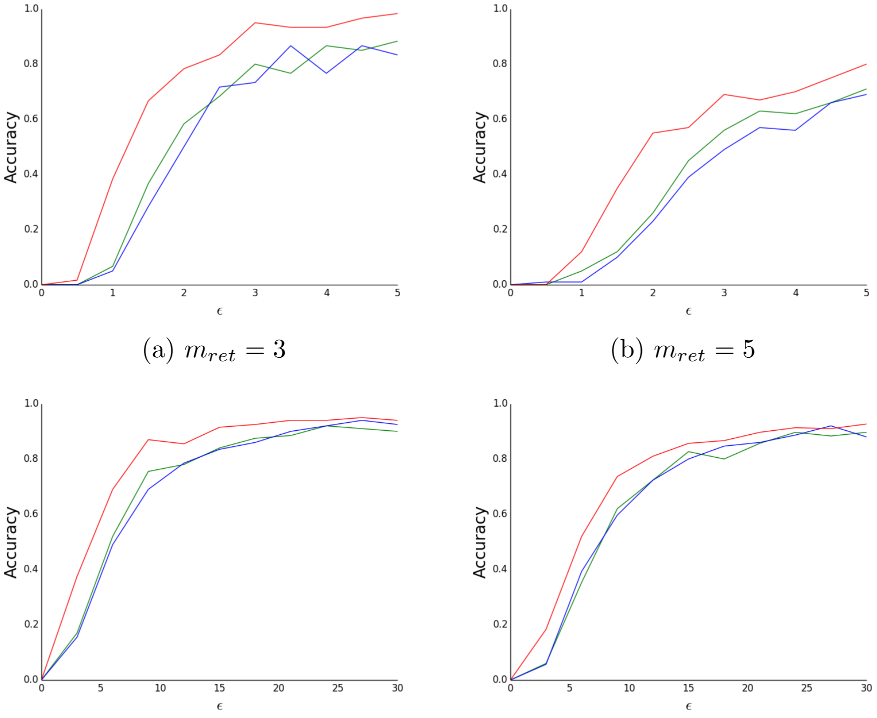
Supplement for “Enabling Privacy Preserving GWAS in Heterogenous Human Populations”

Sean Simmons [1,2,3], Cenk Sahinalp [3,4], and
Bonnie Berger [1,2*]

[1]Department of Mathematics, [2]Computer Science and Artificial Intelligence Laboratory, Massachusetts Institute of Technology, Cambridge, MA
[3] School of Computing Science, Simon Fraser University, Burnaby, BC, Canada and
[4] School of Informatics and Computing, Indiana University, Bloomington, IN

[*]to whom correspondence should be addressed: bab@mit.edu



(a) $m_{ret} = 3$

(b) $m_{ret} = 5$

(c) $m_{ret} = 10$

(d) $m_{ret} = 15$

Figure S1: Comparing approaches for picking SNPs, relates to Figure 2. We measure the accuracy (the percentage of the top SNPs correctly returned) of the three methods for picking top SNPs on simulated GWAS data using score (blue), distance (red) and noise (green) based methods with m_{ret} (the number of SNPs being returned) equal to a. 3, b. 5, c. 10, and d. 15 for varying values of the privacy parameter ϵ. We see that in all four graphs that score and noise based methods outperform the neighbor method. These results are averaged over 20 iterations.

