Package ‘regionalpcs’

January 6, 2024

Title  Summarizing Regional Methylation with Regional Principal Components Analysis

Version  1.0.0

Description Functions to summarize DNA methylation data using regional principal components. Regional principal components are computed using principal components analysis within genomic regions to summarize the variability in methylation levels across CpGs. The number of principal components is chosen using either the Marcenko-Pastur or Gavish-Donoho method to identify relevant signal in the data.

License  MIT + file LICENSE

Encoding UTF-8

Roxygen list(markdown = TRUE)

RoxygenNote  7.2.3

URL  https://github.com/tyeulalio/regionalpcs

BugReports  https://github.com/tyeulalio/regionalpcs/issues

biocViews DNAmethylation, DifferentialMethylation, StatisticalMethod, Software, MethylationArray

Imports dplyr, PCAtools, tibble, GenomicRanges

Suggests knitr, rmarkdown, RMTstat, testthat (>= 3.0.0), BiocStyle, tidyr, minfiData, TxDb.Hsapiens.UCSC.hg19.knownGene, IRanges

VignetteBuilder knitr

Depends R (>= 4.3.0)

LazyData false

git_url  https://git.bioconductor.org/packages/regionalpcs

git_branch  RELEASE_3_18

git_last_commit  00bdaf5

git_last_commit_date  2023-10-24

Repository Bioconductor 3.18
**combine_results**

**Description**
Combine results dataframes across regions

**Usage**
```
combine_results(res, df_name)
```

**Arguments**
- `res` List of lists; contains summarized region results
- `df_name` String; name of result being combined (sig_pcs or percent_var)

**Value**
Data Frame containing results

**Examples**
```
# Create example data for 'sig_pcs' and 'percent_var'
sig_pcs_example <- data.frame(pcs = c("PC1", "PC2"), value = c(0.2, 0.4))
percent_var_example <- data.frame(pcs = c("PC1", "PC2"), value = c(0.7, 0.3))

# Create 'res' list containing both 'sig_pcs' and 'percent_var'
res <- list(region = "Region1", sig_pcs = sig_pcs_example, percent_var = percent_var_example)

# Example function use: Combine 'sig_pcs' across regions
```
```
combined_sig_pcs <- combine_results(res, df_name = "sig_pcs")
print(combined_sig_pcs)
```

---

**compute_dimension**  
*Compute significant dimensions of a matrix using the Marchenko-Pastur or Gavish-Donoho methods*

**Description**  
Compute significant dimensions of a matrix using the Marchenko-Pastur or Gavish-Donoho methods

**Usage**  
```r
compute_dimension(
  x,  
  var_explained,  
  noise_select,  
  pc_method = c("gd", "mp"),  
  verbose = FALSE
)
```

**Arguments**  
- **x**: A data frame or matrix of methylation values; rows = features, columns = samples  
- **var_explained**: A numeric vector containing the variance explained by successive PCs, sorted in decreasing order. (Used for PCAtools)  
- **noise_select**: Numeric scalar specifying the variance of the random noise (Used for PCAtools)  
- **pc_method**: String indicating the method for estimating dimension; "gd" = Gavish-Donoho, "mp" = Marchenko-Pastur  
- **verbose**: Boolean indicating whether to print statements while running, default = FALSE

**Value**  
Numeric scalar representing the optimal number of PCs to retain using the specified method

**Examples**  
```r
x <- diag(4)
pca_res <- PCAtools::pca(x) # Run PCA
eig_sq <- pca_res$sdev^2 # Compute variance explained
compute_dimension(x, eig_sq, 1, "gd")
```
compute_regional_pcs  

Compute regional principal components for methylation data

Description

Compute regional principal components for methylation data

Usage

compute_regional_pcs(
  meth,
  region_map,
  pc_method = c("gd", "mp"),
  verbose = FALSE
)

Arguments

meth               Data frame of methylation beta values, with CpGs in rows and samples in columns
region_map         Data frame mapping CpGs to gene regions
pc_method          Method to use for PC computation, either 'gd' (Gavish-Donoho) or 'mp' (Marchenko-Pastur)
verbose            Logical, should progress messages be displayed?

