Package 'MVR'

January 28, 2016

Type Package

Title Mean-Variance Regularization

Version 1.30.3

Date 2016-01-28

Author Jean-Eudes Dazard [aut, cre], Hua Xu [ctb], Alberto Santana [ctb]

Maintainer Jean-Eudes Dazard <jxd101@case.edu>

Description

This is a non-parametric method for joint adaptive mean-variance regularization and variance stabilization of high-dimensional data. It is suited for handling difficult problems posed by high-dimensional multivariate datasets (p >> n paradigm). Among those are that the variance is often a function of the mean, variable-specific estimators of variances are not reliable, and tests statistics have low powers due to a lack of degrees of freedom. Key features include: (i) Normalization and/or variance stabilization of the data,(ii) Computation of mean-variance-regularized t-statistics (F-statistics to follow),(iii) Generation of diverse diagnostic plots,(iv) Computationally efficient implementation using C/C++ interfacing and an option for parallel computing to enjoy a faster and easier experience in the R environment.

Depends R (>= 3.0.2), parallel, statmod

Imports graphics, grDevices, methods, stats

URL https://github.com/jedazard/MVR

Repository CRAN

License GPL (>= 3) | file LICENSE

LazyLoad yes

LazyData yes

Archs i386, x64

R topics documented:

`MVR-package` *Mean-Variance Regularization Package*

Description

MVR is a non-parametric method for joint adaptive mean-variance regularization and variance stabilization of high-dimensional data.

It is suited for handling difficult problems posed by high-dimensional multivariate datasets ($p \gg n$ paradigm), such as in omics-type data, among which are that the variance is often a function of the mean, variable-specific estimators of variances are not reliable, and tests statistics have low powers due to a lack of degrees of freedom.

Key features include:

1. Normalization and/or variance stabilization of the data
2. Computation of mean-variance-regularized *t*-statistics (*F*-statistics to come)
3. Generation of diverse diagnostic plots
4. Computationally efficient implementation using C/C++ interfacing and an option for parallel computing to enjoy a fast and easy experience in the R environment

Details

The following describes all the end-user functions, and internal R subroutines needed for running a complete MVR procedure. Other internal subroutines are not to be called by the end-user at any time. For computational efficiency, end-user regularization functions offer the option to configure a cluster. This is indicated by an asterisk (* = optionally involving cluster usage). The R functions are categorized as follows:

1. END-USER REGULARIZATION & VARIANCE STABILIZATION FUNCTION
   `mvr` (*) **Function for Mean-Variance Regularization and Variance Stabilization.**
   End-user function for Mean-Variance Regularization (MVR) and Variance Stabilization by similarity statistic under sample group homoscedasticity or heteroscedasticity assumption. The function takes advantage of the R package **parallel**, which allows users to create a cluster of workstations on a local and/or remote machine(s), enabling parallel execution of this function and scaling up with the number of CPU cores available.

2. END-USER REGULARIZED TESTS-STATISTICS FUNCTIONS
   `mvrt.test` (*) **Function for Computing Mean-Variance Regularized T-test Statistic and Its Significance.**
   End-user function for computing MVR t-test statistic and its significance (p-value) under sample group homoscedasticity or heteroscedasticity assumption. The function takes advantage of the R package **parallel**, which allows users to create a cluster of workstations on a local and/or remote machine(s), enabling parallel execution of this function and scaling up with the number of CPU cores available.

3. END-USER DIAGNOSTIC PLOTS FOR QUALITY CONTROL
   `cluster.diagnostic` **Function for Plotting Summary Cluster Diagnostic Plots.**
   Plot similarity statistic profiles and the optimal joint clustering configuration for the means and the variances by group. Plot quantile profiles of means and standard deviations by group and for each clustering configuration, to check that the distributions of first and second moments of the MVR-transformed data approch their respective null distributions under the optimal configuration found, assuming independence and normality of all the variables.

   `target.diagnostic` **Function for Plotting Summary Target Moments Diagnostic Plots.**
   Plot comparative distribution densities of means and standard deviations of the data before and after Mean-Variance Regularization to check for location shifts between observed first and second moments and their expected target values under a target centered homoscedastic model. Plot comparative QQ scatterplots to look at departures between observed distributions of first and second moments of the MVR-transformed data and their theoretical distributions assuming independence and normality of all the variables.

   `stabilization.diagnostic` **Function for Plotting Summary Variance Stabilization Diagnostic Plots.**
   Plot comparative variance-mean plots to check the variance stabilization across variables before and after Mean-Variance Regularization.

   `normalization.diagnostic` **Function for Plotting Summary Normalization Diagnostic Plots.**
   Plot comparative Box-Whisker and Heatmap plots of variables across samples check the effectiveness of normalization before and after Mean-Variance Regularization.

4. OTHER END-USER FUNCTIONS
   `MVR.news` **Display the MVR Package News**
   Function to display the log file `NEWS` of updates of the **MVR** package.

5. END-USER DATASETS
   A `Real` dataset coming from a quantitative proteomics experiment, consisting of $n = 6$ samples split into a control ("*M*") and a treated group ("*S*") with $p = 9052$ unique peptides or predictor variables. This is a balanced design with two sample groups ($G = 2$), under unequal sample group variance.

   A `Synthetic` dataset with $n = 10$ observations (samples) and $p = 100$ variables, where $nvar = 20$ of them are significantly different between the two sample groups. This is a balanced design with two sample groups ($G = 2$), under unequal sample group variance.

Known Bugs/Problems : None at this time.

Author(s)

- "Jean-Eudes Dazard, Ph.D." <jxd101@case.edu>
- "Hua Xu, Ph.D." <huaxu77@gmail.com>
- "Alberto Santana, MBA." <ahs4@case.edu>

Maintainer: "Jean-Eudes Dazard, Ph.D." <jxd101@case.edu>

Acknowledgments: This project was partially funded by the National Institutes of Health (P30-CA043703 to J-E.DAZARD).

References

- Dazard J-E., Hua Xu and J. S. Rao (2011). "*R package MVR for Joint Adaptive Mean-Variance Regularization and Variance Stabilization.*" In JSM Proceedings, Section for Statistical Programmers and Analysts. Miami Beach, FL, USA: American Statistical Association IMS - JSM, 3849-3863.
- Dazard J-E. and J. S. Rao (2012). "*Joint Adaptive Mean-Variance Regularization and Variance Stabilization of High Dimensional Data.*" Comput. Statist. Data Anal. 56(7):2317-2333.

