2019 Russell Sage Foundation Summer Institute in Social-Science Genomics

Problem Set 3

The purpose of this problem set is to teach about the three big problems in gene-discovery research—multiple hypothesis testing, population stratification, and low power—and some ways to address those challenges. In addition, there are several computational questions that will guide you through the process of estimating a GWAS in PLINK and calculating SNP heritability and SNP-based genetic correlation.

This problem set is due at 9:30am on Monday, June 17.

1. Multiple hypothesis testing and genome-wide significance from a frequentist perspective

Let y denote our phenotype of interest, and suppose that we have data on K independent genetic loci. (We reserve J to denote the total number of loci in the genome, whereas we typically have data on only $K \ll J$ loci.) We denote the genotype score of locus k by x_{ik}. For each locus k, denote the population regression equation by

$$y_i = \phi_k + \beta_k x_{ik} + \epsilon_{ik}, \tag{1}$$

where i indexes individuals, and each ϵ_{ik} is an error term that has mean 0 and is independent of x_{ik}. (For this problem only, we use ϕ_k to denote the constant term because we will use α below for the statistical significance threshold. The constant term won't matter in the below.) For each locus k, we adopt the *significance threshold* α_{locus}, meaning that if the p-value for locus k is smaller than α_{locus}, we declare locus k to be statistically significant.

Since we're interested in testing more than one locus for association with the phenotype, we're in a situation called *multiple hypothesis testing*. This problem addresses the question of how to set α_{locus} in a situation of multiple hypothesis testing.

Recall from Problem 3 in Problem Set 1 that when our significance threshold is α_{locus}, the probability that we reject the null hypothesis for a particular locus when the null hypothesis is true—a probability that is called the *type I error rate* for a particular locus—is equal to α_{locus}. Suppose we set the significance threshold at the conventional level: $\alpha_{\text{locus}} = 0.05$.

a. Explain why, under the null hypothesis, the probability that we reject the null hypothesis for at least one of two loci is $1 - (1 - 0.05)^2 = 0.0975$. Generalize this result to show that, under the null hypothesis, if the significance threshold for each locus is α_{locus}, then the probability that we reject the null hypothesis for at least one of the K loci is $1 - (1 - \alpha_{\text{locus}})^K$.

b. If we test $K = 10$ loci, what is the probability that we find a statistically significant result for at least one of them?

c. Explain why the probability of rejecting the null hypothesis for at least one of the K loci approaches 1 as the number of loci gets large.

d. Explain why, for any K, the expected number of statistically significant results is $\alpha_{\text{locus}} \times K$.

The probability of rejecting the null hypothesis for at least one of the K loci is called the *family-wise error rate*, which we will denote by α_{FW}. It corresponds to the type I error rate for the joint hypothesis that *all* of the β_k's are equal to zero: $H_0: \beta_k = 0$ for all $k = 1,2, \ldots, K$. If we reject the null hypothesis for any of the individual loci,

then we will also reject this joint null hypothesis. You showed in part (a) that $\alpha_{\text{FW}} = 1 - (1 - \alpha_{\text{locus}})^K$.

e. Now, suppose we want to control the family-wise error rate α_{FW} at a particular value, and we want to set the significance threshold α_{locus} accordingly. Show that the value of α_{locus} we should set is

$$\alpha_{\text{locus}} = 1 - (1 - \alpha_{\text{FW}})^{\frac{1}{K}}.$$

This value of α_{locus} is called the *Šidák-adjusted significance threshold*.

f. If we test $K = 10$ loci and want to control the family-wise error rate α_{FW} at 0.05, what is the Šidák-adjusted significance threshold α_{locus} that we should use?

Rather than using this correction, it has become conventional to use instead an approximation called the *Bonferroni-adjusted significance threshold*, which was developed earlier and is extremely straightforward to calculate:

$$\alpha_{\text{locus}} = \frac{\alpha_{\text{FW}}}{K}.$$

g. Show that this is the first-order Taylor approximation to the Šidák-adjusted significance threshold taken around the point $\alpha_{\text{FW}} = 0$.

Hint: Recall that the formula for a first-order Taylor approximation of a function $f(x)$ around the point x_0 is $f(x) \approx f(x_0) + f'(x_0)(x - x_0)$. Apply this approximation to the function $f(\alpha_{FW}) = 1 - (1 - \alpha_{\text{FW}})^{\frac{1}{K}}$.

h. If we test $K = 10$ loci and want to control the family-wise error rate α_{FW} at 0.05, what is the Bonferroni-adjusted significance threshold α_{locus} that we should use?

i. If we adjust α_{locus} for multiple hypothesis testing, explain why—to achieve a given level of statistical power for testing each locus—larger samples are needed the more loci we test.

Notice that you relied on the assumption of independent loci in your derivation of the Šidák-adjusted significance threshold in part (e). Because the K loci are independent, the K hypothesis tests are independent.

j. Suppose instead that the K loci were in perfect linkage disequilibrium (LD) with each other, i.e., their genotypes were perfectly correlated. Explain why in that case, the "effective number" of independent tests is 1, no matter what K is.

