TRANSACTIONS ON DATA PRIVACY 9 (2016) 1–13

Genomic Data Privacy Protection using Compressed Sensing

Aminmohammad Roozgard, Nafise Barzigar, Pramode Verma, Samuel Cheng

Department of Electrical and Computer Engineering, University of Oklahoma, 4502 E. 41st St. Tulsa, OK, USA.

E-mail: {roozgard,barzigar,pverma,samuel.cheng}@ou.edu

Received 12 January 2015; received in revised form 22 October 2015; accepted 22 November 2015

Abstract. In this article, we present a privacy preserving genomic data dissemination algorithm based on compressed sensing. We participated in the challenge at the iDASH on March 24, 2014 in La Jolla, California and the result of the challenge are available online [1]. In our proposed method, we are adding noise to the sparse representation of the input vector to make it differentially private. First, we find the sparse representation using SubSpace Pursuit and then perturb it with sufficient Laplacian noise. We also compared our method with a state-of-the-art compressed sensing privacy protection method [1].

Keywords. Genome wide association studies (GWAS), differential privacy, compressed sensing, sparsity, sampling process, reconstruction process

1 Introduction

Most genomic datasets are not publicly accessible, due to privacy concerns. Patients' genomic data contain identifiable markers and can be used to recognize the presence of an individual in a dataset. Prior research shows that this re-identification is possible when only a very small set of Single Nucleotide Polymorphisms (SNPs) is made public, or even only the frequencies of different SNPs across a population are released [2] . To protect patients, the data owners are restricted to a predefined application and evaluation procedure. Then an agreement (like IRB approval) needs to be signed before the use of data is permitted. This process often takes months to complete and imposes significant restrictions to the researchers.

One solution to the problem can be to let each data owner publish a set of pilot data to help data users choose the right datasets based on their needs. Such pilot data are slightly modified (noise-added) from the original data to ensure that individuals information is protected. The data owners release these pilot data with the noise parameters and the privacy protection mechanism that they used [3]. A data user then can download, run any kind of association tests and compare the outcomes with the other datasets outputs to get an idea which datasets may be useful. With such information, the researchers can approach the owners of the most relevant datasets for further research and with proper agreements.

[1] $http://www.humangenomeprivacy.org$

To tackle this pilot data generation problem, numerous privacy protection frameworks have been proposed [4, 5] and differential privacy is one of the most widely-recognized frameworks to provide a guarantee against privacy attacks [6, 7, 8]. In a nutshell, if a query result from a privacy protected dataset is almost identical after modifying or deleting just one of the records, the dataset is considered differentially private [9]. This gives strong guarantee that the existence of a record cannot be inferred, even if one is allowed to make any arbitrary query.

In particular, we propose to generate differentially private datasets using compressed sensing techniques [10]. The inputs to the proposed method are sequences of genomic nucleotides. We first find the location of SNPs and then calculate the empirical distribution of nucleotides for each SNP. Then, we use a compressed sensing technique to find the sparse representations of SNP frequencies and add Laplacian noises to them. The compressive sensing mechanism allows us to use less noise than other differentially private methods under certain conditions [1].

In the following, we review the basics of compressed sensing, which is the inspiration of this work. After that, differential privacy will be reviewed and the proposed genome privacy protection mechanism is explained in detail. We then present experimental results of proposed method and finally give a brief conclusion.

2 Background

2.1 Compressed Sensing

In this section, we briefly review the theory of the compressed sensing and its major processes and elements[2]. Consider an input vector $d \in \mathbb{R}^n$ that we want to represent it by a vector $x \in \mathbb{R}^n$ using an orthonormal basis (a transform matrix) $\Psi \in \mathbb{R}^{n \times n}$ where $d = \Psi x \in \mathbb{R}^n$ and x is a s-sparse vector ($s < n$) which means x has at most s nonzero entries (see Figure 1).

Note that if x is not a sparse vector, by zeroing the very small coefficients of x, we can make it sparse and this new vector still keeps the most amount of information of the original vector [13, 14, 15, 16]. Furthermore, the orthonormal basis Ψ can be a standard transform basis like wavelet basis or discrete cosine transform basis. The vector d is the input of the compressed sensing method.

Compressed sensing is divide into two processes: a "*sampling process*" and a "*reconstruction process*". The sampling process of compressed sensing reduces the size of input from n to $k = O\big(s \log(n/s)\big)$ through projecting the input d into the output y using a random matrix $\Phi \in \mathbb{R}^{k \times n}$, i.e., $y = \Phi d \in \mathbb{R}^k$. The random matrix Φ is typically composed of independent and identically distributed (i.i.d.) entries from a symmetric Bernoulli distribution (see Figure 2).

The reconstruction process of the compressed sensing exactly or approximately reconstructs the original data from the compressed samples y. Using an l_1-Norm minimization method like Orthogonal Matching Pursuit (OMP)[17] or SubSpace Pursuit (SSP)[18], the recovered answer $\hat{x}$ is close to the original x even in the presence of noise (see Figure 3).

