Burden-Testing Pipeline

Overview

This repository provides a **comprehensive, modular R-based pipeline** (`burden_testing_pipeline.R`) for **gene- and region-level burden testing** in case–control cohorts. Designed with flexibility in mind, the pipeline supports multiple statistical methods (SKAT, SKAT-O, CAST, CMC, ACAT) while integrating **frequency filtering, functional annotations, and population-structure correction**. Outputs include detailed TSV/Excel tables, QC visualizations, and execution logs, enabling transparent and reproducible analyses.

> **Why modularity matters:** Each major component—frequency filtering, variant annotation, burden test execution, and result summarization—is implemented as an independent function or script section. This approach allows users to **swap in new tests or filtering thresholds** without modifying core logic, and to **reuse modules** in other projects.

Pipeline Modularity & Key Steps

The pipeline is organized into clearly defined modules, each responsible for a distinct task. The main script (`burden_testing_pipeline.R`) orchestrates these modules in the following order:

1. **Input Parsing & Setup**
   - Load command-line arguments (VCF path, phenotype file, gene annotation, MAF threshold, selected methods, output prefix).
   - Validate that required files exist and the output directory is writable.
   - Initialize logging (timestamped logs capture runtime, warnings, and errors).
2. **Variant Filtering**
   a. **Internal MAF Filtering** (using `data.table`):
   - Read the cohort VCF (cases + controls) and extract allele counts.
   - Exclude any variant with MAF > specified threshold (e.g., 0.01).
     b. **External Frequency (gnomAD) Filtering (Optional):**
   - If `--gnomad-filter TRUE`, load `data/gnomad.vcf.gz`.
   - Use `bcftools` (if installed) or `SeqVarTools` to annotate cohort VCF with external frequencies.
   - Remove variants exceeding external MAF threshold.
3. **Functional Annotation**
   - **Gene Overlap Annotation:** Use `GenomicRanges` to map each variant to gene coordinates (provided in `data/genes.gtf`). Annotate gene symbols, gene biotypes, and transcript IDs.
   - **Predicted Deleteriousness:** Integrate variant-level annotations (e.g., LOF, missense, splice-site) from VEP or precomputed tables. Assign a binary or weighted “deleterious” flag.
   - **Clinical Database Tags:** (Optional) Tag variants present in ClinVar or other local annotation files—adds context for downstream interpretation.
4. **Phenotype & Covariate Processing**
   - Read `data/phenotypes.tsv`, which must include columns: `SampleID`, `Status` (1=case, 0=control), `PC1`, `PC2`, ..., and any additional covariates (e.g., age, sex).
   - Verify matching sample IDs between genotype and phenotype files.

- Compute or load a Genetic Relationship Matrix (GRM) if `--lmm TRUE` to account for relatedness.

5. **Burden Test Execution**
   For each gene or region (defined in `data/genes.gtf`):
   a. **Generate Genotype Matrix** (using `SeqVarTools`): Extract genotypes for all filtered variants within the gene/region.
   b. **Construct Covariate Matrix:** Include intercept, `PC1`, `PC2`, ..., and any user-specified covariates.
   c. **Run Selected Tests:** Depending on `--method`, invoke:

   - **SKAT / SKAT-O**: Kernel-based test suited for rare variant aggregation.
   - **CMC**: Collapsing approach that groups rare variants within a region and tests the burden.
   - **CAST**: Counts aggregated variant alleles per individual and tests via logistic regression.
   - **ACAT**: Combines p-values across multiple tests or regions using the Cauchy method.
   - **Mixed-Model (LMM)**: When `--lmm TRUE`, use `GENESIS` or `lme4` to adjust for relatedness via a GRM-based random effect.
     d. **Store Results:** For each test, record gene/region name, test statistic, p-value, effect size (if applicable), and number of variants tested.

6. **Multiple Testing & Significance**
   - Combine p-values across all genes/regions.
   - Apply false discovery rate (FDR) correction (Benjamini–Hochberg) and/or Bonferroni correction.
   - Classify genes/regions as **Significant** (e.g., FDR < 0.05) or **Suggestive** (nominal p < 0.01).

