Bioinformatics, 32(12), 2016, 1873–1879
doi: 10.1093/bioinformatics/btw073
Advance Access Publication Date: 11 February 2016
Original Paper

Data and text mining

Privacy-preserving microbiome analysis using secure computation

Justin Wagner[1], Joseph N. Paulson[1,†], Xiao Wang[2], Bobby Bhattacharjee[2] and Héctor Corrada Bravo[1,*]

[1]Center for Bioinformatics and Computational Biology and [2]Maryland Cybersecurity Center, Department of Computer Science, University of Maryland, College Park, MD USA

*To whom correspondence should be addressed.
†Present address: Department of Biostatistics and Computational Biology, Dana-Farber Cancer Institute and Department of Biostatistics, Harvard T.H. Chan School of Public Health, Boston, MA, USA

Associate Editor: Jonathan Wren

Received on July 23, 2015; revised on December 4, 2015; accepted on January 31, 2016

Abstract

Motivation: Developing targeted therapeutics and identifying biomarkers relies on large amounts of research participant data. Beyond human DNA, scientists now investigate the DNA of micro-organisms inhabiting the human body. Recent work shows that an individual's collection of microbial DNA consistently identifies that person and could be used to link a real-world identity to a sensitive attribute in a research dataset. Unfortunately, the current suite of DNA-specific privacy-preserving analysis tools does not meet the requirements for microbiome sequencing studies.

Results: To address privacy concerns around microbiome sequencing, we implement metagenomic analyses using secure computation. Our implementation allows comparative analysis over combined data without revealing the feature counts for any individual sample. We focus on three analyses and perform an evaluation on datasets currently used by the microbiome research community. We use our implementation to simulate sharing data between four policy-domains. Additionally, we describe an application of our implementation for patients to combine data that allows drug developers to query against and compensate patients for the analysis.

Availability and implementation: The software is freely available for download at: http://cbcb.umd.edu/~hcorrada/projects/secureseq.html

Supplementary information: Supplementary data are available at *Bioinformatics* online.

Contact: hcorrada@umiacs.umd.edu

1 Introduction

Microbiome sequencing seeks to characterize and classify the composition and structure of microbial communities from metagenomic DNA samples. It is estimated that only 1 in 10 cells in and on a person's body contain that individual's DNA (Turnbaugh *et al.*, 2009), the remainder corresponding to microbial DNA, most from organisms that cannot be cultured and studied in the laboratory.

The Human Microbiome Project (HMP) (Turnbaugh *et al.*, 2007), the Global Enterics Multi-Center Study (MSD)(Pop *et al.*, 2014), the Personal Genome Project (Church, 2005) and the American Gut Project (Blaser *et al.*, 2013) aim to characterize the ecology of human microbiota and its impact on human health. Potentially pathogenic or probiotic bacteria can be identified by detecting significant differences in their distribution across healthy and disease populations. While the biology has led to promising results, privacy concerns of microbiome research are now being identified with no secure analysis tools available.

Recent work by Franzosa *et al.* (2015) shows that microbiome data are an unique identifier across time points in a dataset and could be used to link a sensitive attribute to an individual. Earlier work by Fierer *et al.* (2010) showed that it is possible to identify an object that an individual touched by comparing microbiome samples from the object and the individual's hand. We provide a thorough review of microbiome sequencing and a categorization of microbiome privacy considerations in the Supplementary Materials. To counter these concerns, we present an implementation and evaluation of metagenomic association analyses in a secure multi-party computation (SMC) framework. For this work, we focus on garbled circuits, a cryptographic technique that evaluates a function over private inputs from two parties. In this article, we concentrate on the case where two parties, each holding organism abundances in a set of case and control samples, are interested in performing an association analysis (e.g. determining organisms that are differentially abundant in cases) over their combined data, without revealing organism abundances in any specific sample.

We provide a detailed review of this approach in Section 3 and benchmark our secure implementation of commonly used microbiome analyses on three public datasets. We also quantify the statistical gain of analysis using combined datasets by simulation with a dataset that contains samples from four different countries.

We believe that implementing metagenomic analyses in an SMC framework will prove beneficial to researchers focused on the human microbiome as well as the secure computation community. Computational biologists will benefit from a method that allows efficient and secure function evaluation over datasets which they may be obligated to keep confidential. Security researchers can draw on the findings from our work and construct protocols that enable sharing large, sparse datasets to perform analysis.

2 System and methods

Our secure metagenomic analysis system is built upon garbled circuits (Malkhi *et al.*, 2004), which we describe in this Section. We then detail our system including participants along with alternative approaches in the design space for privacy-preserving analysis.