Relatedly, if the K loci are in LD but not in perfect LD, then the "effective number" of independent tests is larger than 1 but smaller than K.

Consider a genome-wide association study (GWAS), in which each single-nucleotide polymorphism (SNP) measured in the data is tested individually for association with the phenotype. A typical modern genotyping array ("gene chip") measures roughly 2.5 million SNPs, but most of these SNPs are in high LD with their neighbors.

k. Explain why Bonferroni-adjusting the significance threshold for K = 2.5 million tests would be far too conservative (i.e., lead to a significance threshold that is too stringent).

Given the LD structure in the genome in European-descent populations, the "effective number" of independent tests in a GWAS has been estimated to be on the order of 1 million.

l. Explain why the genome-wide significance threshold, $\alpha_{\text{locus}} = 5 \times 10^{-8}$, can be viewed as roughly equal to the Bonferroni-adjusted significance threshold for a family-wise error rate of $\alpha_{\text{FW}} = 0.05$.

As genotyping technology improves, the number of SNPs available in genome-wide data will continue to increase.

m. Explain why the genome-wide significance threshold, $\alpha_{\text{locus}} = 5 \times 10^{-8}$, will *not* have to be made more stringent by much, if at all.

n. Explain why each of the following is an example of multiple hypothesis testing that should be accompanied by adjustment of the significance threshold if we want to control the family-wise error rate:

i. I run an experiment on 40 participants and don't find anything, so I collect data on another 40 participants and then analyze all the data together.

ii. I analyze my data and don't find a treatment effect, but when I examine the results separately by sex, I find an effect among women but not men.

iii. I run a regression of my phenotype on a polygenic score, an environmental variable (that is independent of the polygenic score), and the interaction between the polygenic score and the environmental variable.

Hint: Consider why in cases i and ii, the "effective number" of independent tests is less than the number of tests, while in case iii, the three tests are uncorrelated.

o. Why might it be useful for researchers to *preregister*—i.e., pre-specify and post with a timestamp—the analyses they plan to run? What are some potential downsides of preregistration?

Note that as of the last few years, there are a number of public repositories that researchers can use to preregister their analysis plans. The Open Science Framework is one popular site: https://osf.io/. For discussion of some of the pros and cons of preregistration, see Olken (2015) and Coffman and Niederle (2015).

2. Pre-experimental odds and genome-wide significance from a Bayesian perspective

As in Problem 1, suppose that we will run the regression in Equation 1. We will assume that there are only possible two states of the world:

H_0 (the null hypothesis): Zero effect $\beta_k = 0$.
H_1 (the alternative hypothesis): A true effect of size β_k.

(In realistic applications, we would instead want to consider a prior distribution over a continuum of possible effect sizes, but we assume only two states of the world to keep the math simple and to keep the set-up parallel to the hypothesis-testing set-up from the previous question.)

We will consider the problem of choosing the significance threshold, α, from a Bayesian point of view.

a. Using Bayes' Rule, show that

$$\frac{\Pr(H_1|\text{sig})}{\Pr(H_0|\text{sig})} = \left(\frac{\text{power}}{\alpha}\right) \times \frac{\Pr(H_1)}{\Pr(H_0)}. \quad (2)$$

The *rejection ratio*, $\left(\frac{\text{power}}{\alpha}\right)$, quantifies the evidentiary impact of rejecting the null hypothesis. The *prior odds,* $\frac{\Pr(H_1)}{\Pr(H_0)}$, quantifies the researcher's relative belief in H_1 before having seen the results of the experiment. The *pre-experimental odds*, $\frac{\Pr(H_1|\text{sig})}{\Pr(H_0|\text{sig})}$, quantifies the researcher's relative belief in H_1 after having seen that the results are statistically significant.

(Note that the odds are called "pre-experimental" because we are assuming that the only information we have about the results is whether or not they are statistically

significant. The pre-experimental odds are useful to calculate to help researchers make study design decisions—such as what significance threshold α to set and how large a sample to collect—so as to ensure that statistically significant findings constitute persuasive evidence for H_1 over H_0. Once the data have actually been observed, we have more information than just whether or not the results were statistically significant—for example, we know the estimated effect size $\hat{\beta}_k$ and its corresponding *p*-value. Once the data are in hand, rather than using the pre-experimental odds, a Bayesian's beliefs should correspond to the *post-experimental odds*, more commonly called the *Bayes factor*: $\frac{\Pr(H_1|\text{data})}{\Pr(H_0|\text{data})}$. For discussion and examples, see Bayarri et al. (2016).)

