$ for Algorithm 1.

Input: Genomic statistics data $g \in \mathbb{R}^m$, a threshold η.

Output: The number of nonzero entries in the sparse representation s.

1: $n = 2^{\lceil \log_2 m \rceil}$
2: Create $d \in \mathbb{R}^n$, such that if $i = 0, 1, \ldots, m-1$ then $d_i = g_i$, otherwise $d_i = 0$.
3: Generate a normalized Haar transform matrix $\Psi \in \mathbb{R}^{n \times n}$.
4: Calculate $x^* = \Psi^{\mathrm{T}} d \in \mathbb{R}^n$.
5: Set s to be the number of elements of x^* which are greater than η.

3.2 Compressive Mechanism + Laplace Mechanism

Algorithm 1 assumes that the input vector g has a sparse representation. However, this assumption does not necessarily hold, in which case the introduced noise upon using this method would be greater than that associated with the Laplace mechanism. To overcome this problem, we consider finding the vector d in the compressive mechanism from a sparse input vector x. Specifically, we first create a sparse vector $x \in \mathbb{R}^n$ with the top $\tilde{s}$ entries of the statistics vector $g \in \mathbb{R}^m$ as nonzero elements. We then reconstruct a noisy sparse vector $\hat{x}$ using a methodology similar to that employed with the compressive mechanism. For the other entries, we employed the Laplace mechanism, and finally, we select the top K elements from all the obtained noisy statistics. The above procedure is shown in Algorithm 3.

The value $\tilde{s}$ in Algorithm 3 assumes the number of high statistics in the input statistics. In general, significant statistics are often quite larger than non-significant statistics in genetic statistical tests, as shown in Manhattan plots. Because the general Laplace mechanism cannot capture this feature, we intend to differentiate the significant SNPs from non-significant SNPs by changing the mechanism employed. In

Algorithm 3 ϵ-differentially private algorithm for releasing the top K significant SNPs using the compressive mechanism with Haar wavelet transform and the Laplace mechanism.

Input: Genomic statistics data $g \in \mathbb{R}^m$, the *sensitivity* of the statistic Δ_g, number of data to apply the compressive mechanism $\tilde{s}$, number of data to release K, and privacy budget ϵ.

Output: Top K significant SNPs.

1: $n = 2^{\lceil \log_2 m \rceil}$, $k = \lfloor \tilde{s} \log(n/\tilde{s}) \rfloor$
2: Let t be the $\tilde{s}$-th largest value among the elements of g.
3: Create $x \in \mathbb{R}^n$, such that if $i < m \wedge g_i \geq t$ then $x_i = g_i$, otherwise $x_i = 0$.
4: Generate a random matrix $\Phi \in \mathbb{R}^{k \times n}$ with i.i.d. symmetric Bernoulli distribution and a normalized Haar transform matrix $\Psi \in \mathbb{R}^{n \times n}$.
5: Calculate $y = \Phi \Psi x \in \mathbb{R}^k$.
6: Add Laplace noise with mean 0 and scale $\frac{2K}{\epsilon} \cdot \sqrt{k} \cdot \Delta_g \cdot \frac{1+(\sqrt{n}-1)(\sqrt{2}+1)}{\sqrt{n}}$ to each y_i, and obtain a noisy sample $\hat{y}$.
7: Reconstruct $\hat{x} \in \mathbb{R}^n$, such that $\hat{y} = \Phi \Psi \hat{x}$ and $\hat{x}_i \neq 0$ iff $x_i \neq 0$.
8: Add Laplace noise with mean 0 and scale $\frac{2K\Delta_g}{\epsilon}$ to each g_i, and obtain a noisy vector $\hat{g}$.
9: Set $\hat{g}_i = x_i$ if $g_i \geq t$.
10: Choose the top K SNPs from $\hat{g}$.

addition, the amount of noise in Algorithm 3 is different from that generated upon using Algorithm 1 because the input vector in the compressive mechanism is a sparse vector x. We show the *sensitivity* of the Haar wavelet transform by Theorem 1.

Theorem 1. *When the sensitivity of a vector $x \in \mathbb{R}^n$ is Δ_x, the sensitivity of the Haar wavelet transform for x is $\Delta_x \cdot \frac{1+(\sqrt{n}-1)(\sqrt{2}+1)}{\sqrt{n}}$.*

Proof. The Haar wavelet transform can be defined as the following function $h : \mathbb{R}^n \to \mathbb{R}^n, \quad h(x) = \Psi x$, where Ψ is a normalized Haar transform matrix.

