Package 'MegaCollapsing'

January 23, 2020

Type Package

Title Collapsing Analysis at the MegaGene Level

Version 0.1.0

Date 2020-01-31

Author Quanli Wang and Slave Petrovski

Maintainer Quanli Wang<quanliwang20@gmail.com>

Description Taking results from a gene/sample level varaint count matrix and a set of gene sets, the package collapses gene/sample level variant counts into gene-set (MegaGene)/sample level counts and then allow users to run Fisher Exact Test efficiently.

License GPL(>=3)

LazyData TRUE

R topics documented:

as.mega.matrix *Generate mega collapsing matrix/matrices for a list of gene sets.*

Description

Taking collapsing matrix/matrices and a list of gene sets, this function "collapses" gene/sample level variant information to gene set level varaint information, which can further be used for (mega) collapsing analysis.

Usage

```
as.mega.matrix(gene.sets, input.data)
```

Arguments

gene.sets	The gene sets object that hold list of gene sets. It is typically obtained by calling function read.gene.sets.
input.data	The objects that holds the collapsing matrix/matrices. It is typically obtained by calling function read.collapsing.data.

Details

A lot of explaination here if we wanted. Or provide a link to the method.

Value

Return a list of objects that represent the mega collapsing matrix/matices.

burden.test *Run regression/test using collapsing matrix with covaraiates.*

Description

Add a lot of explainations here to justify why we set up the modeld this way.

Usage

```
burden.test(sample.list, gene.sets, syn.mat, non.syn.mat, X = NULL, Y = NULL)
```

Arguments

sample.list	An object that hold information/covaraites about samples.
gene.sets	The list of gene sets to be anallize.
syn.mat	The syn collsping matrix to be used. We should have an option not to use this one.
non.syn.mat	The non syn collsping matrix to be used.
X	a charactor list of covariates to be used in regression models.
Y	The response varaible. 1 for cases and 0 for controls.

Details

Explain.

Value

Return a list of test results and summary statistics. Will document this further once we decide what to keep.

`burden.test.with.permutation`

Permutation based Burden Test

Description

Don't do this unless we really want to do it. This will run permutation based on Burden Test for given gene list and input mega/collapsing matrix.

Usage

```
burden.test.with.permutation(sample.list, gene.set, syn.mat, non.syn.mat,
                             X = NULL, Y = NULL, n.permutations = 100)
```

Arguments

`sample.list`	An object that hold information/covaraites about samples.[Copied from burden test]
`gene.set`	The list of gene sets to be anallize.
`syn.mat`	The syn collsping matrix to be used. We should have an option not to use this one.
`non.syn.mat`	The non syn collsping matrix to be used.
`X`	a charactor list of covariates to be used in regression models.
`Y`	The response varaible. 1 for cases and 0 for controls.
`n.permutations`	Number of permutations to use in this test. Default to 100.

Value

A list that holds the actual burden test results and permutated results. Add more if we are to keep this function.

`exclude.genes` *Exclude a set of genes from input collapsing matrices before running MegaCollapsing Analysis.*

Description

It might be desire to remove genes that are causing trouble before running MegaCollapsing and this function can be used for that purpose.

Usage

```
exclude.genes(input.data, exclude.file)
```

Arguments

`input.data`	The output from function call to read.collapsing.data.
`exclude.file`	The file contains the list of genes to be excluded before analysis.

Details

Add more about how it was implemented and the reason why we might want to do this.

Value

A list that has the same structure as input.data.

`get.pvalues` *Run permutation based Fisher Test on (Mega)collapsing matrix.*

Description

This function allows one to run permutation based FET tests. Not sure if we need this one in this package though.

Usage

```
get.pvalues(matrix, is.case, n.permutations = 1000)
```

Arguments

`matrix`	A (mega) collapsing matrix.
`is.case`	A boolean vector indicating case/control status.
`n.permutations`	Number of permutations to run the tests.

Details

Running permutation based FET test from (mega)Collapsing matrix. Will explain more if we will keep this function.

Value

A list that holds both FET p values and permutatied p-values. These p-values can be used to do permutation based QQ plot

---

`random.gene.sets` *Generate random gene sets from a list of genes that share similar structure with an input gene set*

Description

This function is used to generate randomized sets of genes that can be used to evaluate test results. This was mainly introduced to calibrate our test results for the giving input data. Need to investigate/explore more about its usage. Might want to remove it from our package if we are not going to use it.