2.1 Garbled circuits

Two parties, one holding input x and another holding input y, wish to compute a public function $F(x, y)$ over their inputs without revealing anything besides the output. The parties could provide their inputs to a trusted third-party that computes the function and reveals the output to each party. However, modern cryptography offers a mechanism to run a protocol between only the two parties while achieving the desired functionality. The main idea behind garbled circuits is to represent the function to be computed as a Boolean circuit over the inputs from both parties and use encryption to hide the input of each party during evaluation by mapping each 0 and 1 bit of the inputs unto random strings that still compute the same result. At the end of circuit evaluation, the resulting random strings can be mapped back to appropriate 0 and 1 bit values that can then be released to each party. In this way, each party learns $F(x, y)$ without learning anything else about the input of x and y. Figure 1 illustrates the garbled circuits protocol.

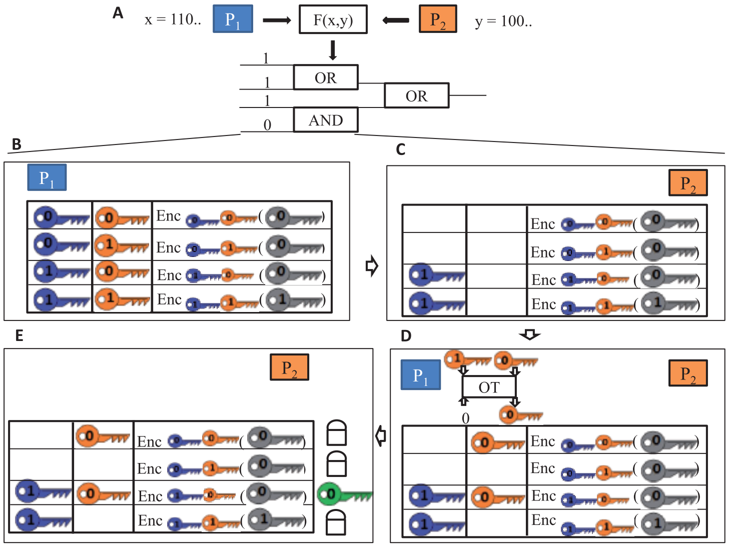
2.2 System participants

We consider the case in which parties located in two policy-domains want to perform metagenomic analyses over shared data. Examples of policy-domains include countries with differing privacy laws or institutions (universities, companies) that stipulate different data disclosure procedures.

For $i \in 1, 2$, denoting PD_i as a policy domain, R_i as a researcher in policy domain i, D_i as the data from R_i, F as the set of functions that a set of R_is would like to compute we consider the following setting:

> R_1 and R_2 would like to compute F over combined D_1 and D_2 but cannot do so by broadcasting the data as either PD_1 or PD_2 does not allow for public release or reception of individual-level microbiome data. We set $|i| = 2$ but this setting could be generalized to any i.

Policy domains naturally arise due to differences in privacy laws. For example, studies currently funded by the NIH are required to release non-human genomic sequences including human microbiome data (http://gds.nih.gov/PDF/NIH_GDS_Policy.pdf). In contrast, the European General Data Protection Regulation, which is currently in draft form, lists biometric data and 'any "data concerning health" means any personal data which relates to the physical or mental health of an individual, or to the provision of health services to the individual' as protected information that is not to be released



A) Parties P1 and P2 agree on a function F(x,y) to compute over inputs x and y each represented as bit strings. F(x,y) will be modeled as a Boolean circuit which takes in two bit strings as input and produces a bit string as output.

B) P1 generates a circuit that computes F(x,y) and "garbles" each of its gates by replacing the input bit values with random strings. These are then used as keys to encrypt the possible logical results of each gate.

C) P1 sends to P2 the circuit wiring diagram, the encrypted logical results for each gate, and the keys mapping to the bits of P1's input.

D) An Oblivious Transfer protocol is used so that P1 can retrieve the encryption keys that map to its input from P2. At this point, P2 holds the encryptions for the possible outputs of the gate, the key for P1's input and the encryption key for to its input. Neither party is able to learn the other party's input from this exchange.

E) P2 use the encryption keys to decrypt the 4 ciphertexts and only that corresponding to the correct output will properly decrypt. This way, P2 computes the output for the circuit holding a set of output keys that map to corresponding bits of F(x,y).

F) P2 sends the output keys to P1 which will use the mapping between the random strings and the bits of F(x,y) to find the value of F(x,y).