See Also

- `makeCluster` (R package **parallel**)
- `justvsn` (R package **vsn**) Variance stabilization and calibration for microarray data *Huber, 2002*
- `eBayes` (R package **limma**) Bayesian Regularized t-test statistic *Smyth, 2004*
- `samr` (R package **samr**) SAM Regularized t-test statistic *Tusher et al., 2001*, *Storey, 2003*
- `matest` (R package **maanova**) James-Stein shrinkage estimator-based Regularized t-test statistic *Cui et al., 2005*
- `ebam` (R package **siggenes**) Empirical Bayes Regularized z-test statistic *Efron, 2001*
- `bayesT` Hierarchical Bayesian Regularized t-test statistic *Baldi et al., 2001*

`cluster.diagnostic` *Function for Plotting Summary Cluster Diagnostic Plots*

Description

Plot similarity statistic profiles and the optimal joint clustering configuration for the means and the variances by group.

Plot quantile profiles of means and standard deviations by group and for each clustering configuration, to check that the distributions of first and second moments of the MVR-transformed data approch their respective null distributions under the optimal configuration found, assuming independence and normality of all the variables.

Usage

```
cluster.diagnostic(obj,
                   span = 0.75,
                   degree = 2,
                   family = "gaussian",
                   title = "Cluster Diagnostic Plots",
                   device = NULL,
                   file = "Cluster Diagnostic Plots",
                   path = getwd(),
                   horizontal = FALSE,
                   width = 8.5,
                   height = 11, ...)
```

Arguments

obj	Object of class "mvr" returned by mvr.
title	Title of the plot. Defaults to "Cluster Diagnostic Plots".
span	Span parameter of the loess() function (R package **stats**), which controls the degree of smoothing. Defaults to 0.75.
degree	Degree parameter of the loess() function (R package **stats**), which controls the degree of the polynomials to be used. Defaults to 2. (Normally 1 or 2. Degree 0 is also allowed, but see the "Note" in loess stats package.)
family	Family distribution in "gaussian", "symmetric" of the loess() function (R package **stats**), used for local fitting . If "gaussian" fitting is by least-squares, and if "symmetric" a re-descending M estimator is used with Tukey's biweight function.
device	Graphic display device in {NULL, "PS", "PDF"}. Defaults to NULL (standard output screen). Currently implemented graphic display devices are "PS" (Postscript) or "PDF" (Portable Document Format).
file	File name for output graphic. Defaults to "Cluster Diagnostic Plots".
path	Absolute path (without final (back)slash separator). Defaults to working directory path.
horizontal	Logical scalar. Orientation of the printed image. Defaults to FALSE, that is potrait orientation.
width	Numeric scalar. Width of the graphics region in inches. Defaults to 8.5.
height	Numeric scalar. Height of the graphics region in inches. Defaults to 11.
...	Generic arguments passed to other plotting functions.

Details

In a plot of a similarity statistic profile, one checks the goodness of fit of the transformed data relative to the hypothesized underlying reference distribution with mean-0 and standard deviation-1 (e.g. $N(0,1)$). The red dashed line depicts the LOESS scatterplot smoother estimator. The subroutine internally generates reference null distributions for computing the similarity statistic under each cluster configuration. The optimal cluster configuration (indicated by the vertical red arrow) is found where the similarity statistic reaches its minimum plus/minus one standard deviation (applying the conventional one-standard deviation rule). A smaller cluster number configuration indicates under-regularization, while over-regularization starts to occur at larger numbers. This over/under-regularization must be viewed as a form of over/under-fitting (see *Dazard, J-E. and J. S. Rao (2012)*

for more details). The quantile diagnostic plots uses empirical quantiles of the transformed means and standard deviations to check how closely they are approximated by theoretical quantiles derived from a standard normal *equal-mean/homoscedastic* model (solid green lines) under a given cluster configuration. To assess this goodness of fit of the transformed data, theoretical null distributions of the mean and variance are derived from a standard normal *equal-mean/homoscedastic* model with independence of the first two moments, i.e. assuming i.i.d. normality of the raw data. However, we do not require i.i.d. normality of the data in general: these theoretical null distributions are just used here as convenient ones to draw from. Note that under the assumptions that the raw data is i.i.d. standard normal ($N(0, 1)$) with independence of first two moments, the theoretical null distributions of means and standard deviations for each variable are respectively: $N(0, \frac{1}{n})$ and $\sqrt{\frac{\chi^2_{n-G}}{n-G}}$, where G denotes the number of sample groups. The optimal cluster configuration found is indicated by the most horizontal red curve. The single cluster configuration, corresponding to no transformation, is the most vertical curve, while the largest cluster number configuration reaches horizontality. Notice how empirical quantiles of transformed pooled means and standard deviations converge (from red to black) to the theoretical null distributions (solid green lines) for the optimal configuration. One should see a convergence towards the target null, after which overfitting starts to occur (see *Dazard, J-E. and J. S. Rao (2012)* for more details). Both cluster diagnostic plots help determine (i) whether the minimum of the *Similarity Statistic* is observed within the range of clusters (i.e. a large enough number of clusters has been accommodated), and (ii) whether the corresponding cluster configuration is a good fit. If necessary, run the procedure again with larger value of the `nc.max` parameter in the `mvr` as well as in `mvrt.test` functions until the minimum of the similarity statistic profile is reached.

Option `file` is used only if device is specified (i.e. non `NULL`).

Value

None. Displays the plots on the chosen `device`.

Note

End-user function.

Author(s)

- "Jean-Eudes Dazard, Ph.D." `<jxd101@case.edu>`
- "Hua Xu, Ph.D." `<huaxu77@gmail.com>`
- "Alberto Santana, MBA." `<ahs4@case.edu>`

Maintainer: "Jean-Eudes Dazard, Ph.D." `<jxd101@case.edu>`

Acknowledgments: This project was partially funded by the National Institutes of Health (P30-CA043703 to J-E.DAZARD).