Mathematically, the l_1-Norm minimizer finds the approximate solution of

$$\hat{x} = \arg \min_{x'} \|x'\|_0, \text{ subject to } \|y - Ax'\|_2 < \eta \tag{1}$$

[2] for more information, please read [10, 11, 12]

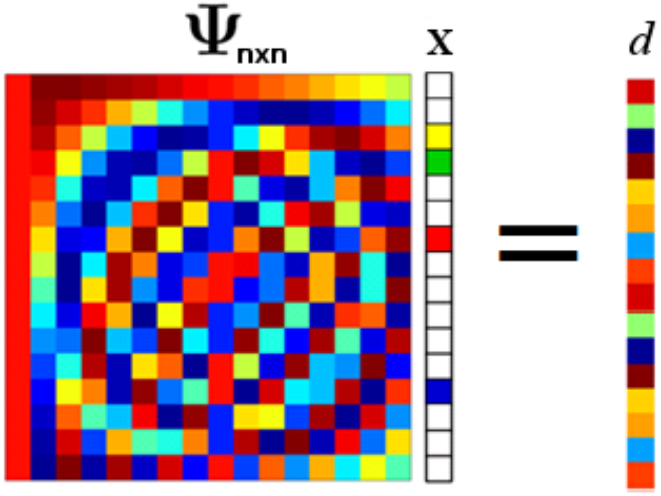


Figure 1: The sparse representation of the natural signals; d is a one dimensional signal that can be represented in a transform domain by a sparse vector x where the transform basis are the columns of the matrix $\Psi \in \mathbb{R}^{n\times n}$. Note that a vector is sparse if most of its elements are equal to zero. Here, the white squares represent zero elements.

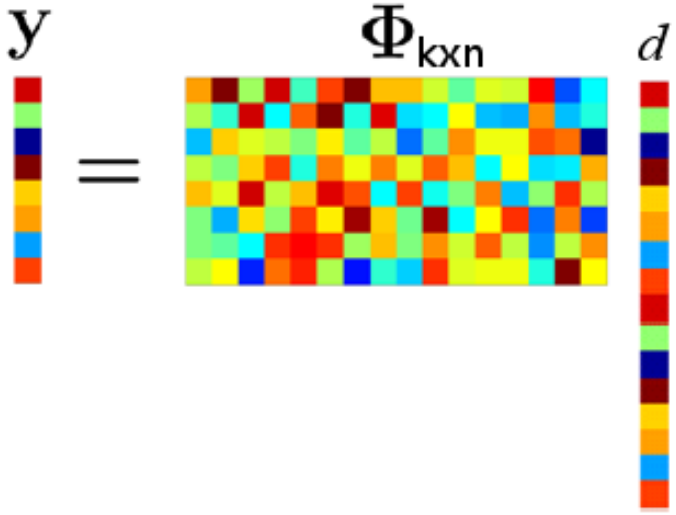


Figure 2: The sampling process of the compressed sensing method; d is an input signal and a random matrix $\Phi \in \mathbb{R}^{k\times n}$ maps d to a measurement vector $y \in \mathbb{R}^k$ which reduces the size of the input signal from n to $k = O\big(s\log(n/s)\big)$.

where the resulting $\hat{x}$ will likely to be a sparse vector (x' is a dummy variable), $A = \Phi\Psi$, $\|x\|_0 := |\{i : x_i \neq 0\}|$[3] and η is a error threshold that determines how close $\hat{x}$ is from the original vector x. A smaller η forces $\hat{x}$ to be closer to x. If we want to guarantee an s-sparse

[3] $\|x\|_1 := \sum_{i=1}^{n} |x_i|$ and $\|x\|_p := \left(\sum_{i=1}^{n} x_i^p\right)^{1/p}$ where $p > 1$

$\hat{x}$, we may consider the following optimization problem instead:

$$\hat{x} = \arg \min_{\|x'\|_0 \leq s} \|y - Ax'\|_2. \quad (2)$$

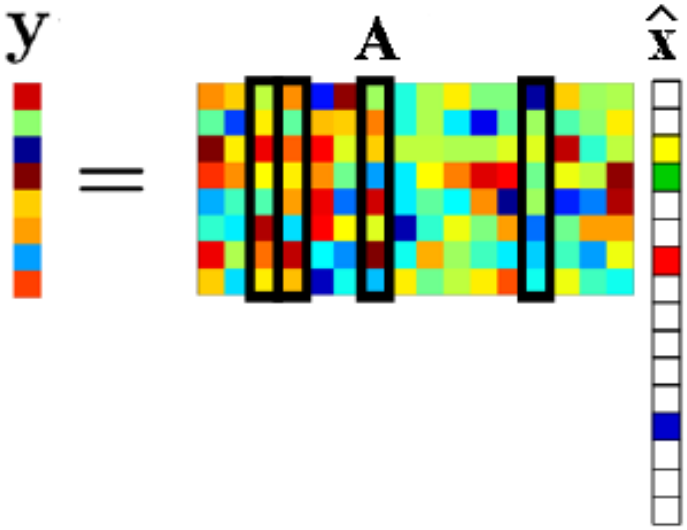


Figure 3: The reconstruction process of the compressed sensing method; y is a measurement vector and $A = \Phi\Psi \in \mathbb{R}^{k \times n}$. The l_1-minimizer "selects" the smallest set of A's columns as the solution which is sparse and its error is less than the threshold η. Note that the columns marked by black are the selected ones.

