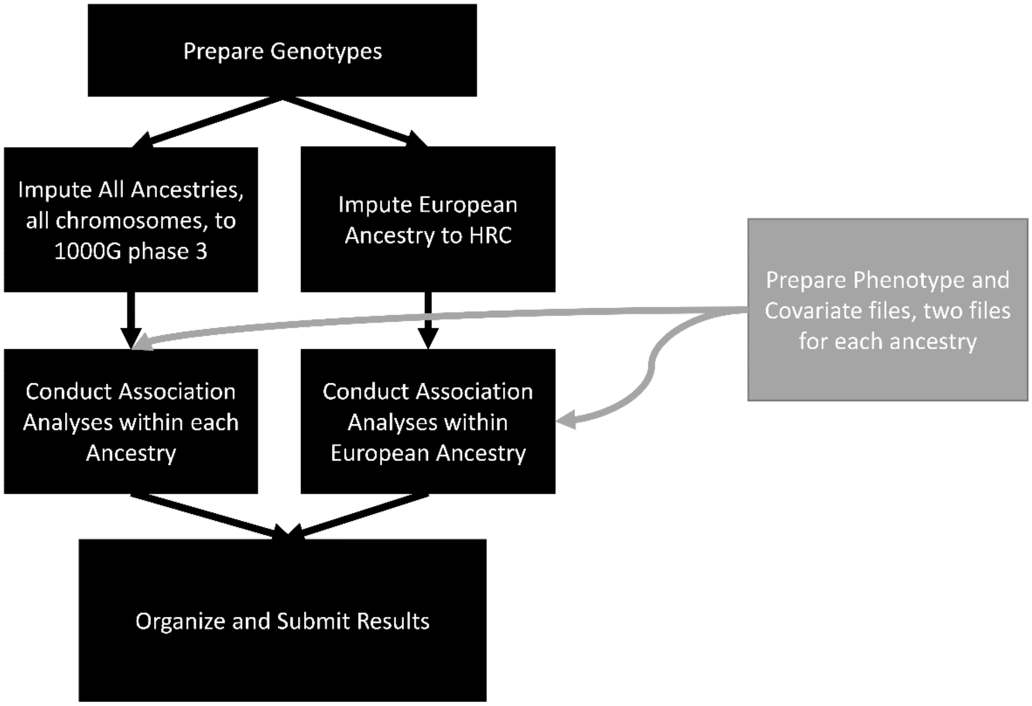
GSCAN GWAS ANALYSIS PLAN, Version 1.2
April 26, 2016

Overview of changes since last version:

a) Modified plan to refer to 8, instead of 7 phenotypes, per changes in the phenotype definitions document.
b) Typo fixes

Overview

There are three major components to this analysis plan. First, genome-wide genotypes must be on the correct build (37/hg19) and correct strand (forward). Second, genotypes must be imputed using the large haplotype panel from the Haplotype Reference Consortium **as well as** 1000 Genomes phase 3 (resulting in two sets of imputed genotypes). Third, association analysis is conducted using specific software tools that will provide all necessary summary statistics for central meta-analysis.


Prepare Genotypes
Impute All Ancestries, all chromosomes, to 1000G phase 3
Impute European Ancestry to HRC
Prepare Phenotype and Covariate files, two files for each ancestry
Conduct Association Analyses within each Ancestry
Conduct Association Analyses within European Ancestry
Organize and Submit Results

The total number of files submitted by any individual study will be

(N ancestries) × (N phenotypes) × (N imputation panels = 2) + (50 QC-related files)

For example, a study that has two ancestries and measured all 8 phenotypes would submit 82 files, unless that study submitted results per-chromosome, in which case the file number would be in excess of 750 files!

As always, there is some chance that there will be an error in the files you submit from this analysis plan. We encourage analysts to organize their scripts, files, and directories just in case re-analysis is required.

Software

All the following software will very likely be needed. SHAPEIT and Minimac are only necessary if you're conducting imputation in-house, rather than using the UMich imputation server.

Needed for generating association summary statistics
rvTests: https://github.com/zhanxw/rvtests
BGZIP and tabix: http://samtools.sourceforge.net/tabix.shtml

Genotypes

Array

All studies must have some version of a genome-wide array, for example with >200,000 genome-wide tag markers. Individual studies will provide information about the manufacturer and version of the array they are using.

What to do with multiple arrays in the same study

Some studies may have data from multiple genotyping arrays, occasionally on the same samples. There are three typical situations:

- No sample overlap: analyze studies separately.
- All samples overlap: you can either a) analyze separately or b) merge genotypes prior to imputation and perform a single analysis.
- Partial, but significant overlap of samples: please contact Scott to customize a plan.
- Finally, please indicate any sample overlap when submitting results.

Genotype QC

We leave calling algorithms, marker filters, and sample filters to the discretion of local sites. Additional QC will be conducted centrally at the meta-analysis stage.

Strand

Strand is an extremely important issue. Without the correct strand meta-analysis doesn't work very well.

All genotypes should be on GRCh37 forward strand. An easy way to get GRCh37 forward strand, if you don't already have it, is to export genotypes from GenomeStudio using TOP allele annotations (typical output from GenomeStudio), which we then ask you to update to the forward strand of build 37 using scripts provided by Will Rayner at Sanger. A description of Illumina's TOP/BOT scheme is here. Will's usage instructions, including scripts, are available here.

Please note: Illumina TOP strand, or Illumina forward strand, ARE NOT the same as GRCh37 forward strand.

Variant Call Format (VCF) files

Nearly all steps in this analysis plan will use VCF files as input and output. If you aren't already using VCF files it's easy to convert from PLINK. Rvtests, which you have already downloaded and installed comes with a plink2vcf function that will convert your PLINK files. An example usage is as follows. Beware, this command can use **a lot of memory**, so keep an eye on that.

Please ensure basic QC has been conducted at this point (e.g., remove all monomorphic variants, variants with low call rates, variants out of Hardy-Weinberg, samples with low call rates, etc.)

```
plink2vcf --inPlink yourplinkfiles --outVcf yourvcffile.vcf
bgzip yourvcffile.vcf
### Note, provide the prefix of your plink file and omit the "bed", "bim" and "fam" suffixes
```

Another easy way to convert your PLINK files to VCF is with PLINK 1.9

```
### PLINK
plink-1.9 --bfile yourplinkfiles --recode vcf-iid --out yourvcffile
bgzip yourvcffile.vcf
```

Either way should work fine with the Michigan Imputation Server.

Finally, if your genotypes are in some format other than PLINK or VCF, and you don't already have a solution for converting them to PLINK or VCF, please contact Scott for advice. (Teemu Palviainen at U Helsinki has a nice step-by-step for bgen files. Scott has the details.)

Note: In VCF format the first allele is referred to as the "reference" allele, and is should to be the same as a human reference genome (in our case, the reference allele from GRCh37). PLINK codes alleles by default as major/minor, which is not the same as reference/alternate. It's not necessary to ensure the reference allele is the same as the reference genome, but if you would like it to be the same you can specify a file for the "—reference" command in plink2vcf, and it will automatically fix the reference allele. Scott (scott.vrieze@colorado.edu) has a mapping of rsID to reference allele he's happy to share.

Analyses will often loop through chromosomes

Many of the steps in this analysis plan can be done chromosome by chromosome (or in smaller chunks – seasoned analysts will no doubt use more sophisticated techniques). This is a good thing! Splitting by chromosome will make everything go faster BUT it will make file management slightly more complex. **Take great care with your filenames and directory structure!**

One very easy way to run by chromosome is with a for loop in bash or c-shell. In bash, running a fake command "yourcommand" across all chromosomes might look like this:

```
for i in {1..22} X; do
    yourcommand --vcf yourfile.chr${i}.vcf --out youroutput.chr${i} >& yourcommand.chr${i}.log &
done
```

(In some cases even looping by chromosome will be too slow (depending on wall-time restrictions for your local cluster or local machine) and analyses can be split into sub-chromosomes quite easily. Ask Scott if you're trying to do this and having questions.)