References

- Dazard J-E., Hua Xu and J. S. Rao (2011). "*R package MVR for Joint Adaptive Mean-Variance Regularization and Variance Stabilization.*" In JSM Proceedings, Section for Statistical Programmers and Analysts. Miami Beach, FL, USA: American Statistical Association IMS - JSM, 3849-3863.
- Dazard J-E. and J. S. Rao (2012). "*Joint Adaptive Mean-Variance Regularization and Variance Stabilization of High Dimensional Data.*" Comput. Statist. Data Anal. 56(7):2317-2333.

See Also

`loess` (R package **stats**) Fit a polynomial surface determined by one or more numerical predictors, using local fitting.

Examples

```
## Not run:
    #=================================================
    # Loading the library and its dependencies
    #=================================================
    library("MVR")

    #=================================================
    # MVR package news
    #=================================================
    MVR.news()

    #=================================================
    # MVR package citation
    #=================================================
    citation("MVR")

    #=================================================
    # Loading of the Synthetic and Real datasets
    # (see description of datasets)
    #=================================================
    data("Synthetic", "Real", package="MVR")
    ?Synthetic
    ?Real

    #=================================================
    # Mean-Variance Regularization (Real dataset)
    # Multi-Group Assumption
    # Assuming unequal variance between groups
    # Without cluster usage
    =================================================
    nc.min <- 1
    nc.max <- 30
    probs <- seq(0, 1, 0.01)
    n <- 6
    GF <- factor(gl(n = 2, k = n/2, len = n),
                 ordered = FALSE,
                 labels = c("M", "S"))
    mvr.obj <- mvr(data = Real,
                   block = GF,
                   log = FALSE,
                   nc.min = nc.min,
                   nc.max = nc.max,
                   probs = probs,
                   B = 100,
                   parallel = FALSE,
                   conf = NULL,
                   verbose = TRUE)

    #=================================================
    # Summary Cluster Diagnostic Plots (Real dataset)
```

```
# Multi-Group Assumption
# Assuming unequal variance between groups
#=================================================
cluster.diagnostic(obj = mvr.obj,
                   title = "Cluster Diagnostic Plots
                   (Real - Multi-Group Assumption)",
                   span = 0.75,
                   degree = 2,
                   family = "gaussian",
                   device = NULL,
                   horizontal = FALSE,
                   width = 8.5,
                   height = 11)

#=================================================
# Mean-Variance Regularization (Real dataset)
# Single-Group Assumption
# Assuming equal variance between groups
# Without cluster usage
#=================================================
nc.min <- 1
nc.max <- 30
probs <- seq(0, 1, 0.01)
n <- 6
mvr.obj <- mvr(data = Real,
               block = rep(1,n),
               log = FALSE,
               nc.min = nc.min,
               nc.max = nc.max,
               probs = probs,
               B = 100,
               parallel = FALSE,
               conf = NULL,
               verbose = TRUE)

#=================================================
# Summary Cluster Diagnostic Plots (Real dataset)
# Single-Group Assumption
# Assuming equal variance between groups
#=================================================
cluster.diagnostic(obj = mvr.obj,
                   title = "Cluster Diagnostic Plots
                   (Real - Single-Group Assumption)",
                   span = 0.75,
                   degree = 2,
                   family = "gaussian",
                   device = NULL,
                   horizontal = FALSE,
                   width = 8.5,
                   height = 11)

## End(Not run)
```

`mvr` *Function for Mean-Variance Regularization and Variance Stabilization*

Description

End-user function for Mean-Variance Regularization (MVR) and Variance Stabilization by similarity statistic under sample group homoscedasticity or heteroscedasticity assumptions.

Return an object of class "mvr". Offers the option of parallel computation for improved efficiency.

Usage

```
mvr(data,
    block = rep(1,nrow(data)),
    tolog = FALSE,
    nc.min = 1,
    nc.max = 30,
    probs = seq(0, 1, 0.01),
    B = 100,
    parallel = FALSE,
    conf = NULL,
    verbose = TRUE)
```

Arguments

data	numeric matrix of untransformed (raw) data, where samples are by rows and variables (to be clustered) are by columns, or an object that can be coerced to such a matrix (such as a numeric vector or a data.frame with all numeric columns). Missing values (NA), NotANumber values (NaN) or Infinite values (Inf) are not allowed.
block	character or numeric vector or factor grouping/blocking variable of length the sample size. Defaults to single group situation (see details).
tolog	logical scalar. Is the data to be log2-transformed first? Optional, defaults to FALSE. Note that negative or null values will be changed to 1 before taking log2-transformation.
nc.min	Positive integer scalar of the minimum number of clusters, defaults to 1
nc.max	Positive integer scalar of the maximum number of clusters, defaults to 30
probs	numeric vector of probabilities for quantile diagnostic plots. Defaults to seq(0, 1, 0.01).
B	Positive integer scalar of the number of Monte Carlo replicates of the inner loop of the sim statistic function (see details).
parallel	logical scalar. Is parallel computing to be performed? Optional, defaults to FALSE.
conf	list of parameters for cluster configuration. Inputs for R package **parallel** function makeCluster (R package **parallel**) for cluster setup. Optional, defaults to NULL. See details for usage.
verbose	logical scalar. Is the output to be verbose? Optional, defaults to TRUE.

Details

Argument block is a vector or a factor grouping/blocking variable. It must be of length sample size with as many different character or numeric values as the number of levels or sample groups. It defaults to single group situation, i.e. under the assumption of equal variance between sample groups. All group sample sizes must be greater than 1, otherwise the program will stop.

Note that argument B is internally reset to conf$cpus*ceiling(B/conf$cpus) in case the parallelization is used (i.e. conf is non NULL), where conf$cpus denotes the total number of CPUs to be used (see below).

Argument nc.max currently defaults to 30. Empirically, we found that this is enough for most datasets tested. This depends on (i) the dimensionality/sample size ratio $\frac{p}{n}$, (ii) the signal/noise ratio, and (iii) whether a pre-transformation has been applied (see *Dazard, J-E. and J. S. Rao (2012)* for more details). See the cluster diagnostic function cluster.diagnostic for more details, whether larger values of nc.max may be required.