When an element x_i changes by t_i, $\|h(x)\|_1$ changes by at most

$$\frac{t_i}{\sqrt{n}} \cdot \left(1 + 1 + \sqrt{2} + \sqrt{4} + \cdots + \sqrt{\frac{n}{2}}\right) = \frac{t_i}{\sqrt{n}} \cdot \left(1 + (\sqrt{n}-1)(\sqrt{2}+1)\right).$$

Here, the maximum change in $\|x\|_1$ is Δ_x, and $\sum_{i=0}^{n-1} t_i \leq \Delta_x$ is satisfied. Therefore, the *sensitivity* of the Haar wavelet transform is $\Delta_x \cdot \frac{1+(\sqrt{n}-1)(\sqrt{2}+1)}{\sqrt{n}}$. □

From Theorem 1, when the *sensitivity* of the statistic is Δ_g, that of $d = \Psi x \in \mathbb{R}^n$ can be expressed as $\Delta_g \cdot \frac{1+(\sqrt{n}-1)(\sqrt{2}+1)}{\sqrt{n}}$. Therefore, the *sensitivity* of $y = \Phi d \in \mathbb{R}^k$ is $\sqrt{k} \cdot \Delta_g \cdot \frac{1+(\sqrt{n}-1)(\sqrt{2}+1)}{\sqrt{n}}$, in accordance with Li *et al.* [25]. The amount of noise in the Laplace mechanism is in agreement with that proposed by Yu *et al.* [38]. In addition, we theoretically prove that Algorithm 3 satisfies ϵ-differential privacy by Theorem 2.

Theorem 2. *Algorithm 3 satisfies ϵ-differential privacy.*

Proof. Let U be the set of data contained in the input vector and W be the set of K data obtained by Algorithm 3. In addition, let $g, g' \in \mathbb{R}^m$ be a neighboring statistics dataset. We set $\mathcal{A}$ as the mechanism represented by Algorithm 3, and we will show $\Pr[\mathcal{A}(g) = W] \leq e^{\epsilon} \cdot \Pr[\mathcal{A}(g') = W]$.

Here, suppose that the input vector is $\mathcal{G}$, and let the noisy statistic for each element $\mathcal{G}_i$ be $\tilde{f}_{\mathcal{G}_i}$ and the noisy vector $\hat{y}$ in the compressive mechanism be $\tilde{h}_{\mathcal{G}_i}$.

Among the K output data, we assume that p data are chosen from the $\tilde{s}$ data which is subjected to the compressive mechanism, which we denote as $\mathcal{G}_1, \mathcal{G}_2, \ldots, \mathcal{G}_p$, and the other data as $\mathcal{G}_{p+1}, \ldots, \mathcal{G}_K$.

Then we let $v_1, v_2, \ldots, v_K \in \mathbb{R}$ be the noisy statistic of $\mathcal{G}_1, \mathcal{G}_2, \ldots, \mathcal{G}_K$, and let $z_1, z_2, \ldots, z_p \in \mathbb{R}^k$ be the vectors for obtaining $v_1, v_2, \ldots, v_p$ by the reconstruction process.

Under the above conditions, the following equation holds:

$$\begin{aligned}
\Pr[\mathcal{A}(g) = W] &= \int_{v_1}\int_{v_2}\cdots\int_{v_K} \Pr[\tilde{f}_{\mathcal{G}_1} = v_1]\cdot\Pr[\tilde{f}_{\mathcal{G}_2} = v_2]\cdots\Pr[\tilde{f}_{\mathcal{G}_K} = v_K] \\
&\quad \cdot \prod_{\mathcal{G}\in U-W} \Pr[\tilde{f}_{\mathcal{G}} < \min\{v_1, v_2, \cdots, v_K\}]\ \mathrm{d}v_1\mathrm{d}v_2\cdots\mathrm{d}v_K \\
&= \int_{z_1}\int_{z_2}\cdots\int_{z_p}\int_{v_{p+1}}\cdots\int_{v_K} \\
&\quad \Pr[\tilde{h}_{\mathcal{G}_1} = z_1]\cdot\Pr[\tilde{h}_{\mathcal{G}_2} = z_2]\cdots\Pr[\tilde{h}_{\mathcal{G}_p} = z_p]\cdot\Pr[\tilde{f}_{\mathcal{G}_{p+1}} = v_{p+1}]\cdots\Pr[\tilde{f}_{\mathcal{G}_K} = v_K] \\
&\quad \cdot \prod_{\mathcal{G}\in U-W} \Pr[\tilde{f}_{\mathcal{G}} < \min\{v_1, v_2, \cdots, v_K\}]\ \mathrm{d}z_1\mathrm{d}z_2\cdots\mathrm{d}z_p\mathrm{d}v_{p+1}\cdots\mathrm{d}v_K.
\end{aligned}$$

Let the sensitivity of $\tilde{f}_{\mathcal{G}}$ and $\tilde{h}_{\mathcal{G}}$ be Δ_f and Δ_h, respectively. In addition, let the scale of the Laplace noise added by the Laplace mechanism and compressive mechanism be λ_f and λ_h, respectively. Then, in the same way as in the discussion of Bhasker *et al.*'s Theorem 4 [5],

$$\log\frac{\Pr[\mathcal{A}(g) = W]}{\Pr[\mathcal{A}(g') = W]} \leq \frac{2p\Delta_h}{\lambda_h} + \frac{2(K-p)\Delta_f}{\lambda_f}. \tag{1}$$

. Thus, if we set $\lambda_f = \frac{2K\Delta_f}{\epsilon}$ and $\lambda_h = \frac{2K\Delta_h}{\epsilon}$, then (1) is equal to ϵ regardless of the value of p.