2.2 SubSpace Pursuit [18]

SubSpace Pursuit (SSP) is an l_1-Norm minimization method which has a reconstruction capability comparable to Linear Programming (LP) methods, and has the benefit of very low reconstruction complexity just as matching pursuit techniques for very sparse signals. For any sampling matrix A satisfying the restricted isometry property (RIP) [19] with a constant parameter independent of K, the SubSpace Pursuit algorithm can recover arbitrary K-sparse signals exactly from its noiseless measurements.

When the measurements are inaccurate and/or the signal is not exactly sparse, the reconstruction distortion is of order a constant multiple of the measurement and/or signal perturbation energy. More precisely, for very sparse signals with $K = O(\sqrt{N})$, the computational complexity of the SubSpace Pursuit algorithm is upper bounded by $O(mNK)$, but can be further reduced to $O(mNlogK)$ when the nonzero entries of the sparse signal decay slowly [4].

[4]We refer readers to [18] for the details of the SubSpace Pursuit.

2.3 Differential Privacy

A randomized algorithm f is ϵ–Differential Private if for all adjacent datasets $\mathcal{D}$ and $\mathcal{D}'$, and any possible output $\hat{\mathcal{D}}$ in the output space of f

$$\frac{Pr[f(\mathcal{D}) = \hat{\mathcal{D}}]}{Pr[f(\mathcal{D}') = \hat{\mathcal{D}}]} \leq e^{\epsilon}. \tag{3}$$

The Laplacian mechanism [20] is commonly used in data disturbing methods to achieve differential privacy, which adds noises generated from a Laplacian distribution to the output of a query on the dataset. The amount of noise to be added is based on the sensitivity of the computed data. The sensitivity represents the maximum change of the output when a single modification happens to a dataset. For any f, and all adjacent datasets $\mathcal{D}$ and $\mathcal{D}'$, the sensitivity of f can be calculated as follows:

$$\Delta f = \max_{\mathcal{D},\mathcal{D}'} \|f(\mathcal{D}) - f(\mathcal{D}')\|_1. \tag{4}$$

3 Proposed Method

The proposed privacy protection method is based on compressed sensing principle. The inputs are sequences of genomic nucleotides (A, C, G, T) from multiple subjects. First, we process the input genomic sequences to identify the locations of the Single Nucleotide Polymorphisms (SNPs). Basically, we go through each location of all sequences and mark the location as SNP if the corresponding nucleotides vary among different sequences. Then we keep only the SNP locations as representatives of the original genome sequences and count the frequencies of nucleotides at each SNP location. At a SNP location, the nucleotide that has minimum counts is called "minor" and the nucleotide occurred most is called "major". The frequencies of these majors and minors will be manipulated in the proposed method below.

To have a differentially private data, we are using the compressed sensing technique and adding noise to the genomic data representation in the transform domain. As it was explained in section 2.1, we need to specify an input vector y, a random matrix Φ and a transform matrix Ψ for the compressed sensing process. The input vector y is the frequency of the minor SNPs which we have obtained from pre-processing step. The random matrix Φ is a binary independent and identically distributed (i.i.d.) Bernoulli distributed matrix which on average, half of the coefficient in each row and each column are "1"s and the rest are equal to "0"s. We choose Haar wavelet transform (HWT) [21, 22] in this work because we can readily allocate an ϵ budget that ensures differential privacy [23, 1, 24].

In our proposed method, we are adding noise on to the sparse representation of the input vector to make it differentially private. We first compute the matrix $A = \Phi\Psi$ and then obtain $\hat{x}$ (the sparse representation of the y) using SubSpace Pursuit [18]. To make sure that the amount of the noise that we are adding is sufficient to guarantee differential privacy, we add Laplacian noise, $Lap\big(2 \times \frac{\sqrt{k}}{\epsilon}\big)$, to all element of the sparse representation according to the geometric noise mechanism [24]. Note that the elements in the sparse representation are corresponding to those on a HWT tree (see Figure 4), therefore based on [23], we double the amount of noise when we move up one step in the wavelet tree from the leaves to the root. We denote the noise-added sparse representation as x^*. Finally, to get the publishable data y^*, we apply the inverse transform on x^* as follows:

