Day 4 AM: Practical

Gene-based rare variant association analysis using SAIGE

In this exercise you will be using the SAIGE software to perform gene-based association analysis of rare variants with quantitative and binary traits. More information about the SAIGE software can be found at: https://saigegit.github.io/SAIGE-doc/.

You should begin by moving to the DAY4-AM directory. The data for this practical are stored in the input directory. The data provided consist of genome-wide genotyping data for 1,000 individuals across 128,916 single nucleotide variants and corresponding phenotype and covariate data to be used for analysis.

There are several input files required for this exercise:

1. Binary PLINK files that include the genotype data: `egdata.bed`, `egdata.bim`, `egdata.fam`.
2. A sample file that includes phenotype and covariate information for each individual: `egdata.phenocov.txt`.
3. A sample list file that includes the sample IDs of individuals to be included in the analysis: `samplelist.txt`.
4. Two group files (one with and one without variant weights): `group_snpid.txt` and `group_snpid_weights.txt`. The group file without weights has two rows for each gene to be analysed: the first provides the variant ID, and the second provides the annotation (in this case lof, missense, synonymous). The group file with weights adds a third row for each gene that provides the weight.

SAIGE is a flexible software tool, which incorporates single variant tests, as well as gene-based tests (SAIGE-GENE). The software can accommodate binary and quantitative traits, and can allow for covariates in the analysis. The methodology implements mixed models through calculation of a genetic relationship matrix (GRM) to allow for relatedness and/or population structure. For gene-based analyses of rare variants, the default is to perform the SKAT-O test, with results presented for the BURDEN and SKAT tests. Analyses can be performed using variants at different MAF thresholds and for different annotations. Ultra-rare variants, defined as having a minor allele count (MAC) ≤20, are collapsed into a single variable within each gene. This variable takes the value 1 if an individual carries a minor allele at any ultra-rare variant within the gene, and 0 otherwise.

SAIGE analyses are conducted in two steps. In the first step, a null mixed model is fitted using the R script `step1_fitNULLGLMM.R`. In the second step, the association model is fitted, and gene-based tests performed, using the R script `step2_SPAtests.R`. The GRM can be calculated “on the fly” using all variants genome-wide. Alternatively, a “sparse” GRM can be pre-constructed using a random subset of variants from across the genome using the R

script `createSparseGRM.R`. This is particularly important when working with large datasets as computation of the GRM is computationally demanding.

There are some long commands in the practical that occasional span multiple pages, so do make sure that you use the complete command! Please note that whilst SAIGE uses R, all the commands in this practical should be run from a linux terminal window (the `Rscript` commands call R from the terminal).

Gene-based test of association with binary outcome

First, we will conduct a SKAT-O test of association with the binary outcome reported for each individual in the dataset, without making use of a sparse GRM. In the first step, we fit the null model using the following command:

```
Rscript step1_fitNULLGLMM.R \
--useSparseGRMtoFitNULL=FALSE \
--plinkFile=./input/egdata \
--phenoFile=./input/egdata.phenocov.txt \
--isCateVarianceRatio=TRUE \
--phenoCol=y_binary \
--covarColList=x1,x2 \
--qCovarColList=x2 \
--sampleIDColinphenoFile=IID \
--traitType=binary \
--outputPrefix=egdata_binary \
--IsOverwriteVarianceRatioFile=TRUE \
--LOCO=TRUE
```

In this command, the `--useSparseGRMtoFitNULL=FALSE` indicates that we are not using a pre-computed GRM. The `--plinkFile` and `--phenoFile` options point to the binary PLINK files and file containing sample phenotype and covariate information. The `--phenoCol` and `--covarColList` options point to the column headers for the phenotype and any covariates to be analysed. The `--qCovarColList` option points to any covariates that are categorical (as opposed to quantitative), and the `--sampleIDColinphenoFile` option indicates the column header for the sample ID. The `--traitType` option is used to define whether the outcome analysed is binary or quantitative. The `--LOCO` option indicates whether the test chromosome should be excluded from the calculation of the GRM, which is important when NOT using a sparse GRM. The output files generated by this analysis are `egdata_binary.rda` and `egdata_binary.varianceRatio.txt`, which include an R data file and results from fitting the null model that are required in the second step.