Suppose you are designing an experiment. You have an anticipated effect size, β_k, and a prior belief about the likelihood that there is an effect: $\Pr(H_1) = 1 - \Pr(H_0)$. Your prior odds are 1:20 against there being an effect: $\frac{\Pr(H_1)}{\Pr(H_0)} = \frac{1}{20}$. You want to choose your significance threshold α so that, if you get a big enough sample for 0.80 power, your pre-experimental odds are 16:1—meaning that if the experimental results lead to you to reject the null hypothesis, then your beliefs will be 16:1 in favor of H_1. (This benchmark of 16:1 is arbitrary, but it is chosen to correspond to the pre-experimental odds that result from the conventions of setting the significance threshold at $\alpha = 0.05$ and of considering 0.80 power, assuming that the null and alternative hypotheses were considered equally likely before the experiment: $\frac{\Pr(H_1)}{\Pr(H_0)} = 1$.)

b. Show that you should set your significance threshold at $\alpha = \frac{0.05}{20} = 0.0025$.

c Suppose that instead, you keep your significance threshold at $\alpha = 0.05$, and you increase your sample size, which increases your power above 0.80. Show that no matter how large your sample, your pre-experimental odds will always be smaller than 1.

Substantively, this means that if the prior odds are 1:20 against there being an effect and we hold $\alpha = 0.05$ or higher, our pre-experimental odds always favor the null hypothesis, regardless of our sample size.

In Problem 1 parts (l) and (m), you provided a frequentist justification for adopting a stringent significance threshold for GWAS. (The justification is called *frequentist* because it is based on ensuring that the frequency of false positives—as measured by the family-wise error rate—is controlled at the desired level.) In an influential paper in 2007, the Wellcome Trust Case-Control Consortium described another justification, this one based on Bayesian reasoning, which you will reproduce in this part of the problem. (The justification is called *Bayesian* because it is based on using Bayes' Rule to calculate appropriate beliefs about possible states of the world given the results of the data analysis.)

In their paper, the Wellcome Trust Case-Control Consortium, who were studying disease phenotypes, reported some of the first large-scale GWAS results. They argued that, given the large number of loci in the genome, the prior odds of a true association for any given locus is only 1:100,000. They supposed that they had 50% power for detecting a true effect at each locus, and they wanted to set the significance threshold so that they would claim a discovery only if the pre-experimental odds were 10:1.

d. Use Equation 2 to explain why they used a significance threshold of $\alpha = 5 \times 10^{-7}$.

(Note: Here is some historical context for the Wellcome Trust paper. When the paper was published in 2007, there was growing concern about the replicability of findings in medical genetics research. Most papers were based on samples of a few hundred individuals. The Wellcome Trust researchers realized that the replicability problems were the result of low power. Their paper demonstrated that with

relatively large samples, a gene-discovery study could produce replicable results. Their sample size—roughly 2,000 cases and roughly 3,000 controls for each of the seven diseases they studied—was small by today's GWAS standards, but it was very large for the time. Moreover, it is large enough to provide reasonable statistical power for some of the disease phenotypes they studied, for which there are genetic loci with larger effects than has been found for behavioral phenotypes.

In 2007, genome-wide data were only beginning to become available, and there were active debates about what significance threshold should be adopted in GWAS. By the early 2010s, $\alpha = 5 \times 10^{-8}$ had become the norm for genome-wide significance.)

e. Write a program (in R, Matlab, Stata, etc.), or create a spreadsheet in Excel, to calculate the pre-experimental odds as a function of α, power, and the prior odds using Equation 2.

f. Consider a GWAS where H_1 corresponds to a locus whose explanatory power is R^2 = 0.0002. Using your program from part (e) and the levels of power you calculated in Problem 3(h) from Problem Set 1, fill in the entries of the table below with the value of the pre-experimental odds corresponding to each scenario.

	Standard significance threshold (α = 0.05)				
	N = 100	N = 1,000	N = 10,000	N = 100,000	N = 1,000,000
$\Pr(H_1)$ = 0.001					
$\Pr(H_1)$ = 0.01					
$\Pr(H_1)$ = 0.1					

	Genome-wide significance threshold ($\alpha = 5\times10^{-8}$)				
	N = 100	N = 1,000	N = 10,000	N = 100,000	N = 1,000,000

$\Pr(H_1) = 0.001$					
$\Pr(H_1) = 0.01$					
$\Pr(H_1) = 0.1$					

3. Winner's Curse Adjustment

Let's begin with the simple model used in GWA studies

$$y_i = \beta_0 + \beta_j x_{i,j} + \varepsilon_i,$$

where y_i is a measure of some phenotype for individual i, $x_{i,j} \in \{0, 1, 2\}$ is the genotype of SNP j for individual i, and ε_i is the residual for individual i. Note that this means that β_j is the "effect" of SNP j on the phenotype not controlling for the effect of other SNPs and therefore may also capture the effect of other SNPs that are in LD with SNP j.

In statistical models of genetic data, we often treat the coefficient in these models as if they are random variables. For this problem, we will assume that β_j has some probability π of being null (i.e., β_j is exactly zero), in which case β_j follows a degenerate distribution with all its mass at zero. We also assume that β_j has some probability $(1 - \pi)$ of being distributed normally with mean zero and variance ω_j^2.

a. Assuming that the choice of the reference allele is random, why is it reasonable to assume that the mean of β_j is zero?