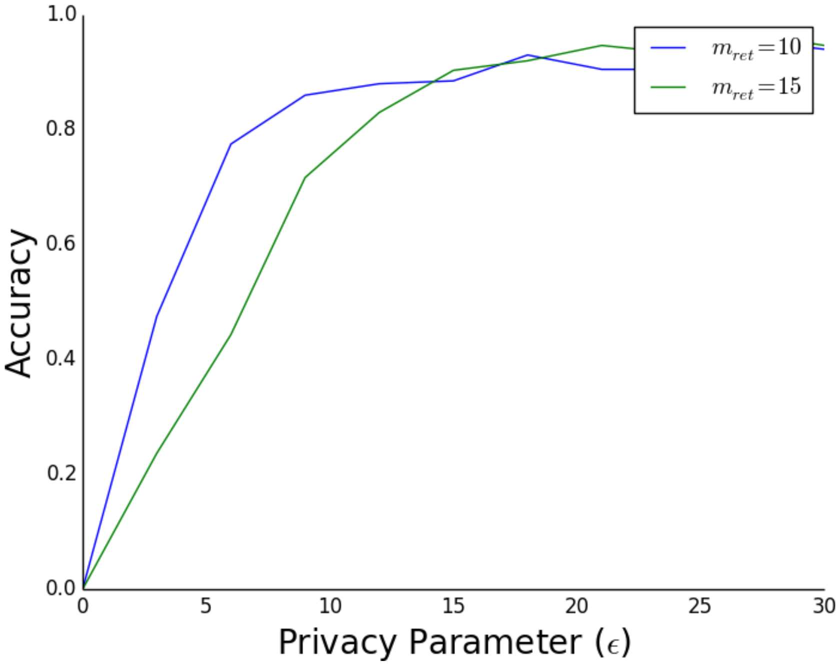
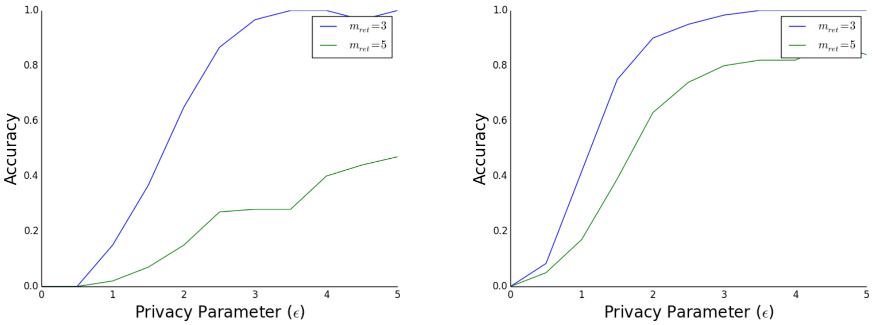


Figure S2: Accuracy when returning larger numbers of SNPs, relates to Figure 2 in the main text. We measure the accuracy (the percentage of the top SNPs correctly returned) of the PrivSTRAT method for picking top SNPs, with m_{ret} (the number of SNPs being returned) equal to (10 and 15 for the RA datasets, with varying values of the privacy parameter ϵ. We see that, in both cases, the accuracy is fairly low for these large values of m_{ret}. These results are averaged over 20 iterations.



(a) Large SNP Set

(b) Small SNP Set

Figure S3: Accuracy on highly stratified population, relates to Figure 2. We measure the accuracy (the percentage of the top SNPs correctly returned) of the PrivSTRAT method for picking top SNPs using with m_{ret} (the number of SNPs being returned) equal to 3 and 5 for two datasets based off HapMap, a (a) larger one and a (b) smaller one, with varying values of the privacy parameter ϵ. These results are averaged over 20 iterations.

Table S1: Accuracy with no stratification, relates to Figure 2. We compare the accuracy of PrivSTRAT with the accuracy of a differentially private version of the allelic test statistic in the absence of population stratification. In particular, we simulate a dataset with one causative SNP, and see what percentage of the time each algorithm returns the causative SNP (see the text for details). We see that, as expected, if there is absolutely no population stratification the privacy preserving allelic test statistic performs better. In practice, however, there will almost always be at least some level of population stratification.

.

Privacy Parameter (ϵ)	.05	.1	.15	.2
PrivSTRAT	**.07**	.61	.87	.99
Differentially Private Allelic Test Statistic	**.07**	**.86**	**1.0**	**1.0**

Table S2: Relationship between sample size and accuracy, relates to Figure 2. We compare the accuracy of our PrivSTRAT method for picking high scoring SNPs for different sample sizes. We used $m_{ret} = 3$, $\epsilon = 1.0$, and averaged over 20 trials. We see that, as the sample size increases, so does accuracy (percentage of top SNPs correctly predicted).

Size	400	600	800	1000	2136 (entire dataset)
Accuracy	0.0	.05	.116	.583	1.0

Supplemental Data and Code

Data S1: Supplementary code and data, related to Experimental Procedure. A zip file containing code and simulated data used in our work. Allows for reconstruction of all figures in our paper that used simulated data, and can be used to test PrivSTRAT and PrivLMM on novel datasets. Contains a readme file explaining the documents included. Also available on github (see main document for more information).

Supplementary Experimental Procedure (Relates to Experimental Procedure)

EIGENSTRAT and LMM

One of the most popular methods for overcoming population stratification is known as EIGENSTRAT. This method is based on the observation that the top few principle components (that is, the top few eigenvectors of the genetic covariance matrix) of the genotype matrix encode information about population stratification.