$$y^* = \Psi x^*, \tag{5}$$

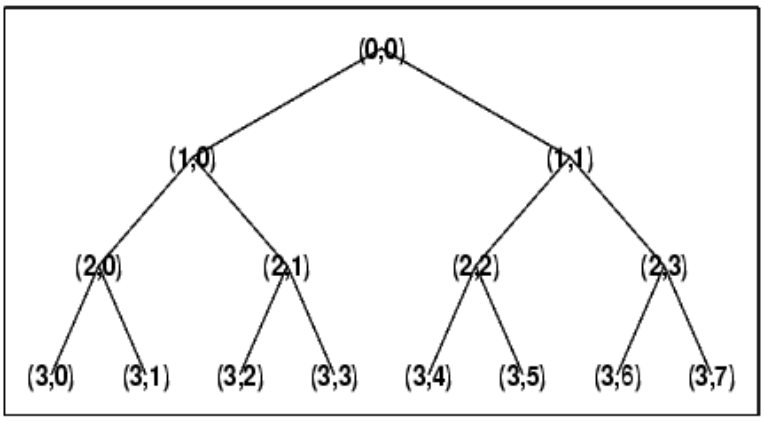


Figure 4: The Haar wavelet tree structure of [$(0,0)$, $(1,0)$, $(1,1)$, $(2,0)$, $(2,1)$, $(2,2)$, $(2,3)$, $(3,0)$, $(3,1)$, $(3,2)$, $(3,3)$, $(3,4)$, $(3,5)$, $(3,6)$, $(3,7)$]. Note that each (x,y) shows one entry in the array. The node $(0,0)$ is the root of the wavelet transform tree and the nodes $(3,t)$, where t is a number between 0 and 7, are the leaves of the wavelet transform tree.

where the Ψ is the Haar wavelet transform matrix. Algorithm 1 shows a summary of the proposed method.

4 Experimental Results

To evaluate the proposed method, we participated in the challenge at the iDASH (UCSD–based National Center for Biomedical Computing: NIH U54HL108460) workshop [5] on March 24, 2014 in La Jolla, California and the results are available on-line at [25]. The challenge was about sharing aggregate human genomic data (i.e., allele frequencies) to preserve the privacy of the data and to maximize the utility of the data for Genome–Wide Association Studies (GWAS). We applied the proposed method on the two datasets from the case and control groups of individuals: the case group includes 411 individuals from the Personal Genome Project [26] [6], and the control data includes 174 participants from the CEU population in HapMap [27] [7]. There are two test sets: the first one consists of 311 SNP sites of the human chromosome 2 and the second one consists of 600 SNP sites of human chromosome 10. Table 1 shows the result of proposed method on these two test sets compared to the SNP–Based baseline of the challenge organizers.

In our proposed method, we are adding noise on to the sparse representation of the input vector to make it differentially private. We first compute the matrix $A = \Phi\Psi$ and then obtain $\hat{x}$ (the sparse representation of the y) using SubSpace Pursuit [18]. To make sure that the amount of the noise that we are adding is sufficient to guarantee differential privacy, we add Laplacian noise, $Lap\big(2 \times \frac{\sqrt{k}}{\epsilon}\big)$, to all element of the sparse representation.

The SNP–Based algorithm treats the allele counts across multiple SNP sites as a histogram, and add Laplacian noises to the allele counts. The privacy budget $\epsilon = 1$ is used for all cases.

[5] http://www.humangenomeprivacy.org
[6] http://www.personalgenomes.org/
[7] http://hapmap.ncbi.nlm.nih.gov/index.html.en

Algorithm 1 : Proposed Differentially Private Protection using Compressed Sensing Algorithm for Genomic Data

Inputs: a set of genome sequences $\mathcal{X}$
Pre-processing: Converting genome sequences to SNP sequences
Initialization: Set the initial parameters:

- Find the frequencies of major and minor for each SNP location
- Generate sampling matrix Φ from a Bernoulli i.i.d. distribution
- Generate Haar wavelet transform matrix Ψ
- Calculate matrix $A = \Phi\Psi$
- Consider an array contains the frequencies of minors as input vector y

Adding Noise: Add Laplacian noise:

- Find $\hat{x}$, a sparse representation of the input vector y using SSP [18] where $y = A\hat{x}$
- Add sufficient Laplacian noise (see last paragraph of proposed method for details)

Post-processing:

- $y^* = \Psi x^*$

Output: the noise-added differentially private version of SNP frequencies y^*

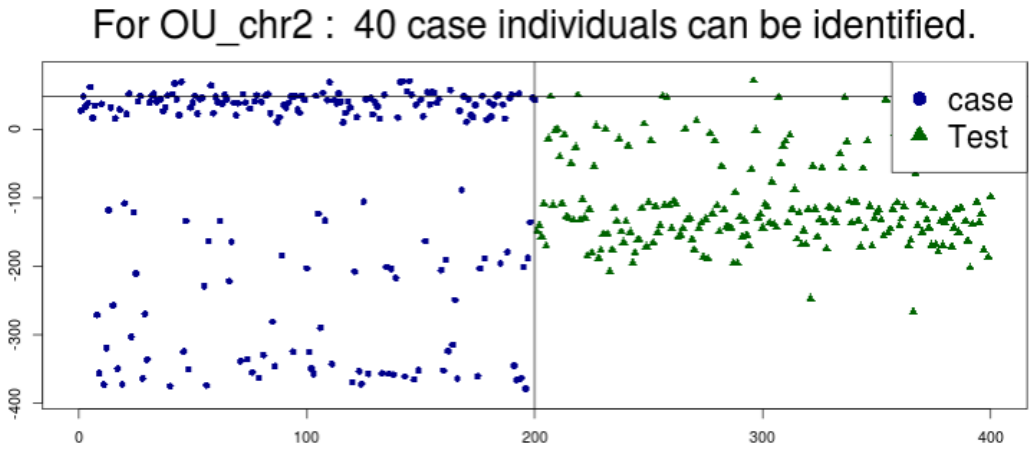


Figure 5: The online privacy evaluation [25] of the proposed method on chromosome 2 with p-value 0.01

