Achieving Differential Privacy of Genomic Data Releasing via Belief Propagation

Zaobo He, Yingshu Li*, Ji Li, Kaiyang Li, Qing Cai, and Yi Liang

Abstract: Privacy preserving data releasing is an important problem for reconciling data openness with individual privacy. The state-of-the-art approach for privacy preserving data release is differential privacy, which offers powerful privacy guarantee without confining assumptions about the background knowledge about attackers. For genomic data with huge-dimensional attributes, however, current approaches based on differential privacy are not effective to handle. Specifically, amount of noise is required to be injected to genomic data with tens of million of SNPs (Single Nucleotide Polymorphisms), which would significantly degrade the utility of released data. To address this problem, this paper proposes a differential privacy guaranteed genomic data releasing method. Through executing belief propagation on factor graph, our method can factorize the distribution of sensitive genomic data into a set of local distributions. After injecting differential-privacy noise to these local distributions, synthetic sensitive data can be obtained by sampling on noise distribution. Synthetic sensitive data and factor graph can be further used to construct approximate distribution of non-sensitive data. Finally, non-sensitive genomic data is sampled from the approximate distribution to construct a synthetic genomic dataset.

Key words: differential privacy; SNP/trait associations; belief propagation; factor graph; data releasing

1 Introduction

With the developing of DNA-genotyping technology, more and more individuals tend to genotype their DNA, in order for genetic services. For example, 23andMe[1], one of the most popular DNA-sequencing service providers, has provided such services for more than 900000 individuals. With genotyped DNA, individuals can learn about their predispositions to disease of genetic tendency. Meanwhile, massive DNA sequences are significantly beneficial to researchers to develop new genetic diagnostic methods or medicines. Furthermore, more and more research groups release the uncovered associations among genotypes, haplotypes, or phenotypes (such as GWAS catalog[2], DisGeNET[3]), which further enrich genetic services and research works.

However, individual privacy is increasingly threatened with more and more genomic data being available online, although significant benefit is brought by them. Combined with publicly available statistics and kin genomic information, individual genomes could be inferred based on machine learning, data mining, or any statistical analysis techniques. On the one hand, the uncovered dependency relationship over genomes by association studies facilitates attackers to launch inference attacks. For example, GWAS catalog reported the trait/SNP (Single Nucleotide Polymorphism) associations, which could undoubtedly help attackers learn about target
SNPs or traits of victims depending on publicly available ones. On the other hand, individual genomes are closely related with their relatives. There are 99.9% similar in DNA sequence between two random individuals and the similar degree is higher between relatives. Therefore, once an individual’s genomic privacy is breached, all its relatives are unavoidably placed into risk. Both SNPs and traits are deeply private, the leak of them may incur serious discrimination. Therefore, how to realize the tradeoff between data openness and genomic privacy becomes a crucial problem.

In this paper, we propose an effective method to address the problem of releasing differentially private kin-genomic data. Implementing differential privacy in genomic data remains a challenging problem. Large scale of noise must be injected to high-dimensional data in order for satisfying differential privacy, which definitely degrade data utility. To address it, we first explore the possibilities for factorizing the joint conditional distribution of sensitive genomes into sets of local probability distributions, by executing belief propagation on factor graph which captures the dependency relationship among family members, SNPs, and traits due to family genetic relationship and SNP/trait associations. To guarantee differential privacy, noise is injected to these local probability distributions to construct approximate distribution for sensitive genomes and then synthetic sensitive genomes can be sampled. Then, synthetic sensitive genomes and factor graph are further used to construct approximate distribution of non-sensitive genomes. Finally, synthetic nonsensitive genomes can be sampled and released. Compared with large body of previous works, which mainly focused on improving the output of differential privacy mechanism (such as optimizing specific query results), we study how to factorize a huge-dimensional distribution into a set of local distributions, so that scale of noise can be reduced by injecting into local distributions.

2 Preliminaries

2.1 Belief propagation on factor graph
Belief propagation is an algorithm to perform approximate inference on probabilistic graph models through passing messages, including Bayesian networks, factor graph, and Markov random fields. It is generally used to calculate marginal probability distribution for target unobserved variables, conditional on observed ones. Belief propagation is generally described by the operations on factor graphs (any Bayesian network and Markov random field can be transformed to factor graph). Factor graph is one type of undirected graphical models, which contains two types of nodes: variable node and factor node. There is an edge between factor node and variable node iff this variable node is an argument of the factor node. In belief propagation, each variable node (factor node) sends message to its neighbor factor nodes (variable nodes). The propagated message is the probability (belief) of one variable nodes being a value (such as 1 (0) represents the presence (absence) of the trait). Given certain initial states and boundary conditions, belief propagation is to propagate such message iteratively until the unobserved variables converge to boundary conditions.