We will impute to two reference panels: HRC and 1000G

Why, you might ask? Because each reference panel has strengths and weaknesses. The Haplotype Reference Consortium **(HRC)** has assembled over 60,000 haplotypes that are European ancestry (with some non-Europeans primarily from 1000 Genomes). While HRC is great for imputing low-frequency variants into individuals of European ancestry, it is only composed of SNPs and chromosome X is not yet available. 1000 Genomes, on the other hand, will work just as well as HRC for imputing into non-European ancestry individuals AND ALSO has indels, CNVs, and chromosome X. A remaining research question is whether and to what extent HRC improves discovery power and resolution.

To make things as simple as possible for analysis, we request that all studies perform imputation twice.

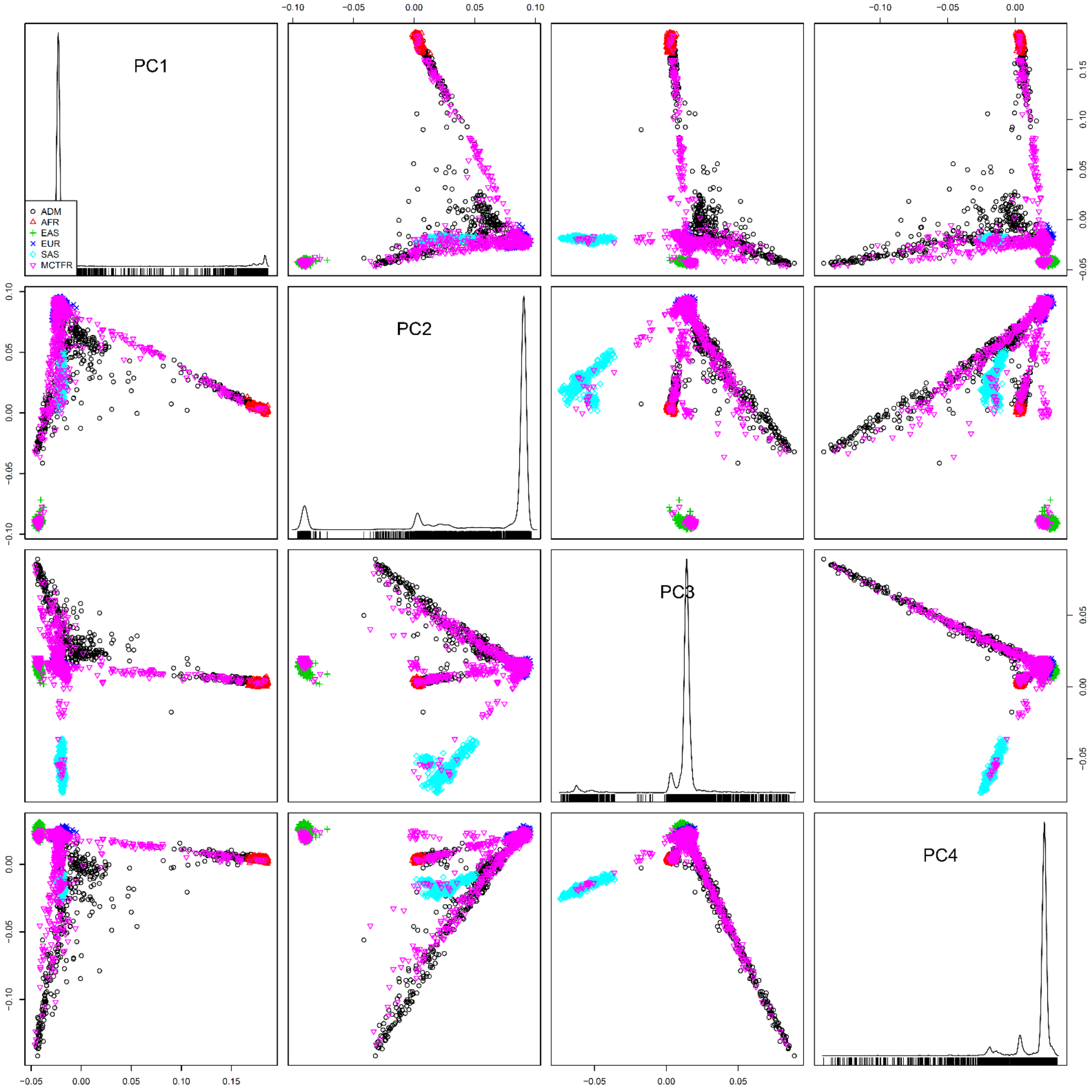
- **<u>First</u>, impute all samples, from all ancestries including European, using 1000 Genomes phase 3**.
- **<u>Second</u>, impute all individuals of European ancestry using the HRC panel.**

Association analyses will be stratified by ancestry

We very much want summary statistics for individuals of all ancestries, including but not limited to African, East Asian, European, Native American, South Asian, Latino/Hispanic, African-American, etc. If you have over 500 samples from one or more of these ancestries, please do include them.

One way to identify ancestry is to group individuals based on PCA projections onto 1000 Genomes PCAs for comparison. If this has not already been done with your sample, please contact Scott with questions. If it's useful to you, a hard-coded script we used to do the 1000 Genomes PCA projection is included in the addendum at the end of this plan. This script was used in the Minnesota Center for

Twin and Family Research to generate the following PCA projection onto 1000G. **Please submit with your results a plot of your samples projected onto 1000 Genomes populations**



Step 1: Imputation

There are two ways to conduct imputation: a) the University of Michigan Imputation Server, or b) in-house imputation. **We recommend that studies use the Imputation Server if possible.**

Step 1a: Imputation Server

The imputation server is available at https://imputationserver.sph.umich.edu. The server uses standard encryption (SSL or SFTP) to help ensure that genotypes are encrypted during upload and download. It does a few things automatically:

1. Ensures we are all imputing to the exact same reference panel
2. Automatically performs quality checks (alleles, MAFs, SNP names, etc.)
3. It phases and imputes, at no cost to the user in either in computational time or analyst time

Using the imputation server (https://imputationserver.sph.umich.edu)

Make an account and follow the instructions on the website https://imputationserver.sph.umich.edu. You will upload VCF genotype files to the server (only VCF file format is supported; instructions on converting to VCF are contained on the imputation server website under "Help"). **One VCF file must be submitted for each chromosome.** The server will automatically phase, impute, and return the imputed genotypes to you. (We prefer that you phase your own genotypes prior to upload, in which case you can control the phasing parameters and the server will only perform imputation.)

If you have a lot of samples to impute (e.g., >6000), please contact Scott at scott.vrieze@colorado.edu to receive increased privileges on the server to submit more samples.