To run a parallel session (and parallel RNG) of the MVR procedures (parallel=TRUE), argument conf is to be specified (i.e. non NULL). It must list the specifications of the folowing parameters for cluster configuration: "names", "cpus", "type", "homo", "verbose", "outfile". These match the arguments described in function makeCluster of the R package **parallel**. All fields are required to properly configure the cluster, except for "names" and "cpus", which are the values used alternatively in the case of a cluster of type "SOCK" (socket), or in the case of a cluster of type other than "SOCK" (socket), respectively.

- "names": names : character vector specifying the host names on which to run the job. Could default to a unique local machine, in which case, one may use the unique host name "localhost". Each host name can potentially be repeated to the number of CPU cores available on the corresponding machine.
- "cpus": spec : integer scalar specifying the total number of CPU cores to be used across the network of available nodes, counting the workernodes and masternode.
- "type": type : character vector specifying the cluster type ("SOCK", "PVM", "MPI").
- "homo": homogeneous : logical scalar to be set to FALSE for inhomogeneous clusters.
- "verbose": verbose : logical scalar to be set to FALSE for quiet mode.
- "outfile": outfile : character vector of the output log file name for the workernodes.

The actual creation of the cluster, its initialization, and closing are all done internally. In addition, when random number generation is needed, the creation of separate streams of parallel RNG per node is done internally by distributing the stream states to the nodes (For more details see function makeCluster (R package **parallel**) and/or http://www.stat.uiowa.edu/~luke/R/cluster/cluster.html.

Value

Xraw	numeric matrix of original data.
Xmvr	numeric matrix of MVR-transformed data.
centering	numeric vector of centering values for standardization (cluster mean of pooled sample mean).
scaling	numeric vector of scaling values for standardization (cluster mean of pooled sample std dev).
MVR	list (of size the number of groups) containing for each group: • membership numeric vector of cluster membership of each variable • nc Positive integer scalar of number of clusters found in optimal cluster configuration • gap numeric vector of the similarity statistic values • sde numeric vector of the standard errors of the similarity statistic values

- mu.std `numeric matrix` (`K` x p) of the vector of standardized means by groups (rows), where `K` = \#groups and `p` = \#variables
- sd.std `numeric matrix` (`K` x p) of the vector of standardized standard deviations by groups (rows), where `K` = \#groups and `p` = \#variables
- mu.quant `numeric matrix` (`nc.max - nc.min` + 1) x (length(`probs`)) of quantiles of means
- sd.quant `numeric matrix` (`nc.max - nc.min` + 1) x (length(`probs`)) of quantiles of standard deviations

`block`	Value of argument `block`.
`tolog`	Value of argument `tolog`.
`nc.min`	Value of argument `nc.min`.
`nc.max`	Value of argument `nc.max`.
`probs`	Value of argument `probs`.

Note

End-user function.

Author(s)

- "Jean-Eudes Dazard, Ph.D." `<jxd101@case.edu>`
- "Hua Xu, Ph.D." `<huaxu77@gmail.com>`
- "Alberto Santana, MBA." `<ahs4@case.edu>`

Maintainer: "Jean-Eudes Dazard, Ph.D." `<jxd101@case.edu>`

Acknowledgments: This project was partially funded by the National Institutes of Health (P30-CA043703 to J-E.DAZARD).

References

- Dazard J-E., Hua Xu and J. S. Rao (2011). "*R package MVR for Joint Adaptive Mean-Variance Regularization and Variance Stabilization.*" In JSM Proceedings, Section for Statistical Programmers and Analysts. Miami Beach, FL, USA: American Statistical Association IMS - JSM, 3849-3863.
- Dazard J-E. and J. S. Rao (2012). "*Joint Adaptive Mean-Variance Regularization and Variance Stabilization of High Dimensional Data.*" Comput. Statist. Data Anal. 56(7):2317-2333.

See Also

- `makeCluster` (R package **parallel**).
- `justvsn` (R package **vsn**) Variance stabilization and calibration for microarray data *Huber, 2002*

Examples

```
#=================================================
# Loading the library and its dependencies
#=================================================
library("MVR")

## Not run:
```

```
    #=================================================
    # MVR package news
    #=================================================
    MVR.news()

    #=================================================
    # MVR package citation
    #=================================================
    citation("MVR")

    #=================================================
    # Loading of the Synthetic and Real datasets
    # (see description of datasets)
    #=================================================
    data("Synthetic", "Real", package="MVR")
    ?Synthetic
    ?Real

## End(Not run)

#=================================================
# Mean-Variance Regularization (Synthetic dataset)
# Single-Group Assumption
# Assuming equal variance between groups
# Without cluster usage
#=================================================
nc.min <- 1
nc.max <- 10
probs <- seq(0, 1, 0.01)
n <- 10
mvr.obj <- mvr(data = Synthetic,
               block = rep(1,n),
               tolog = FALSE,
               nc.min = nc.min,
               nc.max = nc.max,
               probs = probs,
               B = 100,
               parallel = FALSE,
               conf = NULL,
               verbose = TRUE)

## Not run:
    #=================================================
    # Examples of parallelization below with
    # a SOCKET or MPI cluster configuration
    #=================================================
    # 1- WINDOWS multicores PC with SOCKET communication
    #    With a 2-Quad (8-CPUs) PC
    #=================================================
    if (.Platform$OS.type == "windows") {
        cpus <- detectCores()
        conf <- list("names" = rep("localhost", cpus),
                     "cpus" = cpus,
                     "type" = "SOCK",
                     "homo" = TRUE,
                     "verbose" = TRUE,
                     "outfile" = "")
```

```
}
#=====================================================
# 2- LINUX multinodes cluster with SOCKET communication
#    with 4-nodes (32-CPUs) cluster
#    with 1 masternode and 3 workernodes
#    All hosts run identical setups
#    Same number of core CPUs (8) per node
#=====================================================
if (.Platform$OS.type == "unix") {
    masterhost <- Sys.getenv("HOSTNAME")
    slavehosts <- c("compute-0-0", "compute-0-1", "compute-0-2")
    nodes <- length(slavehosts) + 1
    cpus <- 8
    conf <- list("names" = c(rep(masterhost, cpus),
                             rep(slavehosts, cpus)),
                 "cpus" = nodes * cpus,
                 "type" = "SOCK",
                 "homo" = TRUE,
                 "verbose" = TRUE,
                 "outfile" = "")
}
#=====================================================
# 3- LINUX multinodes cluster with MPI communication
#    Here, a file named ".nodes" (e.g. in the home directory)
#    must contain the list of nodes of the cluster
#=====================================================
if (.Platform$OS.type == "unix") {
    hosts <- scan(file=paste(Sys.getenv("HOME"), "/.nodes", sep=""),
                  what="",
                  sep="\n")
    hostnames <- unique(hosts)
    nodes <- length(hostnames)
    cpus <-  length(hosts)/length(hostnames)
    conf <- list("cpus" = nodes * cpus,
                 "type" = "MPI",
                 "homo" = TRUE,
                 "verbose" = TRUE,
                 "outfile" = "")
}
#=====================================================
# Run:
# Mean-Variance Regularization (Real dataset)
# Multi-Group Assumption
# Assuming unequal variance between groups
#=====================================================
nc.min <- 1
nc.max <- 30
probs <- seq(0, 1, 0.01)
n <- 6
GF <- factor(gl(n = 2, k = n/2, len = n),
             ordered = FALSE,
             labels = c("M", "S"))
mvr.obj <- mvr(data = Real,
               block = GF,
               tolog = FALSE,
               nc.min = nc.min,
               nc.max = nc.max,
```

```
                    probs = probs,
                    B = 100,
                    parallel = TRUE,
                    conf = conf,
                    verbose = TRUE)

## End(Not run)
```

