# Package 'GEM'

November 12, 2026

**Type** Package

**Title** GEM: fast association study for the interplay of Gene, Environment and Methylation

**Version** 1.38.0

**Date** 2015-12-05

**Author** Hong Pan, Joanna D Holbrook, Neerja Karnani, Chee-Keong Kwoh

**Maintainer** Hong Pan <pan_hong@sics.a-star.edu.sg>

**Description** Tools for analyzing EWAS, methQTL and GxE genome widely.

**License** Artistic-2.0

**biocViews** MethylSeq, MethylationArray, GenomeWideAssociation, Regression, DNAMethylation, SNP, GeneExpression, GUI

**Depends** R (>= 3.3)

**VignetteBuilder** knitr

**Suggests** knitr, RUnit, testthat, BiocGenerics, rmarkdown, markdown

**Imports** tcltk, ggplot2, methods, stats, grDevices, graphics, utils

**LazyData** TRUE

**RoxygenNote** 5.0.1

**NeedsCompilation** no

**git_url** https://git.bioconductor.org/packages/GEM

**git_branch** RELEASE_3_23

**git_last_commit** 04c7ece

**git_last_commit_date** 2026-04-28

**Repository** Bioconductor 3.23

**Date/Publication** 2026-11-11

# Contents

GEM-package . . . . . 2
GEM_Emodel . . . . . 3
GEM_Gmodel . . . . . 4
GEM_GUI . . . . . 5
GEM_GWASmodel . . . . . 6
GEM_GxEmodel . . . . . 7
SlicedData-class . . . . . 9

**Index** **12**

---

`GEM-package` *GEM: Fast association study for the interplay of Gene, Environment and Methylation*

---

## Description

The GEM package provides a highly efficient R tool suite for performing epigenome wide association studies (EWAS). GEM provides three major functions named `GEM_Emodel`, `GEM_Gmodel` and `GEM_GxEmodel` to study the interplay of Gene, Environment and Methylation (GEM). Within GEM, the existing "Matrix eQTL" package is utilized and extended to study methylation quantitative trait loci (methQTL) and the interaction of genotype and environment (GxE) to determine DNA methylation variation, using matrix based iterative correlation and memory-efficient data analysis. GEM can facilitate reliable genome-wide methQTL and GxE analysis on a standard laptop computer within minutes.

## Author(s)

Hong Pan

## References

https://github.com/fastGEM/GEM

## See Also

`GEM_GUI`

## Examples

```
## Launch GEM GUI
#GEM_GUI()  # remove the hash symbol for running

## Checking the vignettes for more details
if(interactive()) browseVignettes(package = 'GEM')
```

GEM_Emodel *GEM_Emodel Analysis*

## Description

GEM_Emodel is to find the association between methylation and environmental factor genome widely.

## Usage

```
GEM_Emodel(env_file_name, covariate_file_name, methylation_file_name, Emodel_pv,
  output_file_name, qqplot_file_name, savePlot = TRUE)
```

## Arguments

| | |
|---|---|
| env_file_name | Text file with rows representing environment factor and columns representing samples, such as the example data file "env.txt". |
| covariate_file_name | Text file with rows representing covariate factors and columns representing samples, such as the example data file "cov.txt". |
| methylation_file_name | Text file with rows representing methylation profiles for CpGs, and columns representing samples, such as the example data file "methylation.txt". |
| Emodel_pv | The pvalue cut off. Associations with significances at Emodel_pv level or below are saved to output_file_name, with corresponding estimate of effect size (slope coefficient), test statistics and p-value. Default value is 1.0. |
| output_file_name | The result file with each row presenting a CpG and its association with environment, which contains CpGID, estimate of effect size (slope coefficient), test statistics, pvalue and FDR at each column. |
| qqplot_file_name | Image file name to present the QQ-plot for all p-value distribution. |
| savePlot | If save the plot. |

## Details

GEM_Emodel finds the association between methylation and environment genome-wide by performing matrix based iterative correlation and memory-efficient data analysis instead of millions of linear regressions (N = number_of_CpGs). The methylation data are the measurements for CpG probes, for example, 450,000 CpGs from Illumina Infinium HumanMethylation450 Array. The environmental factor can be a particular phenotype or environment factor from,for example, birth outcomes, maternal conditions or disease traits. The output of GEM_Emodel for particular environmental factor is a list of CpGs that are potential epigenetic biomarkers. GEM_Emodel runs linear regression like lm (M ~ E + covt), where M is a matrix with methylation data, E is a matrix with environment factor and covt is a matrix with covariates, and all read from the formatted text data file.