Fig. 1. Schematic illustration of the garbled circuits protocol. For analyses discussed in this paper, parties P1 and P2 are researchers performing a statistical analysis over combined data. They provide metagenomic count matrices, or locally precomputed statistics computed from count matrices, along with case/control status as input. Function $F(x, y)$ is determined by the analysis performed, e.g. test on difference in Alpha Diversity between case and control. The 'garbling' in step (B) also includes randomly permuting the rows of the truth table so that the inputs are not revealed by the ordering - we omit this from the figure for clarity. A review of the Oblivious Transfer protocol used in step (D) is provided in Supplementary Materials Section S3

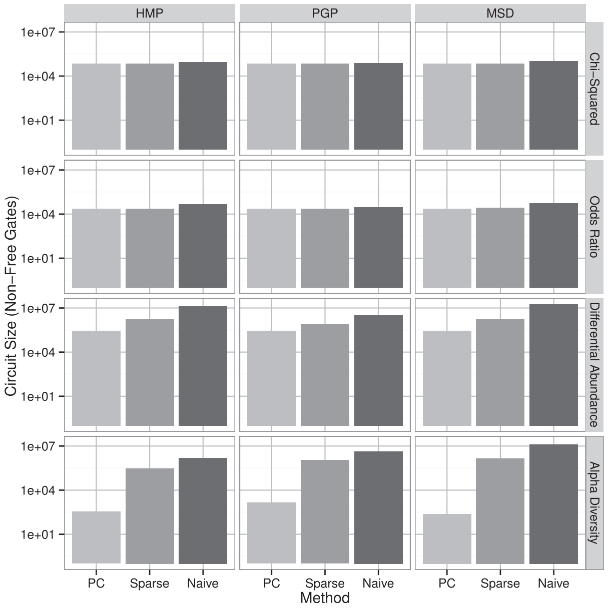


Fig. 2. Circuit size per feature for each implementation and dataset. The feature count for Alpha Diversity is the number of samples. The differences in Alpha Diversity between datasets is explained by the number of samples for PGP (168) being much lower than that of HMP (694) and MSD (992). PC, Pre-compute

publicly (http://www.europarl.europa.eu/sides/getDoc.do?pubRef=-//EP//TEXT+TA+P7-TA-2014-0212+0+DOC+XML+V0//EN). Therefore, researchers in the USA and EU may encounter different policies for data release but still have an interest in computing metagenomic analyses over shared data. Also, given the results published by Franzosa *et al*., some institutions may re-evaluate microbiome data release policies.

2.2.1 Threat model

We consider researcher R_1, who has a microbiome sample from a victim mixed with other samples, to be a semi-honest adversary, or one that follows the protocol but examines the transcript to learn more information than it should. Researcher R_2 is examining an association for a specific trait and would like to expand her study to use samples held by R_1. R_1 wants to determine if the victim is in R_2's dataset and thus learn a sensitive attribute of the victim such as disease status.

The attacks of Fierer *et al*. and Franzosa *et al*. operate over the vector of feature counts for a given sample. For the analyses studied in this article, an adversary will have no better chance of reconstructing the count vector for a specific sample than guessing the majority, or mode, of the count of any specific feature in this system. Through using a garbled circuit implementation of metagenomic analyses, R_2 will be able to keep the vector of microbiome features for any sample private, learn the outputs of functions that she would like to learn over the shared data, and prevent R_1 from completing the attack.

2.3 Solution design approaches

We consider different approaches to allow two parties to compute analyses over data which each must keep confidential.

2.3.1 Access control plus trusted third party

In the USA, the NIH has recognized re-identification through publicly posted genomic data as a realistic threat. Therefore, policy allows for publication of summary statistics and transfer of individual level sequencing data through access control using the Database for Genotypes and Phenotypes (Mailman *et al*., 2007). Once a researcher receives permission to access data, she is provided the data and is required to maintain the access control list for her research group. We look to remove the need for access control by implementing the queries that a researcher would like to run without revealing the data directly.

2.3.2 Differential privacy

Statistical perturbation of analysis results, most widely implemented as differential privacy, is a second approach for researchers to provide privacy guarantees to participants. In this setting, a researcher maintains a data set and allows other researchers to perform queries over the data. Informally, the results of these queries are perturbed in such a manner that an adversary, with access to query results over a data set in which one specific participant has a set of values and results from another data set with that specific participant having a different set of values, will not be able to infer any information about that individual by examining the results (Groce, 2014). Although this approach provides provable privacy guarantees, the introduction of statistical noise has not gained traction in the computational biology research community. Also, recent work showed that learning warfarin dosage models on differentially private data sets introduces enough noise that the dosage recommendation could be fatal to patients (Fredrikson *et al*., 2014).

2.3.3 Secure multiparty computation

An alternative solution which we undertake is using secure computation to perform metagenomic analyses. Other researchers have presented SMC for computing secure genome-wide association studies using secret-sharing, but that particular approach requires the use of three parties for computing tasks (Kamm *et al*., 2013). We address the feasibility of using garbled circuits to implement metagenomic analyses in terms of running time, network traffic, and accuracy. We believe that garbled circuits is the best approach for this scenario as it allows for direct communication between two parties and models research settings well. Further, garbled circuits can handle a variety of adversaries beyond the semi-honest one that we consider in this work.