Formally, assume we have the genotype of n individuals at m SNPs. Let X be the n by m matrix of normalized genotype data and y the n dimensional phenotype vector (Experimental Procedures). The EIGENSTRAT method applies an eigendecomposition to the n by n covariance matrix XX^T. EIGENSTRAT works by forming two new vectors, y^* and x_i^*, where y^* (respectively x_i^*) is given by mean centering y (respectively x_i) and projecting the result onto the vector space orthogonal to the top k eigenvectors of the covariance matrix (k is a user defined parameter; we set $k = 5$). Intuitively, this procedure for producing y^* and x_i^* can be thought of as removing the effects of population stratification. Having removed the population stratification, all that remains is to test if y^* and x_i^* are correlated. This is done using a χ^2-distributed statistic:

$$\chi_i^2 = \frac{(n-k-1)(x_i^* \cdot y^*)^2}{|x_i^*|^2|y^*|^2}$$

Another common method for correcting for population stratification is based on linear mixed models (LMM). LMMs rely on the null model given by $y = X\beta + \epsilon$, where $\epsilon \propto N(0, \sigma_e^2\mathbf{I}_n)$ and $\beta \propto N(0, \frac{\sigma_g^2}{m}\mathbf{I}_m)$. Here, $N(a, B)$ is the normal distribution with covariance matrix B and mean a, σ_e and σ_g are unknown variance parameters, and I_n is the n by n identity matrix.

We consider a slight modification of the LMM based approach used in EMMAX [5]. This approach uses maximum likelihood (ML) to estimate σ_e and σ_g. We can then apply the Wald test to see if a given SNP is associated with our disease phenotype. More specially, if we let $K = \sigma_e^2\mathbf{I}_n + \frac{\sigma_g^2}{m}XX^T$, then we get a χ^2 distributed statistic

$$\chi_{i,LMM}^2 = \frac{(x_i^T K^{-1}(\mathbf{I}_n - \frac{1}{n}\mathbf{1}_n)y)^2}{x_i^T K^{-1} x_i}$$

where $\mathbf{1}_n$ is the n by n matrix of all ones.

Calculating χ^2

Following the discussion in the main text of how to obtain an ϵ-phenotypically differentially private estimate of χ_i^2, we let:

$$u_{dp} = \mu_i \cdot y + Lap(0, \frac{2 \max_j |\mu_{ij}|}{\epsilon})$$

and

$$y_{dp} = |y^*| + Lap(0, \frac{2}{\epsilon})$$

as estimates, where $Lap(0, \lambda)$ is a random variable with distribution:

$$Lap(0, \lambda) \propto exp(\frac{-|x|}{\lambda})$$

We can then estimate the EIGENSTRAT statistic as

$$(n-k-1)\frac{u_{dp}^2}{y_{dp}^2}$$

The approach taken by PrivLMM is almost identical, except there is no need to estimate $|y^*|$, only σ_e and σ_g. (see below).

p-values and Confidence Intervals

In order to calculate p-values based on PrivSTRAT results, it is worth noting that, for fixed ϵ, our PrivSTRAT statistic is asymptotically χ^2 distributed, allowing us to estimate a p-value.

For small ϵ or small population size, however, this p-value tends to be slightly inflated. One way of overcoming this limitation is to select appropriate confidence intervals for the non-private EIGENSTRAT statistic. In particular, when calculating the PrivSTRAT statistic, our algorithm releases