`MVR.news` *Function to Display the NEWS File*

Description

Function to display the NEWS file of the **MVR** package.

Usage

```
MVR.news(...)
```

Arguments

`...`	Further arguments passed to or from other methods.

Value

None.

Note

End-user function.

Author(s)

- "Jean-Eudes Dazard, Ph.D." <jxd101@case.edu>
- "Hua Xu, Ph.D." <huaxu77@gmail.com>
- "Alberto Santana, MBA." <ahs4@case.edu>

Maintainer: "Jean-Eudes Dazard, Ph.D." <jxd101@case.edu>

Acknowledgments: This project was partially funded by the National Institutes of Health (P30-CA043703 to J-E.DAZARD).

References

- Dazard J-E., Hua Xu and J. S. Rao (2011). "*R package MVR for Joint Adaptive Mean-Variance Regularization and Variance Stabilization.*" In JSM Proceedings, Section for Statistical Programmers and Analysts. Miami Beach, FL, USA: American Statistical Association IMS - JSM, 3849-3863.
- Dazard J-E. and J. S. Rao (2012). "*Joint Adaptive Mean-Variance Regularization and Variance Stabilization of High Dimensional Data.*" Comput. Statist. Data Anal. 56(7):2317-2333.

`mvrt.test` *Function for Computing Mean-Variance Regularized T-test Statistic and Its Significance*

Description

End-user function for computing MVR t-test statistic and its significance (p-value) under sample group homoscedasticity or heteroscedasticity assumption.

Return an object of class "`mvrt.test`". Offers the option of parallel computation for improved efficiency.

Usage

```
mvrt.test(data,
          obj=NULL,
          block,
          tolog = FALSE,
          nc.min = 1,
          nc.max = 30,
          pval = FALSE,
          replace = FALSE,
          n.resamp = 100,
          parallel = FALSE,
          conf = NULL,
          verbose = TRUE)
```

Arguments

`data`	`numeric matrix` of untransformed (raw) data, where samples are by rows and variables (to be clustered) are by columns, or an object that can be coerced to such a `matrix` (such as a `numeric vector` or a `data.frame` with all `numeric` columns). Missing values (`NA`), NotANumber values (`NaN`) or Infinite values (`Inf`) are not allowed.
`obj`	Object of class "`mvr`" returned by `mvr`.
`block`	`character` or `numeric vector` or `factor` grouping/blocking variable of length the sample size. (see details).
`tolog`	`logical` scalar. Is the data to be log2-transformed first? Optional, defaults to `FALSE`. Note that negative or null values will be changed to 1 before taking log2-transformation.
`nc.min`	Positive `integer` scalar of the minimum number of clusters, defaults to 1
`nc.max`	Positive `integer` scalar of the maximum number of clusters, defaults to 30
`pval`	`logical` scalar. Shall p-values be computed? If not, `n.resamp` and `replace` will be ignored. If `FALSE` (default), t-statistic only will be computed, If `TRUE`, exact (permutation test) or approximate (bootstrap test) p-values will be computed.
`replace`	`logical` scalar. Shall permutation test (default) or bootstrap test be computed? If `FALSE` (default), permutation test will be computed with null permutation distribution, If `TRUE`, bootstrap test will be computed with null bootstrap distribution.

n.resamp	Positive integer scalar of the number of resamplings to compute (default=100) by permutation or bootstsrap (see details).
parallel	logical scalar. Is parallel computing to be performed? Optional, defaults to FALSE.
conf	list of parameters for cluster configuration. Inputs for R package **parallel** function makeCluster (R package **parallel**) for cluster setup. Optional, defaults to NULL. See details for usage.
verbose	logical scalar. Is the output to be verbose? Optional, defaults to TRUE.

Details

Argument block is a vector or a factor grouping/blocking variable. It must be of length sample size with as many different character or numeric values as the number of levels or sample groups. The number of sample groups must be greater or equal to 2, and all group sample sizes must be greater than 1, otherwise the program will stop.

Argument nc.max currently defaults to 30. Empirically, we found that this is enough for most datasets tested. This depends on (i) the dimensionality/sample size ratio $\frac{p}{n}$, (ii) the signal/noise ratio, and (iii) whether a pre-transformation has been applied (see *Dazard, J-E. and J. S. Rao (2012)* for more details). See the cluster diagnostic function cluster.diagnostic for more details, whether larger values of nc.max may be required.

Argument n.resamp is reset to conf$cpus*ceiling(n.resamp/conf$cpus) in case the cluster is used (i.e. conf is non NULL), where conf$cpus denotes the total number of CPUs to be used (see below).

