# FASTGWA-GLMM Workflow

fastGWA-GLMM is a tool designed for ultra-fast generalized linear mixed model-based association analysis, specifically for binary traits. It uses a sparse Genetic Relationship Matrix (GRM) to conduct Genome-Wide Association Studies (GWAS) efficiently, accounting for population stratification. This approach is particularly useful in large-scale biobank studies.

### **Step 1: Obtain PLINK Files**

**You need to generate .bim, .bed, and .fam files from your genotype data.**

**Code:**

**./plink --bfile UGR --keep output\_suicide\_ugrfam --make-bed --out GPC\_result.suicide --noweb4**

* **UGR: The prefix of your original PLINK files.**
* **output\_suicide\_ugrfam: The .fam file for your study.**
* **GPC\_result.suicide: The output prefix for the new PLINK files.**

### **Step 2: Create a GRM**

Next, generate the Genetic Relationship Matrix (GRM) using GCTA.

**Code:**

**./gcta64 --bfile GPC\_result.suicide --make-grm 0.05 --out suicide\_data\_grm**

This creates the GRM using a cutoff of 0.05.

### **Step 3: Generate a Sparse GRM**

Create a sparse GRM, which is optimized for large-scale genetic analyses.

Code:

**bash**

**./gcta64 --grm suicide\_data\_grm --make-bK-sparse 0.05 --out GPC\_sp\_grm**

* **suicide\_data\_grm: The input GRM.**
* **GPC\_sp\_grm: The output prefix for the sparse GRM.**

### **Step 4: Submit the Job Script**

Now, you can submit the job script to your computing cluster to run the fastGWA-GLMM analysis.

Job Script (SLURM) Example:

**bash**

**Copy code**

**#!/bin/bash**

**#SBATCH --job-name=fastGWA\_GLMM # Job name**

**#SBATCH --nodes=1 # Number of nodes (1 node)**

**#SBATCH --ntasks=1 # Number of tasks (usually 1 for a single job)**

**#SBATCH --cpus-per-task=32 # Number of CPUs/cores per task (up to 32)**

**#SBATCH --mem=128G # Memory per node (reduce from 256GB to 128GB)**

**#SBATCH --time=30:00:00 # Time limit (30 hours)**

**#SBATCH --output=fastGWA\_GLMM\_%j.out # Standard output and error log**

**# Load necessary modules if applicable**

**# module load gcta**

**# Define variables**

**gcta="/mnt/users/s/sh/shakirah/FAST/gcta-1.94.1-linux-kernel-3-x86\_64/gcta64" # Path to GCTA executable**

**geno\_path="/mnt/users/s/sh/shakirah/FAST/GPC\_result.suicide" # Prefix of the PLINK files (.bed, .bim, .fam)**

**path\_to\_sparse\_GRM="/mnt/users/s/sh/shakirah/FAST/GPC\_sp\_grm" # Prefix of the sparse GRM files**

**covar\_file="/mnt/users/s/sh/shakirah/FAST/FAM/ordered.covs" # Path to the quantitative covariate file**

**pheno\_file="/mnt/users/s/sh/shakirah/FAST/FAM/ordered.pheno.txt" # Path to the phenotype file**

**output\_path="/mnt/users/s/sh/shakirah/FAST/RESULTS" # Path to the output directory**

**# Run GCTA fastGWA-GLMM**

**${gcta} --bfile ${geno\_path} \**

**--grm-sparse ${path\_to\_sparse\_GRM} \**

**--fastGWA-mlm-binary \**

**--qcovar ${covar\_file} \**

**--pheno ${pheno\_file} \**

**--threads 32 \**

**--out ${output\_path}/fastGWA\_GLMM\_final**

* **gcta: Path to the GCTA executable.**
* **geno\_path: Path to the PLINK files.**
* **path\_to\_sparse\_GRM: Path to the sparse GRM.**
* **covar\_file: Path to the covariate file.**
* **pheno\_file: Path to the phenotype file.**
* **output\_path: Directory for the final output.**

### **Final Notes:**

* Ensure that all paths provided in the script are correct and accessible.
* Check the cluster’s specific SLURM settings and adjust the script accordingly.

This process will perform a fastGWA-GLLM analysis, producing outputs in the specified directory.