From the above and Theorem 1, Algorithm 3 satisfies ϵ-differential privacy. □

4 Experiments and Discussion

We evaluated the usefulness of Algorithms 1 and 3 in terms of accuracy, rank error of the output, and run time. For the compressed sensing, we used Orthogonal Matching Pursuit [28]. In addition to our algorithms, we also experimented with the Laplace mechanism and the exponential mechanism using χ^2-statistics as the score function and discuss the advantages of our methods. The Laplace mechanism and the exponential mechanism are shown in Algorithm 4 in Appendix A.1 and Algorithm 5 in Appendix A.2, respectively.

For accuracy, we varied the values of the privacy budget ϵ and the number of outputs K and calculated the accuracy rate of the top K data released upon using each mechanism. The value of ϵ was set from 0.01 to 1, because genomic data requires a high privacy guarantee. The value of K was set to values up to 10, because the goal of this study is to extract only significant SNPs. For the rank error, we evaluated the desirability of the output by measuring the error of the output data compared to their actual rank. Finally, we measured the run time of each mechanism by varying the input size. Note that in these experiments, the value of η in Algorithm 2 was set to 10 and the elements below 10 were dropped to 0, so that the information in the statistics vector was approximately represented as a sparse vector. In the future, we should develop a method to optimally determine the value of η under differential privacy, for example, by using the exponential mechanism.

In our experiments, we used simulation data for χ^2-statistics in a χ^2-test using a 3×2 contingency table. Assuming that the number of cases and controls is equal, we can use our algorithms because the *sensitivity* of the statistic is known to be $\frac{4N}{N+2}$, where N is the number of individuals [17].

4.1 Simulation Data

We generated data using two simulations, i.e, a small cohort with 100 individuals and 500 statistics, and a large cohort with 500 individuals and $5,000$ statistics. Here, the χ^2-statistic using a 3×2 contingency table is expressed using the values in Table 1 by the following equation: $\frac{(2a-m)^2}{m} + \frac{(2b-n)^2}{n} + \frac{(2a-m+2b-n)^2}{N-m-n}$.

Let $N = 100$ for a small cohort and $N = 500$ for a large cohort, and we set m, n, a, and b using a binomial distribution with the following equations: $m = Bin\left(N, \frac{1}{3}\right)$, $n = Bin\left(N-m, \frac{1}{2}\right)$, $a = Bin\left(m, \frac{1}{2}\right)$, $b = Bin\left(n, \frac{1}{2}\right)$. In addition, in order to generate 10 significant SNPs, the probabilities in the binomial distribution for calculating a and b were set as $\frac{3}{4}$ for a small cohort and $\frac{2}{3}$ for a large cohort.

Table 1: An example of 3×2 contingency table for χ^2-test.

		Disease Status		Total
		0	1	
Genotype	0	a	$m-a$	m
	1	b	$n-b$	n
	2	$N/2-a-b$	$N/2-m-n+a+b$	$N-m-n$
Total		$N/2$	$N/2$	N

4.2 Results

4.2.1 Accuracy

We varied the value of ϵ (from 0.01 to 1), and the value of K (1, 3, 5, and 10) and calculated the accuracy for the top K output using each mechanism. The results of the simulation data for a small cohort and a large cohort described in Subsection 4.1 are plotted in Figures 1 and 2.

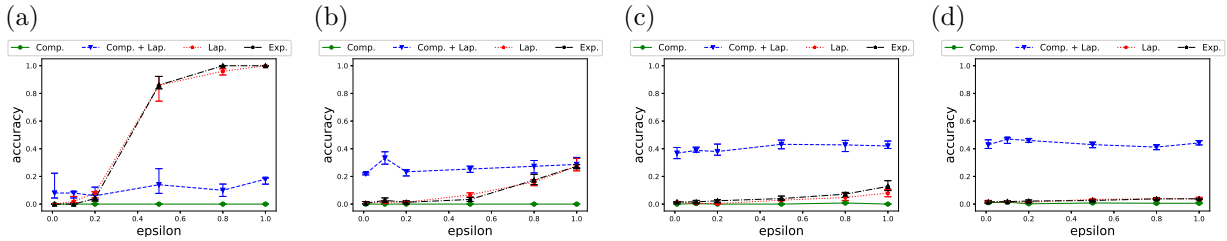

Figure 1: Accuracy of the K outputs when (a) $K = 1$, (b) $K = 3$, (c) $K = 5$, and (d) $K = 10$ in a small cohort. The green, blue, red, and black lines represent Comp., Comp. + Lap., Lap., and Exp., respectively.

