Supplementary Note: Privacy-Preserving Microbiome Analysis
Using Secure Computation

Justin Wagner, Joseph N. Paulson, Xiao Wang, Bobby Bhattacharjee, and Héctor Corrada Bravo*

December 4, 2015

Contents

1 Microbiome Preliminaries 1
   1.1 Microbiome Sequencing ............................................. 2
   1.2 Metagenomic Statistics ................................................ 3

2 Problem Overview 4
   2.1 Forensic Identification .............................................. 4
   2.2 Identification with Metagenomic Codes ............................. 4
   2.3 Genetic Re-identification Attacks ................................... 5

3 Oblivious Transfer 6

4 Implementation 6
   4.1 Pre-computation approach ........................................... 6
   4.2 Sparse matrix approaches ............................................ 6
   4.3 Presence/Absence ..................................................... 6
   4.4 Differential Abundance ............................................... 7
   4.5 Alpha Diversity ...................................................... 8
   4.6 Asymptotic Complexity ............................................... 8

5 Evaluation 9
   5.1 Datasets .............................................................. 9
   5.2 Running Times ....................................................... 9
   5.3 Circuit Sizes ......................................................... 9
   5.4 Network Traffic ..................................................... 10

1 Microbiome Preliminaries

In this section, we provide a background on microbiome sequencing and detail the statistics used in standard metagenomic association analyses.

*to whom correspondence should be addressed
1.1 Microbiome Sequencing

Human microbiome sequencing is carried out in the following steps: 1) A microbial community sample is collected from a bodysite such as the mouth, skin, or gut. 2) DNA is extracted from the sample. 3) The 16s rRNA gene is isolated and sequenced. All bacterial cells which are the same will contain an exact copy of the 16s RNA gene. 4) Sequences that are similar above a threshold (95, 97, or 99 percent similarity) are clustered into an Operational Taxonomic Unit (OTU) 5) OTUs are annotated through comparison to an existing microbial annotation database, 6) the number of times a given OTU is observed for each sample is computed into a count table that serves as the main object of subsequent downstream analysis. Supplementary Figure 1 shows the microbiome sequencing pipeline in more detail. The basic measurement features for metagenomics are OTUs, which are annotated corresponding to specific microbial species or strains.

Supplementary Figure 1: Microbiome Sequencing and Metagenomic Analysis Pipeline

To determine the association between microbiota and certain phenotypes\textsuperscript{1}, multiple statistics are computed from these OTUs: for instance, the presence or absence of a specific OTU across samples with a given phenotype; the abundance or quantity of an OTU across samples with a given phenotype (Paulson \textit{et al.}, 2013); the diversity or the number of distinct OTUs in a sample; and the distribution of OTU abundances in a sample. Each of these statistics reveal a distinct view of the role microbial communities play in healthy and disease individuals. In addition, all of these association statistics can be computed at any level of the OTU taxonomy. In this sense, the data used in microbiome association studies are much richer than the sets of genotypes used to describe an individual in human DNA analysis.

\textsuperscript{1}Physical expression of an individual’s traits.
1.2 Metagenomic Statistics

In this section, we define precisely the statistical measures mentioned in Section 3.3. These are standard statistics in the metagenomic field and we detail them here for completeness. A more thorough review is available by Morgan et al. (Morgan and Huttenhower, 2012). In this paper, we denote a metagenomic dataset \( M \) where \( M_{ij} \) contains the OTU read counts for feature \( i \) in sample \( j \). For each sample \( j \), an entry in a separate database \( D \) contains information regarding its physical characteristics and disease status. Each statistic provides a mechanism to identify associations between groups in \( D \) and trends in \( M \).

**Presence or Absence of an OTU** Identifying the role of an OTU first requires a comparison of presence or absence of that OTU in disease and non-disease groups. A \( \chi^2 \) test is performed to determine the significance of an observed difference in the presence or absence of an OTU between groups. The odds ratio is another measure of association between presence or absence of an OTU and a specific phenotype.