3 Implementation

In this section, we describe how we implemented metagenomic analyses in garbled circuits and detail an evaluation of our system.

3.1 Metagenomics using garbled circuits

3.1.1 FlexSC

FlexSC, the back end of ObliVM, is a framework for secure computation including garbled circuits with a semi-honest adversary (Liu *et al*., 2015). FlexSC allows users to write a function in Java for two parties to compute then compiles and evaluates the garbled circuit representation of that function. We implemented all metagenomic tests as Java packages then compiled and ran each with FlexSC. Our initial work on χ^2-test was based on a χ^2-test implementation using SNP data (https://github.com/wangxiao1254/idash_competition).

3.1.2 Metagenomic analysis assumptions

For this article, we perform all analyses at the species taxonomic level. As detailed in Supplementary Materials Section S1, OTUs are generated from direct pairwise comparison of sequencing reads.

This is a compute-intensive process when performed on clear text (Ghodsi *et al.*, 2011). We do not attempt it in SMC for this work and assume each party performs this operation locally. We assume that each party will annotate each resulting OTU by matching to a common reference database, previously agreed upon by both parties (note that this reference database is orthogonal to sample-specific sequencing results obtained by each party). For illustration we assume that the agreed upon reference database yields annotation at the microbial species level. We also assume that parties can split data into case and control groups based on an agreed upon phenotype. Finally, we do not consider features that have all zeros in the case or control group for either party.

3.2 Design approaches

We took several approaches to implement each statistic. Since the metagenomic datasets we examined are at least 80% sparse and this trend is expected with OTU data (Paulson *et al.*, 2013), we make design choices to make computation with garbled circuits feasible. We now detail each implementation of the χ^2 -test, odds ratio, Differential Abundance and Alpha Diversity. To measure the impact of our design choices we implemented a naive algorithm for each statistic and compared results.

3.2.1 Precomputation

We first developed a method that finds an aggregate statistic at each party so that only those values are circuit inputs. This method is a straightforward approach to reduce the amount of operations and data in the secure computation protocol. As expected, for each statistic this approach had the best performance on all the datasets we evaluated. Supplementary Figure S2 shows the process for calculating a χ^2-test and odds ratio on precomputed contingency table counts. An issue with this approach is not all analyses that researchers are interested in computing may be able to be performed over locally generated aggregates.

3.2.2 Sparse matrix

We devised two methods to account for the sparsity of the feature count matrices we used for evaluation. We first followed an approach introduced by Nikolaenko *et al.* (2013) to perform sparse matrix factorization in garbled circuits. We detail our work with this technique in the Supplementary Materials Section S4. As our contribution, we took a conceptually simpler approach that input the non-zero elements for each feature to the circuit and operated over those elements directly. As shown in Figures 4 and 5, this method significantly reduces the number of operations that need to be performed in the secure protocol and offers reasonable running times compared to the precomputation approach.

3.2.3 Presence/absence

We implemented the χ^2-test and odds ratio to perform presence/absence association testing. We provide a review of χ^2-test and odds ratio in Supplementary Materials Section S1.

For the precomputation technique, each party splits its data into case and control groups on a characteristic determined outside of this protocol. Each party then locally computes the contingency table counts on the split data. These contingency table counts are each party's input into the circuit. Within the circuit, the counts are summed for both case and control groups then the χ^2-statistic along with the odds ratio are computed for each feature.

In the sparse matrix approach, the total number of samples and all non-zero elements for each feature are input to a garbled circuit. The circuit first adds the number of non-zero elements to compute the present contingency table counts then uses the total number of samples to find the absent counts.

3.2.4 Differential abundance

For calculating differential abundance, we implemented a two-sample *t*-test for testing the mean abundance between case and control groups. We assume normalization of sequencing counts can be accomplished in a preprocessing step between both parties. We make this assumption because we use normalized datasets in our evaluation. We leave implementation of normalization techniques in garbled circuits to future work.

For review of two-sample *t*-test we refer the reader to the Supplementary Materials Section S1. We examined the process for calculating mean, variance and the t-statistic to determine what optimizations can be made for computing in a circuit. In order to avoid processing all samples within the computation framework, we observe transformations that reduce the total number of operations. In the Supplementary Materials, we show how mean abundance and variance can be computed using the sum, sum of squares and total number of elements from each party. For precomputation, as each institution only needs to provide three values per feature we calculate them locally. In the circuit, a two-sample *t*-statistic to test difference between case and control groups is computed.

For the sparse matrix approach, the total sum and sum of squares are calculated in the circuit using the non-zero elements for each feature. Mean abundance along with variance can then be calculated and used compute the two-sample *t*-test. We refer the reader to Supplementary Materials Section S4 for more detail.