2.2 Differential privacy
For a sensitive dataset $D$ to be released, to guarantee $D$ satisfies differential privacy, prior to $D$’s release, randomized function $F$ is required to add noise to $D$ so that changing (adding, removing or replacing) an arbitrary entry in $D$ does not significantly change the output of $F$. Differential privacy is formally introduced as follows.

**Definition 1** $\epsilon$-Differential privacy. A randomized function $F$ satisfies $\epsilon$-Differential privacy if for any two neighbor datasets $D$ and $D'$ (differing at most one entry), for any output of $F$, the following rule is held:

$$Pr[F(D) = O] \leq e^\epsilon Pr[F(D') = O].$$

Informally, differential privacy indicates that from the output of $F$, anyone cannot significantly distinguish an arbitrary entry in $D$.

3 Problem Formulation

3.1 Genomic data model
Suppose the set of individuals in the target family is denoted as $F$. The SNP set of an individual is denoted as $S$. The content of SNP $j$ for individual $i$ is denoted as $s^j_i$, where $s^j_i$ takes value from: (1) BB (both alleles inherited from parents are major alleles), (2) Bb (alleles inherited from parents are major allele and minor allele), or (3) bb (both alleles inherited from parents are minor alleles). We assume some individuals in a target family intend to release their part of SNPs or traits (such as diseases, hair color, height, etc.) in order for genetic services or research purpose. However, privacy concerns drive them to just release part of SNPs or non-sensitive traits whereas sensitive part is kept private. We denote the set of non-sensitive variables (including SNPs and traits) as $X_K$, while sensitive variables as $X_U$. 

For SNP/trait associations reported by GWAS catalog, the trait set considered are denoted as $T$. $t_l^i$ is defined to be the value of trait $l$ ($l \in T$) of individual $i$, where each $t_l^i$ is specified by a 0/1 value and each of which indicates the absence/presence of the trait $l$ for individual $i$.

Mendelian inheritance. From Mendel’s First Law, each allele of one individual is inherited from his father or his mother with same probability of 1/2. Let $\mathcal{F}(s_j^F, s_j^M, s_j^U)$ denote the Mendelian inheritance for SNP $j$, in a family (Father, Mother, and Child). The Mendelian inheritance for SNP $j$ in a family is introduced in Table 1.

### 3.2 Problem definition

The studied problem in this paper can be formulated as follows:

**Input:**
1. individual non-sensitive SNPs and traits $X_K$;
2. SNP/trait associations reported by GWAS catalog, $\mathcal{A}$;
3. Mendelian inheritance for target family $\mathcal{F}(s_j^F, s_j^M, s_j^U)$.

**Output:**
$\epsilon$-differential privacy preserving genomic data $X$, $X = X_U \cup X_K$ releasing method.

### 4 Solution

#### 4.1 Solution overview

This section sketches an overview of our method for releasing genomic data with $\epsilon$-differential privacy guarantee. The proposed method runs in five phases:

- **Phase 1:** Calculate the joint conditional distribution of sensitive variables $p(X_U|X_K, A, \mathcal{F})$.
- **Phase 2:** Construct $\epsilon$-differential privacy algorithm to generate the noise version of $p(X_U|X_K, A, \mathcal{F})$ constructed in Phase 1. We denote the noise joint conditional distribution of sensitive variables as $p^\ast(X_U|X_K, A, \mathcal{F})$.
- **Phase 3:** Sample sensitive variables from the noise joint conditional distribution $p^\ast(X_U|X_K, A, \mathcal{F})$ generated in Phase 2 to generate synthetic sensitive variables $X_U^\ast$.
- **Phase 4:** Calculate the joint conditional distribution of non-sensitive variables $p(X_K|X_U^\ast, A, \mathcal{F})$ based on synthetic sensitive variables $X_U^\ast$ generated in Phase 3.
- **Phase 5:** Sample non-sensitive variables from the joint conditional distribution $p(X_K|X_U^\ast, A, \mathcal{F})$ calculated in Phase 4 to generate synthetic sensitive variables $X_K^\ast$.

