# README:

(Updated October 8, 2015)

This is a readme file to explain this repository.

This covers the basic of using PrivSTRAT, PrivLMM, how to generate the figures from our project.

## Idea of project:

This project aims at implementing tools to allow differentially private GWAS statistics while correcting for population stratification in the samples. In order to do this we have generated two methods, PrivLMM based on LMM, and PrivSTRAT based on EIGENSTRAT. Details of both methods to appear!

## Running PrivSTRAT:

PrivSTRAT can perform three different tasks: picking high scoring SNPs, estimating the number of high scoring SNPs, and estimating the Wald test value of a given SNP.

The flag –t is used to signify which task. You can either use:
–t Count
-t Top
-t Wald
if not specified uses Top

The flag –e is used to specify the privacy budget (aka epsilon). For a privacy budget of 2.0 add
–e 2.0
If not specified uses epsilon=1.0

There is also the –k flag that tells the algorithm how many PCs to use for correction. The default is 5.

The –bed flag is given before the name of the binary ped file containing the genotype and phenotype information. For example, if group.bed/group.fam/group.bim are the files, write:

-bed group

The –save tag is used to specify a file to save to. For example, to save to savefile.txt write:

-save savefile.txt

The other flags are all task specific.

**Picking high scoring SNPs:**

For this task we need to specify the number of SNPs to return. This is done by the -mret flag. For example, if one wants the top 10 SNP use
-mret 10.

One can also specify the algorithm--either the neighbor based, noise based or score based (see manuscript)—using the –a tag. Can use either
-a score
-a noise
-a neighbor

*Example:* Assume we want to run on group.bed, with privacy budget epsilon=1.0, returning the top 10 SNPs, with 10 PCs, and the noise algorithm. Then we would run:

python PrivSTRAT.py –t Top -a noise –bed group –k 10 –mret 10 –e 1.0

**Estimating number of significant SNPs:**

We want to estimate the number of SNPs with pval < some threshold, where the pvalue is set using the –s tag. For example, for threshold=.05 use:

-p .05

*Example:* Assume we want to run on group.bed, with privacy budget epsilon=1.0, estimating the number of SNPs with pval < .05, with 10 PCs. Then we would run:

python PrivSTRAT.py –t Count –bed group –k 10 –p .05 –e 1.0

**Estimating Wald:**

Finally, consider the tasks of estimating the Wald statistic. To do this we need to specify the SNPs, using –s tag. For example, if we want it for SNPs snp1 and snp2 write:

-s snp1 snp2

*Example:* Assume we want to run on group.bed, with privacy budget epsilon=1.0, estimating the Wald statistic on rs101 and rs102, with 10 PCs. Then we would run:

python PrivSTRAT.py –t Wald –bed group –k 10 –s rs101 rs102 –e 1.0

## Running PrivLMM:

PrivLMM.py is almost identical to PrivSTRAT.py, except for two main differences. First, instead of specifying as –k flag you specify –se2 and –sg2, the variance components.

*Example:* Assume we want to run on group.bed, with privacy budget epsilon=1.0, estimating the LLM based Wald statistic on rs101 and rs102, with sigma_e^2=.5 and sigma_g^2=.5. Then we would run:

python PrivLMM.py –t Wald –bed group –se2 .5 –sg2 .5 –s rs101 rs102 –e 1.0
In addition to that, the –t flag has one more option: Herit This returns estimates of sigma_e^2 and sigma_g^2

**Estimating Heritability:**

To estimate heritability using PrivLMM, we need to specify a –num parameter (the number of subsets used to estimate). If not specified is set equal to 5.

*Example:* Assume we want to run on group.bed, with privacy budget epsilon=1.0, estimating the heritability with num set to 10. Then we would run:

python PrivLMM.py –t Herit –bed group –num 10 –e 1.0

## Generating Figures:

Note that our paper has numerous figures. The code in Top_STRAT_Fig.py and Top_LMM_Fig.py can be used to produce figures comparing the accuracy of the three algorithms (neighbor, noise and score) for picking top SNPs using PrivSTRAT and PrivLMM. WaldFig.py does something similar for the Wald test.