3.2.5 Alpha diversity

We use a two-sample *t*-test to determine the significance of mean Alpha Diversity difference between case and control groups. Given that FlexSC does not currently compute logarithm, we measure Alpha Diversity as Simpson's index: $D = \frac{\sum n(n-1)}{N(N-1)}$ where n is the number of OTU counts for OTU$_i$ and N is the total number of counts observed in a sample.

For precomputation, we locally compute Simpson's index for each sample. These values are input into the circuit where they are summed, mean and variance is taken, and the *t*-statistic is calculated. In Alpha Diversity, all samples in case and control must be processed together as opposed to Presence/Absence and Differential Abundance which can be computed per feature.

For our sparse computation design, the two values for Simpson's index, $\sum n(n-1)$ and $N(N-1)$ are generated over each sample in the circuit during one pass through the matrix. Then a pass over an array of these values using division yields Simpson's index from which the total sum and sum of squares can be used to compute the two-sample *t*-test between case and control groups.

3.3 Evaluation

We evaluated our implementation using two Amazon EC2 r3.2xLarge instances with 2.5 GHz processors and 61 GB RAM running Amazon Linux AMI 2015.3. We measured the size of the circuit generated, running time and network traffic between both parties for each metagenomic statistic and dataset. Circuit size serves as a useful comparison metric since it depends on the function and input sizes but is independent of hardware. Running time and network traffic are helpful in system-design decisions and benchmarking of deployments.

3.4 Datasets

We used OTU count data from the Personal Genome Project (PGP) (Church, 2005), the HMP (Turnbaugh *et al.*, 2007), and the Global Enterics MSD (Pop *et al.*, 2014). We retrieved the MSD data from the project website (ftp://ftp.cbcb.umd.edu/pub/data/GEMS/MSD1000.biom) as well as the PGP and HMP datasets are from the American-Gut project site (https://github.com/biocore/American-Gut/tree/master/data) (Blaser *et al.*, 2013). We used the tongue as the case and gingiva as control for the HMP data. For PGP, we set forehead as case and left palm as control. Case and control criteria for the MSD dataset were already set by the researchers that publish the data depending on disease phenotype. After aggregating to species and removing features which hold all zeros for either the case or the control group, the PGP contains 168 samples and 277 microbiome features, the HMP has 694 samples and 97 features, and the MSD dataset consists of 992 samples and 754 features. Supplementary Table S2 summarizes the size and sparsity of each dataset.

3.5 Efficiency of secure computation

3.5.1 Circuit size

Figure 2 shows the circuit size per feature for each experiment. As a result of the work by Kolesnikov and Schnieder (2008), XOR gates in each circuit do not require costly network traffic and computation, therefore the total number of non-XOR gates is reported for each statistic and dataset. Using precomputation, the complexity of the equation in terms of arithmetic operations to calculate each statistic determines the circuit size. This explains the circuit sizes for odds ratio and χ^2 test as compared with Differential Abundance. For Alpha Diversity, all rows and columns are preprocessed with only the two sample *t*-test computed in the circuit. With the sparse implementation, the complexity of the test along with the number of non-zero elements in the dataset directly affects circuit size.

3.5.2 Running time

For the sparse implementation, the running time was proportional to the size and number of non-zero elements in each dataset. For precomputation, Alpha Diversity was affected by the number of samples in each dataset. The running time for the χ^2 test, odds ratio, and Differential Abundance were proportional to the number of features (rows) processed. Figure 3 summarizes the effects of input size and algorithm complexity on running time.

3.5.3 Network traffic

Supplementary Table S5 shows the network traffic for each experiment. The increase in network traffic between the precomputation and sparse implementations is more significant than the differences in running times of those approaches. We believe that the network traffic for the precompute implementation is quite good for the security guarantees provided with using garbled circuits while the sparse approach presents an acceptable tradeoff depending on the network resources available.

3.6 Accuracy

We compared the accuracy of our implementation results to computing the statistic using standard R libraries. Table 1 lists the accuracy of results for the χ^2 statistic, odds ratio, as well as the *t*-test results for Differential Abundance and Alpha Diversity. The differences in our garbled circuits results compared to the *R* values appear to be the result of circuit complexity. The floating-point arithmetic operations in FlexSC are software implementations. Therefore the operations are subject to rounding errors that are rarely observed on modern processors which have hardware level support for floating-point arithmetic.