$$u_{dp} = \mu_i \cdot y + Lap(0, \frac{2 \max_j |\mu_{ij}|}{\epsilon})$$

and

$$y_{dp} = |y^*| + Lap(0, \frac{2}{\epsilon})$$

These quantities facilitate selection of confidence intervals for $|\mu_i \cdot y|$ and $|y^*|$, which can be combined to produce a (slightly smaller) confidence interval for χ_i^2. Note, however, this method tends to overestimate the size of the confidence interval, and thus underestimate the level of certainty in our estimate.

```
Algorithm 1 Calculates the neighbor distance
Require: $y, \mu_i, c$
Ensure: Returns the neighbor distance, $d_i(c)$.
  Let $\hat{u}_j = \max(\mu_{ij}(1 - y_j), \mu_{ij}(0 - y_j))$
  Let $\hat{l}_j = \min(\mu_{ij}(1 - y_j), \mu_{ij}(0 - y_j))$
  Let $i_1, \cdots, i_n$ be a permutation on $1, \ldots, n$ such that $\hat{u}_{i_1} \geq \cdots \geq \hat{u}_{i_n}$. Let $u_j = \hat{u}_{i_j}$ for all $j$.
  Let $j_1, \cdots, j_n$ be a permutation on $1, \ldots, n$ such that $\hat{l}_{j_1} \leq \cdots \leq \hat{l}_{j_n}$. Let $l_k = \hat{l}_{j_k}$ for all $k$.
  Let $U_k = \sum_{j=1}^{k} u_j$ and $L_k = \sum_{j=1}^{k} l_j$, $k = 1, \cdots, n$.
  Return $k$ such that $c \in [L_{k+1}, L_k) \cup (U_k, U_{k+1}]$
```

Proofs of correctness

Theorem 1. *Algorithm 1 returns the correct value of $d_i(c)$.*

Proof. Let U_k, L_k, l_k and u_k be as in Algorithm 1.

Assume that y and y' differ in at most k coordinates, then

$$\mu_i y - \mu_i y' = \sum_{j, y_j \neq y'_j} \mu_{ij}(y_j - y'_j) \leq -(l_1 + \cdots + l_k)$$

so

$$\mu_i y' \geq \mu_i y - \sum_{i=1}^{k} l_k = L_k$$

Similarly

$$\mu_i y' \leq \mu_i y + \sum_{i=1}^{k} u_k = U_k$$

so if $d_i(c) \leq k$ than $L_k \leq c \leq U_k$. It is easy to see, however, that if $L_k \leq c \leq U_k$ than $d_i(c) \leq k$, so $d_i(c) = k$ if and only if $c \in [L_k, L_{k-1}) \cup (U_{k-1}, U_k]$. Therefore Algorithm 1 correctly calculates $d_i(c)$.

□

Details About the Distance Based Method

Note that the distance based method for picking high scoring SNPs requires the choice of a boundary value, c. This value is a kind of baseline. Previous work, however, has shown that this arbitrary choice of c can change the accuracy of the method [13].

In order to deal with this we use a slightly modified version of the distance based method [12]. For a given ϵ and choice of m_{ret}, let $x_1, \cdots, x_m$ be a reordering of the list $|\mu_1 \cdot y|, \cdots, |\mu_m \cdot y|$ in decreasing order. Than we can choose c so that

$$c = \frac{|x_{m_{ret}}| + |x_{m_{ret}+1}|}{2} + Lap(0.0, max_{i,j} \frac{|\mu_{i,j}|}{.1\epsilon})$$

We then run the distance based method with a privacy budget of $.9\epsilon$ and a boundary of c. This approach is still ϵ-phenotypically differentially private, and removes some of the accuracy issues of previous approaches.

Simulated dataset

In order to produce simulated data, we used PLINK [11]. The code used to generate this data is available on our website.

We generated two populations of individuals. For each set we first used plink to choose the MAF for 10000 SNPs, each uniformly at random from [.05,.5]. 9900 of the SNPs had no effect on phenotype, 100 had an odds ratio of 1.1. We then generated 5000 people from each of the populations, half of whom where cases, the other half controls. We then combined these two populations to produce our simulated dataset.

The code to do this is present online, as is the simulated data generated in this way.

Estimating the Number of Significant SNPs

The final task we consider is the estimation of the number of significant SNPs in a differentially private way for EIGENSTRAT–in other words, to estimate the number of SNPs with $\chi_i^2 \geq c$ for some user-defined c (often corresponding to a particular p-value cut off). This is equivalent to estimating the number of SNPs with $|\mu_i \cdot y| \geq |y^*|\sqrt{\frac{c}{n-k-1}}$.

In order to do this we first calculate a $.1\epsilon$-phenotypic differential privacy estimate of $|y^*|\sqrt{\frac{c}{n-k-1}}$, denoted c_{dp}, using the Laplacian mechanism.

Since we can calculate $b_i(c_{dp})$ (see the Experimental Procedures) it is easy to apply the method described by Johnson and Shmatikov, 2013, to get a $.9\epsilon$-phenotypic differential privacy estimate of the number of SNPs with $|\mu_i \cdot y| \geq c_{dp}$, which is returned to the researcher. The overall result is an ϵ-phenotypic differential privacy estimate of the number of significant SNPs. Note that the choice of 0.1 and 0.9 are arbitrary, and can be changed to improve results.