In nearly all statistical models that treat β_j as random, we assume that $\omega_j = \frac{1}{\sqrt{2p_j q_j}}$ is a constant, with p_j denoting the allele frequency of the reference allele for SNP j and $q_j = 1 - p_j$.

b. What does this mean about how the variances of effect sizes compare across SNPs? What about the variances of standardized effect sizes? (See Problem 4 from Problem Set 1 for a reminder about what standardized effect sizes are, and see Problem 2 from Problem Set 1 for the formula for the variance of $x_{i,j}$.)

When we estimate a GWAS, we get estimates of β_j for each SNP j. We use $\hat{\beta}_j$ to denote this estimate. We can express this as

$$\hat{\beta}_j = \beta_j + e_j,$$

where e_j is the estimation error, which is independent of β_j. If $\hat{\beta}_j$ is estimated by OLS or logistic regression, then the estimation error will be asymptotically distributed

$$e_j \sim N\left(0, \sigma_j^2\right).$$

Note that since e_j is the estimation error, $\sqrt{\sigma_j^2}$ is the standard error of $\hat{\beta}_j$.

c. In the case that β_j is null, what is the distribution of $\hat{\beta}_j$? What is the distribution of $\hat{\beta}_j$ in the case the β_j is not null?

Very briefly, the *Winner's Curse* means that, even though we may estimate the effect of some SNP on some phenotype to be large, the true effect of the SNP tends to be smaller in magnitude than we estimate it to be. To correct $\hat{\beta}_j$ for the Winner's Curse, we will calculate what we expect the true effect size to be given the estimated effect, or in mathematical terms: $\mathrm{E}\left(\beta_j \mid \hat{\beta}_j\right)$.

Note that

$$\mathrm{E}\left(\beta_j \mid \hat{\beta}_j\right)$$

$$= \mathrm{E}\left(\beta_j \mid \hat{\beta}_j, \beta_j \text{ is null}\right)\Pr\left(\beta_j \text{ is null} \mid \hat{\beta}_j\right)$$
$$+ \mathrm{E}\left(\beta_j \mid \hat{\beta}_j, \beta_j \text{ is not null}\right)\Pr\left(\beta_j \text{ is not null} \mid \hat{\beta}_j\right)$$

$$= 0 \times \Pr\left(\beta_j \text{ is null} \mid \hat{\beta}_j\right) + \mathrm{E}\left(\beta_j \mid \hat{\beta}_j, \beta_j \text{ is not null}\right)\Pr\left(\beta_j \text{ is not null} \mid \hat{\beta}_j\right)$$

$$= \mathrm{E}\left(\beta_j \mid \hat{\beta}_j, \beta_j \text{ is not null}\right)\Pr\left(\beta_j \text{ is not null} \mid \hat{\beta}_j\right).$$

We will need to calculate each of the terms of this product separately.

d. Show that

$$\Pr\left(\beta_j \text{ is not null} \mid \hat{\beta}_j\right) = \frac{\phi\left(\hat{\beta}_j;\ 0, \omega_j^2 + \sigma_j^2\right)(1 - \pi)}{\phi\left(\hat{\beta}_j;\ 0, \omega_j^2 + \sigma_j^2\right)(1 - \pi) + \phi\left(\hat{\beta}_j; 0, \sigma_j^2\right)\pi},$$

where

$$\phi(x;\ A, B) = \frac{e^{-(x-A)^2/2B}}{\sqrt{B2\pi}},$$

which is the pdf of a normal distribution with mean A and variance B.

Hint: Use Bayes' Rule.

It can also be shown that

$$\mathrm{E}\left(\beta_j \mid \hat{\beta}_j, \beta_j \text{ is not null}\right) = \frac{\omega_j^2}{\omega_j^2 + \sigma_j^2}\hat{\beta}_j.$$

(Show this using Bayes' Rule as an optional exercise if you'd like.)

e. What happens to the expression for $\mathrm{E}\left(\beta_j \mid \hat{\beta}_j, \beta_j \text{ is not null}\right)$ as the sample size used to estimate $\hat{\beta}_j$ increases?

The Winner's Curse correction is therefore given by,

$$\mathrm{E}\left(\beta_j \mid \hat{\beta}_j\right) = \mathrm{E}\left(\beta_j \mid \hat{\beta}_j, \beta_j \text{ is not null}\right)\Pr\left(\beta_j \text{ is not null} \mid \hat{\beta}_j\right)$$

$$= \left(\frac{\omega_j^2}{\omega_j^2 + \sigma_j^2}\hat{\beta}_j\right)\left(\frac{\phi\left(\hat{\beta}_j;\ 0, \omega_j^2 + \sigma_j^2\right)(1-\pi)}{\phi\left(\hat{\beta}_j;\ 0, \omega_j^2 + \sigma_j^2\right)(1-\pi) + \phi\left(\hat{\beta}_j; 0, \sigma_j^2\right)\pi}\right).$$

We will next apply the Winner's Curse correction to a set of summary statistics from the recent Okbay et al. (2016) GWAS on depressive symptoms. The most significant SNP in that analysis had an estimated effect of 0.0149 and a standard error of 0.0025 ($N = 162{,}107$). Using a maximum likelihood approach, we estimate that $\hat{\pi} = 0.684$ and $\widehat{\omega}_j^2 = 2.95 \times 10^{-6}$.

f. Using these as the true values, what is the probability that SNP *j* has a non-zero effect on the phenotype given the estimated effect size and standard error?

g. Using the values above, what is the expected, Winner's Curse adjusted estimate of the effect of SNP *j* on the phenotype given the estimated effect? How large is this effect as a fraction of the GWAS estimate, $\hat{\beta}_j$?