## Value

save results automatically

## Examples

```
DATADIR = system.file('extdata',package='GEM')
RESULTDIR = getwd()
env_file = paste(DATADIR, "env.txt", sep = .Platform$file.sep)
covariate_file = paste(DATADIR, "cov.txt", sep = .Platform$file.sep)
methylation_file = paste(DATADIR, "methylation.txt", sep = .Platform$file.sep)
Emodel_pv = 1
output_file = paste(RESULTDIR, "Result_Emodel.txt", sep = .Platform$file.sep)
qqplot_file = paste(RESULTDIR, "QQplot_Emodel.jpg", sep = .Platform$file.sep)
GEM_Emodel(env_file, covariate_file, methylation_file, Emodel_pv, output_file, qqplot_file)
```

---

# GEM_Gmodel *GEM_Gmodel Analysis*

---

## Description

GEM_Gmodel creates a methQTL genome-wide map.

## Usage

```
GEM_Gmodel(snp_file_name, covariate_file_name, methylation_file_name, Gmodel_pv,
  output_file_name)
```

## Arguments

| | |
|---|---|
| snp_file_name | Text file with rows representing genotype encoded as 1,2,3 or any three distinct values for major allele homozygote (AA), heterozygote (AB) and minor allele homozygote (BB) and columns representing samples, such as the example data file "snp.txt". |
| covariate_file_name | Text file with rows representing covariate factors and columns representing samples, such as the example data file "cov.txt". |
| methylation_file_name | Text file with rows representing methylation profiles for CpGs, and columns representing samples, such as the example data file "methylation.txt". |
| Gmodel_pv | The pvalue cut off. Associations with significances at Gmodel_pv level or below are saved to output_file_name, with corresponding estimate of effect size (slope coefficient), test statistics and p-value. Default value is 5.0E-08. |
| output_file_name | The result file with each row presenting a CpG and its association with SNP, which contains CpGID, SNPID, estimate of effect size (slope coefficient), test statistics, pvalue and FDR at each column. |

## Details

GEM_Gmodel creates a methQTL genome-wide map by performing matrix based iterative correlation and memory-efficient data analysis instead of millions of linear regressions (N = number_of_CpGs x number_of_SNPs) between methylation and genotyping. Polymorphisms close to CpGs in the same chromosome (cis-) or different chromosome (trans-) often form methylation quantitative trait loci (methQTLs) with CpGs. In GEM_Gmodel, MethQTLs can be discovered by correlating single nucleotide polymorphism (SNP) data with CpG methylation from the same samples, by linear regression lm (M ~ G + covt), where M is a matrix with methylation data, G is a matrix with genotype data and covt is a matrix with covariates, and all read from the formatted text data file. The methylation data are the measurements for CpG probes, for example, 450,000 CpGs from Illumina Infinium HumanMethylation450 Array. The genotype data are encoded as 1,2,3 or any three distinct values for major allele homozygote (AA), heterozygote (AB) and minor allele homozygote (BB). The linear regression is adjusted by covariates read from covariate data file. The output of GEM_Gmodel is a list of CpG-SNP pairs, where the SNP is the best fit to explain the particular CpG. The significant association between CpG-SNP pair suggests the methylation driven by genotyping variants, which is so called methylation quantitative trait loci (methQTL).

## Value

save results automatically

## Examples

```
DATADIR = system.file('extdata',package='GEM')
RESULTDIR = getwd()
snp_file = paste(DATADIR, "snp.txt", sep = .Platform$file.sep)
covariate_file = paste(DATADIR, "cov.txt", sep = .Platform$file.sep)
methylation_file = paste(DATADIR, "methylation.txt", sep = .Platform$file.sep)
Gmodel_pv = 1e-04
output_file = paste(RESULTDIR, "Result_Gmodel.txt", sep = .Platform$file.sep)
GEM_Gmodel(snp_file, covariate_file, methylation_file, Gmodel_pv, output_file)
```

---

# GEM_GUI *Graphical User Interface (GUI) for GEM*

---

## Description

The user friendly GUI for runing GEM package easily and quickly

## Usage

```
GEM_GUI()
```

## Details

The GEM package provides a highly efficient R tool suite for performing epigenome wide association studies (EWAS). GEM provides three major functions named GEM_Emodel, GEM_Gmodel and GEM_GxEmodel to study the interplay of Gene, Environment and Methylation (GEM). Within GEM, the pre-existing "Matrix eQTL" package is utilized and extended to study methylation quantitative trait loci (methQTL) and the interaction of genotype and environment (GxE) to determine DNA methylation variation, using matrix based iterative correlation and memory-efficient data analysis. GEM can facilitate reliable genome-wide methQTL and GxE analysis on a standard laptop computer within minutes.