To save un-necessary computations, previously computed MVR clustering can be provided through option obj (i.e. obj is fully specified as a mvr object). In this case, arguments data, block, tolog, nc.min, nc.max are ignored. If obj is fully specified (i.e. an object of class "mvr" returned by mvr), the the MVR clustering provided by obj will be used for the computation of the regularized t-test statistics. If obj=NULL, a MVR clustering computation for the regularized t-test statistics and/or p-values will be performed.

To run a parallel session (and parallel RNG) of the MVR procedures (parallel=TRUE), argument conf is to be specified (i.e. non NULL). It must list the specifications of the folowing parameters for cluster configuration: "names", "cpus", "type", "homo", "verbose", "outfile". These match the arguments described in function makeCluster of the R package **parallel**. All fields are required to properly configure the cluster, except for "names" and "cpus", which are the values used alternatively in the case of a cluster of type "SOCK" (socket), or in the case of a cluster of type other than "SOCK" (socket), respectively.

- "names": names : character vector specifying the host names on which to run the job. Could default to a unique local machine, in which case, one may use the unique host name "localhost". Each host name can potentially be repeated to the number of CPU cores available on the corresponding machine.
- "cpus": spec : integer scalar specifying the total number of CPU cores to be used across the network of available nodes, counting the workernodes and masternode.
- "type": type : character vector specifying the cluster type ("SOCK", "PVM", "MPI").
- "homo": homogeneous : logical scalar to be set to FALSE for inhomogeneous clusters.
- "verbose": verbose : logical scalar to be set to FALSE for quiet mode.
- "outfile": outfile : character vector of the output log file name for the workernodes.

Note that the actual creation of the cluster, its initialization, and closing are all done internally. In addition, when random number generation is needed, the creation of separate streams of parallel

RNG per node is done internally by distributing the stream states to the nodes (For more details see function `makeCluster` (R package **parallel**) and/or http://www.stat.uiowa.edu/~luke/R/cluster/cluster.html.

In case p-values are desired (`pval=TRUE`), the use of the cluster is highly recommended. It is ideal for computing embarassingly parallel tasks such as permutation or bootstrap resamplings. Note that in case both regularized t-test statistics and p-values are desired, in order to maximize computational efficiency and avoid multiple configurations (since a cluster can only be configured and used one session at a time, which otherwise would result in a run stop), the cluster configuration will only be used for the parallel computation of p-values, but not for the MVR clustering computation of the regularized t-test statistics.

Value

`statistic`	`vector`, of size the number of variables, where entries are the t-statistics values of each variable.
`p.value`	`vector`, of size the number of variables, where entries are the p-values (if requested, otherwise `NULL` value) of each variable.

Note

End-user function.

Author(s)

- "Jean-Eudes Dazard, Ph.D." `<jxd101@case.edu>`
- "Hua Xu, Ph.D." `<huaxu77@gmail.com>`
- "Alberto Santana, MBA." `<ahs4@case.edu>`

Maintainer: "Jean-Eudes Dazard, Ph.D." `<jxd101@case.edu>`

Acknowledgments: This project was partially funded by the National Institutes of Health (P30-CA043703 to J-E.DAZARD).

References

- Dazard J-E., Hua Xu and J. S. Rao (2011). "*R package MVR for Joint Adaptive Mean-Variance Regularization and Variance Stabilization.*" In JSM Proceedings, Section for Statistical Programmers and Analysts. Miami Beach, FL, USA: American Statistical Association IMS - JSM, 3849-3863.
- Dazard J-E. and J. S. Rao (2012). "*Joint Adaptive Mean-Variance Regularization and Variance Stabilization of High Dimensional Data.*" Comput. Statist. Data Anal. 56(7):2317-2333.

See Also

- `makeCluster` (R package **parallel**)
- `eBayes` (R package **limma**) Bayesian Regularized t-test statistic *Smyth, 2004*
- `samr` (R package **samr**) SAM Regularized t-test statistic *Tusher et al., 2001, Storey, 2003*
- `matest` (R package **maanova**) James-Stein shrinkage estimator-based Regularized t-test statistic *Cui et al., 2005*
- `ebam` (R package **siggenes**) Empirical Bayes Regularized z-test statistic *Efron, 2001*
- `bayesT` Hierarchical Bayesian Regularized t-test statistic *Baldi et al., 2001*