PrivLMM: Privacy-Preserving LMM Association

Note that the above framework can be applied to other GWAS statistics besides EIGENSTRAT. In particular, it can be applied to linear mixed models (LMM).

As was the case with EIGENSTRAT, it is worth noting that, if

$$\mu_{i,LMM} = \mu_{i,LMM}(\sigma_e^2, \sigma_g^2) = \frac{x_i^T K^{-1}(\mathbf{I}_n - \frac{1}{n}\mathbf{1}_n)}{\sqrt{x_i^T K^{-1} x_i}}$$

then

$$\chi^2_{i,LMM} = |\mu_{i,LMM} \cdot y|^2$$

This implies that high-scoring SNPs correspond to SNPs with large values of $|\mu_{i,LMM} \cdot y|$.

This allows us to apply the framework used for PrivSTRAT to this LMM statistic, giving us a method, PrivLMM, that provides phenotypic differential privacy. The one added complication is that we need to calculate σ_e and σ_g in a privacy-preserving way, but this is easily done using the sample-and-aggregate framework (See below).

Runtime

We also assessed the runtime of each step in the PrivSTRAT statistic. Note that, as in EIGENSTRAT, PrivSTRAT calculates the top Principal Components (PCs) by performing singular value decomposition (SVD) on the normalized genotype matrix X. Note that our current implementation of PrivSTRAT uses a fast, approximate method for performing this SVD decomposition by default, though also includes an option for performing an exact SVD. Note that we used the exact SVD method for the Case Study (to ensure greater reproducibility between runs) and the approximate method in all other experiments. This approximate method differs from the standard smartpca algorithm used by default in EIGENSTRAT (note that the newest version of EIGENSTRAT has also implemented a fast approximation similar to the one we use [6]). Therefore, in order to assess the effects of the privacy-preserving nature of PrivSTRAT on runtime, we calculated the runtime of PrivSTRAT using both the exact and approximate methods for calculating the SVD.

Asymptotically, the calculation of the exact SVD is by far the most time-consuming step (O(n^2m) runtime), followed by the calculation of the neighbor distance (O($nm\log(n)$)), which is slightly slower than the approximate SVD calculation (O(nmk)).

Estimating Heritability

Another issue to consider is the estimation of σ_e and σ_g in PrivLMM. This, however, can be done using a sample-and-aggregate based framework [1]. In particular, the works by choosing some integer $K > 1$, and dividing the set of participants into K disjoint sets of equal size. On each of these subsets we can estimate $h^2 = \frac{\sigma_g^2}{\sigma_e^2+\sigma_g^2}$ using FaST-LMM [3], GCTA [15] or a similar tool (our implementation uses FaST-LMM). This gives us K estimates of h^2, namely $h_1^2, \ldots, h_K^2$. Let $\tilde{h}^2$ be the average of these K values. Our ϵ-differentially private estimate of h^2 is then given by calculating $\tilde{h}^2 + Lap(0, \frac{1}{K\epsilon})$ and rounding the result to the interval $[0, 1]$.

Next we want to use the same framework to estimate σ_e^2. Note, however, that this would require a bound on σ_e^2. Note that $\sigma_e^2 \leq Var(y)$, and that we can get a ϵ-differentially private estimate v_{dp} of $Var(y)$ easily using the laplacian mechanism. Then we can easily apply the sample-and-aggregate methodology to $\max\{v_{dp}, \sigma_e^2\}$ to get an ϵ-differentially private estimate. Since $\sigma_g^2 = \sigma_e^2(\frac{1}{1-h^2} - 1)$ this allows us to get a 3ϵ-differentially private estimate of (σ_e^2, σ_g^2). Note that this method relies on a very general methodology, and so it seems likely much more accurate results can be obtained with a little work.

Alternative Methods for Picking High Scoring SNPs

In addition to the distance method used by PrivSTRAT, we implemented two other approaches for picking high scoring SNPs: a noise based method and a score based method. In this section we introduce the algorithmic details. We begin by defining:

$$\Delta = \max_{j\in\{1,\ldots,n\}} \max_{S\subset\{1,\ldots,m\},|S|=m_{ret}} \sum_{i\in S} |\mu_{ij}|$$

Our modified version of the noise based method for picking high scoring SNPs [13] works by calculating, for each i, $s_i = |\mu_i y| + Lap(0, \frac{2\Delta}{\epsilon})$, where

$$Lap(0,\lambda) \propto exp(-\frac{|x|}{\lambda})$$

This method then returns the m_{ret} SNPs with the largest value of s_i.

