

# Package ‘jtGWAS’

August 14, 2017

**Type** Package

**Title** Efficient Jonckheere-Terpstra Test Statistics

**Version** 1.5.1

**Date** 2017-08-14

**Author** Jiaxing Lin, Alexander Sibley, Ivo Shterev, and Kouros Owzar

**Maintainer** Jiaxing Lin <jiaxing.lin@duke.edu>

**Description** The core of this 'Rcpp' based package is a function to compute standardized Jonckheere-Terpstra test statistics for large numbers of dependent and independent variables, e.g., genome-wide analysis. It implements 'OpenMP', allowing the option of computing on multiple threads. Supporting functions are also provided to calculate p-values and summarize results.

**License** GPL (>= 2)

**URL** <https://bitbucket.org/impactp01/jtgwas>

**Imports** Rcpp (>= 0.12.3)

**LinkingTo** Rcpp

**Suggests** knitr

**VignetteBuilder** knitr

**BuildVignettes** yes

**NeedsCompilation** yes

**Repository** CRAN

**Date/Publication** 2017-08-14 18:25:30 UTC

## R topics documented:

jtGWAS-package	2
jtGWAS	3
pvalues	4
summary.jtGWAS	5
<b>Index</b>	<b>7</b>

## Description

The core of this Rcpp-based package is a function to compute standardized Jonckheere-Terpstra test statistics for large numbers of dependent and independent variables, e.g. genome-wide analysis. It implements OpenMP, allowing the option of computing on multiple threads. Supporting functions are also provided to calculate p-values and summarize results.

## Details

Package: jtGWAS  
Type: Package  
Version: 1.5.1  
Date: 2017-08-14  
License: GPL-3

Please see the example function calls below, or refer to the individual function documentation or the included vignette for more information. The package vignette serves as a tutorial for using this package. The technical details are provided in the reference cited below. Specifically, the calculation of the standardized test statistic employs the null variance equation as defined by Hollander and Wolfe (1999, eq. 6.19) to account for ties in the data.

## Author(s)

Jiaying Lin, Alexander Sibley, Ivo Shterev, and Kouros Owzar  
Maintainer: Jiaying Lin <jiaying.lin@duke.edu>

## References

Hollander, M. and Wolfe, D. A. (1999) *Nonparametric Statistical Methods*. New York: Wiley, 2nd edition.

## See Also

[Rcpp](#)

## Examples

```
# Generate dummy data
num_patient <- 100
num_marker <- 10
num_SNP <- 500
set.seed(12345)
X_pat_mak <- matrix(rnorm(num_patient*num_marker), num_patient, num_marker)
```

```
G_pat_SNP <- matrix(rbinom(num_patient*num_SNP, 2, 0.5), num_patient, num_SNP)
colnames(X_pat_mak) <- colnames(X_pat_mak, do.NULL=FALSE, prefix="Mrk:")
colnames(G_pat_SNP) <- colnames(G_pat_SNP, do.NULL=FALSE, prefix="SNP:")

res <- jtGWAS(X_pat_mak, G_pat_SNP, outTopN=15)
res
```

---

jtGWAS

---

*Compute the Jonckheere-Terpstra Test Statistics*


---

## Description

A method to compute the Jonckheere-Terpstra test statistics for large numbers of dependent and independent variables, with optional multi-threaded execution. The calculation of the standardized test statistic employs the null variance equation as defined by Hollander and Wolfe (1999, eq. 6.19) to account for ties in the data.

## Usage

```
jtGWAS(X, G, outTopN=15L, numThreads=1L, standardized=TRUE)
```

## Arguments

X	A matrix of dependent variables, e.g. marker levels. Row names are sample IDs, and column names are marker IDs. Required.
G	A matrix of independent variables, e.g. SNP counts. Row names are sample IDs, and column names are SNP IDs. Required.
outTopN	An integer to indicate the number of top hits to be reported for each marker, based on the standardized Jonckheere-Terpstra test statistics. Optional, and the default value is 15L. If set to NA, all results are returned.
numThreads	A integer to indicate the number of threads used in the computation. Optional, and the default value is 1L(sequential computation).
standardized	A boolean to specify whether to return standardized statistics (TRUE) or non-standardized statistics (FALSE). Optional, the default value is TRUE.

## Value

A list with two objects

J	A matrix of the standardized/non-standardized Jonckheere-Terpstra test statistics, depending on the value of the standardized argument.
gSnipID	If outTopN was specified, this object is a matrix of the column IDs of G corresponding to the top standardized Jonckheere-Terpstra test statistics for each marker. Otherwise this is a vector of column IDs of G.

## Note

Rows (samples) are assumed to be in the same order in X and G.

## References

Hollander, M. and Wolfe, D. A. (1999) *Nonparametric Statistical Methods*. New York: Wiley, 2nd edition.