Examples

```
#================================================
# Loading the library and its dependencies
#================================================
library("MVR")

## Not run:
    #=====================================================
    # MVR package news
    #=====================================================
    MVR.news()

    #================================================
    # MVR package citation
    #================================================
    citation("MVR")

    #=====================================================
    # Loading of the Synthetic and Real datasets
    # (see description of datasets)
    #=====================================================
    data("Synthetic", "Real", package="MVR")
    ?Synthetic
    ?Real

## End(Not run)

#================================================
# Regularized t-test statistics (Synthetic dataset)
# Multi-Group Assumption
# Assuming unequal variance between groups
# With option to use prior MVR clustering results
# Without computation of p-values
# Without cluster usage
#================================================
nc.min <- 1
nc.max <- 10
probs <- seq(0, 1, 0.01)
n <- 10
GF <- factor(gl(n = 2, k = n/2, len = n),
             ordered = FALSE,
             labels = c("G1", "G2"))
mvr.obj <- mvr(data = Synthetic,
               block = GF,
               tolog = FALSE,
               nc.min = nc.min,
               nc.max = nc.max,
               probs = probs,
               B = 100,
               parallel = FALSE,
               conf = NULL,
               verbose = TRUE)
mvrt.obj <- mvrt.test(obj = mvr.obj,
                      pval = FALSE,
                      parallel = FALSE,
                      conf = NULL,
```

```
                      verbose = TRUE)

## Not run:
    #=====================================================
    # Examples of parallelization below with
    # a SOCKET or MPI cluster configuration
    #=====================================================
    # 1- WINDOWS multicores PC with SOCKET communication
    #    With a 2-Quad (8-CPUs) PC
    #=====================================================
    if (.Platform$OS.type == "windows") {
        cpus <- detectCores()
        conf <- list("names" = rep("localhost", cpus),
                     "cpus" = cpus,
                     "type" = "SOCK",
                     "homo" = TRUE,
                     "verbose" = TRUE,
                     "outfile" = "")
    }
    #=====================================================
    # 2- LINUX multinodes cluster with SOCKET communication
    #    with 4-nodes (32-CPUs) cluster
    #    with 1 masternode and 3 workernodes
    #    All hosts run identical setups
    #    Same number of core CPUs (8) per node
    #=====================================================
    if (.Platform$OS.type == "unix") {
        masterhost <- Sys.getenv("HOSTNAME")
        slavehosts <- c("compute-0-0", "compute-0-1", "compute-0-2")
        nodes <- length(slavehosts) + 1
        cpus <- 8
        conf <- list("names" = c(rep(masterhost, cpus),
                                 rep(slavehosts, cpus)),
                     "cpus" = nodes * cpus,
                     "type" = "SOCK",
                     "homo" = TRUE,
                     "verbose" = TRUE,
                     "outfile" = "")
    }
    #=====================================================
    # 3- LINUX multinodes cluster with MPI communication
    #    Here, a file named ".nodes" (e.g. in the home directory)
    #    must contain the list of nodes of the cluster
    #=====================================================
    if (.Platform$OS.type == "unix") {
        hosts <- scan(file=paste(Sys.getenv("HOME"), "/.nodes", sep=""),
                      what="",
                      sep="\n")
        hostnames <- unique(hosts)
        nodes <- length(hostnames)
        cpus <-  length(hosts)/length(hostnames)
        conf <- list("cpus" = nodes * cpus,
                     "type" = "MPI",
                     "homo" = TRUE,
                     "verbose" = TRUE,
                     "outfile" = "")
    }
```

```
#=====================================================
# Mean-Variance Regularization (Real dataset)
# Multi-Group Assumption
# Assuming unequal variance between groups
#=====================================================
nc.min <- 1
nc.max <- 30
probs <- seq(0, 1, 0.01)
n <- 6
GF <- factor(gl(n = 2, k = n/2, len = n),
             ordered = FALSE,
             labels = c("M", "S"))
mvr.obj <- mvr(data = Real,
               block = GF,
               tolog = FALSE,
               nc.min = nc.min,
               nc.max = nc.max,
               probs = probs,
               B = 100,
               parallel = TRUE,
               conf = conf,
               verbose = TRUE)
#=====================================================
# Regularized t-test statistics (Real dataset)
# Multi-Group Assumption
# Assuming unequal variance between groups
# With option to use prior MVR clustering results
# With computation of p-values
#=====================================================
mvrt.obj <- mvrt.test(obj = mvr.obj,
                      pval = TRUE,
                      replace = FALSE,
                      n.resamp = 100,
                      parallel = TRUE,
                      conf = conf,
                      verbose = TRUE)

## End(Not run)
```

`normalization.diagnostic` *Function for Plotting Summary Normalization Diagnostic Plots*

Description

Plot comparative Box-Whisker and Heatmap plots of variables across samples check the effectiveness of normalization before and after Mean-Variance Regularization.

Usage

```
normalization.diagnostic(obj,
                         pal,
                         title = "Normalization Diagnostic Plots",
                         device = NULL,
```

```
                                    file = "Normalization Diagnostic Plots",
                                    path = getwd(),
                                    horizontal = FALSE,
                                    width = 7,
                                    height = 8, ...)
```

Arguments

obj	Object of class "mvr" returned by mvr.
title	Title of the plot. Defaults to "Normalization Diagnostic Plots".
pal	Color palette.
device	Graphic display device in {NULL, "PS", "PDF"}. Defaults to NULL (standard output screen). Currently implemented graphic display devices are "PS" (Postscript) or "PDF" (Portable Document Format).
file	File name for output graphic. Defaults to "Normalization Diagnostic Plots".
path	Absolute path (without final (back)slash separator). Defaults to working directory path.
horizontal	Logical scalar. Orientation of the printed image. Defaults to FALSE, that is potrait orientation.
width	Numeric scalar. Width of the graphics region in inches. Defaults to 7.
height	Numeric scalar. Height of the graphics region in inches. Defaults to 8.
...	Generic arguments passed to other plotting functions.

Details

Option file is used only if device is specified (i.e. non NULL). The argument pal can be any color palette, e.g. as provided by R package **RColorBrewer**.

Value

None. Displays the plots on the chosen device.

Note

End-user function.

Author(s)

- "Jean-Eudes Dazard, Ph.D." <jxd101@case.edu>
- "Hua Xu, Ph.D." <huaxu77@gmail.com>
- "Alberto Santana, MBA." <ahs4@case.edu>

Maintainer: "Jean-Eudes Dazard, Ph.D." <jxd101@case.edu>

Acknowledgments: This project was partially funded by the National Institutes of Health (P30-CA043703 to J-E.DAZARD).

References

- Dazard J-E., Hua Xu and J. S. Rao (2011). "*R package MVR for Joint Adaptive Mean-Variance Regularization and Variance Stabilization.*" In JSM Proceedings, Section for Statistical Programmers and Analysts. Miami Beach, FL, USA: American Statistical Association IMS - JSM, 3849-3863.
- Dazard J-E. and J. S. Rao (2012). "*Joint Adaptive Mean-Variance Regularization and Variance Stabilization of High Dimensional Data.*" Comput. Statist. Data Anal. 56(7):2317-2333.

See Also

`justvsn` (R package **vsn**) Variance stabilization and calibration for microarray data. `loess` (R package **stats**) Fit a polynomial surface determined by one or more numerical predictors, using local fitting.

