Package 'smer'

July 23, 2025

Title Sparse Marginal Epistasis Test

Version 0.0.1

URL https://github.com/lcrawlab/sme, https://lcrawlab.github.io/sme/

BugReports https://github.com/lcrawlab/sme/issues

Description The Sparse Marginal Epistasis Test is a computationally efficient genetics method which detects statistical epistasis in complex traits; see Stamp et al. (2025, <doi:10.1101/2025.01.11.632557>) for details.

License MIT + file LICENSE

Encoding UTF-8

RoxygenNote 7.3.2

LinkingTo BH, Rcpp, RcppEigen, Rhdf5lib, testthat

Imports dplyr, genio, logging, mvMAPIT, Rcpp, tidyr

Suggests GenomicRanges, ggplot2, knitr, rmarkdown, testthat (>= 3.0.0), xml2

Config/testthat/edition 3

SystemRequirements GNU make

VignetteBuilder knitr

Depends R (>= 4.4.0)

LazyData true

biocViews GenomeWideAssociation, Epistasis, Genetics, SNP, LinearMixedModel

NeedsCompilation yes

Author Julian Stamp [cre, aut] (ORCID: <https://orcid.org/0000-0003-3014-6249>), Lorin Crawford [aut] (ORCID: <https://orcid.org/0000-0003-0178-8242>), sriramlab [cph] (Author of included mailman algorithm), Blue Brain Project/EPFL [cph] (Author of included HighFive library)

Maintainer Julian Stamp <julian.d.stamp@gmail.com>

Repository CRAN

Date/Publication 2025-01-16 15:50:01 UTC

Contents

`approximate_memory_requirements` *Estimate Memory Requirements for SME Routine*

Description

This function provides an approximate estimate of the memory requirements (in gigabytes) for running the Sparse Marginal Epistasis (SME) routine based on input parameters such as the number of samples, SNPs, and other configurations.

Usage

```
approximate_memory_requirements(
  n_samples,
  n_snps,
  n_blocks,
  n_randvecs,
  chunksize
)
```

Arguments

`n_samples`	Integer. The number of samples in the dataset.
`n_snps`	Integer. The total number of SNPs in the dataset.
`n_blocks`	Integer. The number of genotype blocks used to partition SNPs. Affects the size of encoded genotype segments.
`n_randvecs`	Integer. The number of random vectors used for stochastic trace estimation. Affects memory for operations involving random vectors.
`chunksize`	Integer. The number of focal SNPs processed per chunk.

Details

The function calculates memory usage by summing the contributions from various components used in the SME routine, including:

- Variance component estimates (`vc_estimates`)
- Phenotype-related matrices
- Random vector-based computations
- Genotype objects and block statistics
- Gene-by-gene interaction masks

The estimated memory requirement is derived from the data dimensions and operational needs, and it provides a guideline for configuring resources for the analysis.

Value

Numeric. The approximate memory requirement (in gigabytes) for the SME routine.

Examples

```
n_samples <- 1e5
n_snps <- 1e6
n_blocks <- 100
n_randvecs <- 100
chunksize <- 10
approximate_memory_requirements(n_samples,
                                n_snps,
                                n_blocks,
                                n_randvecs,
                                chunksize)
```

`create_hdf5_file` *Create an HDF5 File*

Description

This function creates a new, empty HDF5 file at the specified location.

Usage

```
create_hdf5_file(hdf5_file)
```

Arguments

`hdf5_file`	A character string specifying the path and name of the HDF5 file to be created.

Value

No return value; the function creates the HDF5 file at the specified location.

Examples

```
# Create an empty HDF5 file
hdf5_file <- tempfile()
create_hdf5_file(hdf5_file)
```

`getting_started` *Simulated Dataset for Genome-Wide Interaction Analysis*

Description

`getting_started` is a simulated dataset created to demonstrate the use of the `sme()` function for genome-wide interaction analyses. It contains results from a simulated analysis involving additive genetic effects and gene-by-gene (GxG) interactions.