## Examples

```
# Generate dummy data
num_patient <- 100
num_marker  <- 10
num_SNP     <- 500
set.seed(12345)
X_pat_mak <- matrix(rnorm(num_patient*num_marker), num_patient, num_marker)
G_pat_SNP <- matrix(rbinom(num_patient*num_SNP, 2, 0.5), num_patient, num_SNP)
colnames(X_pat_mak) <- colnames(X_pat_mak, do.NULL=FALSE, prefix="Mrk:")
colnames(G_pat_SNP) <- colnames(G_pat_SNP, do.NULL=FALSE, prefix="SNP:")

res <- jtGWAS(X_pat_mak, G_pat_SNP, outTopN=5)
res
res <- jtGWAS(X_pat_mak, G_pat_SNP, outTopN=NA)
head(res)
```

---

pvalues

*Compute P-values Based on Jonckheere-Terpstra Test Statistics*

---

## Description

Method to compute the p-values for results from jtGWAS.

## Usage

```
pvalues(jtGWAS.object)
```

## Arguments

`jtGWAS.object` A jtGWAS object that is the return of method jtGWAS.

## Value

A matrix of p-values with the same dimensions as the standardized statistics from jtGWAS. If jtGWAS was run with `standardized=FALSE`, returns NULL.

## See Also

[jtGWAS](#)

**Examples**

```
# Generate dummy data
num_patient <- 100
num_marker <- 10
num_SNP <- 500
set.seed(12345)
X_pat_mak <- matrix(rnorm(num_patient*num_marker), num_patient, num_marker)
G_pat_SNP <- matrix(rbinom(num_patient*num_SNP, 2, 0.5), num_patient, num_SNP)
colnames(X_pat_mak) <- colnames(X_pat_mak, do.NULL=FALSE, prefix="Mrk:")
colnames(G_pat_SNP) <- colnames(G_pat_SNP, do.NULL=FALSE, prefix="SNP:")

res <- jtGWAS(X_pat_mak, G_pat_SNP, outTopN=5)
pvalues(res)
res <- jtGWAS(X_pat_mak, G_pat_SNP, outTopN=NA)
pvalues(res)
```

summary.jtGWAS

*Summarize Jonckheere-Terpstra Test Statistics and P-Values***Description**

Summary method for jtGWAS results.

**Usage**

```
## S3 method for class 'jtGWAS'
summary(object, marker2Print=1:10, SNP2Print=1:10,
        printP=TRUE, outTopN=NA, subObj=FALSE, ...)
```

**Arguments**

object	A jtGWAS object that is the return of method jtGWAS. Required.
marker2Print	Either a numeric or character vector that indicates the desired dependent variables to print. The default is 1:10. Set to NA to print full results. Optional.
SNP2Print	Either a numeric or character vector that indicates the desired independent variables to print. If outTopN=NA in function jtGWAS, i.e., the results are not sorted, both numeric and character vectors can be used to set the print range. The default range is 1:10. Set to NA to print full results. If outTopN!=NA in function jtGWAS, the range of SNP2Print refers to the range of top normalized statistics computed in jtGWAS and only numeric vectors are processed. Optional.
printP	A boolean indicating whether to print the p-value (TRUE) or the standardized statistics (FALSE). The default value is TRUE.
outTopN	An integer specifying the number of top hits to print in the summary, if the statistics were not sorted during the statistics evaluation (i.e., if outTopN=NA in function jtGWAS). Optional, and the default value is NA.
subObj	A boolean indicating whether to return a jtGWAS object subset per the requested summary (TRUE). Optional. The default value is FALSE (nothing is returned).
...	Additional arguments affecting the summary produced.

**Value**

If subObj=TRUE, this method returns a jtGWAS object matching the subset of statistics from object that are being printed. For example, if object is not sorted by top hits, summary(object, outTopN=10, subObj=TRUE) will print the summary and return a subset of object that contains only the top 10 SNPs for each marker. If subObj=FALSE, nothing is returned.

**Note**

This function prints a matrix or paired columns of SNP IDs and statistics/p-values to the log.

**See Also**

[jtGWAS](#), [pvalues](#)

**Examples**

```
# Generate dummy data
num_patient <- 100
num_marker  <- 10
num_SNP     <- 500
set.seed(12345);
X_pat_mak <- matrix(rnorm(num_patient*num_marker), num_patient, num_marker)
G_pat_SNP <- matrix(rbinom(num_patient*num_SNP, 2, 0.5), num_patient, num_SNP)
colnames(X_pat_mak) <- colnames(X_pat_mak, do.NULL=FALSE, prefix="Mrk:")
colnames(G_pat_SNP) <- colnames(G_pat_SNP, do.NULL=FALSE, prefix="SNP:")

res <- jtGWAS(X_pat_mak, G_pat_SNP, outTopN=5)
summary(res, marker2Print=c("Mrk:1","Mrk:2"), SNP2Print=1:5, printP=FALSE)
summary(res, marker2Print=NA, SNP2Print=1:5, printP=FALSE)

res <- jtGWAS(X_pat_mak, G_pat_SNP, outTopN=NA)
summary(res, marker2Print=1:10, SNP2Print=1:10, printP=TRUE)
summary(res, marker2Print=c("Mrk:1","Mrk:2"), SNP2Print=c("SNP:1","SNP:2"), printP=TRUE)

res <- jtGWAS(X_pat_mak, G_pat_SNP, outTopN=NA, standardized=FALSE)
summary(res, outTopN=10)
```

# Index

\*Topic **package**

jtGWAS-package, [2](#)

jtGWAS, [3](#), [4](#), [6](#)

jtGWAS-package, [2](#)

pvalues, [4](#), [6](#)

Rcpp, [2](#)

summary.jtGWAS, [5](#)