A 2x2 contingency table is populated to compute the \( \chi^2 \) test on exposure to an OTU. The contingency table counts will be calculated from \( M_{ij} \) by first creating a new matrix, Present, as follows: \( \text{Present}_{ij} = M_{ij} > 0 \) : 1; 0. OTU \( i \) Present is the sum of 1s for OTU \( i \) and OTU \( i \) Absent is the sum of 0s.

|       | OTU\(_i\) Present | OTU\(_i\) Absent |
|-------|-------------------|------------------|
| Case  | \( a \)           | \( c \)          |
| Control | \( b \)          | \( d \)          |

The \( \chi^2 \) statistic is calculated as:

\[
\chi^2 = \frac{(a + b + c + d)(bc - ad)^2}{(a + b)(a + c)(b + d)(c + d)}
\]

with one degree of freedom. The odds ratio describing the association between OTU exposure and case or control membership, is defined as:

\[
OR = \frac{ad}{cb}
\]

**Differential Abundance** An OTU may be present in both disease and non-disease groups, but its abundance level may differ between the two groups. Computing differential abundance requires calculating the mean and variance over the counts of a given OTU for each of the two groups to compare (White et al., 2009).

Mean Abundance:

\[
\bar{x}_i = \frac{1}{n} \sum_{j=1}^{j} M_{ij}
\]

Variance:

\[
s_i^2 = \frac{1}{n-1} \sum_{j=1}^{j} (M_{ij} - \bar{x}_i)^2
\]

A two-sample \( t \)-statistic is used to test difference between case and control groups.

\[
t = \frac{\bar{x}_{\text{Feature, case}} - \bar{x}_{\text{Feature, control}}}{\sqrt{\frac{s_i^2}{n_{\text{case}}} + \frac{s_i^2}{n_{\text{control}}}}}
\]
**Alpha Diversity**  While the presence or abundance of specific OTUs may not be associated with disease, differences in microbial community structure as a whole may be associated with disease. Alpha Diversity is commonly used as a statistic to measure the evenness and richness of microbial communities. It is usually computed based on the entropy of the OTU distribution for a single sample (as Shannon’s Index: \( H_j = -\sum_{i} p_{ij} \ln(p_{ij}) \), where \( p_{ij} \) is the fraction of total OTU counts comprised by OTU \( i \) in sample \( j \). Another Alpha Diversity measure is Simpson’s index which is of the form \( D = \sum_{i} p_{ij}^2 \). A two-sample \( t \)-statistic is computed to test the significance of differences in statistics \( H \) or \( D \) between groups.

**Beta Diversity**  The distance of an entire microbiome community structure to that of another sample is the last metagenomic statistic that we will discuss. Beta Diversity is commonly supplied as a check of intra-individual community distance is less than that of inter-individual distance for a specific body site. It is commonly computed as Bray-Curtis dissimilarity \( BC_{ij} = 1 - (2C_{ij} / (S_i - S_j)) \) where \( C_{ij} \) denotes the sum of the counts of species observed at both sites \( i \) and \( j \) while \( S_i \) and \( S_j \) are the total number of species observed at sites \( i \) and \( j \). Another metric for Beta Diversity is UniFrac which builds a phylogenetic tree across samples under study and then computes a pair-wise distance between two samples to determine if two samples are from the same source. Unweighted UniFrac uses presence/absence of an OTU while weighted UniFrac takes in account the abundance of an OTU and weights branch lengths accordingly (Lozupone and Knight, 2005).

2 Problem Overview

In this section we describe the privacy threats of microbiome data and annotate them according to an existing categorization of genome privacy risks. We provide a comprehensive review of microbiome sequencing and metagenomics in the Supplementary Note, Section 1.

2.1 Forensic Identification

One prominent study proved that a person’s hand bacteria can identify objects that individual touched (Fierer et al., 2010). The authors first show the bacteria left after touching a keyboard are separate and unique between individuals. To measure the stability of the bacterial community left behind on the keyboards, the authors compared sequencing results for keyboard samples from the same person stored for 3 to 14 days at -20 degrees C and room temperature. The community makeup for each sample was not significantly different between any sample storage method. Next the authors calculated the UniFrac distance in community membership between keyboard samples from nine people and a database of microbiome samples from 270 individual’s hands. The closest match for each sample was the individual who touched the keyboard. This study was the first to show the identification power of an individual’s microbiome signature.