Usage

```
data("getting_started")
```

Format

A list with results from `sme()`, including the following components:

`summary` A data frame summarizing the analysis results, including p-values for SNP associations (`p`).

`pve` A data frame containing the per SNP variance component estimates normalized to phenotypic variance explained (PVE).

`vc` A data frame containing the per SNP variance component estimates.

`gxg_snps` A vector containing the indices of the SNPs assigned to have epistatic interactions in the trait simulations.

Details

The dataset was generated as follows:

- **Genotype Simulation**: Genotype data for 5000 individuals and 6,000 SNPs was simulated with synthetic allele counts.
- **Phenotype Simulation**: Phenotypic values were simulated with an additive heritability of 0.3 and a GxG interaction heritability of 0.25. A set of 100 SNPs were selected for additive effects, and two groups of 5 SNPs each were used for GxG interactions.
- **PLINK-Compatible Files**: The simulated data was saved in PLINK-compatible `.bed`, `.fam`, and `.bim` files.

- **Interaction Analysis**: The `sme()` function was used to perform genome-wide interaction analyses on a subset of SNP indices, including the GxG SNP groups and 100 additional additive SNPs. Memory-efficient computation parameters (e.g., `chun_ksize`, `n_randvecs`, and `n_blocks`) were applied.

Key Parameters

- **Additive Heritability**: 0.3
- **GxG Heritability**: 0.25
- **Number of Samples**: 5000
- **Number of SNPs**: 6,000
- **Selected Additive SNPs**: 100
- **Selected GxG SNP Groups**:
  - Group 1: 5 SNPs
  - Group 2: 5 SNPs

Source

data-raw/getting_started.R

See Also

sme

Examples

```
data("getting_started")
head(getting_started$summary)
```

`read_hdf5_dataset` *Read Dataset from an HDF5 File*

Description

This function reads a dataset from an existing HDF5 file.

Usage

```
read_hdf5_dataset(file_name, dataset_name)
```

Arguments

`file_name`	A character string specifying the path to the HDF5 file.
`dataset_name`	A character string specifying the name of the dataset within the HDF5 file to read.

Value

The content of the dataset from the HDF5 file, typically in the form of an R object.

Examples

```
data_to_write <- 1:10

# Create an empty HDF5 file
hdf5_file <- tempfile()
create_hdf5_file(hdf5_file)

# Write new data to a dataset in the HDF5 file
write_hdf5_dataset(hdf5_file, "group/dataset", data_to_write)

# Read a dataset from an HDF5 file
hdf5_data <- read_hdf5_dataset(hdf5_file, "group/dataset")
print(hdf5_data)
```

---

simulate_traits *Simulate Quantitative Traits from PLINK Genotypes*

Description

This function simulates a quantitative trait based on additive and epistatic genetic effects using genotype data from a PLINK dataset. The simulated trait is saved to a specified output file in a phenotype format compatible with PLINK.

Usage

```
simulate_traits(
  plink_file,
  output_file,
  additive_heritability,
  gxg_heritability,
  additive_indices,
  gxg_indices_1,
  gxg_indices_2,
  log_level = "WARNING"
)
```

Arguments

plink_file	Character. Path to the PLINK dataset (without file extension). The function will append .bed, .bim, and .fam extensions as needed.
output_file	Character. Path to the output file where the simulated trait will be saved.

`additive_heritability`
: Numeric. A value between 0 and 1 specifying the proportion of trait variance due to additive genetic effects.

`gxg_heritability`
: Numeric. A value between 0 and 1 specifying the proportion of trait variance due to gene-by-gene (epistatic) interactions. The sum of `additive_heritability` and `gxg_heritability` must not exceed 1.

`additive_indices`
: Integer vector. Indices of SNPs contributing to additive genetic effects.

`gxg_indices_1`
: Integer vector. Indices of SNPs in the first group for epistatic interactions.

`gxg_indices_2`
: Integer vector. Indices of SNPs in the second group for epistatic interactions.

`log_level`
: Character. Logging level for messages (e.g., "DEBUG", "INFO", "WARNING"). Default is "WARNING".