Similarly, our modified score based method [13] works by picking m_{ret} SNPs without repetition, where the probability of picking the ith SNP is proportional to $exp(\frac{\epsilon|\mu_i y|}{2\Delta})$. Both the noise and score method are ϵ-phenotypically differentially private, as is proven below.

Theorem 2. *The modified versions of the score and noise based methods for picking high scoring SNPs given in the manuscript are ϵ-phenotypically differentially private.*

Proof. The proofs are similar to those given in previous works [13], except we use a score function where the score of returning SNPs $s_1, \cdots, s_{m_{ret}}$ equals $\sum_{i=1}^{m_{ret}} |\mu_{s_i} \cdot y|$. For completeness we give the details below.

To see that this is true for the score method, let $\mathbb{S}$ be the collection of all ordered sets of exactly m_{ret} SNPs. Define the score function

$$q : \mathbb{S} \times \{0,1\}^n \to \mathbb{R}$$

so that

$$q(s_1, \cdots, s_{m_{ret}}, y) = \sum_{i=1}^{m_{ret}} |\mu_{s_i} \cdot y|$$

Note that, if $y, y' \in \{0,1\}^n$ differ in exactly one coordinate, then for any $s_1, \cdots, s_{m_{ret}}$ we have that:

$$|q(s_1, \cdots, s_{m_{ret}}, y) - q(s_1, \cdots, s_{m_{ret}}, y')| \leq \sum_{i=1}^{m_{ret}} |\mu_{s_i} \cdot (y - y')|$$

$$\leq \Delta$$

where Δ is defined as in the text. Therefore the result follows from the properties of the exponential mechanism [10].

Next consider the noise method. Again, using the fact that

$$|q(s_1, \cdots, s_{m_{ret}}, y) - q(s_1, \cdots, s_{m_{ret}}, y')| \leq \sum_{i=1}^{m_{ret}} |\mu_{s_i} \cdot (y - y')|$$
$$\leq \Delta$$

the result follows from [13].

□

We do not focus on these methods since, consistent with previous work [12], testing suggests that they are not as accurate as the distance based method, even if they are somewhat faster. See, for example, Fig S1.

No Stratification

Here we compare the accuracy of PrivSTRAT with that of differentially private methods for picking high scoring SNPs based off the allelic test statistic, in the case when there is no population stratification. In particular, we compare it to the accuracy of the distance based method for picking high scoring SNPs using the allelic test statistic [12, 9, 8]

In order to do this we generate a simulated dataset consisting of 10000 individuals and 10000 SNPs, where the the genotypes are generated using the same method as in the simulated dataset introduced above, except with only one population instead of two. Moreover, instead of having 100 causative SNPs, this set only has one causative SNP with an odds ratio of 1.5.

We tested both methods for picking high scoring SNPs on this dataset. In particular, for varying values of ϵ and $m_{ret} = 1$, we measured accuracy as the number of times the causative SNP was returned (averaged over 100 trial). The results are show in Table S1. We see that, as might be expected, in this case the allelic test statistic is more accurate. This is not surprising, since in this case the allelic test statistic is testing the correct null hypothesis. Though this is an interesting comparison, in reality there will almost always be at least some level of population stratification.

Difference From Standard GWAS

The privacy preserving framework we introduce here is slightly different than that taken in standard GWAS. In particular, in standard GWAS the quantity m_{ret} (the number of SNPs to be returned) is not known ahead of time. Instead, the user sets some p-value and gets back a list of all SNPs whose p-value is less than that boundary.

If one wants to perform such a study, they can use the method introduced above for calculating the number of significant SNPs, and then use the returned value as m_{ret}. In order to ensure accuracy, however, it seems more reasonable to choose a small m_{ret} ahead of time. This ensures the accuracy of the GWAS on the highest scoring SNPs, even if it comes at a cost to some SNPs near the p-value threshold of interest.