2.2 Identification with Metagenomic Codes

A recent analysis showed that metagenomic data alone can uniquely identify individuals in the Human Microbiome Project dataset (Franzosa et al., 2015). The authors build minimal hitting sets to find a collection of microbiome features that are unique to each individual compared to all others in a dataset. The minimal hitting set algorithm was built using four types of features - OTUs, species, genetic markers, and thousand base windows matching reference genomes. The authors use a greedy algorithm and prioritize features by abundance gap, the difference in abundance between a feature in one sample compared to all other samples. The authors called these sets of features ”metagenomic codes” and used the codes built at the first time point in the Human Microbiome Project dataset to match individuals at a second time point. The genetic marker and base window codes were the best identifiers between the two time points. The OTU and species level codes also identified individuals but had a higher false-positive rate. As the authors note, the discovery of an identifiable microbiome fingerprint substantially changes the considerations for publicly releasing human microbiome data.
2.3 Genetic Re-identification Attacks

Through detailing attacks on genetic datasets, a recent article provided a categorization of techniques to breach participant privacy (Erlich and Narayanan, 2014). The attacks fall into several areas: Identity Tracing defined as determining the identity of an anonymized DNA sample using non-private attributes, Attribute Disclosure which uses a piece of identified DNA to discover phenotypes or activities in other protected databases, and Completion Attacks that use genotype imputation to uncover data that has been removed upon publication of a DNA sequence. To provide a complete overview of microbiome privacy risks, we detail each attack and then expand the categorization to include microbiome specific attacks.

Identity Tracing With Metadata reveals the identity of an anonymized DNA sample by using metadata such as age, pedigree information, geography, sex, ethnicity, and health condition. This attack is a concern with metagenomic comparative analysis as case and control group membership is determined by considering metadata.

Genealogical Triangulation uses genetic genealogy databases which link genealogical information, such as surname, with genetic material to allow an individual to recover ancestral information from his/her own DNA. This attack should not be a concern with microbiome data as microbiome inheritance has not been fully determined.

The microbiome presents three different methods for triangulation of a sample’s identity which we term Location Triangulation, Behavior Tracing, and Rare Disease OTU. As evidence of the first, a recent study detailed the similarity between individuals that occupy the same dwelling (Lax et al., 2014). Therefore, an attacker may be able to reveal the identity of an individual microbiome sample by computing similarity with a sample taken from a specific location.

Further, Behavior Tracing could be used to identify a microbiome. The oral microbiota of romantic partners is more similar than other individuals and it is possible to measure how long the similarity between kissing partners is maintained (Kort et al., 2014). An attack could be mounted using the phylogenetic or feature-level distance between a known person and the sample from a suspected romantic partner.

Rare Disease Feature Tracing takes advantage of attributes of public health disease tracking and microbial disease infections. Some infections, such as antibiotic-resistant cases, are recorded by state health departments and a single microbiome feature could correspond to those infections. If an attacker is able to observe the known microbiome feature of individual in a public health database and use it to link between another dataset, this will reveal any corresponding sensitive attribute.

Identity Tracing by Phenotypic Prediction involves predicting phenotypic information from genotypic information and then using that to match to an individual. Phenotypic prediction with human DNA is quite difficult given that predictions are not currently robust for unique identifiers in the population. For identifiers such as height, weight, and age, the effectiveness of this attack is likely to be low with microbiome data.

Identity Tracing by Side Channel Leaks is possible when an identifier is apparent from the dataset entries either by data preparation techniques or data-id assignment. One example is that Personal Genome Project sequencing files which by default were named with patient first and last name included. This attack is a concern with microbiome sequencing as well given that file uploading of the Personal Genome Project is similar for microbiome results.

Attribute Disclosure With N=1 entails an attacker associating an individual’s identity to a piece of DNA and that piece of DNA to a sensitive attribute, such as an element in a database of drug users. For microbiome data, the forensic identification and the metagenomic codes techniques could be used by an attacker to successfully query a dataset with a sensitive attribute.

Attribute Disclosure from Summary Statistics uses genetic information of one victim and published summary statistics from a case/control study to determine if the victim’s DNA is biased towards the distribution of either the case or control group. If group membership can be determined, then the criteria to split groups (such as disease status) is revealed to the attacker. Linkage disequilibrium, or the probability that portions of DNA are more likely to be inherited together than others, provides a mechanism to increase the power of

---

\(^2\)We use the names for each attack as introduced by Erlich and Narayanan.
the attack. Further, genealogical information can be used to accomplish attribute disclosure.

