

Package ‘PRANA’

July 21, 2025

Type Package

Title Pseudo-Value Regression Approach for Network Analysis (PRANA)

Version 1.0.6

Description A novel pseudo-value regression approach for the differential co-expression network analysis in expression data, which can incorporate additional clinical variables in the model. This is a direct regression modeling for the differential network analysis, and it is therefore computationally amenable for the most users. The full methodological details can be found in Ahn S et al (2023) <[doi:10.1186/s12859-022-05123-w](https://doi.org/10.1186/s12859-022-05123-w)>.

License GPL-3

Encoding UTF-8

LazyData true

RoxygenNote 7.2.1

Depends R (>= 3.5.0)

Imports dplyr, parallel, robustbase, stats, minet

Suggests knitr, rmarkdown, testthat (>= 3.0.0)

Language en-US

VignetteBuilder knitr

Config/testthat/edition 3

NeedsCompilation no

Author Seungjun Ahn [cre, aut, trl] (ORCID:
<<https://orcid.org/0000-0002-4816-8924>>),
Somnath Datta [ctb, ths]

Maintainer Seungjun Ahn <seungjun.ahn@mountsinai.org>

Repository CRAN

Date/Publication 2024-09-22 19:50:07 UTC

Contents

| | |
|--------------------------------|---|
| adjpval | 2 |
| adjpval_specific_var | 3 |

| | |
|-------------------------------|----|
| coeff | 4 |
| coeff_specific_var | 5 |
| combinedCOPDdat_RGO | 6 |
| EBS | 6 |
| PRANA | 7 |
| run_aracne | 8 |
| sigDCGnames | 9 |
| sigDCGtab | 10 |
| thetahats | 11 |

| | |
|--------------|-----------|
| Index | 12 |
|--------------|-----------|

| | |
|---------|----------------|
| adjpval | <i>adjpval</i> |
|---------|----------------|

Description

A function to retrieve a table with adjusted p-values after running PRANA. The table includes all variables that were included in the pseudo-value regression model.

Usage

```
adjpval(PRANAres)
```

Arguments

PRANAres An object called after running PRANA.

Value

A table that includes adjusted p-values for all variables included in the fitted model.

Examples

```
data(combinedCOPDdat_RGO) # A complete data containing expression and clinical data.

# A gene expression data part of the downloaded data.
rnaseqdat = combinedCOPDdat_RGO[ , 8:ncol(combinedCOPDdat_RGO)]
rnaseqdat = as.data.frame(apply(rnaseqdat, 2, as.numeric))

# A clinical data with additional covariates sorted by current smoking groups:
# The first column is ID, so do not include.
phenodat = combinedCOPDdat_RGO[order(combinedCOPDdat_RGO$currentsmoking), 2:7]

# Indices of non-current smoker (namely Group A)
index_grpA = which(combinedCOPDdat_RGO$currentsmoking == 0)
# Indices of current smoker (namely Group B)
index_grpB = which(combinedCOPDdat_RGO$currentsmoking == 1)

# Use PRANA() function to perform the pseudo-value regression analysis.
```

```
PRANares <- PRANA(RNASeqdat = rnaseqdat, clindat = phenodat,
  groupA = index_grpA, groupB = index_grpB)

# Now, we want to keep the table with adjusted p-values only.
adjpval(PRANares)
```

```
adjpval_specific_var  adjpval_specific_var
```

Description

A function to retrieve a vector of adjusted p-values after running PRANA.

Usage

```
adjpval_specific_var(adjptab, varname)
```

Arguments

| | |
|---------|--|
| adjptab | A table that includes adjusted p-values for a specific variable. |
| varname | Specify the name of the variable of interest. |

Value

A vector of adjusted p-values for a single variable from the model.

Examples

```
data(combinedCOPDdat_RGO) # A complete data containing expression and clinical data.

# A gene expression data part of the downloaded data.
rnaseqdat = combinedCOPDdat_RGO[ , 8:ncol(combinedCOPDdat_RGO)]
rnaseqdat = as.data.frame(apply(rnaseqdat, 2, as.numeric))

# A clinical data with additional covariates sorted by current smoking groups:
# The first column is ID, so do not include.
phenodat = combinedCOPDdat_RGO[order(combinedCOPDdat_RGO$currentsmoking), 2:7]

# Indices of non-current smoker (namely Group A)
index_grpA = which(combinedCOPDdat_RGO$currentsmoking == 0)
# Indices of current smoker (namely Group B)
index_grpB = which(combinedCOPDdat_RGO$currentsmoking == 1)

# Use PRANA() function to perform the pseudo-value regression analysis.
PRANares <- PRANA(RNASeqdat = rnaseqdat, clindat = phenodat,
  groupA = index_grpA, groupB = index_grpB)

# Create an object to keep the table with adjusted p-values using adjpval() function.
adjpvaltab <- adjpval(PRANares)
```

```
# Retrieve a vector of adjusted p-values for a single variable of interest.
adjpval_specific_var(adjptab = adjpvaltab, varname = "currentsmoking")
```

| | |
|-------|--------------|
| coeff | <i>coeff</i> |
|-------|--------------|

Description

A function to retrieve a table with coefficient estimates after running PRANA. The table includes all variables that were included in the pseudo-value regression model.

