Package 'omicwas'

October 8, 2020

Type Package

Title Cell-Type-Specific Association Testing in Bulk Omics Experiments

Version 0.8.0

Description In bulk epigenome/transcriptome experiments, molecular expression
is measured in a tissue, which is a mixture of multiple types of cells.
This package tests association of a disease/phenotype with a molecular marker
for each cell type.
The proportion of cell types in each sample needs to be given as input.
The package is applicable to epigenome-wide association study (EWAS) and
differential gene expression analysis.
Takeuchi and Kato (submitted)
``omicwas: cell-type-specific epigenome-wide and transcriptome association study".

URL https://github.com/fumi-github/omicwas

BugReports https://github.com/fumi-github/omicwas/issues

Depends R (>= 3.6.0)

biocViews

License GPL-3

Encoding UTF-8

LazyData true

Imports broom, data.table, dplyr, ff, glmnet, magrittr, MASS,
matrixStats, parallel, purrr, rlang, tidyr

RoxygenNote 7.1.1

Suggests testthat, knitr, rmarkdown

VignetteBuilder knitr

NeedsCompilation no

Author Fumihiko Takeuchi [aut, cre] (<https://orcid.org/0000-0003-3185-5661>)

Maintainer Fumihiko Takeuchi <fumihiko@takeuchi.name>

Repository CRAN

Date/Publication 2020-10-08 12:50:03 UTC

R topics documented:

`ctassoc` *Cell-Type-Specific Association Testing*

Description

Cell-Type-Specific Association Testing

Usage

```
ctassoc(
  X,
  W,
  Y,
  C = NULL,
  test = "full",
  regularize = FALSE,
  num.cores = 1,
  chunk.size = 1000,
  seed = 123
)
```

Arguments

`X`	Matrix (or vector) of traits; samples x traits.
`W`	Matrix of cell type composition; samples x cell types.
`Y`	Matrix (or vector) of bulk omics measurements; markers x samples.
`C`	Matrix (or vector) of covariates; samples x covariates. X, W, Y, C should be numeric.
`test`	Statistical test to apply; either `"full"`, `"marginal"`, `"nls.identity"`, `"nls.log"`, `"nls.logit"`, `"propdiff.identity"`, `"propdiff.log"`, `"propdiff.logit"` or `"reducedrankridge"`.
`regularize`	Whether to apply Tikhonov (ie ridge) regularization to β_{hjk}. The regularization parameter is chosen automatically according to an unbiased version of (Lawless & Wang, 1976). Effective for `nls.*` and `propdiff.*` tests.
`num.cores`	Number of CPU cores to use. Full, marginal and propdiff tests are run in serial, thus num.cores is ignored.

`chunk.size`	The size of job for a CPU core in one batch. If you have many cores but limited memory, and there is a memory failure, decrease num.cores and/or chunk.size.
`seed`	Seed for random number generation.

Details

Let the indexes be h for cell type, i for sample, j for marker (CpG site or gene), k for each trait that has cell-type-specific effect, and l for each trait that has a uniform effect across cell types. The input data are X_{ik}, C_{il}, W_{ih} and Y_{ji}, where C_{il} can be omitted. X_{ik} and C_{il} are the values for two types of traits, showing effects that are cell-type-specific or not, respectively. Thus, calling X_{ik} and C_{il} as "traits" and "covariates" gives a rough idea, but is not strictly correct. W_{ih} represents the cell type composition and Y_{ji} represents the marker level, such as methylation or gene expression. For each tissue sample, the cell type proportion W_{ih} is the proportion of each cell type in the bulk tissue, which is measured or imputed beforehand. The marker level Y_{ji} in bulk tissue is measured and provided as input.

The parameters we estimate are the cell-type-specific trait effect β_{hjk}, the tissue-uniform trait effect γ_{jl}, and the basal marker level α_{hj} in each cell type.

We first describe the conventional linear regression models. For marker j in sample i, the maker level specific to cell type h is

$$\alpha_{hj} + \sum_k \beta_{hjk} * X_{ik}.$$

This is a representative value rather than a mean, because we do not model a probability distribution for cell-type-specific expression. The bulk tissue marker level is the average weighted by W_{ih},

$$\mu_{ji} = \sum_h W_{ih}[\alpha_{hj} + \sum_k \beta_{hjk} * X_{ik}] + \sum_l \gamma_{jl} C_{il}.$$

The statistical model is

$$Y_{ji} = \mu_{ji} + \epsilon_{ji},$$

$$\epsilon_{ji}\ N(0, \sigma_j^2).$$

The error of the marker level is is noramlly distributed with variance σ_j^2, independently among samples.