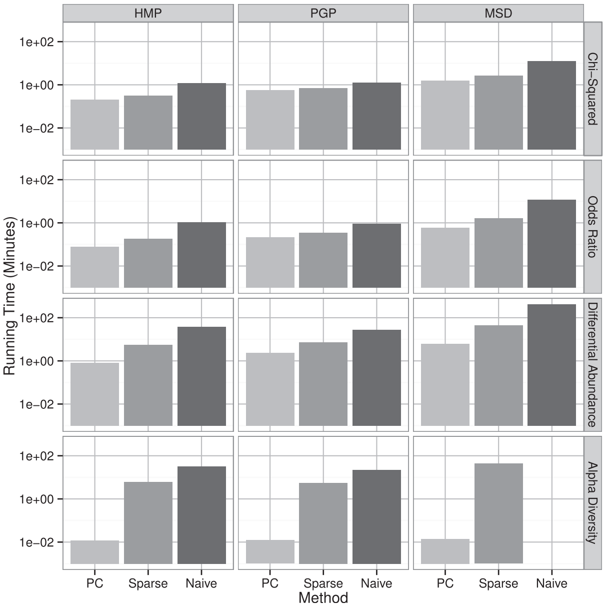


Fig. 3. Running time for each statistic and each dataset in minutes. In each statistic, the number of arithmetic operations determined the running time. The size of the dataset along with sparsity contributed to running time for the sparse implementations. Alpha Diversity MSD Naive did not run to completion on the EC2 instance size due to insufficient memory. Based on the circuit size and the number of gates processed per second for other statistics, we estimate the running time to be 378 min. PC, Pre-compute

Table 1. Computation accuracy

	PGP	HMP	MSD
Chi-square statistic	7.84e-07	7.48e-06	7.02e-08
Chi-square *P*-value	2.00e-07	2.14e-06	9.72e-08
odds ratio	1.60e-13	5.42e-13	2.44e-13
Differential abundance *t*-statistic	0.023	0.0017	0.0012
Differential abundance degrees of freedom	2.7e-4	2.5e-4	0.0028
Differential abundance *P*-value	0.0024	0.0026	0.0011
Alpha Diversity *t*-statistic	0.0038	0.017	0.0049
Alpha Diversity degrees of freedom	1.48e-05	9.7e-4	2.2e-4
Alpha Diversity *P*-value	0.0088	0.044	0.014

Results were generated using the R chisq.test{stats}, odds.ratio{abd}, t.test{stats}, and diversity{vegan} against our implementation in ObliVM for the χ^2-test, odds ratio, differential abundance and Alpha Diversity. We use Normalized Mean Squared Error: $\|x-y\|^2/\|x\|^2$ with x as the value output by R and y the value from our implementation. For comparing *P*-values, we use the $\log_{10}$ *P*-value and exclude any exact matches [since $\log_{10}(0) = -\text{Inf}$ in R] while computing the mean.

We investigated if our implementation yielded any false positives and false negatives with the results from R acting as ground truth. For the *P*-values of Differential Abundance in PGP, HMP, and MSD datasets we found no false positives or false negatives for a significance level of 0.05.

Table 2. Significant features found from sharing data between each country

	Features found	Total increase
Kenya only	47	N/A
Gambia only	84	N/A
Mali only	58	N/A
Bangladesh only	75	N/A
Kenya + The Gambia	133	86
Kenya + Mali	112	65
Kenya + Bangladesh	138	91
Gambia + Bangladesh	166	82
Mali + Gambia	167	109
Mali + Bangladesh	169	111

When computing data with another policy domain, each country saw an increase in the number of features detected to be significantly different between case and control groups.

3.7 Significant features discovered through data-sharing

Researchers in different policy domains may be forced to compute analyses on partial data. We measured the effect of using our implementation for data-sharing between policy domains. The MSD dataset provides a means to simulate secure computation of microbiome analyses between different countries. The data were gathered from Kenya, The Gambia, Bangladesh and Mali. We simulate each country performing secure Differential Abundance pair-wise with the other countries. We observed that sharing data resulted in a substantial increase (at minimum a 98% increase) in the number of species found to be differentially abundant between case and control groups. Table 2 summarizes the results.

3.8 Metagenomic codes

We also evaluated our implementation on the genetic marker data that showed the greatest identification power in the metagenomic codes analysis (Franzosa *et al.*, 2015). The data are also from the HMP and consists of a total of 85 samples and 221 111 features. Due to the large number of features and sparsity of the data, we implemented a filtering garbled circuit in which we first return a vector to each party denoting if a given feature meets a presence cutoff and then have each party input those features into our existing implementations to compute the statistical test. For χ^2, the 1 729 851 751 gate circuit (circuit size of 7823 Non-Free gates per feature) is evaluated in 67.4 min, with 51 926.35 MB sent to the evaluator, and 1 642.53 MB sent to generator. For odds ratio, the 632 918 505 gate circuit is evaluated in 33.18 min, with 20,542.84 MB sent to the evaluator, and 1,642.29 MB sent to generator. This result shows that the secure comparative analyses we would like to perform are possible given the legitimate concerns raised by Franzosa *et al*.

4 Discussion

In this section, we describe related work and provide a context for our contribution. We also discuss a use case for our solution in building datasets and finally present conclusions we formed during the course of our work.