## Value

GEM model analysis results

## See Also

GEM-package

## Examples

```
interactive()
#GEM_GUI()  ## remove the hash symbol to run
```

---

GEM_GWASmodel *GEM_GWASmodel*

---

## Description

GEM_GWASmodel performs genome wide association study (GWAS).

## Usage

```
GEM_GWASmodel(env_file_name, snp_file_name, covariate_file_name, GWASmodel_pv,
  output_file_name, qqplot_file_name)
```

## Arguments

| | |
|---|---|
| env_file_name | Text file with rows representing environment factor and columns representing samples, such as the example data file "env.txt". |
| snp_file_name | Text file with rows representing genotype encoded as 1,2,3 or any three distinct values for major allele homozygote (AA), heterozygote (AB) and minor allele homozygote (BB) and columns representing samples, such as the example data file "snp.txt". |
| covariate_file_name | Text file with rows representing covariate factors, and columns representing samples, such as the example data file "cov.txt". |

GWASmodel_pv The pvalue cut off. Associations with significances at GWASmodel_pv level or below are saved to output_file_name, with corresponding estimate of effect size (slope coefficient), test statistics and p-value. Default value is 5.0E-08.

output_file_name
The result file with each row presenting a SNP and its association with environment, which contains SNPID, estimate of effect size (slope coefficient), test statistics, pvalue and FDR at each column.

qqplot_file_name
Output QQ plot for all pvalues.

## Details

GEM_GWASmodel finds the association between genetic variants and environment genome-wide by performing matrix based iterative correlation and memory-efficient data analysis instead of millions of linear regressions (N = number_of_SNPs). The environmental factor can be a particular phenotype or environment factor from,for example, birth outcomes, maternal conditions or disease traits. The genotype data are encoded as 1,2,3 or any three distinct values for major allele homozygote (AA), heterozygote (AB) and minor allele homozygote (BB). The linear regression is adjusted by covariates read from covariate data file. The output of GEM_GWASmodel is a list of SNPs and their association with environment. GEM_GWASmodel runs linear regression like lm (E ~ G + covt), where G is a matrix with genotype data, E is a matrix with environment factor and covt is a matrix with covariates, and all read from the formatted text data file.

## Value

save results automatically

## Examples

```
DATADIR = system.file('extdata',package='GEM')
RESULTDIR = getwd()
snp_file = paste(DATADIR, "snp.txt", sep = .Platform$file.sep)
covariate_file = paste(DATADIR, "cov.txt", sep = .Platform$file.sep)
env_file = paste(DATADIR, "env.txt", sep = .Platform$file.sep)
GWASmodel_pv = 1e-5
output_file = paste(RESULTDIR, "Result_GxEmodel.txt", sep = .Platform$file.sep)
qqplot_file = paste(RESULTDIR, "Result_GxEmodel.jpeg", sep = .Platform$file.sep)
GEM_GWASmodel(env_file, snp_file, covariate_file, GWASmodel_pv, output_file, qqplot_file)
```

---

GEM_GxEmodel *GEM_GxEmodel*

---

## Description

GEM_GxEmodel tests the ability of the interaction of gene and environmental factor to predict DNA methylation level.

## Usage

```
GEM_GxEmodel(snp_file_name, covariate_file_name, methylation_file_name,
  GxEmodel_pv, output_file_name, topKplot = 10, savePlot = TRUE)
```

## Arguments

| | |
|---|---|
| snp_file_name | Text file with rows representing genotype encoded as 1,2,3 or any three distinct values for major allele homozygote (AA), heterozygote (AB) and minor allele homozygote (BB) and columns representing samples, such as the example data file "snp.txt". |
| covariate_file_name | Text file with rows representing covariate factors and the envirnoment value, and the environment value should be put in the last row, and columns representing samples, such as the example data file "gxe.txt". |
| methylation_file_name | Text file with rows representing methylation profiles for CpGs, and columns representing samples, such as the example data file "methylation.txt". |
| GxEmodel_pv | The pvalue cut off. Associations with significances at GxEmodel_pv level or below are saved to output_file_name, with corresponding estimate of effect size (slope coefficient), test statistics and p-value. Default value is 5.0E-08. |
| output_file_name | The result file with each row presenting a CpG and its association with SNPx-Env, which contains CpGID, SNPID, estimate of effect size (slope coefficient), test statistics, pvalue and FDR at each column. |
| topKplot | The top number of topKplot CpG-SNP-Env triplets will be presented into charts to demonstrate how environment values segregated by SNP groups can explain methylation. |
| savePlot | if save the plot. |