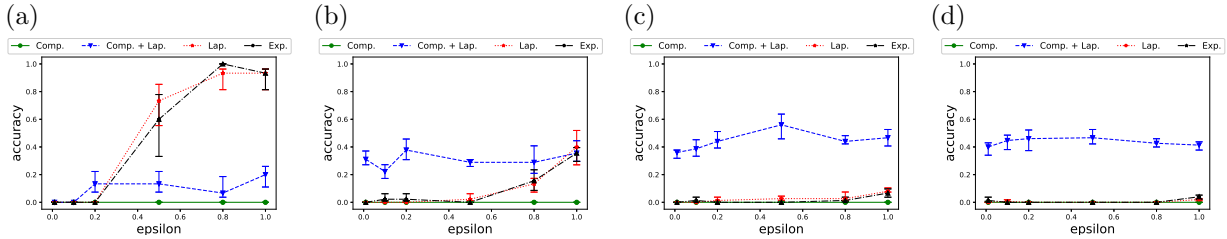

Figure 2: Accuracy of the K outputs when (a) $K = 1$, (b) $K = 3$, (c) $K = 5$, and (d) $K = 10$ in a large cohort. The green, blue, red, and black lines represent Comp., Comp. + Lap., Lap., and Exp., respectively.

Figure 1 shows that when $K = 1$, the Laplace and exponential mechanisms provide higher accuracy than our methods. However, the accuracy of these mechanisms decreases as K increases. The reason for this is that more noise we have to add to each statistic, the less we can distinguish between significant and non-significant SNPs. On the other hand, when $K = 3, 5$, and 10, our methods, which is a combination of the compressive and Laplace mechanisms, outperforms the other methods. More surprisingly, as K increases, higher accuracy is achieved, and when $K = 10$, approximately half of the output is correct. Even in the case of a large cohort, the outline in Figure 2 is like that in Figure 1, confirming that our mechanism achieved a higher accuracy than the others. However, our simple compressive mechanism did not provide an accurate output in any of the cases.

4.2.2 Rank Error

We also evaluated the desirability of the output by calculating the rank error. The rank error represents the deviation from the actual rank and is expressed as $RE = \frac{1}{K}\sum_{i=1}^{K}|R_i - i|$, where R_i is the true rank of the data output as the i-th. Even if the output does not exactly match the true data, when the rank

error is small, it can be expected that the relatively high-rank data will be the output, which is desirable for practical use. The experimental results are shown in Figures 3 and 4, respectively.

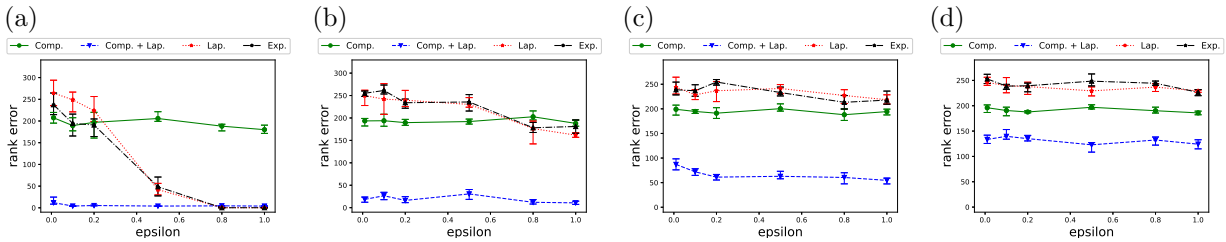

Figure 3: Rank error of the K outputs when (a) $K = 1$, (b) $K = 3$, (c) $K = 5$, and (d) $K = 10$ in a small cohort. The green, blue, red, and black lines represent Comp., Comp. + Lap., Lap., and Exp., respectively.

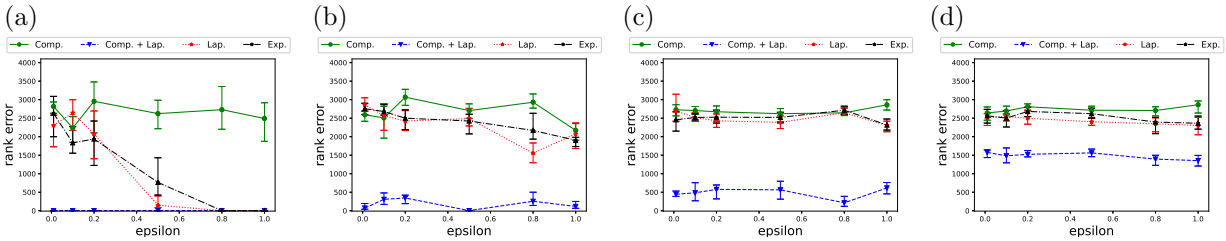

Figure 4: Rank error of the K outputs when (a) $K = 1$, (b) $K = 3$, (c) $K = 5$, and (d) $K = 10$ in a large cohort. The green, blue, red, and black lines represent Comp., Comp. + Lap., Lap., and Exp., respectively.