Usage

```
random.gene.sets(gene.sets, genes)
```

Arguments

`gene.sets`	The actual gene sets that would be used in a MegaCollapsing analysis.
`genes`	A list of genes that would be used to generate randomzied gene sets.

Details

Explain/explore why we had this function.

Value

A randomized sets of genes that share the same set length with the input gene sets but with each gene set populated with random genes.

`read.collapsing.data` *Read all data needed for a "paired burden test.*

Description

This function is currently used by CGR to run MegaCollapsing analsyis that that uses both synonamous and "non-synonamous" collapsing matrices. We need to come up with better names and descriptions for this kind of analysis.

Usage

```
read.collapsing.data(samples, syn, non.syn, sample.column = "IID")
```

Arguments

samples	The input PED file to define the list of samples to be used in analysis.
syn	The collapsing matrix for syn model.
non.syn	The collapsing matrix for matrix to be acutally tested.
sample.column	The column name that is used to identify the sample ID from input sample file.

Details

Need a lot of description here to describe we are trying to do. Obvious bug: the passed in "sample.column"" was not used.

Value

Return a lost of objects that are parsed and matched for "paired burden test"

sample.list	The final sample list used for analysis
syn	The final matching collapsing matrix for syn model
non.syn	The final matching collapsing matrix for non-syn model

---

read.collapsing.matrix

Read gene/sample varaint count (collapsing) matrix.

Description

This is a helper function that parses the input gene/sample varaint count (collapsing) matrix into an R matrix.

Usage

```
read.collapsing.matrix(matrix_file)
```

Arguments

matrix_file	The input collapsing matrix.

Details

Describe what a collapsing matrix looks like.

Value

Returns a matrix that represents the collapsing matrix, with columns for samples and rows for genes.

`read.data` *Read gene/sample variant count matrix.*

Description

This is a helper function to extract gene/sample variant count matrix based on an input sample list and also optionally a list of genes to be used as filter.

Usage

```
read.data(sample.file, matrix.file, filter.list = NULL, sample.column = 2,
          case.control.column = 6)
```

Arguments

`sample.file`	An input sample list file. The default format is a standard PED file that is tab delimited, without column header and has second column indicating the sample ID/name and the 6 column indicating the control/case status, with 1 for control sample and 2 for case sample. User can also provide a tab delimited text file and indicating the column index for sample ID and case/control status.
`matrix.file`	The gene/sample count (collapsing) matrix to be read. A gene/sample count matrix is a tab delimited spreadsheet like text file with each column represents a sample, and each row represents a gene. Gene names will be stored in the first column an the sample names will be stored in the first row.
`filter.list`	Optional. A flat text file with each row gives a gene that is to be extracted from the input matrix file. By default, all genes found in the matrix file will be used.
`sample.column`	Optional. If the sample ID column from the sample file is not the second column, user will need to provide the index(one based) for actutal sample ID column.
`case.control.column`	Optional. If the case/control status column from the sample file is not the second column, user will need to provide the index(one based) for actutal sample ID column.

Value

Return a list containing the extracted case/control status and the selected gene/sample count matrix.

`data`	A matrix for the extracted gene/sample count
`is.case`	A boolean vector indicating the case/control status

`read.gene.sets` *Read the gene set definitition from input text file*

Description

This function read the gene set to be used for mega collapsing analysis.

Usage

```
read.gene.sets(meta_gene_file)
```

Arguments

`meta_gene_file`	The input file that defines gene sets.

Details

Need to add more details about the gene set format we are using here and the minimal information we will need to create the geneset information.

Value

Return a list of gene sets.

`read.sample.list` *Read and format a sample list file.*

Description

This is a helper function to parse and standardize the sample names from sample list file. It assumes that the sample list file is tab demilited, has a column header and the sample ID column is labeled by column header "IID". To do: make this more generic and allow user to change the format.

Usage

```
read.sample.list(sample_file)
```

Arguments

`sample_file`	The input sample list file. Tab delimited, with column header and the sample ID column is indicated by column name IID.

Details

This is a helper function used only when the sample file has column headers and the some sample names are just integers. In this case, we will add an "X" in front the integer to make it a proper sample name. We might want to remove this function later.

Value

Retrun a data frame represent the sample list.

`read.seed.list` *A helper function to read "seed genes"*

Description

Forget what this is supposed to be. Will need to investigate if we need it or not, or if we need to re-implement it to fit our purpose.

Usage

```
read.seed.list(seed_set)
```

Arguments

`seed_set`	An input for "seed set" of genes.

Details

Need to figure how this is used (or not).

Value

Return a list of "seed set"

Index