Large Values of m_{ret}

Our experiments show that, for small m_{ret} ($m_{ret} \leq 5$ or so), our methods are reasonably accurate. It turns out, however, that like previous approaches to differentially private GWAS, these methods do not always scale to large m_{ret} when the number of individuals is small. This is shown for PrivSTRAT on the RA dataset in Fig S2. We see that, though accuracy is comparable to methods that do not correct for population stratification, it is still not as useful as we would like. Luckily, this accuracy should greatly increase as n gets larger.

It is also worth asking if accuracy is the best measure of utility for our method. In particular, using accuracy to measure utility ignores the difference between returning SNPs that score almost as high as the top scoring SNPs versus returning low scoring SNPs. Moreover, GWAS assumes we are using SNPs to tag nearby regions of the genome. This implies that returning a SNP that is near a high scoring SNP can also be useful. Using accuracy as the measurement, however, ignores this as well. Therefore, in order to decide if our method is useful for larger m_{ret} values, we should first decide exactly what makes a given result preferable to another.

Missing Genotype

The above analysis assumed that there was no missing genotype data. In practice, however, many entries in a given genotype matrix will be undefined. There are various ways of dealing with this, most notably imputation. In this work we take a simpler approach (one that is built into the pysnptools package). This approach works by replacing each missing entry in the genotype vector at a given SNP with the mean value taken over all non-missing entries at that SNP. We plan for future versions of PrivSTRAT to make use of imputation based strategies for dealing with missing genotype data.

Calculating PCA

By default, PrivSTRAT uses an approximate version of SVD to perform the PCA in the paper, similar to that suggested in [6]. In particular, we use the TruncatedSVD command in sklearn.decomposition. This is due to the fact that calculating the PCA is by far the most time consuming step in the algorithm. One can, however, use an exact version of the SVD (by setting the -e flag to 1), which uses the SVD method in numpy.linalg.

Sample Size vs Accuracy

As mentioned in the text, it has been noted that there is a tradeoff between sample size and privacy in differentially private statistics. To demonstrate this is true of our method, we subsampled the RA dataset to generate smaller subsets (consisting of $n = 400$, 600, 800 and 1000, half cases, half controls). We ran the PrivSTRAT algorithm for picking high scoring SNPs with $m_{ret} = 3$, $\epsilon = 1.0$. We see that the accuracy (that is to say percentage of top SNPs correctly predicted) increases markedly with sample size, as is expected (Table S2).

Cryptic Relatedness

Though we have focused on the issue of population stratification, another issue in GWAS is that of cryptic relatedness–that is to say pairs of individuals who are more closely related than one would expect at random [14]. As with population stratification this can lead to false positives and inflation of χ^2 statistics.

While the LMM based statistics are designed to deal with this issue, EIGENSTRAT is not. As such, before performing a GWAS with EIGENSTRAT it is important to remove related individuals, using tools such as Plink [11]. Extending this to our framework, one should also remove closely related individuals when using EIGENSTRAT. Luckily, in the large, diverse populations that PrivSTRAT is aimed at, even after removing close relatives there should be a sizable number of individuals remaining on which to perform our GWAS.

Validation

Repeating the experiment in the case study 1000 times shows that the average error for rs9419011 is 1.22 (with the error being less than 0.78 in 50% of the trials) and for rs498422 is 8.70 (with the error being less than 5.91 in

50% of the trials). Notably, the reported p-value for rs9419011 is significant (that is to say it is less than 0.025, the Bonferonni corrected threshold) in all but 15 of the 1000 trials, while the reported p-value for rs498422 is not significant in 932 of the 1000 trials. This means our privacy-preserving validation results agree with the unperturbed validation results in 98.5% and 93.2% of the cases, respectively.

Increased Stratification

Most of the data sets we tested on had small levels of population stratification. Because of this limitation we decided to test PrivSTRAT on a dataset with higher levels of population stratification. In particular, we downloaded the HapMap dataset. After quality control (removing close relatives, etc) we were left with a set consisting of 880 individuals from numerous populations. We sampled around 10,000 of the SNPs at random (9874 to be exact) using the –thin option in Plink.

Using this, we generated simulated phenotype data by picking 5 SNPs at random, and having each SNP correspond to an odds ratio of 1.5. We than used PrivSTRAT on this dataset to return high scoring SNPs. Due to the high population stratification we used $k = 10$ (note that this choice is based off genomic data, not phenotypic data, so does not give away any private phenotypic information). The results, pictured in Fig S3a, are not as striking as in the other cases, but still give reasonable results for $m_{ret} = 3$.