Say that we acquire a sample that is independent from the sample with which we estimated $\hat{\beta}_j$, and we would like to test to see if the SNP replicates in the new sample. (To distinguish this sample from the sample used to obtain $\hat{\beta}_j$, we will refer to the original sample as the "discovery sample" and the new sample as the "replication sample.") Assuming that the replication sample is measuring the same trait with the same heritability, the effect of the SNP on the phenotype should be the same in both samples.

h. Show that with a replication sample of $N = 173{,}600$, we achieve approximately 50% power to replicate our result. That is, under the assumptions that the Winners' Curse corrected estimate (calculated in part (g)) is the true effect size,

show that $N = 173{,}600$ gives us approximately 50% probability that the estimate from the replication sample will have a p-value less than 0.05. (Hint: The variance of the error in the estimate should scale with the size of the sample.) Show that with a sample size of $N = 354{,}700$, we have approximately 80% power to replicate the result, and that with $N = 474{,}700$, we reach about 90% power to replicate the result.

(Note: If we wanted to be more precise in our replication power calculation, we would assume that the true effect is drawn from the posterior distribution rather than assuming that the true effect is equal to the posterior mean, as we do here. A simple way to do the power calculation in that case is by simulation. We would draw the true β_k from the posterior distribution, draw e_j from the distribution of the error given the replication sample size, add them together, and verify if the simulated "estimate" $\hat{\beta}_j$ is large enough to have a p-value less than 0.05. We would repeat this many times, and the fraction of times that we pass the test is the power of the replication. Optional problem: why does Jensen's Inequality imply that this approach is more appropriate?)

Okbay et al. (2016) did try to replicate the effect of this SNP in a sample of $N = 307{,}352$ and got an estimate of 0.0045 ($SE = 0.0018$).

i. Meta-analyze the discovery and replication effect estimates using sample-size weighting. Then, calculate p-values from the discovery, replication, and meta-analyzed samples. Refer to Problem 5 of Problem Set 2 for formulas needed for sample size-weighted meta-analysis.

j. Using the effect estimated in the meta-analysis, what is the probability that the SNP is not null? (Note that you will need to calculate the standard error of this effect using a formula from Problem 5 of Problem Set 2.) How does this compare

to the probability that the SNP is not null based on the estimate from the discovery sample in part (f)?

k. An investigator who uses the p-value as a measure of the strength of evidence would conclude that the evidence of association was weakened by the replication attempt. Using your results from part (i) and the posterior probability results from parts (f) and (j) comment briefly on this conclusion.

Computational Problems 2 and 3

2. Genome-Wide Association Study (GWAS)

Introduction

A GWAS allows one to calculate which SNPs are associated with a given phenotype. The computational demands of GWAS can quickly become unruly with a large number of covariates. Therefore, a common approach is to regress the outcome variable on all covariates (e.g., indicators for age, sex, age*sex, PC's, array indicators, etc.), take residuals, and use these residuals as the outcome variable in the GWAS. While this is technically not equivalent to multivariate GWAS, in practice, it gives similar results.

a. Test GWAS: Type the following into the server (all on one line with spaces separating the flags), taking care to replace USERNAME with your user name. This will run a linear GWAS on test BIM/BED/FAM data and test phenotype data, filtered on MAF, and saved in your directory:

```
plink
--bfile /data/gcta/test
--maf 0.05
--linear
--pheno /data/gcta/test.phen
--allow-no-sex
--out /home/USERNAME/test_plink_assoc_nocov
```

Notice that PLINK creates two files when you execute this command. One is a .log file, and the other includes the results from the GWAS (its file name is appended with assoc.linear).

Here we provide explanations of the various components of the above example:

- -- bfile /data/gcta/test
    - Specifies that the genetic data is in BED/BIM/FAM format; "--bfile file" in the above command will look for files with the .bed, .bim, and .fam filename extensions: test.bed, test.bim, test.fam in the folder /data/gcta

- --linear
    - Specifies that we'll be running an additive, linear GWAS

- --pheno /data/gcta/test.phen
    - Causes phenotype values to be read from the 3rd column of the specified space- or tab-delimited file, instead of the .fam or .ped file. The first and second columns of that file must contain family and within-family IDs, respectively. Take a look at this file on the server by typing "less /data/gcta/test.phen" or "head –n10 /data/gcta/test.phen" and note the residualized phenotype values in the 3rd column (Remember: type "q" to exit less.)