Details

The function uses the following components to simulate the trait:

- Additive genetic effects: Determined by `additive_indices` and the specified `additive_heritability`.
- Epistatic interactions: Simulated using pairs of SNPs from `gxg_indices_1` and `gxg_indices_2`, contributing to the `gxg_heritability`.
- Environmental effects: Any remaining variance not explained by genetic effects is assigned to random environmental noise.

The output file is in PLINK-compatible phenotype format with three columns: Family ID (`FID`), Individual ID (`IID`), and the simulated trait (`TRAIT`).

Value

None. The simulated trait is written to the specified `output_file`.

Examples

```
plink_file <- gsub("\\.bed", "", system.file("testdata", "test.bed", package = "smer"))
out_file <- tempfile()
additive_heritability <- 0.3
gxg_heritability <- 0.1
additive_snps <- sort(sample(1:100, 50, replace = FALSE))
gxg_group_1 <- sort(sample(additive_snps, 10, replace = FALSE))
gxg_group_2 <- sort(sample(setdiff(additive_snps, gxg_group_1), 10, replace = FALSE))
n_samples <- 200
simulate_traits(
  plink_file,
  out_file,
  additive_heritability,
  gxg_heritability,
  additive_snps,
  gxg_group_1,
  gxg_group_2
```

```
)
from_file <- read.table(out_file, header = TRUE)
head(from_file)
```

---

sme *Sparse Marginal Epistasis Test (SME)*

Description

SME fits a linear mixed model in order to test for marginal epistasis. It concentrates the scans for epistasis to regions of the genome that have known functional enrichment for a trait of interest.

Usage

```
sme(
  plink_file,
  pheno_file,
  mask_file = NULL,
  gxg_indices = NULL,
  chunk_size = NULL,
  n_randvecs = 10,
  n_blocks = 100,
  n_threads = 1,
  gxg_h5_group = "gxg",
  ld_h5_group = "ld",
  rand_seed = -1,
  log_level = "WARNING"
)
```

Arguments

plink_file	Character. File path to the PLINK dataset (without *.bed extension). The function will append .bim, .bed, and .fam extensions automatically. The genotype data must not have any missing genotypes. Use PLINK to remove variants with missing genotypes or impute them.
pheno_file	Character. File path to a phenotype file in PLINK format. The file should contain exactly one phenotype column.
mask_file	Character or NULL. File path to an HDF5 file specifying per-SNP masks for gene-by-gene interaction tests. This file informs which SNPs are tested for marginal epistasis. Defaults to NULL, indicating no masking. Masking impacts the scaling of memory and time.
gxg_indices	Integer vector or NULL. List of indices corresponding to SNPs to test for marginal epistasis. If NULL, all SNPs in the dataset will be tested. These indices are **1-based**.

chunk_size	Integer or NULL. Number of SNPs processed per chunk. This influences memory usage and can be left NULL to automatically determine the chunk size based on gxg_indices and number of threads.
n_randvecs	Integer. Number of random vectors used for stochastic trace estimation. Higher values yield more accurate estimates but increase computational cost. Default is 10.
n_blocks	Integer. Number of blocks into which SNPs are divided for processing. This parameter affects memory requirements. Default is 100.
n_threads	Integer. Number of threads for OpenMP parallel processing. Default is 1.
gxg_h5_group	Character. Name of the HDF5 group within the mask file containing gene-by-gene interaction masks. SNPs in this group will be included in the gene-by-gene interactions. Defaults to "gxg".
ld_h5_group	Character. Name of the HDF5 group within the mask file containing linkage disequilibrium masks. SNPs in this group are excluded from analysis. Defaults to "ld".
rand_seed	Integer. Seed for random vector generation. If -1, no seed is set. Default is -1.
log_level	Character. Logging level for messages. Must be in uppercase (e.g., "DEBUG", "INFO", "WARNING", "ERROR"). Default is "WARNING".

Details

This function integrates PLINK-formatted genotype and phenotype data to perform marginal epistasis tests on a set of SNPs. Using stochastic trace estimation, the method computes variance components for gene-by-gene interaction and genetic relatedness using the MQS estimator. The process is parallelized using OpenMP when n_threads > 1.