In the second step, we fit the association model using the following command:

```
Rscript step2_SPAtests.R \
--bedFile=./input/egdata.bed \
--bimFile=./input/egdata.bim \
--famFile=./input/egdata.fam \
--SAIGEOutputFile=egdata_binary_group.txt \
--chrom=1 \
--AlleleOrder=ref-first \
--minMAF=0 \
--minMAC=0.5 \
```

```
--LOCO=TRUE \
--sampleFile=./input/samplelist.txt \
--GMMATmodelFile=egdata_binary.rda \
--varianceRatioFile=egdata_binary.varianceRatio.txt \
--groupFile=./input/group_snpid.txt \
--annotation_in_groupTest=lof,missense:lof,missense:lof:synonymous \
--maxMAF_in_groupTest=0.001,0.01 \
--is_fastTest=TRUE
```

In this command, `--bedFile`, `--bimFile`, and `--famFile` point to the binary PLINK files containing the genotype data, and `--SAIGEOutputFile` points to the name of the output file for the analysis results. It is also important to specify the chromosome for analysis with the `--chrom` option and the `--LOCO=TRUE` option to ensure that this chromosome is excluded from the GRM construction, which is important when not using a sparse GRM, as for the null model. The `--sampleFile` option points to the file with the IDs of individuals to be included in the analysis, and the `--GMMATmodelFile` and `--varianceRatioFile` options point to the output files generated from fitting the null model. The `--groupFile` option points to the group file for analysis. The `--annotation_in_groupTest` option provides a comma separated list of different annotations (or groups of annotations to be analysed), where the annotation of each variant is reported in the group file. Finally, the `--maxMAF_in_groupTest` option specifies the maximum MAF threshold of variants to be included in the analysis. Note that multiple thresholds can be specified using a comma separated list.

The output file generated by this analysis is `egdata_binary_group.txt`. For each gene specified in the group file, results are presented for each MAF threshold and annotation combination. For each combination, the following results are provided:

Pvalue: the p-value for association from the SKAT-O test.
Pvalue_Burden: the p-value for association from the burden test.
Pvalue_SKAT: the p-value for association from the SKAT test.
BETA_Burden: effect size for the burden test.
SE_Burden: standard error of the effect size for the burden test.
MAC: total MAC across variants (with MAF below threshold) in all individuals.
MAC_case: total MAC across variants (with MAF below threshold) in cases.
MAC_control: total MAC across variants (with MAF below threshold) in controls.
Number_rare: number of variants (with MAF below threshold).
Number_ultra_rare: number of variants with MAC ≤20 (collapsed as a single variable).

The final row of output for each gene is a combined test of association across all annotations and MAF thresholds. This provides a joint test p-value for the SKAT-O, burden, and SKAT tests.

In this analysis, you will notice that for GENE1, for the “lof” group with maximum MAF <0.001, the p-values for the three tests are the same. This is because there are no variants at this MAF threshold with MAC >20, and the three variants with MAC ≤20 have been collapsed into a single variable. The association test is therefore performed with a single variant (the collapsed variable), which will always give the same p-value for association for the burden test and SKAT test (and consequently the SKAT-O test).