Please run two jobs on the server. For the first job, upload VCF files containing **all genome-wide genotyped individuals**, regardless of ancestry. Select the following options:

- **Reference Panel:** 1000G Phase 3 v5
- **Phasing:** SHAPEIT (If you have not already phased)
- **Population:** Mixed (this parameter is for quality control purposes)
- **Mode:** Quality Control & Imputation

For the second job, upload VCF files containing individuals of **European ancestry only**. Then select the following options:

- **Reference Panel:** HRC
- **Phasing:** SHAPEIT (If you have not already phased)
- **Population:** EUR (this parameter is for quality control purposes)
- **Mode:** Quality Control & Imputation

Tip: How to split your VCF files by Chromosome. To upload your genotypes to the imputation server they will need to be in vcf format (see above for using rvtests for converting PLINK to VCF). Then run this command in a UNIX environment to get 23 files (one per chromosome) that can then be uploaded to the server

```
#Command is for the full sample, including individuals of all ancestries (the "ALL" in the
filenames)
for i in {1..22} X; do
    zgrep "#\|^${i}\s" yourvcffile.ALL.vcf.gz | bgzip -c yourvcffile.ALL.chr${i}.vcf.gz
    tabix -p vcf yourvcffile.ALL.chr${i}.vcf.gz
done

#Command is for individuals of primarily EURopean ancestry (the "EUR" in the filenames)
for i in {1..22} X; do
    zgrep "#\|^${i}\s" yourvcffile.EUR.vcf.gz | bgzip -c yourvcffile.EUR.chr${i}.vcf.gz
    tabix -p vcf yourvcffile.EUR.chr${i}.vcf.gz
done
```

DOWNLOADING YOUR IMPUTED GENOTYPES, INFO files, AND QC REPORT

When imputation has finished you will receive an email alert. The imputation server will automatically encrypt all your imputed genotypes (for protection during download). The password to decrypt the files will be in the email notification, so don't delete that email!

When you download your imputed genotypes please be sure to download all available files (the qcreport, statistics, zip files, and all the log files). We will ask that you submit most of these files to us along with all other files generated as part of this analysis plan.

Please send the qcreport.html and statistics.txt files to Scott for a quick check scott.vrieze@colorado.edu **before proceeding** further in the analysis plan.

NOTE on chromosome X: You will receive two output files for chromosome X, one for males and one for females. Please merge these files into a single vcf for chromosome X using vcf-merge from vcftools.

Step 1b: Imputation In-House

If you have chosen to use the UMich imputation server, then you can ignore this step.

Some studies will not be able to use the imputation server, for example if their original participant consents forbid it. These studies will have to conduct imputation on their own. It is currently unclear which studies will have to use this option, so the current plan is to deal with this on a case-by-case basis, working closely with those studies who must conduct imputation.

The Haplotype Reference Consortium haplotype panel is expected to be released through EGA in late 2015, but a definitive timeline is not in place. Unfortunately, the panel cannot be distributed beforehand for studies to impute in house. We'll update these studies as soon as the panel is available.

Step 2: Define Phenotypes & Covariates

Phenotype and Covariate definitions are described in a separate document, the latest version is available here:
http://gscan.sph.umich.edu/gwas/analysis_plan

Phenotype abbreviations used throughout the example code below are:

- CPD = Cigarettes per day
- SI = Smoking initiation
- SC = Smoking cessation
- AI = Age of smoking initiation
- DPW = Drinks per week
- DND = Drinker versus nondrinker
- BDE = Binge drinking in everyone
- BDL = Binge drinking in lifetime drinkers only (if applicable – see phenotype definitions)

Step 3: Generating Summary Statistics

PLEASE NOTE!

- If your sample is composed of primarily unrelated individuals, proceed to **Step 3a**
- If your sample is composed of a significant number of related individuals (e.g., it is a family study), proceed to **Step 3b**

Step 3a: **UNRELATED** Individuals

Create one ped file for each ancestry group (study_gscan_ANCESTRY_phen.ped)

If your study is composed only of individuals of European ancestry, then you would create only one ped file and call it "study_gscan_EUR_phen.ped". If your study is composed of two ancestry groups, say African-Americans and Europeans, then you would create two ped files and call them "study_gscan_EUR_phen.ped" and "study_gscan_AFR_phen.ped", the first containing only individuals of European ancestry; the second containing only those of African-American ancestry. Repeat this process for other ancestral groups.

Here is an example tab-delimited file with three participants and "x" to denote missing data:

```
fid    iid    patid matid sex    cpd    si     sc     ai     dpw    dnd    bde    bdl
f1     i1     x     x     1      3      2      2      2.71   2.30   2      2      2
f2     i2     x     x     2      x      1      x      x      0      2      1      1
f3     i3     x     x     2      1      2      1      2.83   x      1      1      x
```

Key:
fid = family ID, **iid** = individual ID, **patid** = father ID, **matid** = mother ID
cpd = **cigarettes per day** (binned according to phenotype definitions)
si = **smoking initiation** (2=does/has smoked, 1=denies ever smoking)
sc = **smoking cessation** (2=has quit; 1=has not quit)
ai = **age of initiation of smoking** (normal log of age of initiation)
dpw = **drinks per week** (normal log of reported number of drinks per week)
dnd = **drinker versus non-drinker** (2=drinker, 1=non-drinker)
bde = **binge drinking in everyone** (2=has reported binge drinking, 1=denied binge drinking)
bdl = binge drinking in lifetime drinkers (2=has reported binge drinking, 1=denied binge drinking)

This example is useful because it shows what values you would expect to have in your pedigree file if you followed the phenotype definition and scale transformations correctly from the Phenotype Definition document.

In this example individual i1 is

- Male (**sex = 1**),
- a FORMER smoker (**si=2**; **sc=2**) who smokes 16-25 cigarettes per day (**cpd = 3**),
- started smoking at 15 [**ai = ln(15) = 2.71**], and
- has 10 drinks per week [**dpw = ln(10) = 2.30**; **dnd=2**]
- and reports binge drinking (**bde=2, bdl=2**)

Individual i2 is

- female (**sex = 2**),
- a lifelong nonsmoker (**cpd = x**; **sc = x**; **ai = x**; **si = 2**),
- drinks 1 drink per week [**dpw = ln(1) = 0; dnd = 2**],
- and denies binge drinking (**bde=1, bdl=1**)

Individual i3 is

- female (**sex = 2**),
- a CURRENT smoker (**si=2**; **sc=1**) who smokes 1-5 cigarettes per day (**cpd = 1**),

- started smoking at age 17 [**ai = ln(17) = 2.83**],
- and denies drinking alcoholic beverages (**dpw = x**, **dnd = 1**; **bde =1, bdl=x**)

Create one covariate file for each ancestry group (study_gscan_ANCESTRY_cov.ped)

For each phenotype file you create you will also create a covariates file, one for each ancestry group in your study.

Here is an example with fake data for individuals i1 and i2:

```
fid   iid   patid matid sex   age   age2  PC1   PC2   PC3 ... (additional covariates)
f1    i1    x     x     1     25    625   1.2   0.8   0.9 ... (additional covariates)
f2    i2    x     x     2     40    1600  0.4   0.5   1.0 ... (additional covariates)
f3    i3    x     x     2     59    3481  -0.3  1.2   1.4 ... (additional covariates)
```

*Again, missing values are denoted as "x". **age2** = age squared, **PC[1-3]** = genetic principal components (if applicable)

Analysis of 1000 genomes imputed genotypes: Run rvTests for each ancestry and trait separately