- -- out /home/USERNAME/test_plink_assoc_nocov
    - Specifies the location of the output files (in your home directory)

- --allow-no-sex
    - Allows for missing sex codes

- --maf 0.05
    - Filters out SNPs with minor allele frequency < 0.05

Here are a few additional options that might come in handy at some point:

- --covar filename
    - Designates the file to load covariates from. The file format is the same as for --pheno (optional header line, FID and IID in first two columns with covariates in the remaining columns). This might be helpful if you are not using residualized phenotypes, but instead need to specify each covariate in your GWAS regression model.

- --covar-name name-of-covariates
    - This command allows you to specify a subset of covariates to load, by column name. This could be useful if your phenotypes and covariates are all in the same file. Separate multiple column names with spaces or commas, and use dashes to designate ranges. (Spaces are not permitted immediately before or after a range-denoting dash.) --covar-number lets you use column numbers instead

Remember: There are many uses and options in PLINK. As you begin to learn or continue to use PLINK, the online manual will become invaluable: https://www.cog-genomics.org/plink2

PLINK GWAS Problem

b. Generating Principal Components: When estimating a GWAS, researchers often wish to control for genetic ancestry in order to reduce bias due to stratification. They do so with principal components. In this part of the problem, you will learn how to construct principal components from genetic data in PLINK.

   Use PLINK's "--bfile" and "--pca 10" flags to generate the first 10 principal components for those genotyped in the following test files: /data/gcta/test.bim, /data/gcta/test.bed, /data/gcta/test.fam. We are not asking you to perform these on the actual Add Health data, because this is a memory-intensive process which would crash our system if you were all to run it for the actual genetic data

at the same time. The operations should take a few minutes to complete. Be sure to specify an output file to your personal directory, /home/USERNAME/, using the "--out" flag.

Notice that there are two files created with this command. One ends in ".eigenvec" and the other ends in ".eigenval".

Compare your results to those found in /data/clean/test_pcs.eigenvec and /data/clean/test_pcs.eigenval to make sure you're doing this correctly. For example, type "less /data/clean/test_pcs.eigenvec" or "head –n10 /data/clean/test_pcs.eigenvec" to print out the first 10 lines of that file, do the same with your file, and compare.

You can also view the PC's generated from identical commands for the two Add Health chips in:

- /data/clean/omni2_5_pcs.eigenvec and /data/clean/omni2_5_pcs.eigenval
- /data/clean/omni1_pcs.eigenvec and /data/clean/omni1_pcs.eigenval

c. Residualizing the Phenotype: Remember from above that a common computational shortcut (which isn't technically equivalent to multivariate GWAS) for running a GWAS is regressing the phenotype of interest on all relevant covariates, and taking the residual from that regression as the phenotype (i.e., a univariate GWAS with the residuals). In this part of the problem, we use R to residualize our phenotype, EA, on a set of covariates that is often used with GWAS.

Begin by opening /data/clean/ah_ea_sex_byear_pcs_euros.csv in R. (The code below should help if you are unfamiliar with R.)

```
df <- as.data.frame(read.csv("FILEPATH/FILENAME"))
```

Then, run the following linear regression of educational attainment (EA) on sex (BIO_SEX4, transformed to 0/1 binary), year of birth (H4OD1Y), the interaction between sex and year of birth, year of birth squared and its interaction with sex, and the first 10 principal components. Save the residuals as a new variable.

```
df$BIO_SEX4 <- df$BIO_SEX4 - 1
df$H4OD1Y2 <- df$H4OD1Y^2

regr <- lm(EA ~ BIO_SEX4 + H4OD1Y + BIO_SEX4*H4OD1Y + H4OD1Y2 +
BIO_SEX4*H4OD1Y2 + PC1 + PC2 + PC3 + PC4 + PC5 + PC6 + PC7 + PC8 + PC9
+ PC10, data = df)

summary(regr)
df$resid <- resid(regr)
```

Output a file with three columns: family identifier (FID), within-family identifier (AID), and the residual from the regression estimated above (resid).

```
df_abbrev <- data.frame(df$FID, df$AID, df$resid)

write.csv(df_abbrev, file = "home/USERNAME/FILENAME.csv",
row.names=F, col.names=F, sep=" ")
```

Note that /data/clean/ah_ea_resid_euros.csv contains these saved results. Be sure you can replicate these residuals. Again, use the “less” or “head” command to check that the first 10 lines or so match.

d. Running the GWAS: Now, we’re ready to run a GWAS of EA (residualized on sex, age, interactions, and PCs) on the omni2_5 genetic data in PLINK. The genetic

data can be found here: /data/genotypes/orig/omni2_5_sample_AID_sex_dupQC_hapmapQC_maf_hwe_1000GI_fwd_het_plateQC. Use the phenotype residuals in /data/clean/ah_ea_resid_euros.csv (this file is identical to what you just generated above).

You should use the flags “--bfile", “--linear hide-covar”, “--pheno", and “--allow-no-sex” (for this problem, omit the “--maf 0.05” filter). The GWAS should take about 5 minutes to run. Be sure to specify an output file in /home/USERNAME/

You should compare your results to /data/ah_gwas/ah_ea_resid_plink_gwas_omni2_5_euros.assoc.linear, again using less to view the first 10 lines or so.