The `full` model is the linear regression

$$Y_{ji} = (\sum_h \alpha_{hj} * W_{ih}) + (\sum_{hk} \beta_{hjk} * W_{ih} * X_{ik}) + (\sum_l \gamma_{jl} * C_{il}) + error.$$

The `marginal` model tests the trait association only in one cell type h, under the linear regression,

$$Y_{ji} = (\sum_{h'} \alpha_{h'j} * W_{ih'}) + (\sum_k \beta_{hjk} * W_{ih} * X_{ik}) + (\sum_l \gamma_{jl} * C_{il}) + error.$$

The nonlinear model simultaneously analyze cell type composition in linear scale and differential expression/methylation in log/logit scale. The normalizing function is the natural logarithm f = log for gene expression, and f = logit for methylation. Conventional linear regression can be formulated by defining f as the identity function. The three models are named `nls.log`, `nls.logit` and

`nls.identity`. We denote the inverse function of f by g; g = exp for gene expression, and g = logistic for methylation. The mean normalized marker level of marker j in sample i becomes

$$\mu_{ji} = f(\sum_h W_{ih} g(\alpha_{hj} + \sum_k \beta_{hjk} * X_{ik})) + \sum_l \gamma_{jl} C_{il}.$$

The statistical model is

$$f(Y_{ji}) = \mu_{ji} + \epsilon_{ji},$$
$$\epsilon_{ji}\ N(0, \sigma_j^2).$$

The error of the marker level is is noramlly distributed with variance σ_j^2, independently among samples.

The ridge regression aims to cope with multicollinearity of the interacting terms $W_{ih} * X_{ik}$. Ridge regression is fit by minimizing the residual sum of squares (RSS) plus $\lambda \sum_{hk} \beta_{hjk}^2$, where $\lambda > 0$ is the regularization parameter.

The propdiff tests try to cope with multicollinearity by, roughly speaking, using mean-centered W_{ih}. We obtain, instead of β_{hjk}, the deviation of β_{hjk} from the average across cell types. Accordingly, the null hypothesis changes. The original null hypothesis was $\beta_{hjk} = 0$. The null hypothesis when centered is $\beta_{hjk} - (\sum_{ih'} W_{ih'} \beta_{h'jk}) / (\sum_{ih'} W_{ih'}) = 0$. It becomes difficult to detect a signal for a major cell type, because β_{hjk} would be close to the average across cell types. The tests `propdiff.log` and `propdiff.logit` include an additional preprocessing step that converts Y_{ji} to $f(Y_{ji})$. Apart from the preprocessing, the computations are performed in linear scale. As the preprocessing distorts the linearity between the dependent variable and (the centered) W_{ih}, I actually think `propdiff.identity` is better.

Value

A list with one element, which is named "coefficients". The element gives the estimate, statistic, p.value in tibble format. In order to transform the estimate for α_{hj} to the original scale, apply `plogis` for `test = nls.logit` and `exp` for `test = nls.log`. The estimate for β_{hjk} by `test = nls.log` is the natural logarithm of fold-change, not the log2. If numerical convergence fails, `NA` is returned for that marker.

References

Lawless, J. F., & Wang, P. (1976). A simulation study of ridge and other regression estimators. Communications in Statistics - Theory and Methods, 5(4), 307–323. https://doi.org/10.1080/03610927608827353

See Also

ctcisQTL

Examples

```
data(GSE42861small)
X = GSE42861small$X
W = GSE42861small$W
Y = GSE42861small$Y
```

```
C = GSE42861small$C
result = ctassoc(X, W, Y, C = C)
result$coefficients
```

`ctcisQTL` *Cell-Type-Specific QTL analysis*

Description

Cell-Type-Specific QTL analysis

Usage

```
ctcisQTL(
  X,
  Xpos,
  W,
  Y,
  Ypos,
  C = NULL,
  max.pos.diff = 1e+06,
  outdir = tempdir(),
  outfile = "ctcisQTL.out.txt"
)
```

Arguments

`X`	Matrix (or vector) of SNP genotypes; SNPs x samples.
`Xpos`	Vector of the physical position of X
`W`	Matrix of cell type composition; samples x cell types.
`Y`	Matrix (or vector) of bulk omics measurements; markers x samples.
`Ypos`	Vector of the physical position of Y
`C`	Matrix (or vector) of covariates; samples x covariates. X, Xpos, W, Y, Ypos, C should be numeric.
`max.pos.diff`	Maximum positional difference to compute cis-QTL. Association analysis is performed between a row of X and a row of Y, only when they are within this limit. Since the limiting is only by position, the function needs to be run separately for each chromosome.
`outdir`	Output directory.
`outfile`	Output file.