Figure 3 indicates that for a small cohort, our methods have much smaller output errors than existing mechanisms. For the Laplace and exponential mechanisms, as in Section 4.2.1, the utility decreased as ϵ decreased and as K increased. Our mechanisms did not significantly alter the rank error much, depending on the value ϵ. In particular, our method that combines the compressive and Laplace mechanisms performed better than the other methods in all cases. Even in the case of $K = 1$, where the accuracy was not very high, the rank error was quite small, and we can expect a significant SNP output. In addition, the algorithm that involved only the compressive mechanism resulted in smaller rank errors than those that involved the Laplace and exponential mechanisms. However, in the case of a large cohort shown in Figure 4, a small rank error was not achieved upon using the compressive mechanism alone, instead a very high utility could be achieved upon using the combination with the Laplace mechanism.

In Appendix A.3, we show the results of additional experiments on accuracy and rank error for a larger cohort containing $25,000$ SNPs. The results also indicate that our method outperforms the other mechanisms.

4.2.3 Run Time

We varied the data size of the statistics data g, and measured the run time of each mechanism. Here, note that the parameters that affect the run time are only the dataset size $|g|$, i.e., the number of SNPs, and the number of outputs K. Thus, it does not depend on the number of individuals in the dataset or the value of ϵ. We conducted five runs for each case, and the average values are listed in Table 2.

Table 2: Run time [s] of each mechanism for each input size.

$\|g\|$	500	$1,000$	$2,500$	$5,000$	$10,000$	$25,000$
Comp.	6.52	28.37	567.2	9.2×10^3	—	—
Comp. + Lap.	2.96	10.03	136.6	588.2	2.1×10^3	7.9×10^3
Lap.	2.9×10^{-4}	3.9×10^{-4}	6.1×10^{-4}	1.1×10^{-3}	1.9×10^{-3}	5.6×10^{-3}
Exp.	1.6×10^{-3}	3.4×10^{-3}	8.7×10^{-3}	2.0×10^{-2}	3.2×10^{-2}	7.8×10^{-2}

With respect to the run time, our algorithms are inferior to the Laplace mechanism and the exponential mechanism owing to matrix multiplication and compressed sensing. Moreover, the run time of our method

using only the compressive mechanism highly depends on the method to determine the number of nonzero entries. In this experiment, we considered elements larger than the threshold η in Algorithm 2 as nonzero entries, and the sparsity decreases with the increase in the data size. Consequently, the run time becomes longer. In contrast, the method that involved combining the compressive mechanism with the Laplace mechanism, which performed well in terms of accuracy and rank error, required only approximately 10 minutes to process $5,000$ data. However, when the data size was much larger than that shown in Tabke 2, e.g. when $|g| = 100,000$, it took about two days. In the future, we will need to employ faster calculation methods in compressed sensing and more effective ways to determine the number of nonzero entries.

4.3 Real Data

We also evaluated the proposed mechanism using real data. The data we used were TDT statistics by Justice *et al.* [24]. Here, the *sensitivity* of TDT statistics for N families is $\frac{8(N-1)}{N}$ [32]. From their data, we took $2,000$ statistics and subjected them to our method of combining the compressive and Laplace mechanisms, Laplace mechanism, and the exponential mechanism to them. We also evaluated the exponential mechanism using the SHD score [32], which is applicable to datasets of the TDT statistics. Since there are six significant SNPs based on their statistics, we considered the cases where $K = 1, 2$, and 5. For each K, we calculated the probability of releasing those significant SNPs. The values of ϵ were set to $1, 2$, and 5, which are larger than those used in the simulation studies, in order to also check the variation in the utility when using the Laplace and exponential mechanisms. The results are shown in Table 3.

Table 3: Probability of extracting significant SNPs.

ϵ		1			2			5		
K		1	2	5	1	2	5	1	2	5
Significance Rate (%)	Comp. + Lap.	40.8	40.4	40.5	45.6	47.6	41.9	56.0	44.8	43.2
	Lap.	4.00	0.80	0.32	12.8	3.60	0.32	82.4	28.0	1.76
	Exp.	3.20	0.80	0.16	9.60	2.00	0.64	76.8	18.4	0.96
	Exp. (SHD score) [32]	27.2	7.60	1.28	49.6	16.4	5.44	93.6	69.2	25.3

The results indicate that our method is better than the Laplace and exponential mechanisms even for real data. We also found that our method can release significant SNPs with high probability even when ϵ becomes smaller and the number of outputs K increases. However, as with the results in Section 4.2, when $K = 1$, the Laplace and exponential mechanisms provide relatively high utility because the scale of adding noise is small compared to the difference among the original statistics. The exponential mechanism using the SHD score achieved relatively high accuracy, but was not as accurate as out method when the privacy level was high or when the number of outputs was large.

5 Conclusion

In this study, we proposed privacy-preserving methods using the compressive mechanism to release the top K significant SNPs based on a genomic dataset. Our first algorithm, which involves only the simple conmpressive mechanism, achieved smaller rank errors than existing algorithms involving the Laplace and exponential mechanisms in a small cohort. In our second method, we considered genomic data as a sparse vector and combined the compressive and Laplace mechanisms. This method outperformed the other methods in terms of accuracy as well as rank error by distinguishing significant SNPs from non-significant SNPs. We also theoretically proved that this method satisfies ϵ-differential privacy. In addition, we also obtained high performance in our experiments using real data, indicating that our method can ensure both high utility and privacy assurance.