While the authors cite Attribute Disclosure from Summary Statistics as an attack possible with all 'omic' data, linkage disequilibrium and genealogical triangulation are not applicable to microbiome sequencing. The release of summary statistics may be used to determine if a metagenomic code for an individual is present in a case/control group, but the probability of this attack needs be determined.

Completion Attacks reveal portions of a DNA sample that are not released publicly by using linkage disequilibrium to uncover the hidden SNPs. Genealogical information, such as a pedigree and the SNPs of relatives, can also be used in genotype imputation. For metagenomic data, a cohabitation mapping of individuals from the same household to distinct features could be used to mount a completion attack.

3 Oblivious Transfer

Oblivious Transfer is a subroutine that allows one party known as the sender (P1) to offer two messages and for the other party, referred to as the receiver (P2), to input a bit selecting one of the messages. Oblivious Transfer guarantees that the sender learns nothing about the receiver's selection and the receiver learns nothing about the other inputs beyond the one selected. In the semi-honest setting, one approach to Oblivious Transfer is for the receiver to generate two public-private key-pairs but with one of the public keys to not have a valid private key. The receiver then sends both keys to the sender, who encrypts its inputs with the public keys and sends them to the receiver. The receiver will only be able to properly decrypt one of the ciphertexts (Snyder, 2014).

4 Implementation

In this section we provide details on each implementation approach. In the pre-computation approach, we compute over values that are locally computed by each party. In the sparse matrix approach, we operate on the non-zero elements from each party directly.

4.1 Pre-computation approach

This method is a straightforward approach to reduce the amount of operations and data in the circuit. Supplementary Figure 2 shows the process for calculating a $\chi^2$-test and odds ratio on pre-computed contingency table counts.

4.2 Sparse matrix approaches

The main idea of the technique from Nikolaenko et al. is to create a counting circuit using Bitonic Sort, a sorting algorithm that can be implemented as an oblivious circuit with $O(n \log^2(n))$ running time, to operate over tuples consisting of (row, column, matrix element) (Nikolaenko et al., 2013). Supplementary Figure 3 shows the scheme in greater depth. We use the counting mechanism to implement each statistic. Surprisingly, the sorting operation for this approach outweighed the naive approach for chi-square and odds ratio. For completeness, we provide a description of this approach for implementing each statistical test.

4.3 Presence/Absence

Sparse Computation. We assume that parties first locally split their datasets on case and control criteria. For the scheme described in Supplementary Figure 4, each party will then only input the non-zero elements of the respective case and control matrices as tuples. For the $\chi^2$ test and odds ratio, the counter can be used to find contingency table counts. The oblivious counter will be used to calculate OTU, Present for each group. The number of samples for each parties case and control groups is shared obliviously and OTU, Absent can be calculated. With $a, b, c, d$, $\chi^2$ can be calculated as in Equation (1) and odds ratio as Equation (2).
Supplementary Figure 2: Diagram of Pre-computation for Presence/Absence. The inputs to the garbled circuit are locally generated row sums from each party. Presence/Absence \( \chi^2 \)-test statistic and Odds Ratio are calculated in the circuit.

4.4 Differential Abundance

For calculating differential abundance, the sequencing counts need to be normalized and that can be accomplished per sample in the pre-computation phase. We examine Equations (3) and (4) to determine what optimizations can be accomplished for computing in secure computation. In order to avoid processing all samples within the computation framework, we observe transformations that reduce the total number of operations. With \( k \) for party 1 and party 2 the following can be computed:

Mean Abundance

\[
\bar{\pi}_{OTU_{i,k}} = \frac{1}{n_{k_1} + n_{k_2}} \left( \sum_{j}^{n_{k_1}} M_{ij} + \sum_{j}^{n_{k_2}} M_{ij} \right)
\]

Variance

\[
s^2_{OTU_{i,k}} = \frac{\sum^{n_{k_1}}_{j} M_{ij}^2 + \sum^{n_{k_2}}_{j} M_{ij}^2 + \left( \sum_{j}^{n_{k_1}} M_{ij} + \sum_{j}^{n_{k_2}} M_{ij} \right)^2}{n_{k_1} + n_{k_2}}
\]

Sparse Computation. With Nikolanko et al. sparse matrix approach, the oblivious counter is used to calculate the total sum and augmented to compute the sum of squares for each feature. Then a two-sample
t-test can be performed using those values as described in Equations (6) and (7).