The power of the likelihood ratio test [28, 29] is a privacy risk measure which is the number of case individuals who can be recognized with confidence level more than a threshold. The lower likelihood ratio power level shows that the perturbed data has less risk of re-

Data Set		SNP–Based baseline		Proposed Method		Significant SNPs #
Dataset 1	Power	0.05		0.61		
	Cutoffs	TPR	FPR	TPR	FPR	
	5×10^{-2}	0.864	0.844	1.0	0.941	22
	10^{-3}	0.632	0.774	1.0	0.884	19
	10^{-5}	0.642	0.700	1.0	0.879	14
Dataset 2	Power	0.4		0.005		
	Cutoffs	TPR	FPR	TPR	FPR	
	5×10^{-2}	0.933	0.924	1.0	0.958	45
	10^{-3}	0.800	0.862	1.0	0.909	15
	10^{-5}	0.625	0.788	1.0	0.876	8

Table 1: Results of the proposed method on the challenge datasets. The Dataset 1 refers to 200 participants with 311 SNPs on chromosome 2 and Dataset 2 refers to 200 participants with 610 SNPs on chromosome 10. The power is defined as the ratio of identifiable individuals in the case group using the likelihood ratio test. The false positive rate (FPR) and true positive rate (TPR) based on χ^2 test are listed per different cutoff thresholds. In addition, the last column corresponds to the number of significant SNPs.

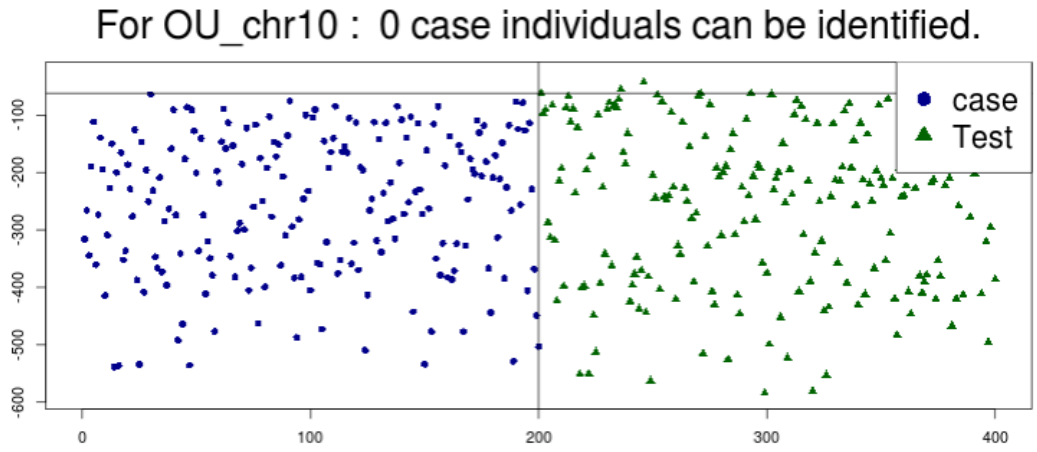


Figure 6: The online privacy evaluation [25] of the proposed method on chromosome 10 with p-value 0.01

identification. Figures 5 and 6 show the power of the proposed method evaluated on two different datasets. The dots in those figures represent the log likelihood ratio statistic T_i [28] for each individual i in the case and test group. The horizontal line shows the 0.99 confidence level for re-identifying of an individual from the case group based on the estimated log likelihood test statistic values of test group individuals. As shown in these figures, 40 case individuals can be identified in the chromosome 2 with p–value 0.01 but none of the case individuals is identifiable in chromosome 2 with the same p–value. Therefore the proposed method has better performance on the chromosome 10 compare to the chromosome 2.

Moreover, the utility of a privacy protection method can be estimated based on the χ^2 test

to enumerate the significant SNPs with different cutoff p–values. As Table 1 shows, the proposed method has higher true positive on the perturbed data compared to the SNP–Based baseline algorithm. Figures 7 and 8 show the utility of the proposed method obtained from the online evaluation tool WIDGET [25]. In particular, they show the box plot of the positive (P), negative (N), true positive (TP), false positive (FP), true negative (TN) and false negative (FN) SNPs detection after running 1000 times on chromosomes 2 and 10, respectively. In addition, these figures also show true negative rate (SPC), false predictive value (PPV), negative predictive value (NPV), false positive rate (FPR), true positive rate (TPR) and accuracy (ACC).