Gene-based test of association with quantitative outcome

Next, we will conduct a SKAT-O test of association with the quantitative outcome reported for each individual in the dataset, without making use of a sparse GRM. As before, in the first step, we fit the null model, this time using the following command:

```
Rscript step1_fitNULLGLMM.R \
--useSparseGRMtoFitNULL=FALSE \
--plinkFile=./input/egdata \
--phenoFile=./input/egdata.phenocov.txt \
--isCateVarianceRatio=TRUE \
--phenoCol=y_quantitative \
--covarColList=x1,x2 \
--qCovarColList=x2 \
--sampleIDColinphenoFile=IID \
--traitType=quantitative \
--invNormalize=TRUE \
--outputPrefix=egdata_quantitative \
--IsOverwriteVarianceRatioFile=TRUE \
--LOCO=TRUE
```

The structure of the command is the same as for the binary outcome, but this time we specify `--traitType=quantitative` and `--invNormalize=TRUE` (to enable inverse normalised quantitative trait analysis).

As before, in the second step, we fit the association model, this time using the following command, which has the same structure as for the binary outcome:

```
Rscript step2_SPAtests.R \
--bedFile=./input/egdata.bed \
--bimFile=./input/egdata.bim \
--famFile=./input/egdata.fam \
--SAIGEOutputFile=egdata_quantitative_group.txt \
--chrom=1 \
--AlleleOrder=ref-first \
--minMAF=0 \
--minMAC=0.5 \
--LOCO=TRUE \
--sampleFile=./input/samplelist.txt \
--GMMATmodelFile=egdata_quantitative.rda \
--varianceRatioFile=egdata_quantitative.varianceRatio.txt \
--groupFile=./input/group_snpid.txt \
--annotation_in_groupTest=lof,missense:lof,missense:lof:synonymous \
--maxMAF_in_groupTest=0.001,0.01 \
--is_fastTest=TRUE
```

The output file generated by this analysis is egdata_quantitative_group.txt. The structure of this output file is similar to that for the binary outcome, but the MAC in cases and controls is not reported.

Gene-based test of association with binary outcome and sparse GRM

Next, we will conduct a SKAT-O test of association with the binary outcome reported for each individual in the dataset, but this time making use of a pre-constructed sparse GRM. To construct the sparse GRM, we use the following command:

```
Rscript createSparseGRM.R \
--plinkFile=./input/egdata \
--nThreads=4 \
--outputPrefix=sparseGRM \
--numRandomMarkerforSparseKin=2000 \
--relatednessCutoff=0.125
```

In this command, the `--plinkFile` option points to the binary PLINK files, and the `--outputPrefix` option specifies the prefix for output files. The `--nThreads` option enables parallel computing over multiple CPUs. The `--numRandomMarkerforSparseKin` option specifies the number of randomly selected variants across the genome for constructing the GRM, and the value specified for the `--relatednessCutoff` option defines the kinship value at which pairs of samples are treated as unrelated.

This command will generate two output files, one that includes the matrix (`sparseGRM_relatednessCutoff_0.125_2000_randomMarkersUsed.sparseGRM.mtx`) and one that lists the IDs of individuals used to construct the matrix (`sparseGRM_relatednessCutoff_0.125_2000_randomMarkersUsed.sparseGRM.mtx.sampleIDs.txt`). Note that the sparse GRM needs to be pre-constructed only once, irrespective of the downstream association analysis.

We can then repeat the first step in the analysis using the sparse GRM, where we fit the null model using the following command:

```
Rscript step1_fitNULLGLMM.R \
--
sparseGRMFile=sparseGRM_relatednessCutoff_0.125_2000_randomMarkersUsed.sparseGRM.mtx \
--
sparseGRMSampleIDFile=sparseGRM_relatednessCutoff_0.125_2000_randomMarkersUsed.sparseGRM.mtx.sampleIDs.txt \
--useSparseGRMtoFitNULL=TRUE \
--plinkFile=./input/egdata \
--phenoFile=./input/egdata.phenocov.txt \
--isCateVarianceRatio=TRUE \
--phenoCol=y_binary \
--covarColList=x1,x2 \
--qCovarColList=x2 \
--sampleIDColinphenoFile=IID \
--traitType=binary \
--outputPrefix=egdata_binary_sparseGRM \
--IsOverwriteVarianceRatioFile=TRUE
```

The structure of the command is the same as for the binary outcome without a sparse GRM, except that now we include the `--useSparseGRMtoFitNULL=TRUE` option, and specify the GRM files with the options `--sparseGRMFile` and `--sparseGRMSampleIDFile`. Because we

are using a sparse GRM, we do not need to use the `--LOCO=TRUE` option (we could specify `--LOCO=FALSE`, but this is the default).