4.5 Alpha Diversity

We compute Alpha Diversity for each sample, then use a two-sample t-test to determine the significance of a difference between case and control groups. Given that ObliVM does not currently compute logarithm, we measure Alpha Diversity as Simpson’s index: 

\[ D = \sum \frac{n(n-1)}{N(N-1)} \]

where \( n \) is the number of OTU counts for OTU\(_i\) and \( N \) is the total number of counts observed in a sample.

**Sparse Computation** The two values for Simpson’s index, \( \sum n(n-1) \) and \( N(N-1) \) are generated over each sample using the sparse matrix counter technique. Then a pass over the array using division yields Simpson’s index from which the total sum and sum of squares can be computed for case and control groups.

4.6 Asymptotic Complexity

Since secure computation is orders of magnitude slower than cleartext computation, we carefully designed our protocols so that we either operate only on a sparse representation of matrix elements or can put as much computation as possible outside of the secure computation. In fact for our pre-computation approach, as shown in Supplementary Table 1 for all test cases we evaluated, we achieved asymptotic improvement compared with a generic solution that performs all operations in secure computation directly.
Supplementary Table 1: **Running Time Complexity.** Speedup of our approaches using local computation and sparse matrix computation compared with generic solution. For the analysis of the sparse approach, the running time is proportional to a constant $k$, which is the proportion of samples ($n$) which have a non-zero element for a given feature. For a given dataset, the total number of non-zero elements will be $(k \times n)n$. In our experiments, $k$ took a value of $\leq 0.2$ for all datasets used as shown in Supplementary Table 2. While the asymptotic complexity is the same, our sparse approach ran faster than the naive approach for each dataset considered.

## 5 Evaluation

### 5.1 Datasets

Supplementary Table 2 summarizes the number of features, samples, file size, and sparsity. The datasets provide a good array of input sizes and sparsity to evaluate our implementations.

|                  | Samples | Features | Size(kB) | Sparsity |
|------------------|---------|----------|----------|----------|
| MSD Case P1      | 254     | 754      | 549.7    | 91%      |
| MSD Case P2      | 254     | 754      | 543.7    | 92%      |
| MSD Control P1   | 242     | 754      | 527.6    | 91%      |
| MSD Control P2   | 242     | 754      | 526.9    | 91%      |
| PGP Case P1      | 43      | 277      | 57.0     | 80%      |
| PGP Case P2      | 43      | 277      | 52.4     | 83%      |
| PGP Control P1   | 42      | 277      | 53.5     | 82%      |
| PGP Control P2   | 41      | 277      | 49.7     | 85%      |
| HMP Case P1      | 173     | 97       | 63.6     | 85%      |
| HMP Case P2      | 173     | 97       | 61.3     | 87%      |
| HMP Control P1   | 174     | 97       | 58.1     | 88%      |
| HMP Control P2   | 174     | 97       | 51.2     | 92%      |

Supplementary Table 2: **Dataset Sizes.** Dimensions and sparsity of each dataset used for evaluation. P1 is Party 1 and P2 is Party 2 for secure computation. Sparsity is defined as 1-(Percent of Non-Zero entries).

### 5.2 RunningTimes

Supplementary Table 3 lists the running times for each statistic and dataset.

### 5.3 Circuit Sizes

Supplementary Table 4 lists the circuit size per feature for each statistic and dataset.
Supplementary Table 3: **Running Times.** Running time for each statistic and each dataset in seconds (PC stands for Pre-Compute). 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 minutes.

Supplementary Table 4: **Circuit Size Per Feature.** Circuit size for each implementation and dataset (PC stands for Pre-Compute). The number of samples is considered as feature count for calculating Alpha Diversity circuit size. 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).