Finally, the target individual releases the synthetic genomic data $X^\ast$, $X^\ast = X_K^\ast \cup X_U^\ast$. In short, our method is to use synthetic SNPs and traits $X^\ast$ of individuals to approximate the real SNPs and traits $X$. Relatively, sampling operations (Phase 3 and Phase 5) are straightforward. However, calculating the joint conditional distribution (Phase 1 and Phase 4) and constructing $\epsilon$-differential privacy algorithm in Phase 2 are non-trivial. In the following subsections, we detail these phases and prove our method satisfies $\epsilon$-differential privacy.

#### 4.2 Generation of joint conditional distribution

The computation complexity grows exponentially with the number of variables so that it is infeasible to calculate joint conditional distribution in both Phase 1 and Phase 4 directly considering human’s genomes generally contain tens of million of SNPs. Therefore, we expect the joint conditional distribution can be factorized into the product of several local functions, and each one captures the dependency relationship among variables (because of trait/SNP association $\mathcal{A}$, and Mendelian inheritance $\mathcal{F}$), by supporting a subset of variables. Through running belief propagation on factor graph, the computation complexity can be improved from exponential to linear complexity.

For this, we develop a factor graph to capture the dependency relationship among variables. A factor graph uses Variable Node and Factor Node to represent variables and dependency relationship among them, respectively. In our factor graph, four types of nodes are considered:

- **SNP variable node:** taking each SNP as a variable node.
- **trait variable node:** taking each trait as a variable node.
- **familial node:** to represent variables associated with trait/SNP association $\mathcal{A}$, and Mendelian inheritance $\mathcal{F}$, by supporting a subset of variables. Through running belief propagation on factor graph, the computation complexity can be improved from exponential to linear complexity.

For example, a factor graph with three trait variable nodes and four SNP variable nodes is shown in Fig. 1. As shown in Fig. 1, for traits 1, 2, and 3, the associated SNPs are $\{1, 2\}$, $\{2, 3\}$, and $\{4\}$, respectively.

<table>
<thead>
<tr>
<th>Father</th>
<th>Mother</th>
</tr>
</thead>
<tbody>
<tr>
<td>BB</td>
<td>BB</td>
</tr>
<tr>
<td>(1, 0, 0)</td>
<td>(1/2, 1/2, 0)</td>
</tr>
<tr>
<td>Bb</td>
<td>Bb</td>
</tr>
<tr>
<td>(1/2, 1/2, 0)</td>
<td>(1/4, 1/2, 1/4)</td>
</tr>
<tr>
<td>bb</td>
<td>BB</td>
</tr>
</tbody>
</table>
To obtain the joint conditional distribution of an arbitrary set of variables $X'$, there are three cases to be considered:

- **Case 1**: $X'$ is the set of variables involved in a factor node, i.e., the entire neighbors of this factor node.
- **Case 2**: $X'$ is only the subset of variables involved in a factor node.
- **Case 3**: $X'$ is the set of variables involved in more than one factor node.

For Case 1, running belief propagation on factor graph, the joint conditional distribution of the set of variables $X'$ involved in a factor node $q$ (where $q$ can be a familial factor node $f$ and trait factor node $g$) can be calculated:

$$p(X') \propto q(X') \prod_{x \in X'} \mu_{x \rightarrow q}(x)$$

where $\mu_{x \rightarrow q}(x)$ is the message passing from variable node $x$, $x \in X'$ to $q$. For example, if $X' = \{s_1^1, s_1^2, s_2^1\}$ in Fig. 1, $X'$ applies to Case 1, since the nodes in $X'$ entirely involve with $f_1^{a}$. In this case, we can obtain $p(X') \propto f_1^{a}(s_1^1, s_1^2, s_2^1)\mu_{s_2^1 \rightarrow f_1}(s_1^2)\mu_{s_2^1 \rightarrow f_1}(s_2^1)\mu_{s_2^1 \rightarrow f_1}(s_1^2)\mu_{s_2^1 \rightarrow f_1}(s_2^1)$. Furthermore, if $X' = \{t_1^1, t_2^1\}$, $X'$ also applies to Case 1, since the nodes in $X'$ entirely involve with $g_2^{a1}$. In this case, we can obtain $p(X') \propto g_2^{a1}(t_1^1, s_2^1)\mu_{t_1^1 \rightarrow g_2}(t_1^1)\mu_{t_1^1 \rightarrow g_2}(s_2^1)\mu_{t_1^1 \rightarrow g_2}(s_2^1)$.