Note that this performance is not as good as for the datasets presented in the paper. One reason for this is size: each of the other datasets is at least twice as large as the HapMap dataset. There are, however, other factors in play: the size of each SNPs effect (aka the odds ratio), the number of SNPs used, etc. For example, if we run the same test with a smaller set of SNPs (2470, just a little over one fourth), the results are much better (see Fig S3b).

Details About Differential Privacy

As an aside, it is worth noting that differential privacy is only meant to ensure that by choosing to participate in a given study, the individual does not lose much more privacy than they would if they had not chosen to participate. It is still possible that the results of the study can lead to a loss of privacy due to the resulting scientific discoveries. For example, if the results of the study determine that a given allele is associated with a disease of interest, than this reveals that anyone who has that allele has an increased risk of the given disease. Privacy lost to such discoveries, however, is considered unavoidable–the only way to avoid such privacy loss is to curtail the growth of science, something that even the most privacy minded researchers are not likely to advocate.

Note that phenotypic differential privacy does not entail releasing genomic data. Rather, it guarantees that even when an adversary deduces genotypic information about a participant, phenotypic information will not be deducible. This choice is motivated by the fact that, in many cases, our main concern is preventing the leakage of private phenotype information. By focusing on protecting phenotypic data–that is to say by using phenotypic differential privacy instead of standard differential privacy– we are able to achieve increased accuracy.

Future Work

Potential future studies include: settings where stronger privacy guarantees (beyond just protecting private phenotype data); recent theoretical work that employs differential privacy to help prevent false positives due to overfitting in adaptive data analysis (looking at data to determine the optimal analysis techniques), overcoming a major problem in medical research [2]; and recent work that shows background knowledge about haplotypes [7] and population genetics [4] can improve accuracy in privacy-preserving genomic analysis, perhaps improving the accuracy of PrivSTRAT and PrivLMM and diminishing false GWAS results which can lead to wasted time and resources.

References

[1] J Abowd, M Schneider, and L Vilhuber. Differential privacy applications to bayesian and linear mixed model estimation. *Journal of Privacy and Condentiality*, 5(1):73–105, 2013.

[2] C Dwork, V Feldman, M Hardt, T Pitassi, O Reingold, and A Roth. The reusable holdout: preserving validity in adaptive data analysis. *Science*, 349:636–638, 2015.

[3] C Lippert et al. Fast linear mixed models for genome-wide association studies. *Nature Methods*, 8:833–835, 2011.

[4] F Tramer et al. Differential privacy with bounded priors: Reconciling utility and privacy in genome-wide association studies. ACM Conference on Computer and Communications Security 2015, 2015.

[5] H Kang et al. Variance component model to account for sample structure in genome-wide association studies. *Nat. Genet.*, 42:348–54, 2010.

[6] K Galinsky et al. Fast principal components analysis reveals independent evolution of adh1b gene in europe and east asia. *BioRxiv*, 2015.

[7] Y Zhao et al. Choosing blindly but wisely: differentially private solicitation of DNA datasets for disease marker discovery. *JAMIA*, 22:100–108, 2015.

[8] Yu et al. Scalable privacy-preserving data sharing methodology for genome-wide association studies. *JBI*, 50:133–141, 2014.

[9] A Johnson and V. Shmatikov. Privacy-preserving data exploration in genome-wide association studies. KDD, pages 1079–1087, 2013.

[10] F McSherry and K Talwar. Mechanism design via differential privacy. Proceedings of the 48th Annual Symposium of Foundations of Computer Science, 2007.

[11] S Purcell, B Neale, K Todd-Brown, L Thomas, M Ferreira, D Bender, J Maller, P Sklar, P de Bakker, M Daly, and P Sham. PLINK: a toolset for whole-genome association and population-based linkage analysis. *AJHG*, 81:559–575, 2007.

[12] S Simmons and B Berger. Realizing privacy preserving genome-wide association studies. *Bioinformatics*, to appear, 2016.

[13] C Uhler, S Fienberg, and A Slavkovic. Privacy-preserving data sharing for genome-wide association studies. *Journal of Privacy and Confidentiality*, 5(1):137–166, 2013.

[14] B Voight and J Pritchard. Confounding from cryptic relatedness in case-control association studies. *PLOS Genetics*, 1, 2013.

[15] J Yang, S Lee, and M Goddard a P Visscher. GCTA: a tool for genome-wide complex trait analysis. *AJHG*, 88:76–82, 2011.