4.1 Related work

As we are the first, to our knowledge, to approach secure microbiome analysis, we review related work on privacy-preserving operations over human DNA.

4.1.1 Secure DNA sequence matching and searching

Comparing two DNA segments is essential to genome alignment and identifying the presence of a disease causing mutation. One approach is to use an oblivious finite state machine for privacy-preserving approximate string matching (Troncoso-Pastoriza *et al.*, 2007). FastGC, the predecessor of the FlexSC library, was benchmarked by computing Levenstein distance and the Smith-Waterman algorithm between private strings held by two parties (Huang *et al.*, 2011). More recently, Wang *et al.* (2015) compute approximate edit-distance using whole genome sequences.

4.1.2 Privacy-preserving Genome-wide association studies

Prior work has shown that secure computation between two institutions on biomedical data is possible by using a three-party secret-sharing scheme (Kamm *et al.*, 2013). The authors present an implementation of a χ^2-test over SNP data using the Sharemind framework. Other researchers have presented a modification of functional encryption that enables a person to provide her genome and phenotype to a study but only for a restricted set of functions based on a policy parameter (Naveed *et al.*, 2014).

Prior works have built systems for genomic studies using different cryptographic protocols, including systems using additive homomorphic encryption (Dachman-Soled *et al.*, 2011) and systems using fully homomorphic encryption (Lauter *et al.*, 2014). When compared with these works, we use a garbled circuit protocol with circuits for floating-point operations. Our system has two unique advantages compared to these prior works: (i) We can benefit from a long line of work on improving the practicality of garbled circuits (Huang *et al.*, 2011; Kolesnikov and Schneider, 2008; Zahur *et al.*, 2015) (ii) Floating-point operations ensure us a small and bounded error even after multiple operations.

4.1.3 Secure genetic testing

For using sequencing results in the clinical realm, paternity determination and patient-matching is possible using private set intersection (Baldi *et al.*, 2011). Also, it is feasible to utilize homomorphic encryption for implementing disease-risk calculation without revealing the value of any genomic variant (Ayday *et al.*, 2013).

4.2 Patient pool

A novel application of multi-party secure computation approaches to genomic analysis are patient pool designs that can benefit patient groups, specifically those suffering from rare diseases or those with insufficient data in existing repositories for association studies. The recent announcement by 23andMe to begin drug development on its genome variant datasets highlights the value of biomarker data. We imagine a scenario where individuals can use our solution to create and manage datasets in order to charge drug developers to run analysis functions over the data. The companies will have to be non-colluding as otherwise all function results could be shared among companies. The current regulatory process for drug development allows a mechanism to enforce this constraint.

The patient pool can be paid to compute a function to over its data and sign the output. Upon requesting drug trial permission in the USA, a company is required to hand over all data from research, which in this case would include the output of the patient pool analysis and signatures over those results. The FDA could verify the signatures to enforce non-collusion between companies. This provides a mechanism to create high-quality datasets that are accessible to a variety of companies and ensure patients are compensated for their efforts.

5 Conclusions

In this article, we show that it is possible to perform metagenomic analyses in a secure computation framework. Our implementation made use of precomputation steps to minimize the number of operations performed in secure computation making the use of garbled circuits feasible. We also implemented sparse-matrix methods for each statistic. We took this step in order to prove the applicability of this solution for other analyses when the data itself acts as sufficient statistics, such as for the Wilcoxon rank-sum test. We also explored potential applications of our implementation in patient pool designs.

Although the storage and sharing of medical data is ultimately a policy matter, providing a technical solution is useful to forming good policy. We believe that given the time costs associated with re-consenting patients to release data to another researcher or creating a legal contract stipulating a data receiver's responsibility, that the running times we presented for metagenomic analyses are a reasonable tradeoff.

DNA-sequencing technologies are entering a period of unprecedented applicability in clinical and medical settings with a concomitant need for regulatory oversight over each individual's sequencing data. We believe that addressing privacy concerns through computational frameworks similar to those used in this article is paramount for patients while allowing researchers to have access to the largest and most descriptive datasets possible. We expect that secure computation and storage of DNA sequencing data, both the individual's DNA and their metagenomic DNA, will play an increasingly important role in the biomedical research and clinical practice landscape.

Funding

This work was partially supported by the National Institutes of Health [R01HG005220 and R01GM114267 to H.C.B. and J.W.]; the US National Science Foundation Graduate Research Fellowship [DGE0750616 to J.N.P.]; and the US National Science Foundation [1111599 to X.W., 1314857 to X.W., 0964541 to B.B.].

Conflict of Interest: none declared.