Note that there are related individuals in Add Health, and that for a real GWAS analysis, we would need to generate and control for a genetic relatedness matrix (GRM) using a Linear Mixed Model approach. If you are curious about how to do this, please ask Hui!

3. LD Score Regression

Introduction

In this problem, we will implement LD score regression (LDSC) analyses using Python on real data. With LDSC, we can calculate SNP-based heritability, calculate genetic correlations between phenotypes, and calculate stratification statistics.

LDSC is run with Python, which is increasingly the de facto programming language these days. So long as the appropriate libraries are installed, then running a python application is as simple as:

python app.py --parameter1 parametervalue1 --parameter2 parametervalue2 …

Two python applications will be used in our LDSC analyses, which we will review in greater detail below.

- munge_sumstats.py, which will "munge" GWAS summary statistics data to get them in a format that can be directly read by LDSC
- ldsc.py, which will run the actual analyses

Be sure to review the git and wiki, which are great for setup instructions and for troubleshooting LDSC:

- https://github.com/bulik/ldsc
- https://github.com/bulik/ldsc/wiki

Using munge_sumstats.py

Munge (pronounced MUHNJ) is "a verb, used in a derogatory sense, meaning to imperfectly transform information"

- This application will make sure you have all the necessary variables from GWAS summary statistics to run LDSC, and puts in a format ldsc.py recognizes
- To run LDSC, we'll need columns that have: (1) a SNP identifier (rsID); (2) sample size; (3) effect allele; (4) "other" (non-effect) allele; (4) a signed summary statistic (beta, OR, z-score, etc); and (6) a p-value
- Munge is an "intelligent" application; it will guess column names associated with each required column, but if it doesn't recognize a necessary one, you need to supply the column header explicitly (always check logs to make sure Munge guessed correctly!)

a. Type the following into the server (all on one line with spaces separating the flags), taking care to replace USERNAME with your user name. This code will

munge the Okbay et al. 2016 educational attainment GWAS results and save the resulting file in your directory. This should take less than a minute to run:

```
python /data/ldsc/munge_sumstats.py
--sumstats /data/ea2/EduYears_Main.txt
--N 328917
--out /home/USERNAME/EduYears_Main.munged.hm3
--merge-alleles /data/ldsc/w_hm3.snplist
```

Notice that there are two files produced, a .log file and a file that includes the munged summary statistics (appended with ".sumstats.gz").

Here we provide explanations of the various components of the above example:

- -- sumstats /data/ea2/EduYears_Main.txt
    - Specifies GWAS summary stats file named "EduYears_Main.text" found in the folder /data/ea2

- --N 328917
    - Provides the overall sample size of the GWAS summary statistics, overriding what might already be a column in --sumstats (if you less EduYears_Main.txt, you'll see that for these EA summary statistics, sample size was not already a column)

- --out /home/USERNAME/EduYears_Main.munged.hm3
    - Specifies the location of the output files (in your home directory)

- --merge-alleles /data/ldsc/w_hm3.snplist
    - Restrict analysis to HM3 SNPs given these are usually well-imputed across studies

Note that there are a few things that Munge automatically does, but which can be changed with flags if the need arises. Munge will automatically filter out SNPs with INFO <= 0.9 if the INFO column exists. INFO is a measure of imputation accuracy. You can override this filter with --info-min INFO_MIN. Next, Munge will automatically filter out SNPs with MAF <= 0.01. You can override this filter with --maf-min MAF_MIN. Munge will also automatically remove variants that are not SNPs (e.g., indels), strand ambiguous SNPs, and SNPs with duplicated rsID numbers. It will also automatically check that the median value of the signed summary statistic column (beta, Z, OR, log OR) is close to the null median, in order to make sure that this column is not mislabeled. Finally, if the mean chi-square is below 1.02, it will warn you that the data probably are not suitable for LD Score regression.

Here are some additional flags that you can use to specify column names not automatically detected by the software (where the argument of the flag is the actual column name in your GWAS summary statistics file):

- --snp SNP
- --signed-sumstats SIGNED_SUMSTATS
- --info INFO
- --a1 A1
- --a2 A2
- --p P
- --frq FRQ
- --N-col N_COL
- --N-cas-col N_CAS_COL
- --N-con-col N_CON_COL
- --N-cas N_CAS
- --N-con N_CON

Using ldsc.py

b. Type the following into the server (all on one line with spaces separating the flags). Take care to replace USERNAME with your user name. This code will calculate the LD score heritability from the munged Okbay et al. 2016 educational attainment GWAS results (EduYears_Main), and will save the results in your directory. This should take a minute or two to run:

```
python /data/ldsc/ldsc.py
--h2 /home/USERNAME/EduYears_Main.munged.hm3.sumstats.gz
--ref-ld-chr /data/ldsc/eur_w_ld_chr/
--w-ld-chr /data/ldsc/eur_w_ld_chr/
--out /home/USERNAME/EduYears_Main.munged.hm3.sumstats.h2
```

Notice that the file produced is appended with ".log"; this file contains the same information as is shown in the command-line output.