## Details

GEM_GxEmodel explores how the genotype can work in interaction with environment (GxE) to influence specific DNA methylation level, by performing matrix based iterative correlation and memory-efficient data analysis among methylation, genotyping and environment. This has greatly released the computational burden for GxE study from billions of linear regression (N = number_of_CpGs x number_of_SNPs x number_of_environment) and made it possible to be accomplished in an efficient way. The linear regression is lm (M ~ G x E + covt), where M is a matrix with methylation data, G is a matrix with genotype data, E is environment data and covt is covariate matrix. E values is combined into covariate file as the last row, and all read from the formatted text data file. The output of GEM_GxEmodel is a list of CpG-SNP-Env triplets, where the environment factor segregated in genotype group fits to explain the particular CpG. The significant association suggests the association between methylation and environment can be better explained by segregation in genotype groups (GxE).

## Value

save results automatically

## Examples

```
DATADIR = system.file('extdata',package='GEM')
RESULTDIR = getwd()
snp_file = paste(DATADIR, "snp.txt", sep = .Platform$file.sep)
covariate_file = paste(DATADIR, "gxe.txt", sep = .Platform$file.sep)
methylation_file = paste(DATADIR, "methylation.txt", sep = .Platform$file.sep)
GxEmodel_pv = 1e-4
output_file = paste(RESULTDIR, "Result_GxEmodel.txt", sep = .Platform$file.sep)
GEM_GxEmodel(snp_file, covariate_file, methylation_file, GxEmodel_pv, output_file)
```

---

# SlicedData-class *Class* SlicedData *for storing large matrices*

---

## Description

This class is created for fast and memory efficient manipulations with large datasets presented in matrix form. It is used to load, store, and manipulate large datasets, e.g. genotype and gene expression matrices. When a dataset is loaded, it is sliced in blocks of 1,000 rows (default size). This allows imputing, standardizing, and performing other operations with the data with minimal memory overhead.

## Usage

```
# x[[i]] indexing allows easy access to individual slices.
# It is equivalent to x$GetSlice(i) and x$SetSlice(i,value)
## S4 method for signature 'SlicedData'
x[[i]]
## S4 replacement method for signature 'SlicedData'
x[[i]] <- value

# The following commands work as if x was a simple matrix object
## S4 method for signature 'SlicedData'
nrow(x)
## S4 method for signature 'SlicedData'
ncol(x)
## S4 method for signature 'SlicedData'
dim(x)
## S4 method for signature 'SlicedData'
rownames(x)
## S4 method for signature 'SlicedData'
colnames(x)
## S4 replacement method for signature 'SlicedData'
rownames(x) <- value
## S4 replacement method for signature 'SlicedData'
colnames(x) <- value

# SlicedData object can be easily transformed into a matrix
```

```
# preserving row and column names
## S4 method for signature 'SlicedData'
as.matrix(x)

# length(x) can be used in place of x$nSlices()
# to get the number of slices in the object
## S4 method for signature 'SlicedData'
length(x)
```

## Arguments

| | |
|---|---|
| x | SlicedData object. |
| i | Number of a slice. |
| value | New content for the slice / new row or column names. |

## Value

check the resutls

## Extends

SlicedData is a reference classes (envRefClass). Its methods can change the values of the fields of the class.

## Fields

**dataEnv:** environment. Stores the slices of the data matrix. The slices should be accessed via getSlice() and setSlice() methods.

**nSlices1:** numeric. Number of slices. For internal use. The value should be access via nSlices() method.

**rowNameSlices:** list. Slices of row names.

**columnNames:** character. Column names.

**fileDelimiter:** character. Delimiter separating values in the input file.

**fileSkipColumns:** numeric. Number of columns with row labels in the input file.

**fileSkipRows:** numeric. Number of rows with column labels in the input file.

**fileSliceSize:** numeric. Maximum number of rows in a slice.

**fileOmitCharacters:** character. Missing value (NaN) representation in the input file.

## Methods

**initialize(mat):** Create the object from a matrix.

**nSlices():** Returns the number of slices.

**nCols():** Returns the number of columns in the matrix.

**nRows():** Returns the number of rows in the matrix.

**Clear():** Clears the object. Removes the data slices and row and column names.

`Clone()`: Makes a copy of the object. Changes to the copy do not affect the source object.