Example commands for individuals of European ancestry imputed with 1000 Genomes phase 3. Note that there are two separate commands for continuous (e.g., CPD) and binary traits (e.g., SI).

```
####################################################
### 1000 Genomes imputation Association Analyses ###
####################################################
### CONTINUOUS TRAITS (cpd, ai, dpw)
ancestry=EUR #replace this as needed with the appropriate ancestry
imputation_version=i1000G
for cont_trait in cpd ai dpw; do #Loop over phenotypes
                            #Include only the continuous traits in your phenotype pedigree file
    for i in {1..22} X; do #Loop over chromosomes
        rvtest --inVcf your${imputation_version}vcffile.chr${i}.vcf.gz \  #vcf ( 1000G-imputed)
            --pheno study_gscan_${ancestry}_phen.ped \  #Input phenotype ped file
            --pheno-name ${cont_trait} \      #Name of phenotype (cpd in this case)
            --covar study_gscan_${ancestry}_cov.ped  \  #Name of covariate file
            --meta score \                    #Generate score stats for meta-analysis
            --covar-name sex,age,age2,PC1,PC2,PC3,PC4,PC5,PC6,PC7,PC8,PC9,PC10 \
            --xLabel X  \                     #Label used for X-chromosome ("X")
            --useResidualAsPhenotype         #Residualize before testing variants
            --inverseNormal \                 #Inverse normalize the resid distr.
            --qtl \                           #Specify pheno is continuously distr.
            --dosage DS \                     #Specify vcf dosage field (here EC)
            --out  STUDY_${ancestry}_${imputation_version}_${cont_trait}_chr${i} &  #Specify
output file name (here assuming
                                                #EURopean ancestry and cigarettes per day)
    done
    wait # this wait command will restrict the command to running 23 jobs at a time, maximum
done
###
###BINARY traits (si sc dnd bd)
for binary_trait in si sc dnd bd; do #Loop over phenotypes
                              #Include only those binary traits in your phenotype pedigree file
    for i in {1..22} X; do
        rvtest --inVcf your${imputation_version}vcffile.chr${i}.vcf.gz \
            --pheno study_gscan_${ancestry}_phen.ped \
            --pheno-name ${binary_trait} \
            --covar study_gscan_${ancestry}_cov.ped \
            --meta score \
            --covar-name sex,age,age2,PC1,PC2,PC3,PC4,PC5,PC6,PC7,PC8,PC9,PC10 \
            --xLabel X \
            --dosage DS \
            --out STUDY_${ancestry}_${imputation_version}_${binary_trait}_chr${i} &
    done
    wait # this wait command will restrict the command to running 23 jobs at a time, maximum
done
```

The following example command concatenates per-chromosome results into one output file for each ancestry x trait combination. Our hope is that having fewer numbers of files will make file management and transfer easier.

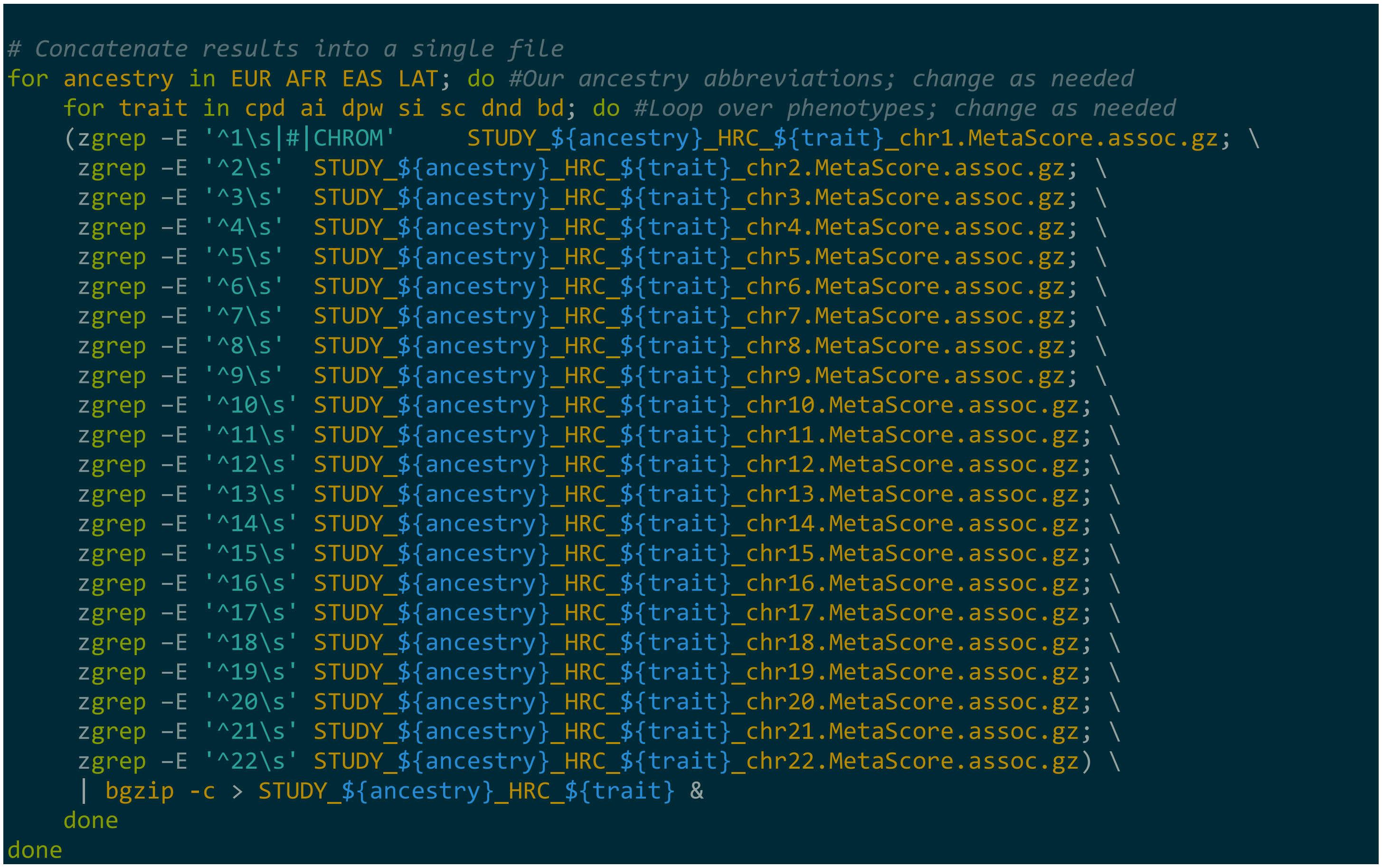
```
# Concatenate results into a single file
for ancestry in EUR AFR EAS LAT; do #Our ancestry abbreviations; change as needed
    for trait in cpd ai dpw si sc dnd bd; do #Loop over phenotypes; change as needed
    (zgrep -E '^1\s|#|CHROM' STUDY_${ancestry}_1000G_${trait}_chr1.MetaScore.assoc.gz; \
        zgrep -E '^2\s'  STUDY_${ancestry}_1000G_${trait}_chr2.MetaScore.assoc.gz; \
        zgrep -E '^3\s'  STUDY_${ancestry}_1000G_${trait}_chr3.MetaScore.assoc.gz; \
        zgrep -E '^4\s'  STUDY_${ancestry}_1000G_${trait}_chr4.MetaScore.assoc.gz; \
        zgrep -E '^5\s'  STUDY_${ancestry}_1000G_${trait}_chr5.MetaScore.assoc.gz; \
        zgrep -E '^6\s'  STUDY_${ancestry}_1000G_${trait}_chr6.MetaScore.assoc.gz; \
        zgrep -E '^7\s'  STUDY_${ancestry}_1000G_${trait}_chr7.MetaScore.assoc.gz; \
        zgrep -E '^8\s'  STUDY_${ancestry}_1000G_${trait}_chr8.MetaScore.assoc.gz; \
        zgrep -E '^9\s'  STUDY_${ancestry}_1000G_${trait}_chr9.MetaScore.assoc.gz; \
        zgrep -E '^10\s' STUDY_${ancestry}_1000G_${trait}_chr10.MetaScore.assoc.gz; \
        zgrep -E '^11\s' STUDY_${ancestry}_1000G_${trait}_chr11.MetaScore.assoc.gz; \
        zgrep -E '^12\s' STUDY_${ancestry}_1000G_${trait}_chr12.MetaScore.assoc.gz; \
        zgrep -E '^13\s' STUDY_${ancestry}_1000G_${trait}_chr13.MetaScore.assoc.gz; \
        zgrep -E '^14\s' STUDY_${ancestry}_1000G_${trait}_chr14.MetaScore.assoc.gz; \
        zgrep -E '^15\s' STUDY_${ancestry}_1000G_${trait}_chr15.MetaScore.assoc.gz; \
        zgrep -E '^16\s' STUDY_${ancestry}_1000G_${trait}_chr16.MetaScore.assoc.gz; \
        zgrep -E '^17\s' STUDY_${ancestry}_1000G_${trait}_chr17.MetaScore.assoc.gz; \
        zgrep -E '^18\s' STUDY_${ancestry}_1000G_${trait}_chr18.MetaScore.assoc.gz; \
        zgrep -E '^19\s' STUDY_${ancestry}_1000G_${trait}_chr19.MetaScore.assoc.gz; \
        zgrep -E '^20\s' STUDY_${ancestry}_1000G_${trait}_chr20.MetaScore.assoc.gz; \
        zgrep -E '^21\s' STUDY_${ancestry}_1000G_${trait}_chr21.MetaScore.assoc.gz; \
        zgrep -E '^22\s' STUDY_${ancestry}_1000G_${trait}_chr22.MetaScore.assoc.gz; \
        zgrep -E '^X\s'  STUDY_${ancestry}_1000G_${trait}_chrX.MetaScore.assoc.gz) \
        | bgzip -c > STUDY_${ancestry}_1000G_${trait} &
    done
done
```