One limitation of this study is that we could not achieve high accuracy using the compressive mechanism alone. We need to further improve the method by, for example, considering the position of nonzero elements in a sparse representation. In addition, for much larger datasets containing millions of SNPs, we would like to investigate additional methods to conduct calculation without using matrices and faster algorithms using the Fourier transform or other techniques. For further work, we need to consider wavelet transforms other than the Haar matrix and develop new score functions for the exponential mechanism.

Acknowledgements
This work was supported by JSPS KAKENHI Grant 20H05967, 20K21827, and 21H05052. The supercomputing resource was provided by Human Genome Center, the Institute of Medical Science, the University of Tokyo.

References

[1] Abadi, M., Chu, A., Goodfellow, I., McMahan, H. B., Mironov, I., Talwar, K., and Zhang, L. Deep learning with differential privacy. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, CCS '16, page 308–318, New York, NY, USA, 2016.

[2] Acs, G., Castelluccia, C., and Chen, R. Differentially private histogram publishing through lossy compression. In *2012 IEEE 12th International Conference on Data Mining*, pages 1–10, 2012.

[3] Alnemari, A., Raj, R. K., Romanowski, C. J., and Mishra, S. Interactive range queries for healthcare data under differential privacy. In *2021 IEEE 9th International Conference on Healthcare Informatics (ICHI)*, pages 228–237, 2021.

[4] Baraniuk, R., Davenport, M., DeVore, R., and Wakin, M. A simple proof of the restricted isometry property for random matrices. *Constructive Approximation*, 28:253–263, 2008.

[5] Bhaskar, R., Laxman, S., Smith, A., and Thakurta, A. Discovering frequent patterns in sensitive data. In *KDD'10*, pages 503–512, Washington, DC, USA, July 2010.

[6] Blatt, M., Gusev, A., Polyakov, Y., and Goldwasser, S. Secure large-scale genome-wide association studies using homomorphic encryption. *PNAS*, 117(21):11608–11613, 2020.

[7] Bonte, C., Makri, E., Ardeshirdavani, A., Simm, J., Moreau, Y., and Vercauteren, F. Towards practical privacy-preserving genome-wide association study. *BMC Bioinformatics*, 19:537, 2018.

[8] Candès, E. J., Romberg, J., and Tao, T. Robust uncertainty principles: exact signal reconstruction from highly incomplete frequency information. *IEEE Transactions on Information Theory*, 52(2):489–509, 2006.

[9] Candès, E. J. and Tao, T. Near-optimal signal recovery from random projections: Universal encoding strategies? *IEEE Transactions on Information Theory*, 52(12):5406–5425, 2006.

[10] Chang, Y. H., Gray, J. W., and Tomlin, C. J. Exact reconstruction of gene regulatory networks using compressive sensing. *BMC Bioinformatics*, 15(1):400, 2014.

[11] Cho, H., Wu, D. J., and Berger, B. Secure genome-wide association analysis using multiparty computation. *Nat. Biotechnol.*, 36:547–551, 2018.

[12] Cortés, J., Dullerud, G. E., Han, S., Le Ny, J., Mitra, S., and Pappas, G. J. Differential privacy in control and network systems. In *2016 IEEE 55th Conference on Decision and Control (CDC)*, pages 4252–4272, 2016.

[13] Dai, W. and Milenkovic, O. Subspace pursuit for compressive sensing signal reconstruction. *IEEE Transactions on Information Theory*, 55(5):2230–2249, 2009.

[14] Donoho, D. L. Compressed sensing. *IEEE Transactions on Information Theory*, 52(4):1289–1306, 2006.

[15] Dwork, C. Differential privacy. *Michele Bugliesi, Bart Preneel, Vladimiro Sassone, and Ingo Wegener, (eds) Automata, Languages and Programming*, 4052, 2006.

[16] Dwork, C., McSherry, F., Nissim, K., and Smith, A. Calibrating noise to sensitivity in private data analysis. *S. Halevi and T. Rabin, (eds) Theory of Cryptography*, 3876:265–284, 2006.

[17] Fienberg, S. E., Slavkovic, A., and Uhler, C. Privacy preserving GWAS data sharing. In *IEEE 11th International Conference on Data Mining Workshops*, pages 628–635, Vancouver, Canada, December 2011.

[18] Haar, A. Zur theorie der orthogonalen funktionensysteme. *Math. Ann.*, 69:331–371, 1910.

[19] Hsu, J., Gaboardi, M., Haeberlen, A., Khanna, S., Narayan, A., Pierce, B. C., and Roth, A. Differential privacy: An economic method for choosing epsilon. In *2014 IEEE 27th Computer Security Foundations Symposium*, pages 398–410, Vienna, Austria, July 2014.