For Case 2, the joint conditional distribution of $X'$ is equal to the marginal distribution of $X'$ since $X'$ only contains one variable node in this case (because a factor node just connects two variable nodes). The marginal distribution of $X'$ can be calculated by multiplying all messages passed to such variable node in $X'$:

$$p(X') \propto \prod_{p \in ne(x), x \in X'} \lambda_{y \rightarrow x}(x)$$

where $ne(x)$ is the set of neighbor factor nodes of $x$, and $\lambda_{p \rightarrow x}(x)$ is the message passed from factor node $p, p \in ne(x)$ to variable node $x$, $x \in X'$. For example, if $X' = \{t_1^1\}$ in Fig. 1, we can obtain $p(X') \propto \lambda_{g_1 \rightarrow t_1^1}(t_1^1)$.

For Case 3, it is the superset of Case 1 and Case 2, in which the joint conditional distributions of those variable nodes applying to Case 1 and Case 2 are first calculated with Eqs. (1) and (2), respectively. The joint conditional distribution is then equal to the product of the values calculated previously. Assuming $X' = \{s_1^1, s_1^2, s_1^3, t_1^1, t_2^1, t_3^1\}$ in Fig. 1, in which $\{s_1^1, s_1^2, s_1^3\}$ applies to Case 1, $\{t_1^1, t_2^2\}$ also applies to Case 1; meanwhile, $\{t_1^1\}$ applies to Case 2.

We then can calculate

$$p(X'|X_K, A, F) \propto f_1^{a}(s_1^1, s_1^2, s_1^3)\mu_{s_1^2 \rightarrow f_1}(s_1^2)\mu_{s_1^3 \rightarrow f_1}(s_1^3)\mu_{t_1^1 \rightarrow f_1}(s_1^2)\times$$

$$g_2^{a1}(t_1^1, s_2^1)\mu_{t_1^1 \rightarrow g_2}(t_1^1)\mu_{t_1^1 \rightarrow g_2}(s_2^1)\times \lambda_{g_1 \rightarrow t_1^1}(t_1^1) \quad (3)$$

### 4.3 Generation of noise joint conditional distribution

Given the joint conditional distribution $p(X_U|X_K, A, F)$, to construct approximate distribution $p^*(X_U|X_K, A, F)$, we need to inject $\varepsilon$-differential privacy noise. In contrast, the calculation of $p(X_K|X_U^*, A, F)$ based on $X_U^*$ in Phase 4, however, does not need any additional information from the original data. The joint conditional distribution of $X_K$ can be derived from $X_U^*$ directly.

Without loss of generality, we assume $X_U$ applies to Case 3. For the joint conditional distribution of variables $X_U$, the numbers of familial factor node $f_j^i(s_j^i, s_j^{i+1}, s_j^{i+2})$ and trait factor node $g_{jk}^{i}(t_k^i, s_j^i)$ are $m$ and $n$, respectively. The following shows how to drive $p(X_U|X_K, A, F)$ in an $\varepsilon$-differential privacy manner.

For the set of familial factor node $f_j^{i}(s_j^i, s_j^{i+1}, s_j^{i+2})$ presenting in $p(X_U|X_K, A, F)$, such as $f_1^{a}(s_1^1, s_1^2, s_1^3)$ in Eq. (3), we need to inject Laplace noise into the set of $f_j^{i}(s_j^i, s_j^{i+1}, s_j^{i+2})$ with scale of $\frac{\varepsilon}{m}$ to guarantee that the noise version of sets of $f_j^{i}(s_j^i, s_j^{i+1}, s_j^{i+2})$ satisfies ($\frac{\varepsilon}{m}$)-differential privacy since set of $f_j^{i}(s_j^i, s_j^{i+1}, s_j^{i+2})$ has sensitivity $\frac{1}{m}$.

For the set of trait factor node $g_{jk}^{i}(t_k^i, s_j^i)$ presenting in $p(X_U|X_K, A, F)$, such as $g_2^{a1}(t_1^1, s_2^1)$ in Eq. (3), we need to inject Laplace noise into the set of $g_{jk}^{i}(t_k^i, s_j^i)$ with scale of $\frac{\varepsilon}{n}$, in order to guarantee that the noise version of sets of $g_{jk}^{i}(t_k^i, s_j^i)$ satisfies ($\frac{\varepsilon}{n}$)-differential privacy since set of $g_{jk}^{i}(t_k^i, s_j^i)$ has sensitivity $\frac{1}{n}$.