In addition to the challenge, we select 180 SNPs of the Personal Genome Project [26] and partition it into two subsets. The first subset is used as a control group. We modified the frequencies of randomly selected SNPs on second subset and generated 3 new datasets A, B, and C with different levels of utility (with 27, 9 and 4 significant SNPs, respectively). We then perturbed each resulting dataset with different privacy protection methods, and computed the p-value of each SNP using the χ^2 test. We then ranked the utility of the perturbed datasets based on the test result. We expect that the order of utility should preserve after applying privacy protection. Namely, dataset A should have the highest utility and C the lowest. However, we observed that such order did not always preserve. Actually, dataset A did not even always maintain its highest utility after applying privacy protection. We summarize these observations in Table 2. "Best pick" percentage counts the fraction of cases when A maintains a highest utility after privacy protection. Similar, "correct order" percentage specifies the fraction of cases when the order of utility is preserved. In this experiment, the proposed method appears to dominate the control methods.

	Correct Order %	Best Pick %
SNP–Based baseline	19.34	34.54
Reference [1]	24.85	47.52
Proposed Method	25.02	75.12

Table 2: Percentages of "best pick" and "correct order" for the SNP–Based baseline, Reference [1] and the proposed method.

5 Conclusion

In this work, we present a privacy preserving genomic data dissemination algorithm based on the compressed sensing. The inputs to the proposed method are sequences of the genomic nucleotides from multiple subjects. The method first estimates the locations of the SNPs, then it transforms the frequencies of SNPs into the sparse domain. Finally, Laplacian noise is added to all element of the sparse representation according to [24] to ensure differential privacy of the output. The proposed method is evaluated through the iDASH 2014 privacy challenge and with the SNP data from the Personal Genome Project.

Acknowledgements

The authors would like to thank iDASH health care privacy protection challenge organizers for giving us this opportunity to participate and present our proposed method. They would

also like to thank Dr. Xiaoqian Jiang and Dr. Shuang Wang for helpful discussions and suggestions and Ms. Sepideh Darbandi for editing the manuscript.

References

[1] Li, Y.D., Zhang, Z., Winslett, M., Yang, Y.: Compressive mechanism: Utilizing sparse representation in differential privacy. In: Proceedings of the 10th Annual ACM Workshop on Privacy in the Electronic Society, pp. 177–182 (2011). ACM

[2] Wjst, M.: Caught you: threats to confidentiality due to the public release of large-scale genetic data sets. BMC medical ethics **11**(1), 21 (2010)

[3] Karr, A.F.: The role of transparency in statistical disclosure limitation. Joint UNECE/Eurostat Work Session on Statistical Data Confidentiality (2009)

[4] Chen, Y., Peng, B., Wang, X., Tang, H.: Large-scale privacy-preserving mapping of human genomic sequences on hybrid clouds. In: NDSS (2012)

[5] Kifer, D., Machanavajjhala, A.: A rigorous and customizable framework for privacy. In: Proceedings of the 31st Symposium on Principles of Database Systems, pp. 77–88 (2012). ACM

[6] Dwork, C., Smith, A.: Differential privacy for statistics: What we know and what we want to learn. Journal of Privacy and Confidentiality **1**(2), 2 (2010)

[7] Dwork, C.: Differential privacy: A survey of results. Theory and Applications of Models of Computation **4978**(1), 1–19 (2008)

[8] Bambauer, J., Muralidhar, K., Sarathy, R.: Fool's gold: an illustrated critique of differential privacy. Vand. J. Ent. & Tech. L. **16**, 701 (2013)

[9] Gabriel, S.B., Schaffner, S.F., Nguyen, H., Moore, J.M., Roy, J., Blumenstiel, B., Higgins, J., DeFelice, M., Lochner, A., Faggart, M., *et al.*: The structure of haplotype blocks in the human genome. Science **296**(5576), 2225–2229 (2002)

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

[11] Barzigar, N., Roozgard, A., Verma, P., Cheng, S.: A video super resolution framework using scobep. IEEE Transactions on circuits and systems for video technology (2013)

[12] Roozgard, A., Barzigar, N., Cheng, S., Verma, P.: Medical image registration using sparse coding and belief propagation. In: Engineering in Medicine and Biology Society (EMBC), 2012 Annual International Conference of the IEEE, pp. 1141–1144 (2012). IEEE

[13] Candes, E.J.: The restricted isometry property and its implications for compressed sensing. Comptes Rendus Mathematique **346**(9), 589–592 (2008)

[14] Roozgard, A., Barzigar, N., Wang, S., Jiang, X., Ohno-Machado, L., Cheng, S.: Nucleotide sequence alignment using sparse coding and belief propagation. In: Engineering in Medicine and Biology Society (EMBC), 2013 35th Annual International Conference of the IEEE, pp. 588–591 (2013). IEEE

[15] Barzigar, N., Roozgard, A., Cheng, S., Verma, P.: Scobep: Dense image registration using sparse coding and belief propagation. Journal of Visual Communication and Image Representation **24**(2), 137–147 (2013)

[16] Roozgard, A., Barzigar, N., Cheng, S., Verma, P.: Dense image registration using sparse coding and belief propagation. In: Signal Processing and Communication Systems (ICSPCS), 2011 5th International Conference On, pp. 1–5 (2011). IEEE

[17] Pati, Y.C., Rezaiifar, R., Krishnaprasad, P.: Orthogonal matching pursuit: Recursive function approximation with applications to wavelet decomposition. In: Signals, Systems and Computers, 1993. 1993 Conference Record of The Twenty-Seventh Asilomar Conference On, pp. 40–44 (1993). IEEE