### 5.4 Network Traffic

Supplementary Table 5 lists the traffic from each computation party. The pre-computation approach requires the least amount of traffic with the sparse implementation requiring several more orders of magnitude. The most costly approach is the naive approach. The increase in network traffic between the sparse and pre-computation implementations is significant as compared to the differences in running times of those approaches.
### Supplementary Table 5: Network Traffic

Left column details traffic in MB sent from evaluator (PC stands for Pre-compute). Right column is MB sent from garbler.

| PC From Evaluator | HMP  | PGP  | MSD  | PC From Garbler | HMP  | PGP  | MSD  |
|-------------------|------|------|------|-----------------|------|------|------|
| Alpha Diversity   | 0.02 | 0.02 | 0.02 | Alpha Diversity | 6.86 | 6.86 | 6.86 |
| Chi Square        | 0.31 | 0.86 | 2.32 | Chi Square      | 184.90 | 527.98 | 1437.13 |
| Odds Ratio        | 0.31 | 0.86 | 2.32 | Odds Ratio      | 65.96 | 188.32 | 512.57 |
| Differential Abundance | 0.99 | 2.80 | 7.60 | Differential Abundance | 780.26 | 2228.12 | 6064.97 |
| Sparse From Evaluator | HMP  | PGP  | MSD  | Sparse From Garbler | HMP  | PGP  | MSD  |
| Alpha Diversity   | 5.87 | 5.22 | 43.78 | Alpha Diversity | 5862.87 | 5305.95 | 41568.30 |
| Chi Square        | 15.78 | 17.07 | 145.20 | Chi Square      | 203.64 | 543.32 | 1644.50 |
| Odds Ratio        | 15.78 | 17.07 | 145.20 | Odds Ratio      | 79.91 | 194.89 | 718.15 |
| Differential Abundance | 18.26 | 19.84 | 167.90 | Differential Abundance | 5055.84 | 6593.47 | 41020.37 |
| Naive From Evaluator | HMP  | PGP  | MSD  | Naive From Garbler | HMP  | PGP  | MSD  |
| Alpha Diversity   | 43.71 | 30.32 | N/A  | Alpha Diversity | 30577.40 | 21118.01 | N/A  |
| Chi Square        | 146.42 | 103.76 | 1626.68 | Chi Square      | 402.44 | 680.05 | 3856.70 |
| Odds Ratio        | 146.42 | 103.76 | 1626.67 | Odds Ratio      | 283.50 | 340.40 | 2932.14 |
| Differential Abundance | 169.89 | 122.26 | 1883.51 | Differential Abundance | 35500.60 | 25777.89 | 393164.20 |
References

Erlich, Y. and Narayanan, A. (2014). Routes for breaching and protecting genetic privacy. *Nature Reviews Genetics.*

Fierer, N., Lauber, C. L., Zhou, N., McDonald, D., Costello, E. K., and Knight, R. (2010). Forensic identification using skin bacterial communities. *Proceedings of the National Academy of Sciences.*

Franzosa, E., Huang, K., Meadow, J., Gevers, D., Lemon, K., Bohannan, B., and Huttenhower, C. (2015). Identifying personal microbiomes using metagenomic codes. *Proceeding of the National Academy of Sciences.*

Kort, R., Caspers, M., van de Graaf, A., van Egmond, W., Keijser, B., and Roeselers, G. (2014). Shaping the oral microbiota through intimate kissing. *Microbiome.*

Lax, S., Smith, D. P., Hampton-Marcell, J., Owens, S. M., Handley, K. M., Scott, N. M., Gibbons, S. M., Larsen, P., Shogan, B. D., Weiss, S., et al. (2014). Longitudinal analysis of microbial interaction between humans and the indoor environment. *Science.*

Lozupone, C. and Knight, R. (2005). Unifrac: a new phylogenetic method for comparing microbial communities. *Applied and environmental microbiology.*

Morgan, X. C. and Huttenhower, C. (2012). Human microbiome analysis. *PLoS computational biology.*

Nikolaenko, V., Ioannidis, S., Weinsberg, U., Joye, M., Taft, N., and Boneh, D. (2013). Privacy-preserving matrix factorization. In *CCS.*

Paulson, J. N., Stine, O. C., Bravo, H. C., and Pop, M. (2013). Differential abundance analysis for microbial marker-gene surveys. *Nature methods.*

Snyder, P. (2014). Yaos garbled circuits: Recent directions and implementations.

White, J. R., Nagarajan, N., and Pop, M. (2009). Statistical methods for detecting differentially abundant features in clinical metagenomic samples. *PLoS computational biology.*