### 4.4 Privacy guarantee

According to compensability property\cite{4}, our method satisfies $\varepsilon$-differential privacy. Specifically, noises injected to all familial factor nodes and all trait factor nodes are $\varepsilon/2$, respectively.

### 5 Related Work

Privacy preserving genomic data release has received much attention in recent years. Caroline et al.\cite{15} proposed
methods for differential privacy preserving release of GWAS catalog statistics, including \( \chi^2 \) -statistics, minor allele frequency, and \( p \)-values. Wang et al.\cite{6} proposed a method sharing data with differential privacy manner by splitting original genomes in a top-down way, and then added noise to each block. Simmons and Berger\cite{7} stated that current methods for identifying high scoring SNPs with differential privacy guarantee have low accuracy and high computational complexity, so that they proposed a new neighbor distance definition for performing private GWAS. Johnson and Shmatikov\cite{8} proposed privacy preserving algorithms for supporting exploratory analysis, including the location of SNPs with strong association with specific disease, correlations among SNPs.

In addition to differential privacy, Sankararaman et al.\cite{9} stated how to combat against the statistical analysis attack (Homer’s attack\cite{10}), by restricting data release scale. Existing works have shown that personal information is threaten by attackers that usually launch attacks by exploiting data correlations and effective privacy preserving methods have also been proposed, such as location, social attributes\cite{11–17}, or mobile wireless networks\cite{18–23}. Reference \cite{24} releases certain number of most crucial SNPs. However, their method makes several unfeasible assumptions, such as taking only \( \chi^2 \) tack into consideration, fixed individual size, and the attacker knows the number of SNPs to release as background knowledge. For example, Johnson and Shmatikov\cite{8} proposed privacy preserving algorithms in order for calculating the statistical information about SNPs involving number and location which significantly imply the association between SNPs and diseases. Humbert et al.\cite{25} proposed two privacy metrics, adversary incorrectness, and uncertainty, to quantify the privacy loss due to inference attacks on released genomic data.

6 Conclusion

We have proposed a differential-privacy preserving kin-genomic data releasing method. Based on factor graph, which has been proven an effective model to incorporate high-dimensional data and multiple correlations among them, our method can factorize the joint conditional distribution of sensitive genomes into sets of local probability distributions by executing belief propagation on factor graph which captures the dependency relationship among family members, SNPs, and traits due to family genetic relationship and SNP/trait associations. A key part of our method is that, to ensure differential privacy, noise can be directly injected into low-dimensional local distributions rather than huge-dimensional genomic data, which significantly improve data utility.

Acknowledgment

This work was partly supported by the National Natural Science Foundation of China (Nos. 61632010 and 61602129).

References


**Zaobo He** is currently a PhD student in the Department of Computer Science at Georgia State University. He received the MS degree from Shannxi Normal University in 2014, and BS degree from Yan’an University in 2011. He is the recipient of Best Paper Award from PAMCO 2016, Outstanding Graduate Research Award from Georgia State University, Molecular Basis of Disease (MBD) Graduate Fellowship from Georgia State University, and IEEE Travel Grant from ICDCS 2015. He is now a student member of the IEEE. His research interests involve privacy preserving data publishing.

**Kaiyang Li** received the MS degree from the University of Electronic Science and Technology of China, Chengdu, China, in 2016. He is currently a PhD student in the School of Computer Science and Engineering, University of Electronic Science and Technology of China. His research interest includes game theory and big data.

**Yingshu Li** received the PhD and MS degrees from University of Minnesota-Twin Cities in 2005 and 2003, respectively. She received the BS degree from the Beijing Institute of Technology, China in 2001. Dr. Li is currently an associate professor in the Department of Computer Science at Georgia State University. Her research interests include wireless networking, sensor networks, sensory data management, social networks, and optimization. Dr. Li is the recipient of an NSF CAREER Award. She is a senior member of the IEEE.

**Ji Li** received the BS degree from Heilongjiang University in 2012. He is currently a PhD student in the Department of Computer Science at Georgia State University. His research area focuses on mobile crowd-sensing.
Yi Liang is a PhD student in the Department of Computer Science at Georgia State University. He received the MS degree in computer science from University of Science and Technology of China in 2014. His research interests include privacy preservation and social networking. He is now a student member of IEEE.