Usage

```
coeff(PRANAres)
```

Arguments

PRANAres An object called after running PRANA.

Value

A table that includes coefficient estimates for all variables included in the fitted model.

Examples

```
#' data(combinedCOPDdat_RGO) # A complete data containing expression and clinical data.

# A gene expression data part of the downloaded data.
rnaseqdat = combinedCOPDdat_RGO[ , 8:ncol(combinedCOPDdat_RGO)]
rnaseqdat = as.data.frame(apply(rnaseqdat, 2, as.numeric))

# A clinical data with additional covariates sorted by current smoking groups:
# The first column is ID, so do not include.
phenodat = combinedCOPDdat_RGO[order(combinedCOPDdat_RGO$currentsmoking), 2:7]

# Indices of non-current smoker (namely Group A)
index_grpA = which(combinedCOPDdat_RGO$currentsmoking == 0)
# Indices of current smoker (namely Group B)
index_grpB = which(combinedCOPDdat_RGO$currentsmoking == 1)

# Use PRANA() function to perform the pseudo-value regression analysis.
# Then, create an object called `res` to call results later.
PRANAres <- PRANA(RNASEqdat = rnaseqdat, clindat = phenodat,
  groupA = index_grpA, groupB = index_grpB)

# Now, we want to keep the table with coefficient estimates only.
coeff(PRANAres)
```

| | |
|--------------------|---------------------------|
| coeff_specific_var | <i>coeff_specific_var</i> |
|--------------------|---------------------------|

Description

A function to retrieve a vector of coefficient estimates for a specific variable of interest after running PRANA.

Usage

```
coeff_specific_var(coefftab, varname)
```

Arguments

| | |
|----------|--|
| coefftab | A table that includes adjusted p-values for a specific variable. |
| varname | Specify the name of the variable of interest. |

Value

A vector of coefficient estimates for a single variable from the model.

Examples

```
#' data(combinedCOPDdat_RGO) # A complete data containing expression and clinical data.

# A gene expression data part of the downloaded data.
rnaseqdat = combinedCOPDdat_RGO[ , 8:ncol(combinedCOPDdat_RGO)]
rnaseqdat = as.data.frame(apply(rnaseqdat, 2, as.numeric))

# A clinical data with additional covariates sorted by current smoking groups:
# The first column is ID, so do not include.
phenodat = combinedCOPDdat_RGO[order(combinedCOPDdat_RGO$currentsmoking), 2:7]

# Indices of non-current smoker (namely Group A)
index_grpA = which(combinedCOPDdat_RGO$currentsmoking == 0)
# Indices of current smoker (namely Group B)
index_grpB = which(combinedCOPDdat_RGO$currentsmoking == 1)

# Use PRANA() function to perform the pseudo-value regression analysis.
# Then, create an object called `res` to call results later.
PRANAres <- PRANA(RNASEqdat = rnaseqdat, clindat = phenodat,
  groupA = index_grpA, groupB = index_grpB)

# Create an object to keep the table with coefficient estimates using coeff() function.
coefftab <- coeff(PRANAres)

# Lastly, we use coeff_specific_var function to retrieve
# adjusted p-values for a single variable of interest.
coeff_specific_var(coefftab = coefftab, varname = "currentsmoking")
```

combinedCOPDdat_RGO *Chronic Obstructive Pulmonary Disease (COPD) data*

Description

A subset of the COPDGene study data, obtained from the Gene Expression Omnibus database. This contains expression and clinical covariates data. All of the 28 genes contained in this data are identified as COPD-related by Sakornsakolpat et al. Nat Genet, 2019.

Usage

```
data(combinedCOPDdat_RGO)
```

Format

```
combinedCOPDdat_RGO
```

References

Sakornsakolpat P, Prokopenko D, Lamontagne M, et al. Genetic landscape of chronic obstructive pulmonary disease identifies heterogeneous cell-type and phenotype associations. Nat Genet. 2019;51(3):494–505. ([PubMed](#)) Ahn S, Grimes T, Datta S. A pseudo-value regression approach for differential network analysis of co-expression data. BMC Bioinformatics, 2023;24(1):8. ([PubMed](#))

Examples

```
data(combinedCOPDdat_RGO)
combinedCOPDdat_RGO
```

EBS

EBS

Description

A function to calculate the adjusted p-value for each gene (Datta and Datta, 2005)

Usage

```
EBS(pvo, alpha = 0.05, B = 500, h = 1)
```

Arguments

| | |
|-------|--|
| pvo | An object with p-values estimated from the user-provided expression data |
| alpha | a level of significance (default is 0.05) |
| B | size of bootstrapping (default is 500) |
| h | a bandwidth (default is 1) |

Value

A vector of adjusted p-values for each gene.