Details

A function for analyses of QTL, such as eQTL, mQTL, pQTL. The statistical test is almost identical to `ctassoc(test = "nls.identity",regularize = "TRUE")`. Association analysis is performed between each row of Y and each row of X. Usually, the former will be a methylation/expression marker, and the latter will be a SNP. To cope with the large number of combinations, the testing is limited to pairs whose position is within the difference specified by `max.pos.diff`; i.e., limited to cis-QTL. In detail, this function performs linear ridge regression, whereas `ctassoc(test = "nls.identity",regularize = "TRUE")` actually is nonlinear regression but with f = identity as normalizing transformation. In order to speed up computation, first, the parameters α_{hj} and γ_{jl} are fit by ordinary linear regression assuming $\beta_{hjk} = 0$. Next, β_{hjk} are fit and tested by linear ridge regression (see documentation for ctassoc).

Value

The estimate, statistic, p.value are written to the specified file.

See Also

ctassoc

Examples

```
data(GSE79262small)
X    = GSE79262small$X
Xpos = GSE79262small$Xpos
W    = GSE79262small$W
Y    = GSE79262small$Y
Ypos = GSE79262small$Ypos
C    = GSE79262small$C
X    = X[seq(1, 3601, 100), ] # for brevity
Xpos = Xpos[seq(1, 3601, 100)]
ctcisQTL(X, Xpos, W, Y, Ypos, C = C)
```

GSE42861small *Small Subset of GSE42861 Dataset From GEO*

Description

The dataset includes 336 rheumatoid arthritis cases and 322 controls. A subset of 500 CpG sites were randomly selected from the original EWAS dataset.

Usage

```
data(GSE42861small)
```

Format

An object of class `list` of length 4.

Source

GEO

See Also

ctassoc

Examples

```
data(GSE42861small)
X = GSE42861small$X
W = GSE42861small$W
Y = GSE42861small$Y
Y = Y[seq(1, 20), ] # for brevity
C = GSE42861small$C
result = ctassoc(X, W, Y, C = C)
result$coefficients
```

GSE79262small *Small Subset of GSE79262 Dataset From GEO*

Description

The dataset includes 53 samples. A subset of 737 CpG sites and 3624 SNPs within Chr1:100,000,000-110,000,000 were selected from the original EWAS dataset. DNA methylation was measured in T cells. The estimated proportion of CD4T, CD8T, NK cells are saved in W.

Usage

```
data(GSE79262small)
```

Format

An object of class `list` of length 6.

Source

GEO

See Also

ctcisQTL

Examples

```
data(GSE79262small)
X    = GSE79262small$X
Xpos = GSE79262small$Xpos
W    = GSE79262small$W
Y    = GSE79262small$Y
Ypos = GSE79262small$Ypos
C    = GSE79262small$C
X    = X[seq(1, 3001, 100), ] # for brevity
Xpos = Xpos[seq(1, 3001, 100)]
Y    = Y[seq(1, 501, 100), ]
Ypos = Ypos[seq(1, 501, 100)]
ctcisQTL(X, Xpos, W, Y, Ypos, C = C)
```

GTExsmall *Small Subset of GTEx Dataset*

Description

The dataset includes gene expression measured in whole blood for 389 samples. A subset of 500 genes were randomly selected from the original dataset.

Usage

```
data(GTExsmall)
```

Format

An object of class `list` of length 4.

Source

GTEx

See Also

ctassoc

Examples

```
data(GTExsmall)
X = GTExsmall$X
W = GTExsmall$W
Y = GTExsmall$Y + 1
Y = Y[seq(1, 20), ] # for brevity
C = GTExsmall$C
result = ctassoc(X, W, Y, C = C)
result$coefficients
```

`rrs.fit` *Fitting reduced-rank ridge regression with given rank and shrinkage penalty*

Description

Fitting reduced-rank ridge regression with given rank and shrinkage penalty This is a modification of rrs.fit in rrpack version 0.1-6. In order to handle extremely large q = ncol(Y), generation of a q by q matrix is avoided.

Usage

```
rrs.fit(Y, X, nrank = min(ncol(Y), ncol(X)), lambda = 1, coefSVD = FALSE)
```

Arguments

`Y`	a matrix of response (n by q)
`X`	a matrix of covariate (n by p)
`nrank`	an integer specifying the desired rank
`lambda`	tunging parameter for the ridge penalty
`coefSVD`	logical indicating the need for SVD for the coeffient matrix int the output

Value

S3 `rrr` object, a list consisting of

`coef`	coefficient of rrs
`coef.ls`	coefficient of least square
`fitted`	fitted value of rrs
`fitted.ls`	fitted value of least square
`A`	right singular matrix
`Ad`	sigular value vector
`nrank`	rank of the fitted rrr

References

Mukherjee, A. and Zhu, J. (2011) Reduced rank ridge regression and its kernal extensions.

Mukherjee, A., Chen, K., Wang, N. and Zhu, J. (2015) On the degrees of freedom of reduced-rank estimators in multivariate regression. *Biometrika*, 102, 457–477.

Examples

```
Y <- matrix(rnorm(400), 100, 4)
X <- matrix(rnorm(800), 100, 8)
rfit <- rrs.fit(Y, X)
```

Index