The memory requirements and computation time scaling can be optimized through the parameters chunk_size, n_randvecs, and n_blocks.

Mask Format Requirements

The mask file format is an HDF5 file used for storing index data for the masking process. This format supports data retrieval by index. Below are the required groups and datasets within the HDF5 file:

The required group names can be configured as input parameters. The defaults are described below.

- **Groups**:
  - ld: Stores SNPs in LD with the focal SNP. These SNPs will be **excluded**.
  - gxg: Stores indices of SNPs that the marginal epistasis test is conditioned on. These SNPs will be **included**.
- **Datasets**:
  - ld/<j>: For each focal SNP <j>, this dataset contains indices of SNPs in the same LD block as that SNP. These SNPs will be **excluded** from the gene-by-gene interaction covariance matrix.
  - gxg/<j>: For each focal SNP <j>, this dataset contains indices of SNPs to **include** in the the gene-by-gene interaction covariance matrix for focal SNP <j>.

Important: All indices in the mask file data are **zero-based**, matching the zero-based indices of the PLINK .bim file.

Value

A list containing:

- summary: A tibble summarizing results for each tested SNP, including:
  - id: Variant ID.
  - index: Index of the SNP in the dataset.
  - chromosome: Chromosome number.
  - position: Genomic position of the SNP.
  - p: P value for the gene-by-gene interaction test.
  - pve: Proportion of variance explained (PVE) by gene-by-gene interactions.
  - vc: Variance component estimate.
  - se: Standard error of the variance component.
- pve: A long-format tibble of PVE for all variance components.
- vc_estimate: A long-format tibble of variance component estimates.
- vc_se: A long-format tibble of standard errors for variance components.
- average_duration: Average computation time per SNP.

References

Stamp, J., Pattillo Smith, S., Weinreich, D., & Crawford, L. (2025). Sparse modeling of interactions enables fast detection of genome-wide epistasis in biobank-scale studies. bioRxiv, 2025.01.11.632557.

Stamp, J., DenAdel, A., Weinreich, D., & Crawford, L. (2023). Leveraging the genetic correlation between traits improves the detection of epistasis in genome-wide association studies. G3: Genes, Genomes, Genetics, 13(8), jkad118.

Crawford, L., Zeng, P., Mukherjee, S., & Zhou, X. (2017). Detecting epistasis with the marginal epistasis test in genetic mapping studies of quantitative traits. PLoS genetics, 13(7), e1006869.

Examples

```
plink_file <- gsub("\\.bed", "", system.file("testdata", "test.bed", package="smer"))
pheno_file <- system.file("testdata", "test_h2_0.5.pheno", package="smer")
mask_file <- ""

# Parameter inputs
chunk_size <- 10
n_randvecs <- 10
n_blocks <- 10
n_threads <- 1

# 1-based Indices of SNPs to be analyzed
n_snps <- 100
snp_indices <- 1:n_snps

sme_result <- sme(
  plink_file,
  pheno_file,
  mask_file,
```

```
  snp_indices,
  chunk_size,
  n_randvecs,
  n_blocks,
  n_threads
)
head(sme_result$summary)
```

`write_hdf5_dataset` *Write Data to an HDF5 Dataset*

Description

This function writes new data to an existing HDF5 file. If the dataset already exists, it will be replaced with the new data.

Usage

```
write_hdf5_dataset(file_name, dataset_name, new_data)
```

Arguments

`file_name`	A character string specifying the path to the HDF5 file.
`dataset_name`	A character string specifying the name of the dataset to be written in the HDF5 file.
`new_data`	The new data to write to the dataset. The data must be compatible with the dataset's structure.

Value

No return value; the function modifies the specified dataset in the HDF5 file.

Examples

```
data_to_write <- 1:10

# Create an empty HDF5 file
hdf5_file <- tempfile()
create_hdf5_file(hdf5_file)

# Write new data to a dataset in the HDF5 file
write_hdf5_dataset(hdf5_file, "group/dataset", data_to_write)
```

Index