Package ‘aSPC’
October 12, 2022

Type Package
Title An Adaptive Sum of Powered Correlation Test (aSPC) for Global Association Between Two Random Vectors
Version 0.1.2
Date 2016-10-26
Author Zhiyuan (Jason) Xu and Wei Pan
Maintainer Zhiyuan (Jason) Xu <xuxx0284@umn.edu>
Description The aSPC test is designed to test global association between two groups of variables potentially with moderate to high dimension (e.g. in hundreds). The aSPC is particularly useful when the association signals between two groups of variables are sparse.
License GPL-3
RoxygenNote 5.0.1
Imports mvtnorm, energy, stats
NeedsCompilation no
Repository CRAN
Date/Publication 2017-04-27 16:12:05 UTC

R topics documented:

aSPC .................................................. 1

Description
An adaptive sum of powered correlation test (aSPC) for association between two random vectors
Usage

aSPC(df1, df2, pow = c(1:6, Inf), B = 100, Z.transform = TRUE, method = "pearson")

Arguments

df1, first sample matrix
df2, second sample matrix
pow, power integer candidates, default c(1:8, Inf)
B, number of permutations to calculate a P-value. Default is 100.
Z.transform, whether to do Fisher's z-transformation on Pearson correlation, default is TRUE.
metho(d, one of "pearson", "spearman", or "dcor". Default is "pearson".

Value

the P-values of SPC and aSPC tests

References

Xu Z., Pan W. 2017. Adaptive testing for association between two random vectors in moderate to high dimensions. Submitted to Genetic Epidemiology

Examples

library(mvtnorm)
sigma = diag(0.9, 10) + 0.1
n = 50  # sample size
Z = rmvnorm(n=n, mean=rep(0,10), sigma=sigma)
X = rmvnorm(n=n, mean=rep(0,15), sigma=diag(1, 15))
Y = rmvnorm(n=n, mean=rep(0,15), sigma=diag(1, 15))
X = as.data.frame(cbind(Z[,1:5], X))
Y = as.data.frame(cbind(Z[,6:10], Y))
set.seed(123)  # to ensure we can replicate the permutation P-value
p = 2; q = 2; n=50
X = rmvnorm(n=n, mean=rep(0,p), sigma=diag(1, p))
Y = rmvnorm(n=n, mean=rep(0,q), sigma=diag(1, q))
a = proc.time()
aSPC(X, Y, pow = c(1:8, Inf), B = 1000, method = "pearson")
proc.time() - a

#' a = proc.time()
aSPC(X, Y, pow = c(1:8, Inf), B = 1000, method = "spearman")
proc.time() - a

a = proc.time()
aSPC(X, Y, pow = c(1:8, Inf), B = 500, method = "dcor")
proc.time() - a
Index

aSPC, 1