As before, in the second step, we fit the association model, this time using the following command, which has the same structure as for the binary outcome without a sparse GRM:

```
Rscript step2_SPAtests.R \
--bedFile=./input/egdata.bed \
--bimFile=./input/egdata.bim \
--famFile=./input/egdata.fam \
--SAIGEOutputFile=egdata_binary_sparseGRM_group.txt \
--chrom=1 \
--AlleleOrder=ref-first \
--minMAF=0 \
--minMAC=0.5 \
--LOCO=FALSE \
--sampleFile=./input/samplelist.txt \
--GMMATmodelFile=egdata_binary_sparseGRM.rda \
--varianceRatioFile=egdata_binary_sparseGRM.varianceRatio.txt \
--
sparseGRMFile=sparseGRM_relatednessCutoff_0.125_2000_randomMarkersUsed.spar
seGRM.mtx \
--
sparseGRMSampleIDFile=sparseGRM_relatednessCutoff_0.125_2000_randomMarkersU
sed.sparseGRM.mtx.sampleIDs.txt \
--groupFile=./input/group_snpid.txt \
--annotation_in_groupTest=lof,missense:lof,missense:lof:synonymous \
--maxMAF_in_groupTest=0.001,0.01 \
--is_fastTest=TRUE
```

The structure of the command is the same as for the binary outcome without a sparse GRM, except that now we specify the GRM files with the options `--sparseGRMFile` and `--sparseGRMSampleIDFile`. Because we are using a sparse GRM, we specify the `--LOCO=FALSE` option.

The structure of this output file is the same that for the binary outcome without a sparse GRM. How do the results compare between analyses with and without a sparse GRM?

Weighted gene-based test of association with binary outcome and sparse GRM

Finally, we will conduct a weighted SKAT-O test of association with the binary outcome reported for each individual in the dataset, making use of a pre-constructed sparse GRM. Note that the null model is the same as for the unweighted analysis, so we need only conduct the second step in the analysis, where we fit the association, this time using the following command, which has the same structure as for the unweighted analysis:

```
Rscript step2_SPAtests.R \
--bedFile=./input/egdata.bed \
--bimFile=./input/egdata.bim \
--famFile=./input/egdata.fam \
--SAIGEOutputFile=egdata_binary_sparseGRM_weights_group.txt \
--chrom=1 \
--AlleleOrder=ref-first \
--minMAF=0 \
```

```
--minMAC=0.5 \
--LOCO=FALSE \
--sampleFile=./input/samplelist.txt \
--GMMATmodelFile=egdata_binary_sparseGRM.rda \
--varianceRatioFile=egdata_binary_sparseGRM.varianceRatio.txt \
--
sparseGRMFile=sparseGRM_relatednessCutoff_0.125_2000_randomMarkersUsed.spar
seGRM.mtx \
--
sparseGRMSampleIDFile=sparseGRM_relatednessCutoff_0.125_2000_randomMarkersU
sed.sparseGRM.mtx.sampleIDs.txt \
--groupFile=./input/group_snpid_weights.txt \
--annotation_in_groupTest=lof,missense:lof,missense:lof:synonymous \
--maxMAF_in_groupTest=0.001,0.01 \
--is_fastTest=TRUE
```

The only difference in this command is the specification of the group file, which now includes weights (in addition to annotation).

The structure of this output file is the same that for the unweighted analysis. How do the results compare between analyses with and without weights?