Here we provide explanations of the various components of the above example:

- --h2 /home/USERNAME/EduYears_Main.munged.hm3.sumstats.gz
    - This gives the location of munged file you created above and placed in your home directory with the --out Munge command, called "EduYears_Main.munged.hm3.sumstats.gz"

- --ref-ld-flag /data/ldsc/eur_w_ld_chr/
    - The --ref-ld flag tells LDSC which LD Score files to use as the independent variable in the LD Score regression. The --ref-ld-chr flag is used for LD Score files split across chromosomes (which we have).

- --w-ld-chr /data/ldsc/eur_w_ld_chr/
    - This flag tells LDSC which LD Scores to use for the regression weights. Ideally, the --w-ld LD Score for SNP j should be the sum over all SNPs k included in the regression of r^2_jk. In practice, LD Score Regression

is not very sensitive to the precise choice of LD Scores used for the--w-ld flag.

- --out /home/USERNAME/EduYears_Main.munged.hm3.sumstats.h2
    - This again tells LDSC where to place the output file. Once your command has run, call less on EduYears_Main.munged.hm3.sumstats.h2 and use the space bar to scroll through the output produced by LDSC. Here is a truncated example of what this heritability output should look like:

        Total Observed scale h2: 0.1109 (0.0036)
        Lambda GC: 1.4781
        Mean Chi^2: 1.648
        Intercept: 0.9382 (0.0092)
        Ratio < 0 (usually indicates GC correction).
        Analysis finished at Wed Jun 1 14:13:50 2017
        Total time elapsed: 23.62s

LDSC problem

We want to calculate the LDSC correlations between the Add Health educational attainment GWAS you generated in the previous problem and the EA2 results from Okbay et al. Unfortunately, as you might have noticed in your above GWAS results, only the reference allele is listed in the summary statistics (not the other, non-reference allele). However, both alleles are needed to run munge_sumstats.py and use LDSC. To rectify, we used /data/ldsc/w_hm3.snplist to merge in these missing non-reference alleles.

Results including these non-reference alleles can be found in /data/ah_gwas/ah_ea_resid_plink_gwas_omni2_5_euros.assoc.linear.A2. The code used to do this cleaning is /data/ah_code/clean_ah_gwas_add_A2_from_hm3.py.

c. Munge these results (/data/ah_gwas/ah_ea_resid_plink_gwas_omni2_5_euros.assoc.linear.A2) using /data/ldsc/munge_sumstats.py. Be sure to use --merge-alleles /data/ldsc/w_hm3.snplist. Also, you may need to specify what the sample size column is labeled in these summary statistics (check using less or head) with "--N-col LABEL".

   Next, munge /data/ea2/EduYears_Main.txt, with --merge-alleles data/ldsc/w_hm3.snplist. Be sure to specify --N 328917, as you did in part (a).

d. Calculate the heritabilities from each of the two samples using /data/ldsc/ldsc.py, the --h2 flag, and your munged results files. Use the same --ref-ld-chr and --w-ld-chr flags that you did in the example above.

   Verify that you get the following point estimates and standard errors:

   - 0.1109 (SE=0.0036) for Okbay et al.
   - 0.2086 (SE=0.468) for the Omni2.5 Chip

e. Calculate the pairwise genetic correlation between the Okbay et al. and AddHealth Omni2.5 Chip samples, using the --rg flag rather than the --h2 flag. Note that you'll probably need to use the LDSC resources listed above to adapt your heritability code for estimating genetic correlation.

   Finally, calculate the pairwise genetic correlation between the Okbay et al. and AddHealth Omni1 Chip samples. (Remember to munge the GWAS summary statistics for the AddHealth Omni1 Chip data [located in /data/ah_gwas/] first!)

   Verify that the point estimates and standard errors are as follows:

- Between Okbay et al. and Omni2.5: 0.7326 (0.9533)
- Between Okbay et al. and Omni1: 0.8017 (0.1116)

References

Bayarri, M.J., Daniel J. Benjamin, James O. Berger, and Thomas M. Sellke (2016). "Rejection odds and rejection ratios: A proposal for statistical practice in testing hypotheses." *Journal of Mathematical Psychology*, 72, 90–103.

Coffman, Lucas C., and Muriel Niederle (2015). "Pre-analysis plans have limited upside especially where replications are feasible." *Journal of Economic Perspectives*, 29(3), 81–98.

Olken, Benjamin (2015). "Promises and perils of pre-analysis plans." *Journal of Economic Perspectives*, 29(3), 61–80.

Price, Alkes L., et al. (2006). "Principal components analysis corrects for stratification in genome-wide association studies." *Nature Genetics*, 38(8), 904-909.

Wellcome Trust Case-Control Consortium (2007). "Genome-wide association study of 14,000 cases of seven common diseases and 3000 shared controls." *Nature*, 447(7145), 661–678.