Value

A list containing several elements, including the regional PCs, percent variance, and other information

Examples

# Create synthetic methylation data
meth_data <- matrix(rnorm(1000), nrow = 100, ncol = 10)
rownames(meth_data) <- paste0("CpG", 1:100)
colnames(meth_data) <- paste0("Sample", 1:10)

# Create a synthetic region map
region_map_data <- data.frame(
  region_id = rep(c("Gene1", "Gene2"), each = 50),
  cpg_id = rownames(meth_data)
)

# Run the function
compute_regional_pcs(meth_data, region_map_data, pc_method = 'gd')
Create a Region Map Between CpGs and Gene Regions

**Description**

This function generates a map that assigns CpG sites to gene regions, establishing a linkage based on their genomic coordinates and providing a foundation for subsequent region-specific analyses.

**Usage**

```r
create_region_map(cpg_gr, genes_gr, verbose = FALSE)
```

**Arguments**

- `cpg_gr`: A `GRanges` object containing the genomic positions of CpG sites.
- `genes_gr`: A `GRanges` object containing the genomic positions of gene regions (e.g., promoters) of interest.
- `verbose`: Boolean; print output statements

**Value**

A `data.frame` with mappings between gene IDs and CpG IDs, facilitating associating CpG sites with their corresponding gene regions for downstream analyses.

**Examples**

```r
library(GenomicRanges)

# Creating dummy GRanges objects for CpG sites and gene regions
cpg_gr <- GRanges(seqnames=c("chr1", "chr1", "chr2"),
                  ranges=IRanges(start=c(100, 200, 150),
                                end=c(100, 200, 150)))
genes_gr <- GRanges(seqnames=c("chr1", "chr2", "chr2"),
                  ranges=IRanges(start=c(50, 100, 130),
                                 end=c(150, 180, 160)))

# Creating a region map using the function
region_map <- create_region_map(cpg_gr, genes_gr)
```
get_sig_pcs  

*Get significant principal components*

**Description**

Get significant principal components

**Usage**

```r
get_sig_pcs(x, pc_method = c("mp", "gd"), verbose = FALSE)
```

**Arguments**

- `x`: A data frame or matrix of methylation values; rows = features, columns = samples
- `pc_method`: String indicating the method for estimating dimension; "gd" = Gavish-Donoho (default), "mp" = Marchenko-Pastur
- `verbose`: Boolean; print output statements

**Value**

List containing four elements; `sig_pcs` = significant PCs, `percent_var` = percent variance explained, `loadings` = PC loadings, `est_dim` = estimated dimension

**Examples**

```r
x <- diag(4)
get_sig_pcs(x, "gd")
```

---

summarize_region  

*Summarize a region using regional principal components*

**Description**

Summarize a region using regional principal components

**Usage**

```r
summarize_region(region, region_map, meth, pc_method, verbose = FALSE)
```
Arguments

- **region**
  - String; name of region being processed

- **region_map**
  - Data frame; Mapping of CpGs to regions, column 1 should be regions, column 2 should be CpGs with the same names as the rows of meth

- **meth**
  - Data frame or matrix; Methylation values to summarize; rows=CpGs, columns=samples

- **pc_method**
  - String; indicating the method for estimating dimension; "gd"=Gavish-Donoho (default), "mp"=Marchenko-Pastur

- **verbose**
  - Boolean; print output statements

Value

- list containing PC results

Examples

```r
# Create the region map with just one region containing 10 CpGs
region_map <- data.frame(region_id = rep(1, 10), cpg_id = seq(1, 10))

# Create methylation data frame
set.seed(123)
meth <- as.data.frame(matrix(runif(10 * 20, min = 0, max = 1), nrow = 10))
rownames(meth) <- seq(1, 10)

# Call the function
summarize_region(1, region_map, meth, 'gd')
```
Index

combine_results, 2
compute_dimension, 3
compute_regional_pcs, 4
create_region_map, 5

get_sig_pcs, 6

summarize_region, 6