7. **Result Summarization & QC Plots**
   - **TSV/Excel Report:** Use `openxlsx::write.xlsx()` to generate a workbook containing:
     a. **`Burden_Results`** sheet: All genes/regions with columns: `Gene`, `Test`, `PValue`, `EffectSize`, `NumVariants`, `FDR`, `SignificanceStatus`.
     b. **`Summary_Stats`** sheet: Counts of significant genes per test, distribution of p-values.
   - **QC Plots (via `ggplot2`)**:
     - **PCA Plot:** Scatterplot of `PC1` vs. `PC2`, colored by `Status` (case/control). Ensures no major stratification.
     - **QQ Plot:** Observed vs. Expected –log10(p-values) to assess inflation.
     - **Histogram of Variant Counts:** Distribution of number of variants per gene tested.
     - **Volcano Plot (optional):** –log10(p-value) vs. effect size for top hits.

8. **Logging & Cleanup**
   - Write a comprehensive log file (`results/<prefix>_log.txt`) capturing:
     - Start/end times, runtime per module, memory usage.
     - Number of variants filtered at each step (internal vs. external).
     - Any warnings or errors encountered.
   - Save session info (R version, package versions) at the end for reproducibility.

Prerequisites

- **R ≥ 4.0** (tested on R 4.1+)

- **Required R packages:**

```
install.packages(c("data.table", "dplyr", "optparse", "openxlsx", "ggplot2", "lme4"))
if (!requireNamespace("BiocManager", quietly = TRUE)) install.packages("BiocManager")
BiocManager::install(c("SKAT", "SeqVarTools", "GenomicRanges", "GENESIS", "gaston"))
```

- **Optional**:
  - **PLINK/PLINK2** (for genotype conversions)
  - **bcftools** (for variant filtering/export)
  - **SLURM / HPC environment** (for large-scale runs)

Input Files & Structure

1. **`data/cohort.vcf.gz`**
   - Combined VCF of case and control samples (indexed with `.tbi`).
   - Ensure `FORMAT` fields: `GT` (genotype), `AD` (allele depth) if performing allele count filtering.
2. **`data/phenotypes.tsv`**
   - Tab-delimited table with header; columns must include:
     - `SampleID`: Unique sample identifiers matching VCF sample IDs.
     - `Status`: 1 = Case, 0 = Control.
     - `PC1`, `PC2`, ..., `PCn`: Principal components for population structure (optional).
     - Additional covariates (e.g., `Sex`, `Age`).
3. **`data/genes.gtf`**
   - Gene annotation file (GTF format) containing gene coordinates.
   - The pipeline extracts gene boundaries to define collapsing regions.
4. **`data/gnomad.vcf.gz`** (Optional)
   - Reference VCF with allele frequencies from gnomAD or genoMAD.
   - Use for external MAF filtering if `--gnomad-filter TRUE`.
5. **Directory Expectations**

```
Burden-Testing-Pipeline/
├── docs/
│   └── README.pdf             # Full documentation (this file)
├── data/
│   ├── cohort.vcf.gz
│   ├── cohort.vcf.gz.tbi      # index
│   ├── phenotypes.tsv
│   ├── genes.gtf
│   └── gnomad.vcf.gz          # optional external frequency reference
├── scripts/
│   └── burden_testing_pipeline.R
├── results/                   # auto-created after running (> 1 per run)
└── README.md                  # higher-level quick-start guide
```

Quick Start

1. **Clone & navigate**

   ```
   git clone https://github.com/<YOUR-USERNAME>/Burden-Testing-Pipeline.git
   cd Burden-Testing-Pipeline
   ```

2. **Install R dependencies**

   ```
   install.packages(c("data.table","dplyr","optparse","openxlsx","ggplot2","lme4"))
   if (!requireNamespace("BiocManager", quietly = TRUE)) install.packages("BiocManager")
   BiocManager::install(c("SKAT","SeqVarTools","GenomicRanges","GENESIS","gaston"))
   ```

3. **Prepare inputs**

   - Place `cohort.vcf.gz` (and its index `cohort.vcf.gz.tbi`), `phenotypes.tsv`, `genes.gtf` under `data/`.
   - If using external frequency filtering, place `gnomad.vcf.gz` (and its index `gnomad.vcf.gz.tbi`) in `data/`.