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

[19] Baraniuk, R., Davenport, M., DeVore, R., Wakin, M.: A simple proof of the restricted isometry property for random matrices. Constructive Approximation **28**(3), 253–263 (2008)

[20] Dwork, C., McSherry, F., Nissim, K., Smith, A.: Calibrating noise to sensitivity in private data analysis **3876**, 265–284 (2006)

[21] Heil, C.E., Walnut, D.F.: Continuous and discrete wavelet transforms. SIAM review **31**(4), 628–666 (1989)

[22] Mulcahy, C.: Image compression using the haar wavelet transform. Spelman Science and Mathematics Journal **1**(1), 22–31 (1997)

[23] Xiao, X., Wang, G., Gehrke, J.: Differential privacy via wavelet transforms. Knowledge and Data Engineering, IEEE Transactions on **23**(8), 1200–1214 (2011)

[24] Cormode, G., Procopiuc, M., Srivastava, D., Tran, T.T.: Differentially private publication of sparse data. arXiv preprint arXiv:1103.0825 (2011)

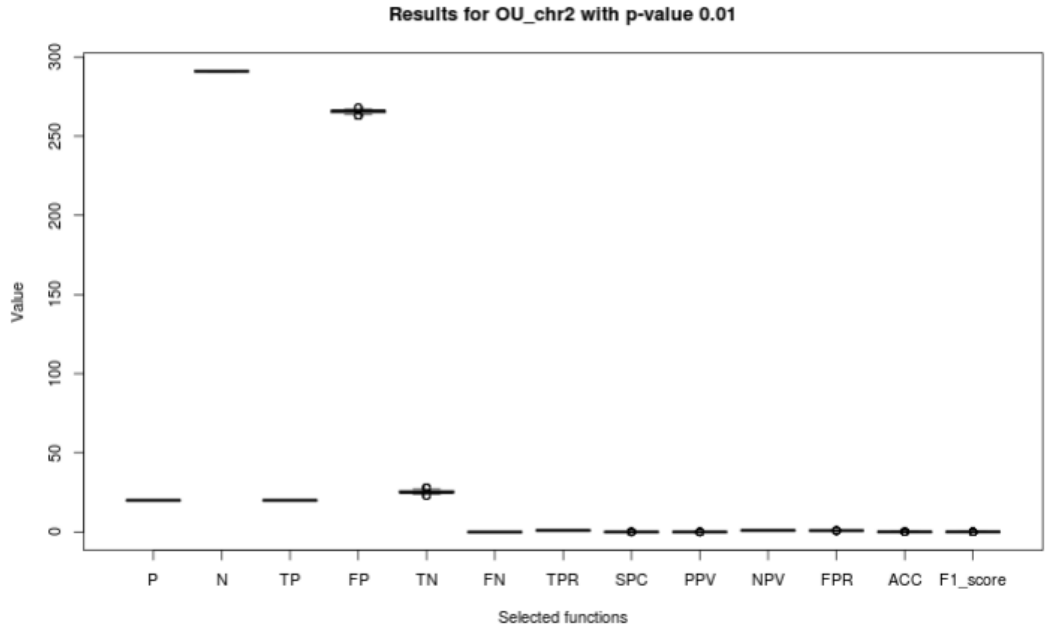
[25] Wang, S., Wei, W., Ji, Z., Zhao, Y., Jiang, X., Wang, X., Tang, H., Ohno-Machado, L.: WIDGET: a Web Interface for Dynamic Genome-privacy EvaluaTion. https://humangenomeprivacy.ucsd-dbmi.org/

[26] Church, G.M.: The personal genome project. Molecular Systems Biology **1**(1) (2005)

[27] Gibbs, R.A., Belmont, J.W., Hardenbol, P., Willis, T.D., Yu, F., Yang, H., Ch'ang, L.-Y., Huang, W., Liu, B., Shen, Y., *et al.*: The international hapmap project. Nature **426**(6968), 789–796 (2003)