References

Ayday,E. *et al.* (2013). Privacy-preserving computation of disease risk by using genomic, clinical, and environmental data. In: *Proceedings of the 2013 USENIX Conference on Safety, Security, Privacy and Interoperability of Health Information Technologies, HealthTech '13*, Berkeley, CA, USA. USENIX Association, p. 1.

Baldi,P. *et al.* (2011). Countering gattaca: efficient and secure testing of fully-sequenced human genomes. In: *Proceedings of the 18th ACM Conference on Computer and Communications Security, CCS '11*, New York, NY, USA. ACM, pp. 691–702.

Blaser,M. *et al.* (2013) The microbiome explored: recent insights and future challenges. *Nat. Rev. Microbiol*, **11**, 213–7.

Church,G.M. (2005) The personal genome project. *Mol. Syst. Biol.*, **1**, 2005.0030

Dachman-Soled,D. *et al.* (2011). Secure efficient multiparty computing of multivariate polynomials and applications. In: *Applied Cryptography and Network Security*, pp. 130–46. Springer.

Fierer,N. *et al.* (2010) Forensic identification using skin bacterial communities. *Proc. Natl. Acad. Sci. USA*, **107**, 6477–81.

Franzosa,E. *et al.* (2015) Identifying personal microbiomes using metagenomic codes. *Proc. Natl. Acad. Sci. USA*, **112**, E2930–8.

Fredrikson,M. *et al.* (2014). Privacy in pharmacogenetics: an end-to-end case study of personalized warfarin dosing. In: *23rd USENIX Security Symposium (USENIX Security 14)*, USENIX Association, pp. 17–32.

Ghodsi,M. *et al.* (2011) Dnaclust: accurate and efficient clustering of phylogenetic marker genes. *BMC Bioinformatics*, **12**, 271.

Groce,A.D. (2014). *New notions and mechanisms for statistical privacy*. Ph.D. Thesis, University of Maryland. ProQuest. 3644113.

Huang,Y. *et al.* (2011). Faster secure two-party computation using garbled circuits. In: *Proceedings of the 20th USENIX Conference on Security, SEC'11*, Berkeley, CA, USA. USENIX Association, pp. 35–35.

Kamm,L. *et al.* (2013) A new way to protect privacy in large-scale genome-wide association studies. *Bioinformatics*, **29**, 886–93.

Kolesnikov,V. and Schneider,T. (2008) Improved garbled circuit: free XOR gates and applications. In: *Automata, Languages and Programming*, pp. 486–98. Springer.

Lauter,K. *et al.* (2014) Private computation on encrypted genomic data. In: *Progress in Cryptology-LATINCRYPT 2014*, pp. 3–27. Springer.

Liu,C. *et al.* (2015) Oblivm: A programming framework for secure computation. In: *2015 IEEE Symposium on Security and Privacy (SP)*, IEEE, pp. 359–376.

Mailman,M.D. *et al.* (2007) The NCBI dbgap database of genotypes and phenotypes. *Nat. Genet.*, **30**, 1181–86.

Malkhi,D. *et al.* (2004) Fairplay-secure two-party computation system. In: *Proceedings of the 13th Conference on USENIX Security Symposium, SSYM'04, Berkeley*, CA, USA. USENIX Association, p. 20.

Naveed,M. *et al.* (2014). Controlled functional encryption. In: *Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security, CCS '14*, New York, NY, USA. ACM, pp. 1280–1291.

Nikolaenko,V. *et al.* (2013). Privacy-preserving matrix factorization. In: *Proceedings of the 2013 ACM SIGSAC Conference on Computer and Communications Security, CCS '13*, New York, NY, USA. ACM, pp. 801–812.

Paulson,J.N. *et al.* (2013) Differential abundance analysis for microbial marker-gene surveys. *Nat. Methods*, **10**, 1200–2.

Pop,M. *et al.* (2014) Diarrhea in young children from low-income countries leads to large-scale alterations in intestinal microbiota composition. *Genome Biol.*, **15**, R76.

Troncoso-Pastoriza,J.R. *et al.* (2007). Privacy preserving error resilient DNA searching through oblivious automata. In: *Proceedings of the 14th ACM Conference on Computer and Communications Security, CCS '07*, New York, NY, USA. ACM, pp. 519–528.

Turnbaugh,P.J. *et al.* (2007). The human microbiome project: exploring the microbial part of ourselves in a changing world. *Nature*, **449**, 801–10.

Turnbaugh,P.J. *et al.* (2009) A core gut microbiome in obese and lean twins. *Nature*, **457**, 480–4.

Wang,X.S. *et al.* (2015). Efficient genome-wide, privacy-preserving similar patient query based on private edit distance. In: *Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security*, CCS '15, pp. 492–503.

Zahur,S. *et al.* (2015). Two halves make a whole. In: *Advances in Cryptology-EUROCRYPT 2015*, pp. 220–50. Springer.