Analysis of HRC imputed genotypes: Run rvTests for each ancestry and trait separately

Example commands for individuals of European ancestry imputed with HRC. Note there are separate commands for continuous and binary traits.

```
######################
### HRC imputation ###
######################
### CONTINUOUS TRAITS (cpd, ai, dpw)
ancestry=EUR #replace this as needed with the appropriate ancestry
imputation_version=HRC
for cont_trait in cpd ai dpw; do #Loop over phenotypes
                            #Include only the continuous traits in your phenotype pedigree file
    for i in {1..22}; do #Loop over chromosomes
        rvtest --inVcf your${imputation_version}vcffile.chr${i}.vcf.gz \  #Input vcf (in this case
HRC-imputed)
            --pheno study_gscan_${ancestry}_phen.ped \  #Input phenotype ped file
            --pheno-name ${cont_trait} \      #Name of phenotype (cpd in this case)
            --covar study_gscan_${ancestry}_cov.ped  \  #Name of covariate file
            --meta score \                    #Generate score stats for meta-analysis
            --covar-name sex,age,age2,PC1,PC2,PC3,PC4,PC5,PC6,PC7,PC8,PC9,PC10 \
            --useResidualAsPhenotype         #Residualize before testing variants
            --inverseNormal \                 #Inverse normalize the resid distr.
            --qtl \                           #Specify pheno is continuously distr.
            --dosage DS \                     #Specify vcf dosage field (here EC)
            --out  STUDY_${ancestry}_${imputation_version}_${cont_trait}_chr${i} &
    done
    wait # this wait command will restrict the command to running 23 jobs at a time, maximum
done
###
###BINARY traits (si sc dnd bd)
for binary_trait in si sc dnd bd; do #Loop over phenotypes
                              #Include only those binary traits in your phenotype pedigree file
    for i in {1..22}; do
        rvtest --inVcf your${imputation_version}vcffile.chr${i}.vcf.gz \
            --pheno study_gscan_${ancestry}_phen.ped \
            --pheno-name ${binary_trait} \
            --covar study_gscan_${ancestry}_cov.ped \
            --meta score \
            --covar-name sex,age,age2,PC1,PC2,PC3,PC4,PC5,PC6,PC7,PC8,PC9,PC10 \
            --dosage DS \
            --out STUDY_${ancestry}_${imputation_version}_${binary_trait}_chr${i} &
    done
    wait # this wait command will restrict the command to running 23 jobs at a time, maximum
done
```

The following example command concatenates per-chromosome results for HRC-imputed files into one output file for each ancestry x trait combination. Our hope is that having fewer numbers of files will make file management and transfer easier.



```
# Concatenate results into a single file
for ancestry in EUR AFR EAS LAT; do #Our ancestry abbreviations; change as needed
    for trait in cpd ai dpw si sc dnd bd; do #Loop over phenotypes; change as needed
    (zgrep -E '^1\s|#|CHROM'     STUDY_${ancestry}_HRC_${trait}_chr1.MetaScore.assoc.gz; \
     zgrep -E '^2\s'  STUDY_${ancestry}_HRC_${trait}_chr2.MetaScore.assoc.gz; \
     zgrep -E '^3\s'  STUDY_${ancestry}_HRC_${trait}_chr3.MetaScore.assoc.gz; \
     zgrep -E '^4\s'  STUDY_${ancestry}_HRC_${trait}_chr4.MetaScore.assoc.gz; \
     zgrep -E '^5\s'  STUDY_${ancestry}_HRC_${trait}_chr5.MetaScore.assoc.gz; \
     zgrep -E '^6\s'  STUDY_${ancestry}_HRC_${trait}_chr6.MetaScore.assoc.gz; \
     zgrep -E '^7\s'  STUDY_${ancestry}_HRC_${trait}_chr7.MetaScore.assoc.gz; \
     zgrep -E '^8\s'  STUDY_${ancestry}_HRC_${trait}_chr8.MetaScore.assoc.gz; \
     zgrep -E '^9\s'  STUDY_${ancestry}_HRC_${trait}_chr9.MetaScore.assoc.gz; \
     zgrep -E '^10\s' STUDY_${ancestry}_HRC_${trait}_chr10.MetaScore.assoc.gz; \
     zgrep -E '^11\s' STUDY_${ancestry}_HRC_${trait}_chr11.MetaScore.assoc.gz; \
     zgrep -E '^12\s' STUDY_${ancestry}_HRC_${trait}_chr12.MetaScore.assoc.gz; \
     zgrep -E '^13\s' STUDY_${ancestry}_HRC_${trait}_chr13.MetaScore.assoc.gz; \
     zgrep -E '^14\s' STUDY_${ancestry}_HRC_${trait}_chr14.MetaScore.assoc.gz; \
     zgrep -E '^15\s' STUDY_${ancestry}_HRC_${trait}_chr15.MetaScore.assoc.gz; \
     zgrep -E '^16\s' STUDY_${ancestry}_HRC_${trait}_chr16.MetaScore.assoc.gz; \
     zgrep -E '^17\s' STUDY_${ancestry}_HRC_${trait}_chr17.MetaScore.assoc.gz; \
     zgrep -E '^18\s' STUDY_${ancestry}_HRC_${trait}_chr18.MetaScore.assoc.gz; \
     zgrep -E '^19\s' STUDY_${ancestry}_HRC_${trait}_chr19.MetaScore.assoc.gz; \
     zgrep -E '^20\s' STUDY_${ancestry}_HRC_${trait}_chr20.MetaScore.assoc.gz; \
     zgrep -E '^21\s' STUDY_${ancestry}_HRC_${trait}_chr21.MetaScore.assoc.gz; \
     zgrep -E '^22\s' STUDY_${ancestry}_HRC_${trait}_chr22.MetaScore.assoc.gz) \
     | bgzip -c > STUDY_${ancestry}_HRC_${trait} &
    done
done
```

Step 3b: Sample of **RELATED** Individuals (e.g., families)

CREATE PHENOTYPE/COVARIATE FILES (study_gscan_ANCESTRY_phen.ped & study_gscan_ANCESTRY_cov.ped)

Follow the instruction under Step 3a to create these files.