4. **Run the pipeline**

   ```
   Rscript scripts/burden_testing_pipeline.R \
       --vcf data/cohort.vcf.gz \
       --pheno data/phenotypes.tsv \
       --genes data/genes.gtf \
       --maf 0.01 \
       --method SKAT,SKATO,CAST,CMC,ACAT \
       --gnomad-filter TRUE \
       --covars PC1,PC2,Sex \
       --lmm TRUE \
       --grm data/grm.Rds \
       --out results/analysis1
   ```

   - This will generate `results/analysis1_burden_results.tsv`, `results/analysis1_QC_plots.pdf`, and `results/analysis1_log.txt`.

5. **Inspect results**

   - Open `results/analysis1_burden_results.tsv` in a spreadsheet or R to review p-values, effect sizes, and significance.
   - View `results/analysis1_QC_plots.pdf` for PCA and QQ plots to assess quality.
   - Check `results/analysis1_log.txt` for runtime metrics and warnings.

Usage Notes & Customization

- **Adding New Tests:**
  1. Open `scripts/burden_testing_pipeline.R`.
  2. Under the **Run Tests** section, add a new function to implement your desired test (e.g., Madsen-Browning).
  3. Append the test name to the `--method` argument when running.
- **Adjusting Frequency Filters:**
  - **Internal MAF:** Change `--maf <value>` (e.g., `--maf 0.005`).
  - **External gnomAD:** Use `--gnomad-filter TRUE`; ensure `data/gnomad.vcf.gz` is provided and indexed.
- **Population-Structure Correction:**
  - By default, the pipeline uses `PC1` and `PC2` from `phenotypes.tsv`.
  - To include more or fewer PCs, set `--covars` accordingly (e.g., `--covars PC1,PC2,PC3`).
- **Mixed-Model Extension (LMM):**

- Use `--lmm TRUE` to enable relatedness correction via a Genetic Relationship Matrix (GRM).
- Provide `--grm <path to GRM.rds>` (constructed via external tools or within R).

- **Output Directory Customization:**
  - By default, outputs go to `results/<prefix>_*`.
  - To override, set `OUTDIR` prior to running:

    ```
    export OUTDIR="/my/custom/path/analysis1"
    Rscript scripts/burden_testing_pipeline.R --vcf ... --out $OUTDIR
    ```

- **SLURM/HPC Integration:**
  - Create a Slurm job script (e.g., `burden_job.sbatch`) containing:

    ```
    #!/bin/bash
    #SBATCH --job-name=BurdenTest
    #SBATCH --cpus-per-task=8
    #SBATCH --mem=32G
    #SBATCH --time=04:00:00

    module load R/4.0
    Rscript /path/to/Burden-Testing-Pipeline/scripts/burden_testing_pipeline.R \
        --vcf /path/to/data/cohort.vcf.gz \
        --pheno /path/to/data/phenotypes.tsv \
        --genes /path/to/data/genes.gtf \
        --maf 0.01 \
        --method SKAT,SKATO,CAST,CMC,ACAT \
        --lmm TRUE \
        --grm /path/to/data/grm.Rds \
        --out /path/to/results/analysis1
    ```

  - Submit with:

    ```
    sbatch burden_job.sbatch
    ```

Full Methods & Scoring Criteria

Germline Filtering (Case–Control Comparison)

The first stage ensures that only **rare, potentially pathogenic variants** proceed to burden testing:

1. **Load VCF & Compute Allele Counts:**
   - Use `SeqVarTools` or `data.table` to extract allele counts per variant across all samples.
   - Calculate Minor Allele Frequency (MAF) = `min(AC/AN, 1-AC/AN)`.
2. **Internal MAF Filtering:**
   - Exclude variants with MAF > threshold (e.g., 0.01).
3. **External gnomAD Filtering (Optional):**
   - Annotate cohort VCF with allele frequencies from `data/gnomad.vcf.gz`.
   - Remove any variant with gnomAD MAF > threshold.

Functional Annotation

Adds biological context to each variant:

1. **Gene Overlap Annotation:**

- Import `data/genes.gtf` into `GenomicRanges`.
- For each variant, find overlapping gene(s) and assign gene symbols.

2. **Predicted Deleteriousness:**
   - Integrate variant consequence annotations (e.g., via VEP).
   - Classify variants as **LOF**, **missense**, or **other**.
3. **Clinical Database Tagging (Optional):**
   - If local ClinVar or other annotation files are present, tag variants with known clinical significance.

