

Package ‘permtest’

February 15, 2012

Title Permutation test to compare variability within and distance between two groups

Version 1.2

Author Douglas Hayden

Description The permtest uses permutations in order to compare the variability within and distance between two groups within a set of microarray data.

Maintainer <plazar@partners.org>

License GPL

Repository CRAN

Date/Publication 2011-11-10 12:59:06

R topics documented:

permtest 1

Index 6

permtest	<i>Permutation test for testing two groups for a variability or location difference.</i>
----------	--

Description

Permtest compares two groups of microarrays, or more generally, two groups of high dimensional signal vectors derived from microarrays, for a difference in location or variability. It reduces the microarray data to a matrix of pairwise distances, computes summary statistics based on the mean within and between group distances, and performs four hypothesis tests by permuting the group labels. Two distances are supported, squared Euclidean and minus log Pearson correlation. Three experimental designs are supported, completely random, paired, and blocked.

Usage

```
permtest(datamatrix, designmatrix, distance = "euclid", nperms = 1000, maxperms = 1e+06, designtype = "r
```

Arguments

datamatrix	An all numeric matrix (double or integer type) containing the microarray or signal vector data to be analysed. The data should be arranged into one column per data vector. The number of rows should be equal across all of data columns and there should be no NA or Nan missing items. Each column needs to have a column name as given by colnames(mymatrix). Each of these column names must have a matching item in column 1 of the designmatrix described below.
designmatrix	A matrix with information describing the columns of the datamatrix. It has a number of rows equal to the number columns of the datamatrix. Each row has three items: a string that corresponds to the column name in datamatrix, a string for group "1" or "2", and a string referring to another column name string (for paired mode) or a block name (for blocked mode)
distance	Distance is either Euclidean: "euclid" or logarithmic:"logr". Logarithmic distance uses the negative log Pearson correlation to make a distance matrix while Euclidean uses traditional mean squared Euclidean distance.
nperms	The number of permutations to perform. This is ignored if in paired mode with less than 24 columns, in which case, permtest will enumerate all the possible permutations instead.
maxperms	Maximum permutations to perform. This is used primarily during paired testing that attempts to enumerate all possible permutations. Any permutation count beyond this threshold will be set to the threshold.
designtype	There are three experimental designs for the datamatrix data: completely random: "random"; a paired design where each data column in group 2 has a corresponding partner in group 1:"paired"; and a blocked design where certain columns from each group are part of blocks: "blocked".
savedist	A flag specifying whether the permutation distributions obtained from the permutation tests should be saved and returned. The default is to not save them to preserve memory.
savedmat	A flag specifying whether the distance matrix should be saved and returned. The default is to not save the distance matrix.

Details

Permtest compares two groups of microarrays, or more generally, two groups of high dimensional signal vectors derived from microarrays, for a difference in location or variability. It reduces the microarray data to a matrix of pairwise distances, computes summary statistics based on the mean within and between group distances, and performs four hypothesis tests by permuting the group labels. Two distances are supported, mean squared Euclidean and minus log Pearson correlation. If there are negative correlations between microarrays then Euclidean distance must be used. Three experimental designs are supported, completely random, paired, and blocked.

Input consists of two matrices: a matrix of microarray data and a design matrix. The microarray data matrix must be numeric with no NaNs and must have column names corresponding to each

microarray or signal vector. The design matrix must be strings. The first column of the design matrix lists the column names of the data matrix. The second column gives the group assignment of each microarray as either '1' or '2'. The third column gives either the column name of the paired microarray if the design is paired, or a string designating the block if the design is blocked.

Three summary statistics are computed:

1. d11 is the mean distance within group 1.
2. d22 is the mean distance within group 2.
3. d12 is the mean distance between groups 1 and 2.