`CreateFromMatrix(mat)`: Creates `SlicedData` object from a `matrix`.

`LoadFile(filename, skipRows = NULL, skipColumns = NULL, sliceSize = NULL, omitCharacters = NULL, delimiter`
Loads data matrix from a file. `filename` should be a character string. The remaining parameters specify the file format and have the same meaning as `file*` fields. Additional `rowNamesColumn` parameter specifies which of the columns of row labels to use as row names.

`SaveFile(filename)`: Saves the data to a file. `filename` should be a character string.

`getSlice(sl)`: Retrieves `sl`-th slice of the matrix.

`setSlice(sl, value)`: Set `sl`-th slice of the matrix.

`ColumnSubsample(subset)`: Reorders/subsets the columns according to `subset`.
Acts as `M = M[ ,subset]` for a matrix `M`.

`RowReorder(ordr)`: Reorders rows according to `ordr`.
Acts as `M = M[ordr, ]` for a matrix `M`.

`RowMatrixMultiply(multiplier)`: Multiply each row by the `multiplier`.
Acts as `M = M %*% multiplier` for a matrix `M`.

`CombineInOneSlice()`: Combines all slices into one. The whole matrix can then be obtained via `$getSlice(1)`.

`IsCombined()`: Returns `TRUE` if the number of slices is 1 or 0.

`ResliceCombined(sliceSize = -1)`: Cuts the data into slices of `sliceSize` rows. If `sliceSize` is not defined, the value of `fileSliceSize` field is used.

`GetAllRowNames()`: Returns all row names in one vector.

`RowStandardizeCentered()`: Set the mean of each row to zero and the sum of squares to one.

`SetNanRowMean()`: Impute rows with row mean. Rows full of NaN values are imputed with zeros.

`RowRemoveZeroEps()`: Removes rows of zeros and those that are nearly zero.

`FindRow(rowname)`: Finds row by name. Returns a pair of slice number an row number within the slice. If no row is found, the function returns `NULL`.

`rowMeans(x, na.rm = FALSE, dims = 1L)`: Returns a vector of row means. Works as rowMeans but requires `dims` to be equal to `1L`.

`rowSums(x, na.rm = FALSE, dims = 1L)`: Returns a vector of row sums. Works as rowSums but requires `dims` to be equal to `1L`.

`colMeans(x, na.rm = FALSE, dims = 1L)`: Returns a vector of column means. Works as colMeans but requires `dims` to be equal to `1L`.

`colSums(x, na.rm = FALSE, dims = 1L)`: Returns a vector of column sums. Works as colSums but requires `dims` to be equal to `1L`.

## Author(s)

Andrey Shabalin <ashabalin@vcu.edu>

## References

The package website: http://www.bios.unc.edu/research/genomic_software/Matrix_eQTL/

## Examples

```
# Create a SlicedData variable
```

# Index


∗ **classes**
SlicedData-class, 9
[[,SlicedData-method
(SlicedData-class), 9
[[<-,SlicedData-method
(SlicedData-class), 9

as.matrix,SlicedData-method
(SlicedData-class), 9

colMeans, *11*
colMeans,SlicedData-method
(SlicedData-class), 9
colnames,SlicedData-method
(SlicedData-class), 9
colnames<-,SlicedData-method
(SlicedData-class), 9
colSums, *11*
colSums,SlicedData-method
(SlicedData-class), 9

dim,SlicedData-method
(SlicedData-class), 9

envRefClass, *10*

GEM-package, 2
GEM_Emodel, *2*, 3, *6*
GEM_Gmodel, *2*, 4, *6*
GEM_GUI, *2*, 5
GEM_GWASmodel, 6
GEM_GxEmodel, *2*, *6*, 7

length,SlicedData-method
(SlicedData-class), 9

matrix, *11*

NCOL,SlicedData-method
(SlicedData-class), 9
ncol,SlicedData-method
(SlicedData-class), 9
NROW,SlicedData-method
(SlicedData-class), 9
nrow,SlicedData-method
(SlicedData-class), 9

rowMeans, *11*
rowMeans,SlicedData-method
(SlicedData-class), 9
rownames,SlicedData-method
(SlicedData-class), 9
rownames<-,SlicedData-method
(SlicedData-class), 9
rowSums, *11*
rowSums,SlicedData-method
(SlicedData-class), 9

show,SlicedData-method
(SlicedData-class), 9
SlicedData, *10*
SlicedData *(*SlicedData-class*)*, 9
SlicedData-class, 9
summary.SlicedData *(*SlicedData-class*)*, 9