Generate kinship matrices

To account for familial relatedness and population stratification rvtests uses an empirical kinship matrix. We need only generate this kinship matrix once, on the full 1000 Genomes imputed VCF files. That matrix can then be used in all association analyses.

rvTests generates an empirical kinship matrix from the VCF file. Within the rvtest folder there is a script called "vcf2kinship". However, we want to run it on all common markers genome-wide, so we first must concatenate

```
### Concatenate per-chromosome imputed genotypes VCF file into single VCF file
(zcat yourvcffile.1000Gimputed.chr1.vcf.gz;
    zgrep -v '^#' yourvcffile.1000Gimputed.chr2.vcf.gz; \
    zgrep -v '^#' yourvcffile.1000Gimputed.chr3.vcf.gz; \
    zgrep -v '^#' yourvcffile.1000Gimputed.chr4.vcf.gz; \
    zgrep -v '^#' yourvcffile.1000Gimputed.chr5.vcf.gz; \
    zgrep -v '^#' yourvcffile.1000Gimputed.chr6.vcf.gz; \
    zgrep -v '^#' yourvcffile.1000Gimputed.chr7.vcf.gz; \
    zgrep -v '^#' yourvcffile.1000Gimputed.chr8.vcf.gz; \
    zgrep -v '^#' yourvcffile.1000Gimputed.chr9.vcf.gz; \
    zgrep -v '^#' yourvcffile.1000Gimputed.chr10.vcf.gz; \
    zgrep -v '^#' yourvcffile.1000Gimputed.chr11.vcf.gz; \
    zgrep -v '^#' yourvcffile.1000Gimputed.chr12.vcf.gz; \
    zgrep -v '^#' yourvcffile.1000Gimputed.chr13.vcf.gz; \
    zgrep -v '^#' yourvcffile.1000Gimputed.chr14.vcf.gz; \
    zgrep -v '^#' yourvcffile.1000Gimputed.chr15.vcf.gz; \
    zgrep -v '^#' yourvcffile.1000Gimputed.chr16.vcf.gz; \
    zgrep -v '^#' yourvcffile.1000Gimputed.chr17.vcf.gz; \
    zgrep -v '^#' yourvcffile.1000Gimputed.chr18.vcf.gz; \
    zgrep -v '^#' yourvcffile.1000Gimputed.chr19.vcf.gz; \
    zgrep -v '^#' yourvcffile.1000Gimputed.chr20.vcf.gz; \
    zgrep -v '^#' yourvcffile.1000Gimputed.chr21.vcf.gz; \
    zgrep -v '^#' yourvcffile.1000Gimputed.chr22.vcf.gz; \
    zgrep -v '^#' yourvcffile.1000Gimputed.chrX.vcf.gz) \
    | bgzip -c > yourvcffile.1000Gimputed.chrALL.vcf.gz


### Generate kinship matrix (--threads controls the number of parallel threads, adjust as needed)
vcf2kinship --inVcf yourvcffile.1000Gimputed.chrALL.vcf.gz \
    --bn \  #balding-nichols method
    --out kinship_matrix \  #output file name prefix
    --xLabel X \  #Label we used for the X chromosome
    --xHemi \  #create kinship for hemizygous region
    --minMAF .05 \  #min MAF of variants that contribute to the kinship matrix
    --threads 12
```

Analysis of 1000 genomes imputed genotypes: Run rvTests for each ancestry and trait separately

Example commands for individuals of East Asian ancestry imputed with 1000 Genomes phase 3. Note that there are two separate commands for continuous and binary traits.

```
########################
### 1000G imputation ###
########################
### CONTINUOUS AND BINARY TRAITS USE SAME COMMAND
ancestry=EAS #replace this as needed with the appropriate ancestry
imputation_version=1000G
for trait in cpd ai dpw si sc dnd bd; do #Loop over phenotypes
    for chr in {1..22} X; do #Loop over chromosomes
        rvtest --inVcf your${imputation_version}vcffile.chr${i}.vcf.gz \  #Input vcf
            --pheno study_gscan_${ancestry}_phen.ped \    #Input phenotype ped file
            --pheno-name ${trait} \                       #Name of phenotype
            --covar study_gscan_${ancestry}_cov.ped  \    #Name of covariate file
            --meta score \                                #Generate score stats for meta-analysis
            --covar-name sex,age,age2,PC1,PC2,PC3,PC4,PC5,PC6,PC7,PC8,PC9,PC10 \
            --kinship kinship_matrix.kinship \            #Kinship filename
            --xLabel X \                                  #Label for X chromosome in vcf file
            --xHemiKinship kinship_matrix.xHemiKinship \  #Kinship in hemizygous region
            --useResidualAsPhenotype \                    #Residualize before testing variants
            --inverseNormal \                             #Inverse normalize the resid distr.
            --qtl \                                       #Force analysis as if pheno is continuous
            --dosage DS \                                 #Specify vcf dosage field (here EC)
            --out  STUDY_${ancestry}_${imputation_version}_${trait}_chr${i} &
    done
    wait
done
```

Analysis of HRC imputed genotypes: Run rvTests for each ancestry and trait separately

Example commands for individuals of European ancestry imputed with HRC. Note there are separate commands for continuous and binary traits.

```
######################
### HRC imputation ###
######################
### CONTINUOUS AND BINARY TRAITS USE SAME COMMAND
ancestry=EUR #replace this as needed with the appropriate ancestry
imputation_version=HRC
for trait in cpd ai dpw si sc dnd bd; do #Loop over phenotypes
    for chr in {1..22} X; do #Loop over chromosomes
        rvtest --inVcf your${imputation_version}vcffile.chr${i}.vcf.gz \  #Input vcf
            --pheno study_gscan_${ancestry}_phen.ped \            #Input phenotype ped file
            --pheno-name ${trait} \                   #Name of phenotype (cpd in this case)
            --covar study_gscan_${ancestry}_cov.ped  \            #Name of covariate file
            --meta score \                            #Generate score stats for meta-analysis
            --covar-name sex,age,age2,PC1,PC2,PC3,PC4,PC5,PC6,PC7,PC8,PC9,PC10 \
            --kinship kinship_matrix.kinship \            #Kinship filename
            --useResidualAsPhenotype                  #Residualize before testing variants
            --inverseNormal \                         #Inverse normalize the resid distr.
            --qtl \                                   #Force analysis as if pheno is continuous
            --dosage DS \                            #Specify vcf dosage field (here EC)
            --out  STUDY_${ancestry}_${imputation_version}_${trait}_chr${i} &
    done
    wait
done
```