Four test statistics are computed from the summary statistics:

1. A one sided variability test, $d11-d22$, of the null hypothesis that the groups 1 and 2 have equal variability. If the observed value, STAT, of $d11-d22$ is positive then the COUNT is the number of permutations with a value greater than or equal to the observed value and the PVALUE is the COUNT divided by the number of permutations. If the p-value is small then reject the null hypothesis of equality and accept the alternative hypothesis that group 1 has more variability than group 2. If the observed value, STAT, of $d11-d22$ is negative (or zero) then the COUNT is the number of permutations with a value less than or equal to the observed value and the PVALUE is the COUNT divided by the number of permutations. If the p-value is small then reject the null hypothesis of equality and accept the alternative hypothesis that group 1 has less variability than group 2.
2. A two sided variability test, $abs(d11-d22)$, of the null hypothesis that the groups 1 and 2 have equal variability. The COUNT is the number of permutations with a value greater than or equal to the observed value, STAT, and the PVALUE is the COUNT divided by the number of permutations. If the p-value is small then reject the null hypothesis of equality and accept the alternative hypothesis that group 1 and group 2 have unequal variability.
3. A one sided location test, $d12-((d22+d11)/2)$, of the null hypothesis that group 1 and 2 have the same the location. This statistic estimates the distance between the two groups that is not due to within group variability. The COUNT is the number of permutations with a value greater than or equal to the observed value, STAT, and the PVALUE is the COUNT divided by the number of permutations. If the p-value is small then reject the null hypothesis and accept the alternative hypothesis that group 1 and group 2 differ in location.
4. A one sided omnibus equivalence test, $d12-d11$, of the null hypothesis that group 2 is equivalent in location and variability to group 1 with group 1 considered a known "Gold Standard" to which group 2 is being compared. The COUNT is the number of permutations with a value greater than or equal to the observed value, STAT, and the PVALUE is the COUNT divided by the number of permutations. If the p-value is small then reject the null hypothesis and accept the alternative hypothesis that either the groups differ in location or group 2 has greater variability.

Value

Permtest returns a list object with a series of named elements

out\$table	A table displaying the summary statistics and the hypothesis tests and associated pvalues
out\$distributions	Observed permutation distributions
out\$distributions\$v1	Observed permutation distribution from the one sided variability test

```

out$distributions$v2
    Observed permutation distribution from the two sided variability test
out$distributions$loc
    Observed permutation distribution from the location test
out$distributions$equiv
    Observed permutation distribution from the equivalence test
out$distancematrix
    The distance matrix

```

Author(s)

Douglas Hayden and Peter Lazar

Examples

```

#Make a data set with no pairing, slight group mean differences and slight differences in variation
datamatrix<-matrix(data=c(
#group 1
rnorm(4000,.1,1.25),
#group 2
rnorm(4000,0,1) ),ncol=10,nrow=800);
designmatrix<-matrix(data=as.matrix(list("ad1","1","0","ad2","1","0","ad3","1","0","ad4","1","0","ad5","1","0"
colnames(designmatrix)<-designmatrix[1,];
colnames(datamatrix)<-colnames(designmatrix);
designmatrix<-t(designmatrix);
#run as a randomized model with default 1000 permutations.
permtest(datamatrix,designmatrix)

#Add pairing information, we can run as paired
#each pair item follows it's group's mean and std dev
#each pair has a common mean and deviation deflection
## Not run:
datamatrix<-matrix(data=c(

#group 1 mean is .1, std dev 1.25
rnorm(800,.11,1.25),#pair 1
rnorm(800,.2 ,1.55),#pair 2
rnorm(800,.1 ,1.25),#pair 3
rnorm(800,.1 ,1.15),#pair 4
rnorm(800,.13,1.25),#pair 5
#group 2 mean is 0, std dev 1
rnorm(800,.01, 1),#pair 1
rnorm(800,.1 , 1.3),#pair 2
rnorm(800, 0 , 1),#pair 3
rnorm(800, 0 ,.9),#pair 4
rnorm(800,.03, 1),#pair 5

),ncol=10,nrow=800);
designmatrix<-matrix(data=as.matrix(list("ad1","1","nd1","ad2","1","nd2","ad3","1","nd3","ad4","1","nd4","ad5"
colnames(designmatrix)<-designmatrix[1,];
colnames(datamatrix)<-colnames(designmatrix);

```

```
designmatrix<-t(designmatrix);  
permtest(datamatrix,designmatrix,designtype="paired")  
  
## End(Not run)
```

Index

*Topic **misc**
permtest, 1

*Topic **utilities**
permtest, 1

permtest, 1