References

Datta, S and Datta, S (2005). Empirical Bayes screening of many p-values with applications to microarray studies. *Bioinformatics*, 21(9), 1987-1994

 PRANA

PRANA

Description

A pseudo-value regression approach for differential network analysis that adjusts for additional covariates (PRANA).

Usage

```
PRANA(RNASeqdat, clindat, groupA, groupB)
```

Arguments

| | |
|-----------|--|
| RNASeqdat | An RNA-Seq data with subjects in rows and genes in columns. |
| clindat | A data with clinical variables to be included in the regression (e.g., binary group variable indicating current smoking status, continuous age, ...) |
| groupA | Indices of the subjects in the first category (e.g., non-current smoker) of binary group variable. |
| groupB | Indices of the subjects in the second category (e.g., current smoker) of binary group variable. |

Value

A list containing three data frame objects that summarize the results of PRANA. This includes beta coefficients, p-values, and adjusted p-values via the empirical Bayes approach for each predictor variables that are included in the regression model.

References

Ahn S, Grimes T, Datta S. A pseudo-value regression approach for differential network analysis of co-expression data. *BMC Bioinformatics*, 2023;24(1):8

Examples

```

data(combinedCOPDdat_RGO) # A complete data containing expression and clinical data.

# A gene expression data part of the downloaded data.
rnaseqdat = combinedCOPDdat_RGO[, 8:ncol(combinedCOPDdat_RGO)]
rnaseqdat = as.data.frame(apply(rnaseqdat, 2, as.numeric))

# A clinical data with additional covariates sorted by current smoking groups:
# The first column is ID, so do not include.
phenodat = combinedCOPDdat_RGO[order(combinedCOPDdat_RGO$currentsmoking), 2:7]

# Indices of non-current smoker (namely Group A)
index_grpA = which(combinedCOPDdat_RGO$currentsmoking == 0)
# Indices of current smoker (namely Group B)
index_grpB = which(combinedCOPDdat_RGO$currentsmoking == 1)

PRANAres <- PRANA(RNASeqdat = rnaseqdat, clindat = phenodat,
groupA = index_grpA, groupB = index_grpB)

```

run_aracne

run_aracne

Description

A function to conduct co-expression analysis using ARACNE (Margolin et al. 2006). Uses the implementation from the minet package (Meyer et al. 2008). This function is from dnapath R package, which is archived in May 2024.

Usage

```

run_aracne(
  x,
  estimator = "spearman",
  disc = "none",
  nbins = NULL,
  eps = 0,
  ...
)

```

Arguments

| | |
|-----------|---|
| x | A n by p matrix of gene expression data (n samples and p genes) |
| estimator | Argument is passed into minet::build.mim. |
| disc | Argument is passed into minet::build.mim. |
| nbins | Argument is passed into minet::build.mim. |
| eps | Argument is passed into minet::aracne. |
| ... | Additional arguments are ignored. |

Value

A p by p matrix of association scores.

References

Margolin AA, Nemenman I, Basso K, Wiggins C, Stolovitzky G, Dalla Favera R, Califano A (2006). “ARACNE: An Algorithm for the Reconstruction of Gene Regulatory Networks in a Mammalian Cellular Context.” In BMC Bioinformatics, volume 7(1), S7. BioMed Central.

Meyer PE, Lafitte F, Bontempi G (2008). “minet: A R/Bioconductor Package for Inferring Large Transcriptional Networks using Mutual Information.” BMC Bioinformatics, 9(1), 461.

sigDCGnames

sigDCGnames

Description

A function to retrieve the name of genes that are significantly differentially connected (DC). between two biological/clinical states (aka the main binary indicator) with the presence of additional covariate information.

Usage

```
sigDCGnames(adjptab, groupvar, alpha)
```

Arguments

| | |
|----------|---|
| adjptab | A table with adjusted p-values for all variables that were included in the pseudo-value regression model. |
| groupvar | Specify the name of binary indicator variable. |
| alpha | A level of significance (e.g. 0.05). |

Value

Names of significantly DC genes (e.g. gene IDs) from PRANA. If you need both adjusted p-values and names, please use sigDCGtab() instead.