CONCATENATING RESULTS INTO ONE OUTPUT FILE PER PHENOTYPE

The following example command concatenates per-chromosome results for HRC-imputed files into one output file for each ancestry x trait combination. Our hope is that having fewer numbers of files will make file management and transfer easier.

```
# Concatenate results into a single file
for ancestry in EUR AFR EAS LAT; do #Our ancestry abbreviations; change as needed
    for trait in cpd ai dpw si sc dnd bd; do #Loop over phenotypes; change as needed
    (zgrep –E '^1\s|#|CHROM'     STUDY_${ancestry}_HRC_${trait}_chr1.MetaScore.assoc.gz; \
     zgrep –E '^2\s'  STUDY_${ancestry}_HRC_${trait}_chr2.MetaScore.assoc.gz; \
     zgrep –E '^3\s'  STUDY_${ancestry}_HRC_${trait}_chr3.MetaScore.assoc.gz; \
     zgrep –E '^4\s'  STUDY_${ancestry}_HRC_${trait}_chr4.MetaScore.assoc.gz; \
     zgrep –E '^5\s'  STUDY_${ancestry}_HRC_${trait}_chr5.MetaScore.assoc.gz; \
     zgrep –E '^6\s'  STUDY_${ancestry}_HRC_${trait}_chr6.MetaScore.assoc.gz; \
     zgrep –E '^7\s'  STUDY_${ancestry}_HRC_${trait}_chr7.MetaScore.assoc.gz; \
     zgrep –E '^8\s'  STUDY_${ancestry}_HRC_${trait}_chr8.MetaScore.assoc.gz; \
     zgrep –E '^9\s'  STUDY_${ancestry}_HRC_${trait}_chr9.MetaScore.assoc.gz; \
     zgrep –E '^10\s' STUDY_${ancestry}_HRC_${trait}_chr10.MetaScore.assoc.gz; \
     zgrep –E '^11\s' STUDY_${ancestry}_HRC_${trait}_chr11.MetaScore.assoc.gz; \
     zgrep –E '^12\s' STUDY_${ancestry}_HRC_${trait}_chr12.MetaScore.assoc.gz; \
     zgrep –E '^13\s' STUDY_${ancestry}_HRC_${trait}_chr13.MetaScore.assoc.gz; \
     zgrep –E '^14\s' STUDY_${ancestry}_HRC_${trait}_chr14.MetaScore.assoc.gz; \
     zgrep –E '^15\s' STUDY_${ancestry}_HRC_${trait}_chr15.MetaScore.assoc.gz; \
     zgrep –E '^16\s' STUDY_${ancestry}_HRC_${trait}_chr16.MetaScore.assoc.gz; \
     zgrep –E '^17\s' STUDY_${ancestry}_HRC_${trait}_chr17.MetaScore.assoc.gz; \
     zgrep –E '^18\s' STUDY_${ancestry}_HRC_${trait}_chr18.MetaScore.assoc.gz; \
     zgrep –E '^19\s' STUDY_${ancestry}_HRC_${trait}_chr19.MetaScore.assoc.gz; \
     zgrep –E '^20\s' STUDY_${ancestry}_HRC_${trait}_chr20.MetaScore.assoc.gz; \
     zgrep –E '^21\s' STUDY_${ancestry}_HRC_${trait}_chr21.MetaScore.assoc.gz; \
     zgrep –E '^22\s' STUDY_${ancestry}_HRC_${trait}_chr22.MetaScore.assoc.gz) \
     | bgzip -c > STUDY_${ancestry}_HRC_${trait} &
    done
done
```

Step 4. Upload Results

Congratulations! You made it! Please upload to sftp server at the University of Michigan for central analysis -- please email Scott (scott.vrieze@colorado.edu) for the hostname, username, and password.

Rename all of the following files and place them in a single directory

Imputation Server Log Files. You can find these under the "Results" tab on the imputation server.
qcreport.html (2 of these, one for HRC and one for 1000G)
statistics.txt (2 of these, one for HRC and one for 1000G)

Imputation Info Files. You can find these among the files downloaded from the imputation server, with the name chr1.info.gz (for example). Please upload both the male and female chromosome X info files. There will be 22 HRC info files and 24 1000G info files.

PCA Plot. Please submit a projection of your samples onto 1000 Genomes PCA space.

rvTests. Please submit each of the results files. The total number will be a function of the number of ancestries times the number of phenotypes times the number of imputation panels (=2). Use the following naming convention.

STUDY_ANCESTRY_IMPUTATION_TRAIT_DDMMYY_INITIALS.MetaScore.assoc.gz

README
STUDY_DDMMYY_INITIALS.readme

Please submit the README file with the following information:

1. Name, email
2. Study name
3. Array version(s)
4. Server versus in-house imputation
5. Basic information about genotype calling procedures (e.g., GenomeStudio, etc.)
6. The actual survey questions and recall periods (last year, last month, period of heaviest drinking, etc.) used to build the phenotypes and any irregularities encountered in phenotype definition
7. Other concerns or uncertainties that arose during the analysis

Key:
STUDY = your study name (please also add any strata - e.g., COGA_AfricanAmerican)
ANCESTRY = AFR, EUR, EAS, SAS, LAT, AME (for African, European, East Asian, South Asian, Latino, and (Native) American respectively)
TRAIT = CPD, AI, SI, SC, DPW, DND, BDE, BDL (if applicable)
DDMMYY = Day, Month, and Year of submission (January 1, 2016 would be "010116")
INITIALS = initials of the analyst

Tarball the directory and transfer to sftp server (ask Scott for log-in details)

Please place all relevant output files into a folder, make a zipped tarball, and upload it. Let Scott know when you've done so.

```
### Make tarball to hold all the results
tar -zcvf STUDY_DDMMYY_INITIALS.tar.gz  yourresultsdirectory
```