[20] International Human Genome Sequencing Consortium. Finishing the euchromatic sequence of the human genome. *Nature*, 431:931–945, 2004.

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

[22] Jälkö, J., Lagerspetz, E., Haukka, J., Tarkoma, S., Honkela, A., and Kaski, S. Privacy-preserving data sharing via probabilistic modeling. *Patterns (N Y)*, 2(7):100271, 2021.

[23] Johnson, A. and Shmatikov, V. Privacy-preserving data exploration in genome-wide association studies. In *KDD'13*, pages 1079–1087, Chicago, Illinois, USA, August 2013.

[24] Justice, C. M., Cuellar, A., Bala, K., Sabourin, J. A., Cunningham, M. L., Crawford, K., Phipps, J. M., Zhou, Y., Cilliers, D., Byren, J. C., Johnson, D., Wall, S. A., Morton, J. E. V., Noons, P., Sweeney, E., Weber, A., Rees, K. E. M., Wilson, L. C., Simeonov, E., Kaneva, R., Yaneva, N., Georgiev, K., Bussarsky, A., Senders, C., Zwienenberg, M., Boggan, J., Roscioli, T., Tamburrini, G., Barba, M., Conway, K., Sheffield, V. C., Brody, L., Mills, J. L., Kay, D., Sicko, R. J., Langlois, P. H., Tittle, R. K., Botto, L. D., Jenkins, M. M., LaSalle, J. M., Lattanzi, W., Wilkie, A. O. M., Wilson, A. F., Romitti, P. A., and Boyadjiev, S. A. A genome-wide association study implicates the BMP7 locus as a risk factor for nonsyndromic metopic craniosynostosis. *Hum. Genet.*, 139(8):1077–1090, 2020.

[25] Li, Y. D., Zhang, Z., Winslett, M., and Yang, Y. Compressive mechanism: Utilizing sparse representation in differential privacy. In *Proceedings of the 10th Annual ACM Workshop on Privacy in the Electronic Society*, page 177–182, New York, NY, USA, 2011.

[26] Lionnie, R. and Alaydrus, M. An analysis of haar wavelet transformation for androgenic hair pattern recognition. In *2016 International Conference on Informatics and Computing (ICIC)*, pages 22–26, 2016.

[27] McSherry, F. and Talwar, K. Mechanism design via differential privacy. In *48th Annual IEEE Symposium on Foundations of Computer Science*, pages 94–103, Providence, RI, USA, October 2007.

[28] Pati, Y. C., Rezaiifar, R., and Krishnaprasad, P. S. Orthogonal matching pursuit: recursive function approximation with applications to wavelet decomposition. In *Proceedings of 27th Asilomar Conference on Signals, Systems and Computers*, volume 1, pages 40–44, 1993.

[29] Porwik, P. and Lisowska, A. The Haar wavelet transform in digital image processing: its status and achievements. *Machine GRAPHICS & VISION*, 13(1/2):79–98, 2004.

[30] Sankararaman, S., Obozinski, G., Jordan, M. I., and Halperin, E. Genomic privacy and limits of individual detection in a pool. *Nat. Genet.*, 41:965–967, 2009.

[31] Simmons, S., Berger, B., and Sahinalp, C. S. Protecting genomic data privacy with probabilistic modeling. In *Biocomputing 2019: Proceedings of the Pacific Symposium, The Big Island of Hawaii, Hawaii, USA, January 3-7, 2019*, pages 403–414, 2019.

[32] Wang, M., Ji, Z., Wang, S., Kim, J., Yang, H., Jiang, X., and Ohno-Machado, L. Mechanisms to protect the privacy of families when using the transmission disequilibrium test in genome-wide association studies. *Bioinformatics*, 33(23):3716–3725, 2017.

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

[34] Wjst, M. Caught you: threats to confidentiality due to the public release of large-scale genetic data sets. *BMC Med. Ethics.*, 11(21), 2010.

[35] Xiao, X., Wang, G., and Gehrke, J. Differential privacy via wavelet transforms. *IEEE Transactions on Knowledge and Data Engineering*, 23:1200–1214, 2011.

[36] Yamamoto, A. and Shibuya, T. More practical differentially private publication of key statistics in GWAS. *Bioinformatics Advances*, 1(1):vbab004, May 2021.

[37] Yamamoto, A. and Shibuya, T. Differentially private linkage analysis with TDT — the case of two affected children per family. In *2021 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*, pages 765–770, 2021.

[38] Yu, F. and Ji, Z. Scalable privacy-preserving data sharing methodology for genome-wide association studies: an application to iDASH healthcare privacy protection challenge. *BMC Med. Inform. Decis. Mak.*, 14(S3), 2014.

[39] Yuan, X. and Haimi-Cohen, R. Image compression based on compressive sensing: End-to-end comparison with jpeg. *IEEE Transactions on Multimedia*, 22(11):2889–2904, 2020.

[40] Zerhouni, E. A. and Nabel, E. G. Protecting aggregate genomic data. *Science*, 322(5898):44, 2008.