Examples

```
#' data(combinedCOPDdat_RGO) # A complete data containing expression and clinical data.

# A gene expression data part of the downloaded data.
rnaseqdat = combinedCOPDdat_RGO[ , 8:ncol(combinedCOPDdat_RGO)]
rnaseqdat = as.data.frame(apply(rnaseqdat, 2, as.numeric))

# A clinical data with additional covariates sorted by current smoking groups:
# The first column is ID, so do not include.
phenodat = combinedCOPDdat_RGO[order(combinedCOPDdat_RGO$currentsmoking), 2:7]
```

```

# Indices of non-current smoker (namely Group A)
index_grpA = which(combinedCOPDdat_RGO$currentsmoking == 0)
# Indices of current smoker (namely Group B)
index_grpB = which(combinedCOPDdat_RGO$currentsmoking == 1)

# Use PRANA() function to perform the pseudo-value regression analysis.
# Then, create an object called PRANA_Results to call results.
PRANAres <- PRANA(RNASeqdat = rnaseqdat, clindat = phenodat,
groupA = index_grpA, groupB = index_grpB)

# Next, we want to call the table with adjusted p-values only.
adjptab <- adjpval(PRANAres)

# Please specify the name of binary group indicator in sigDCGnames(groupvar = ).
sigDCGnames <- sigDCGnames(adjptab = adjptab, groupvar = "currentsmoking", alpha = 0.05)
sigDCGnames

```

sigDCGtab

sigDCGtab

Description

A function to retrieve the data frame that are significantly differentially connected (DC) between two biological/clinical states (aka the main binary indicator) with the presence of additional covariate information.

Usage

```
sigDCGtab(adjptab, groupvar, alpha)
```

Arguments

| | |
|----------|--|
| adjptab | A table with adjusted p-values and names for the variable that the user specifies in the groupvar. |
| groupvar | Specify the name of binary indicator variable. |
| alpha | A level of significance (e.g. 0.05). |

Value

Adjusted p-values and names of significantly DC genes (e.g. gene IDs) from PRANA.

Examples

```

#' data(combinedCOPDdat_RGO) # A complete data containing expression and clinical data.

# A gene expression data part of the downloaded data.
rnaseqdat = combinedCOPDdat_RGO[ , 8:ncol(combinedCOPDdat_RGO)]
rnaseqdat = as.data.frame(apply(rnaseqdat, 2, as.numeric))

```

```
# A clinical data with additional covariates sorted by current smoking groups:
# The first column is ID, so do not include.
phenodat = combinedCOPDdat_RGO[order(combinedCOPDdat_RGO$currentsmoking), 2:7]

# Indices of non-current smoker (namely Group A)
index_grpA = which(combinedCOPDdat_RGO$currentsmoking == 0)
# Indices of current smoker (namely Group B)
index_grpB = which(combinedCOPDdat_RGO$currentsmoking == 1)

# Use PRANA() function to perform the pseudo-value regression analysis.
# Then, create an object called PRANA_Results to call results.
PRANAres <- PRANA(RNASeqdat = rnaseqdat, clindat = phenodat,
  groupA = index_grpA, groupB = index_grpB)

# Next, we want to call the table with adjusted p-values.
adjptab <- adjpval(PRANAres)

# Please specify the name of variable in sigDCGtab(groupvar = ).
sigDCGtab <- sigDCGtab(adjptab = adjptab, groupvar = "currentsmoking", alpha = 0.05)
sigDCGtab
```

thetahats

thetahats

Description

A function to compute the total connectivity of each gene from the association matrix.

Usage

```
thetahats(asso.matinput)
```

Arguments

`asso.matinput` An association matrix that is estimated from the user-provided expression data is used as an input to compute the total connectivity of each gene.

Value

A vector containing total connectivity of each gene (i.e. continuous version of centrality measure of a network)

Index

- * **2023.**
 - combinedCOPDdat_RGO, 6
- * **Ahn**
 - combinedCOPDdat_RGO, 6
- * **Application**
 - combinedCOPDdat_RGO, 6
- * **BMC**
 - combinedCOPDdat_RGO, 6
- * **Bioinformatics.**
 - combinedCOPDdat_RGO, 6
- * **al.**
 - combinedCOPDdat_RGO, 6
- * **data**
 - combinedCOPDdat_RGO, 6
- * **et**
 - combinedCOPDdat_RGO, 6
- * **in**
 - combinedCOPDdat_RGO, 6
- * **used**
 - combinedCOPDdat_RGO, 6

adjpval, 2

adjpval_specific_var, 3

coeff, 4

coeff_specific_var, 5

combinedCOPDdat_RGO, 6

EBS, 6

PRANA, 7

run_aracne, 8

sigDCGnames, 9

sigDCGtab, 10

thetahats, 11