Addendum: Sample R commands for PCA projection onto 1000 Genomes

```
### This is an R script, but some of the following shell commands may be useful to you.
### Get 1000 genomes phase3 autosomal vcfs (run in bash in termal window)
# for i in {1..22}; do
#       wget ftp://ftp-
trace.ncbi.nih.gov/1000genomes/ftp/release/20130502/ALL.chr${i}.phase3_shapeit2_mvncall_integrated
_v5a.20130502.genotypes.vcf.gz
# done

### Extract site list from my target genotypes (run in bash in terminal window)
# for i in {1..22}; do
#      zgrep "^${i}" MCTFR.vcf.gz | cut -f-2 > chr${i}.MCTFR_sites.txt &
# done

### Command to extract relevant SNPs from 1000 genomes phase three (run in bash in terminal window
# for i in {1..22}; do
#      vcftools --gzvcf
ALL.chr${i}.phase3_shapeit2_mvncall_integrated_v5a.20130502.genotypes.vcf.gz \
#               --positions chr${i}.MCTFR_sites.txt \
#               --recode \
#               --recode-INFO-all \
#               --out chr${i}.1000g.MCTFR_sites &
# done


system("(cat chr1.1000g.MCTFR_sites.recode.vcf;   grep -v '#' chr2.1000g.MCTFR_sites.recode.vcf;
grep -v '#' chr3.1000g.MCTFR_sites.recode.vcf;    grep -v '#' chr4.1000g.MCTFR_sites.recode.vcf;
grep -v '#' chr5.1000g.MCTFR_sites.recode.vcf;     grep -v '#' chr6.1000g.MCTFR_sites.recode.vcf;
grep -v '#' chr7.1000g.MCTFR_sites.recode.vcf;     grep -v '#' chr8.1000g.MCTFR_sites.recode.vcf;
grep -v '#' chr9.1000g.MCTFR_sites.recode.vcf;     grep -v '#' chr10.1000g.MCTFR_sites.recode.vcf;
grep -v '#' chr11.1000g.MCTFR_sites.recode.vcf;     grep -v '#'
chr12.1000g.MCTFR_sites.recode.vcf;     grep -v '#' chr13.1000g.MCTFR_sites.recode.vcf;     grep -
v '#' chr14.1000g.MCTFR_sites.recode.vcf;      grep -v '#' chr15.1000g.MCTFR_sites.recode.vcf;
grep -v '#' chr16.1000g.MCTFR_sites.recode.vcf;   grep -v '#' chr17.1000g.MCTFR_sites.recode.vcf;
grep -v '#' chr18.1000g.MCTFR_sites.recode.vcf;     grep -v '#'
chr19.1000g.MCTFR_sites.recode.vcf;     grep -v '#' chr20.1000g.MCTFR_sites.recode.vcf;     grep -
v '#' chr21.1000g.MCTFR_sites.recode.vcf;     grep -v '#' chr22.1000g.MCTFR_sites.recode.vcf) |
bgzip -c > chrALL.1000g.MCTFR_sites.recode.vcf.gz")

### Calculate principal components using plink (Careful! MAF filter is applied after the SNPs are
thinned to 30,000)
system("plink-1.9   --vcf chrALL.1000g.MCTFR_sites.recode.vcf.gz  --indep-pairwise 100kb 5 .1")
system("plink-1.9   --vcf chrALL.1000g.MCTFR_sites.recode.vcf.gz  --extract plink.prune.in --pca
var-wts --make-bed --out chrALL.1000g.MCTFR_sites.pruned")


### Extract per-SNP weights for each of the first 20 PCs and get the correct effect allele for
later scoring
```

```
PCA <- read.table("chrALL.1000g.MCTFR_sites.pruned.eigenvec.var", header=F, col.names=c("CHROM",
"RS", paste("PC", 1:20, sep="")))
bim <- read.table("chrALL.1000g.MCTFR_sites.pruned.bim", header=F, col.names=c("CHROM", "RS",
"unknown", "POS", "A1", "A2"))
x <- merge(PCA, bim)
write.table(subset(x, select=c("RS","A1",paste("PC",1:20, sep=""))), file="PCA-score-file.txt",
quote=F)
### End: extract per-SNP weights for each of the first 20 PCs


### Score 1000g subjects on the 1000g-defined PC1 and 2
system("plink-1.9 --vcf chrALL.1000g.MCTFR_sites.recode.vcf.gz --score PCA-score-file.txt 2 3 4
header --out 1000g-PC1-score")
system("plink-1.9 --vcf chrALL.1000g.MCTFR_sites.recode.vcf.gz --score PCA-score-file.txt 2 3 5
header --out 1000g-PC2-score")
system("plink-1.9 --vcf chrALL.1000g.MCTFR_sites.recode.vcf.gz --score PCA-score-file.txt 2 3 6
header --out 1000g-PC3-score")
system("plink-1.9 --vcf chrALL.1000g.MCTFR_sites.recode.vcf.gz --score PCA-score-file.txt 2 3 7
header --out 1000g-PC4-score")


### Score MCTFR subjects on the 1000g-defined PC1 and 2
system("plink-1.9 --vcf ../../MCTFR_imputation/raw_genotypes/gedi5-660WQuad-b37-forwardstrand-
correctreferenceallele-final-vcf.vcf.gz --score PCA-score-file.txt 2 3 4 header --out MCTFR-PC1-
score")
system("plink-1.9 --vcf ../../MCTFR_imputation/raw_genotypes/gedi5-660WQuad-b37-forwardstrand-
correctreferenceallele-final-vcf.vcf.gz --score PCA-score-file.txt 2 3 5 header --out MCTFR-PC2-
score")
system("plink-1.9 --vcf ../../MCTFR_imputation/raw_genotypes/gedi5-660WQuad-b37-forwardstrand-
correctreferenceallele-final-vcf.vcf.gz --score PCA-score-file.txt 2 3 6 header --out MCTFR-PC3-
score")
system("plink-1.9 --vcf ../../MCTFR_imputation/raw_genotypes/gedi5-660WQuad-b37-forwardstrand-
correctreferenceallele-final-vcf.vcf.gz --score PCA-score-file.txt 2 3 7 header --out MCTFR-PC4-
score")

### Read in scores for PC1 and PC2 in 1000g
kg_PC1 <- read.table("1000g-PC1-score.profile", header=T)
kg_PC2 <- read.table("1000g-PC2-score.profile", header=T)
kg_PC3 <- read.table("1000g-PC3-score.profile", header=T)
kg_PC4 <- read.table("1000g-PC4-score.profile", header=T)

### "Self-reported" ancestry of 1000g participants
kg_sf <- read.table("20130502.sequence.index", header=T, sep="\t", fill=T, stringsAsFactors=F)
sample_ids <- unique(data.frame(IID=kg_sf$SAMPLE_NAME, POPULATION=kg_sf$POPULATION,
stringsAsFactors=F))
sample_ids$CONTINENT <- ifelse(sample_ids$POPULATION=="CHB" |
                               sample_ids$POPULATION=="JPT" |
                               sample_ids$POPULATION=="CHS" |
                               sample_ids$POPULATION=="CDX" |
                               sample_ids$POPULATION=="KHV" |
                               sample_ids$POPULATION=="CHD", "EAS", sample_ids$POPULATION)
sample_ids$CONTINENT <- ifelse(sample_ids$CONTINENT=="CEU" |
                               sample_ids$CONTINENT=="TSI" |
                               sample_ids$CONTINENT=="GBR" |
                               sample_ids$CONTINENT=="FIN" |
                               sample_ids$CONTINENT=="IBS", "EUR", sample_ids$CONTINENT)
sample_ids$CONTINENT <- ifelse(sample_ids$CONTINENT=="YRI" |
                               sample_ids$CONTINENT=="LWK" |
                               sample_ids$CONTINENT=="GWD" |
                               sample_ids$CONTINENT=="MSL" |
                               sample_ids$CONTINENT=="ESN", "AFR", sample_ids$CONTINENT)
sample_ids$CONTINENT <- ifelse(sample_ids$CONTINENT=="ASW" |
                               sample_ids$CONTINENT=="ACB" |
                               sample_ids$CONTINENT=="MXL" |
                               sample_ids$CONTINENT=="PUR" |
                               sample_ids$CONTINENT=="CLM" |
                               sample_ids$CONTINENT=="PEL", "ADM", sample_ids$CONTINENT)
sample_ids$CONTINENT <- ifelse(sample_ids$CONTINENT=="GIH" |
                               sample_ids$CONTINENT=="PJL" |
                               sample_ids$CONTINENT=="BEB" |
                               sample_ids$CONTINENT=="STU" |
                               sample_ids$CONTINENT=="ITU", "SAS", sample_ids$CONTINENT)

sample_ids$CONTINENT_numeric <- ifelse(sample_ids$CONTINENT == 'EAS', 1, sample_ids$CONTINENT)
sample_ids$CONTINENT_numeric <- ifelse(sample_ids$CONTINENT == 'EUR', 1,
sample_ids$CONTINENT_numeric)
sample_ids$CONTINENT_numeric <- ifelse(sample_ids$CONTINENT == 'AFR', 1,
sample_ids$CONTINENT_numeric)
sample_ids$CONTINENT_numeric <- ifelse(sample_ids$CONTINENT == 'ADM', 1,
sample_ids$CONTINENT_numeric)
sample_ids$CONTINENT_numeric <- ifelse(sample_ids$CONTINENT == 'SAS', 1,
sample_ids$CONTINENT_numeric)

kg_PC1 <- merge(kg_PC1, sample_ids)


MCTFR_PC1 <- read.table("MCTFR-PC1-score.profile", header=T)
MCTFR_PC2 <- read.table("MCTFR-PC2-score.profile", header=T)
MCTFR_PC3 <- read.table("MCTFR-PC3-score.profile", header=T)
MCTFR_PC4 <- read.table("MCTFR-PC4-score.profile", header=T)

kg <- data.frame(PC1 = -kg_PC1$SCORE,
                 PC2 = -kg_PC2$SCORE,
                 PC3 = -kg_PC3$SCORE,
                 PC4 = -kg_PC4$SCORE,
                 ancestry = kg_PC1$CONTINENT)

all <- rbind(kg, data.frame(PC1=-MCTFR_PC1$SCORE,
                            PC2=-MCTFR_PC2$SCORE,
                            PC3=-MCTFR_PC3$SCORE,
                            PC4=-MCTFR_PC4$SCORE,
                            ancestry=rep("MCTFR", nrow(MCTFR_PC1))))

### Scatterplot matrix
library(car)
scatterplotMatrix(~PC1+PC2+PC3+PC4 | ancestry, data=all, smoother=FALSE, reg.line=FALSE, cex=.7)
```