[41] Zhang, Y., Blanton, M., and Almashaqbeh, G. Secure distributed genome analysis for GWAS and sequence comparison computation. *BMC Med. Inform. Decis. Mak.*, 15:Suppl 5(Suppl 5):S4, 2015.

A Appendix

A.1 Laplace Mechanism

An algorithm for releasing the top K significant SNPs based on genomic statistics data using the Laplace mechanism is shown in Algorithm 4.

Algorithm 4 ϵ-differentially private algorithm for releasing the top K significant SNPs using the Laplace mechanism.

Input: Genomic statistics data $g \in \mathbb{R}^m$, the *sensitivity* of the statistic Δ_g, number of data to release K, and privacy budget ϵ.

Output: Top K significant SNPs.

1: Add Laplace noise with mean 0 and scale $\frac{2K\Delta_g}{\epsilon}$ to each g_i, and get a noisy vector $\hat{g}$.
2: Choose the top K SNPs from $\hat{g}$.

A.2 Exponential Mechanism

An algorithm for releasing the top K significant SNPs based on genomic statistics data using the exponential mechanism is shown in Algorithm 5.

Algorithm 5 ϵ-differentially private algorithm for releasing the top K significant SNPs using the exponential mechanism.

Input: Genomic statistics data $g \in \mathbb{R}^m$, the *sensitivity* of the statistic Δ_g, number of data to release K, and privacy budget ϵ.

Output: Top K significant SNPs.

1: Let $S = \emptyset$.
2: For each $i \in \{1, \ldots, m\}$, set the weight $w_i = \exp\left(\frac{\epsilon g_i}{2K\Delta_g}\right)$ and the probability $p_i = \frac{w_i}{\sum_{i=1}^{m} w_i}$ for sampling the i-th SNP.
3: Sample k from $\{1, \ldots, m\}$ with probabilities $\{p_1, \ldots, p_m\}$; add k-th data to S and set $g_k = -\infty$.
4: Repeat steps 2 and 3 until the size of S reaches K.

A.3 Additional Experiments

We also conducted experiments on accuracy and rank error for the case where the number of SNPs in the simulation dataset was $25,000$. The results are shown in Figures 5 and 6. When $K = 1$, the existing Laplace and exponential mechanisms achieved high accuracy because the noise is small and the rank of the original statistics varied little. However, when K was increased or when ϵ was small, i.e., higher privacy was achieved, our combining method outperformed the other methods.

A.3.1 Accuracy

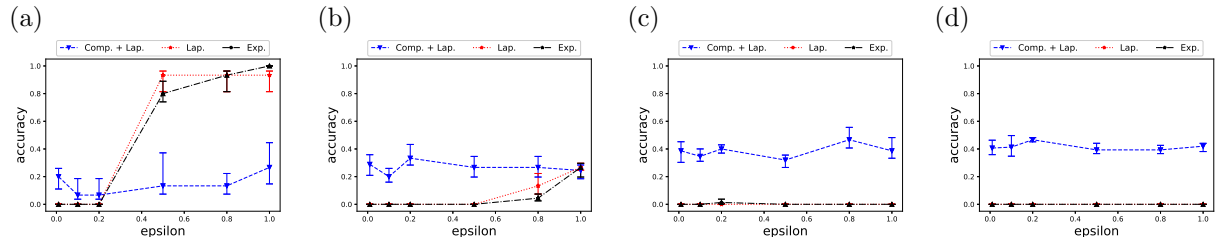

Figure 5: Accuracy of the K outputs when (a) $K = 1$, (b) $K = 3$, (c) $K = 5$, and (d) $K = 10$ in a large cohort with $25,000$ SNPs. The blue, red, and black lines represent Comp. + Lap., Lap., and Exp., respectively.

A.3.2 Rank Error

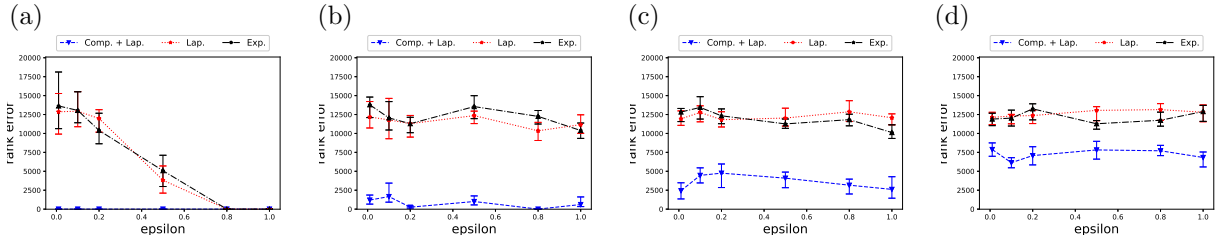

Figure 6: Rank error of the K outputs when (a) $K = 1$, (b) $K = 3$, (c) $K = 5$, and (d) $K = 10$ in a large cohort with $25,000$ SNPs. The blue, red, and black lines represent Comp. + Lap., Lap., and Exp., respectively.