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

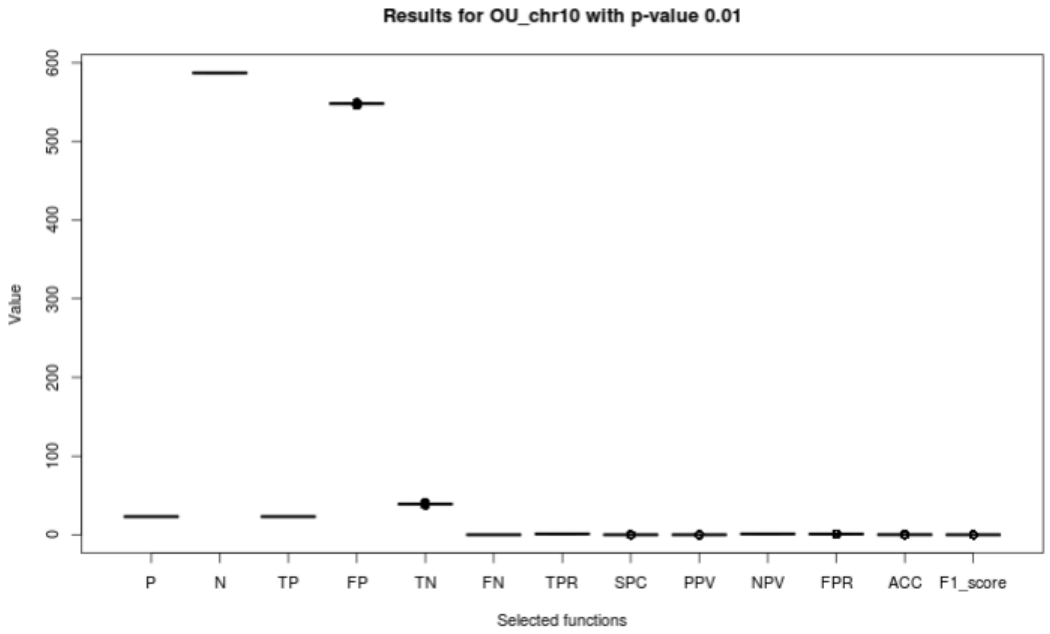
[29] Homer, N., Szelinger, S., Redman, M., Duggan, D., Tembe, W., Muehling, J., Pearson, J.V., Stephan, D.A., Nelson, S.F., Craig, D.W.: Resolving individuals contributing trace amounts of dna to highly complex mixtures using high-density snp genotyping microarrays. PLoS genetics **4**(8), 1000167 (2008)




```
Notations:   P - Positive;                          N - Negative;
            TP - True Positive;                    FP - False Positive;
            TN - True Negative;                    FN - False Negative;
           SPC - True Negative Rate (Specificity); PPV - positive predictive value (Precision)
           NPV - Negative Predictive Value;        FPR - False Positive Rate
           TPR - True Positive Rate (Sensitivity); ACC - Accuracy;
Summarize results:
  1 result was selected.
    Result "OU_chr2" has 311 SNPs over 1000 runs with p-value:0.01

==============Statistics for results: OU_chr2=========================
For p-value < 0.01, number of significant SNPs in Case Group is 20
After adding noise:
              mean    median       min       max        sd    n
P         20.00000  20.00000  20.00000  20.00000 0.0000000 1000
N        291.00000 291.00000 291.00000 291.00000 0.0000000 1000
TP        20.00000  20.00000  20.00000  20.00000 0.0000000 1000
FP       265.59300 266.00000 263.00000 268.00000 0.7320430 1000
TN        25.40700  25.00000  23.00000  28.00000 0.7320430 1000
FN         0.00000   0.00000   0.00000   0.00000 0.0000000 1000
TPR        1.00000   1.00000   1.00000   1.00000 0.0000000 1000
SPC        0.08731   0.08591   0.07904   0.09622 0.0025156 1000
PPV        0.07003   0.06993   0.06944   0.07067 0.0001794 1000
NPV        1.00000   1.00000   1.00000   1.00000 0.0000000 1000
FPR        0.91269   0.91409   0.90378   0.92096 0.0025156 1000
ACC        0.14600   0.14469   0.13826   0.15434 0.0023538 1000
F1_score   0.13089   0.13072   0.12987   0.13201 0.0003134 1000
```

Figure 7: The online utility evaluation [25] of the proposed method on chromosome 2 with p-value 0.01




```
Notations:   P - Positive;                          N - Negative;
            TP - True Positive;                    FP - False Positive;
            TN - True Negative;                    FN - False Negative;
           SPC - True Negative Rate (Specificity); PPV - positive predictive value (Precision)
           NPV - Negative Predictive Value;        FPR - False Positive Rate
           TPR - True Positive Rate (Sensitivity); ACC - Accuracy;
Summarize results:
  1 result was selected.
    Result "OU_chr10" has 610 SNPs over 1000 runs with p-value:0.01

===============Statistics for results: OU_chr10=========================
For p-value < 0.01, number of significant SNPs in Case Group is 23
After adding noise:
              mean    median       min       max         sd    n
P         23.00000  23.00000  23.00000  23.00000 0.00000000 1000
N        587.00000 587.00000 587.00000 587.00000 0.00000000 1000
TP        23.00000  23.00000  23.00000  23.00000 0.00000000 1000
FP       548.05400 548.00000 546.00000 550.00000 0.45308861 1000
TN        38.94600  39.00000  37.00000  41.00000 0.45308861 1000
FN         0.00000   0.00000   0.00000   0.00000 0.00000000 1000
TPR        1.00000   1.00000   1.00000   1.00000 0.00000000 1000
SPC        0.06635   0.06644   0.06303   0.06985 0.00077187 1000
PPV        0.04028   0.04028   0.04014   0.04042 0.00003199 1000
NPV        1.00000   1.00000   1.00000   1.00000 0.00000000 1000
FPR        0.93365   0.93356   0.93015   0.93697 0.00077187 1000
ACC        0.10155   0.10164   0.09836   0.10492 0.00074277 1000
F1_score   0.07743   0.07744   0.07718   0.07770 0.00005912 1000
```

Figure 8: The online utility evaluation [25] of the proposed method on chromosome 10 with p-value 0.01