Phenotype & Covariate Processing

Ensures robust population correction:

- **Load `phenotypes.tsv`:** Must contain `SampleID`, `Status`, `PC1`, `PC2`, etc.
- **Verify Matches:** Check that sample IDs in VCF match those in phenotype file.
- **Compute Additional PCs (Optional):** If fewer/more PCs needed, run PCA on genotype data (not implemented by default).
- **GRM Construction (Optional):** If `--lmm TRUE`, build or load a GRM (`.rds`) using `gaston` or other tools.

Statistical Tests

For each gene/region, a suite of tests is performed on the filtered, annotated variant set:

- **SKAT / SKAT-O:** Optimal sequence kernel association test using `SKAT` package.
- **CMC:** Collapsing test grouping rare variants; results from a logistic regression on burden counts.
- **CAST:** Binary burden test counting alleles per individual.
- **ACAT:** Combines p-values across tests or regions through a Cauchy combination.
- **Mixed-Model (LMM):** Adjust for relatedness using `GENESIS` or `lme4`, requiring GRM.

Multiple Testing Correction & Classification

1. **Combine p-values** across all genes for each test.
2. **Adjust for multiple comparisons:**
   - **FDR (Benjamini–Hochberg)** for exploratory discovery.
   - **Bonferroni correction** for stringent genome-wide significance.
3. **Classify Significance:** Label genes as **Significant** (e.g., FDR < 0.05) or **Suggestive** (nominal p < 0.01).

Result Summarization & QC Plots

After test execution, the pipeline produces:

- **Excel Workbook** (`.xlsx`) with two sheets:
  1. **`Burden_Results`**: Detailed results table with `Gene`, `Test`, `NumVariants`, `PValue`, `EffectSize`, `FDR`, `SignificanceStatus`.
  2. **`Summary_Stats`**: Aggregated counts of significant hits per test and distribution summaries.
- **QC Plots**:

- **PCA Plot (`ggplot2`)**: Displaying `PC1` vs. `PC2` colored by case/control status.
- **QQ Plot**: Observed vs. expected –log10(p-values).
- **Variant Count Histogram**: Distribution of variant counts per gene.
- **Volcano Plot (Optional)**: Highlighting top significant genes by effect size and p-value.

Logging & Reproducibility

- **Log File** (`results/<prefix>_log.txt`): Tracks timeline of each module, memory usage, number of variants filtered, and any errors.
- **Session Info**: Captured at script end, including R version and package versions for reproducibility.

Example Outputs

`results/analysis1_burden_results.tsv` (TSV format):

Gene	Test	NumVariants	PValue	EffectSize	FDR	SignificanceStatus
BRCA1	SKAT	12	2.0e-06	1.52	5.3e-04	Significant
TP53	SKAT-O	9	1.0e-04	1.32	1.2e-02	Suggestive
...	...	...	...	...	...	...

`results/analysis1_QC_plots.pdf`: Contains:

- **PCA Plot:** Samples plotted by `PC1` and `PC2`, colored by case vs. control.
- **QQ Plot:** Observed vs. Expected –log10(p-values).
- **Histogram:** Distribution of number of variants tested per gene.

`results/analysis1_log.txt`:

```
[2025-06-05 14:22:10] Starting burden_testing_pipeline.R
[2025-06-05 14:22:12] Input files verified, cohort.vcf.gz and phenotypes.tsv loaded.
[2025-06-05 14:22:15] Internal MAF filtering: 250,000 variants -> 10,000 rare variants.
[2025-06-05 14:22:25] External gnomAD filtering (MAF < 0.01): 10,000 -> 8,500 variants.
[2025-06-05 14:22:30] Functional annotation completed; 8,500 variants annotated.
[2025-06-05 14:22:35] Loaded phenotype file; 500 cases, 500 controls.
[2025-06-05 14:22:40] Running SKAT: completed 20,000 genes in 120 seconds.
[2025-06-05 14:24:40] Running SKAT-O: completed 20,000 genes in 115 seconds.
...
[2025-06-05 14:32:10] Writing Excel report: results/analysis1_burden_results.xlsx.
[2025-06-05 14:32:15] Generating QC plots.
[2025-06-05 14:32:30] Completed. Total runtime: 600 seconds.
```

Credits & Contact

- **Author**: Sally L. Yepes Torres
- **Email**: sallyepes233@gmail.com

- **Last Updated**: June 2025
- **License**: MIT