Examples

```
## Not run:
    #=====================================================
    # Loading the library and its dependencies
    #=====================================================
    library("MVR")
    library("RColorBrewer")

    #=====================================================
    # MVR package news
    #=====================================================
    MVR.news()

    #=================================================
    # MVR package citation
    #=================================================
    citation("MVR")

    #=====================================================
    # Loading of the Synthetic and Real datasets
    # (see description of datasets)
    #=====================================================
    data("Synthetic", "Real", package="MVR")
    ?Synthetic
    ?Real

    #=====================================================
    # Mean-Variance Regularization (Real dataset)
    # Multi-Group Assumption
    # Assuming unequal variance between groups
    # Without cluster usage
    #=====================================================
    nc.min <- 1
    nc.max <- 30
    probs <- seq(0, 1, 0.01)
    n <- 6
    GF <- factor(gl(n = 2, k = n/2, len = n),
                 ordered = FALSE,
                 labels = c("M", "S"))
    mvr.obj <- mvr(data = Real,
```

```
                block = GF,
                log = FALSE,
                nc.min = nc.min,
                nc.max = nc.max,
                probs = probs,
                B = 100,
                parallel = FALSE,
                conf = NULL,
                verbose = TRUE)

#=================================================
# Summary Normalization Diagnostic Plots (Real dataset)
# Multi-Group Assumption
# Assuming unequal variance between groups
#=================================================
normalization.diagnostic(obj = mvr.obj,
                         title = "Normalization Diagnostic Plots
                         (Real - Multi-Group Assumption)",
                         pal = brewer.pal(n=11, name="RdYlGn"),
                         device = NULL,
                         horizontal = FALSE,
                         width = 7,
                         height = 8)

#=================================================
# Mean-Variance Regularization (Real dataset)
# Single-Group Assumption
# Assuming equal variance between groups
# Without cluster usage
#=================================================
nc.min <- 1
nc.max <- 30
probs <- seq(0, 1, 0.01)
n <- 6
mvr.obj <- mvr(data = Real,
               block = rep(1,n),
               log = FALSE,
               nc.min = nc.min,
               nc.max = nc.max,
               probs = probs,
               B = 100,
               parallel = FALSE,
               conf = NULL,
               verbose = TRUE)

#=================================================
# Summary Normalization Stabilization Diagnostic Plots (Real dataset)
# Single-Group Assumption
# Assuming equal variance between groups
#=================================================
normalization.diagnostic(obj = mvr.obj,
                         title = "Normalization Diagnostic Plots
                         (Real - Single-Group Assumption)",
                         pal = brewer.pal(n=11, name="RdYlGn"),
                         device = NULL,
                         horizontal = FALSE,
                         width = 7,
```

```
                                height = 8)

## End(Not run)
```

`Real` *Real Proteomics Dataset*

Description

The dataset comes from a quantitative Liquid Chromatography/Mass-Spectrometry (LC/MS) shotgun (bottom-up) proteomics experiment. It consists of $n = 6$ independent cell cultures of human of Myeloid Dendritic Cells (MDCs) from normal subjects. Samples were split into a control ("*M*") and a treated group ("*S*"), stimulated with either media alone or a Toll-Like receptor-3 Ligand respectively. The goal was to identify differentially expressed peptides (or proteins) between the two groups involved in the immune response of human MDCs upon TLR-3 Ligand binding.

The dataset is assumed to have been pre-processed for non-ignorable missing values, leaving a complete dataset with $p = 9052$ unique peptides or predictor variables.

This is a balanced design with two sample groups ($G = 2$), under unequal sample group variance.

Usage

```
Real
```

Format

A numeric matrix containing $n = 6$ observations (samples) by rows and $p = 9052$ variables by columns, named after peptide names ($diffset_1, ..., diffset_p$). Samples are balanced ($n_1 = 3$,$n_2 = 3$) between the two groups ("*M*", "*S*"). Compressed Rda data file.

Author(s)

- "Jean-Eudes Dazard, Ph.D." <jxd101@case.edu>
- "Hua Xu, Ph.D." <huaxu77@gmail.com>
- "Alberto Santana, MBA." <ahs4@case.edu>

Maintainer: "Jean-Eudes Dazard, Ph.D." <jxd101@case.edu>

Acknowledgments: This project was partially funded by the National Institutes of Health (P30-CA043703 to J-E.DAZARD).

Source

See real proteomics data application in Dazard et al., 2011, 2012.

References

- Dazard J-E., Hua Xu and J. S. Rao (2011). "*R package MVR for Joint Adaptive Mean-Variance Regularization and Variance Stabilization.*" In JSM Proceedings, Section for Statistical Programmers and Analysts. Miami Beach, FL, USA: American Statistical Association IMS - JSM, 3849-3863.
- Dazard J-E. and J. S. Rao (2012). "*Joint Adaptive Mean-Variance Regularization and Variance Stabilization of High Dimensional Data.*" Comput. Statist. Data Anal. 56(7):2317-2333.

stabilization.diagnostic *Function for Plotting Summary Variance Stabilization Diagnostic Plots*

Description

Plot comparative variance-mean plots to check the variance stabilization across variables before and after Mean-Variance Regularization.

Usage

```
stabilization.diagnostic(obj,
                         span = 0.5,
                         degree = 2,
                         family = "gaussian",
                         title = "Stabilization Diagnostic Plots",
                         device = NULL,
                         file = "Stabilization Diagnostic Plots",
                         path = getwd(),
                         horizontal = FALSE,
                         width = 7,
                         height = 5, ...)
```

Arguments

obj	Object of class "mvr" returned by mvr.
title	Title of the plot. Defaults to "Stabilization Diagnostic Plots".
span	Span parameter of the loess() function (R package **stats**), which controls the degree of smoothing. Defaults to 0.75.
degree	Degree parameter of the loess() function (R package **stats**), which controls the degree of the polynomials to be used. Defaults to 2. (Normally 1 or 2. Degree 0 is also allowed, but see the "Note" in loess stats package.)
family	Family distribution in "gaussian", "symmetric" of the loess() function (R package **stats**), used for local fitting . If "gaussian" fitting is by least-squares, and if "symmetric" a re-descending M estimator is used with Tukey's biweight function.
device	Graphic display device in {NULL, "PS", "PDF"}. Defaults to NULL (standard output screen). Currently implemented graphic display devices are "PS" (Postscript) or "PDF" (Portable Document Format).
file	File name for output graphic. Defaults to "Stabilization Diagnostic Plots".
path	Absolute path (without final (back)slash separator). Defaults to working directory path.
horizontal	Logical scalar. Orientation of the printed image. Defaults to FALSE, that is potrait orientation.
width	Numeric scalar. Width of the graphics region in inches. Defaults to 7.
height	Numeric scalar. Height of the graphics region in inches. Defaults to 5.
...	Generic arguments passed to other plotting functions.

Details

In the plots of standard deviations vs. means, standard deviations and means are calculated in a feature-wise manner from the expression matrix. The scatterplot allows to visually verify whether there is a dependence of the standard deviation (or variance) on the mean. The black dotted line depicts the LOESS scatterplot smoother estimator. If there is no variance-mean dependence, then this line should be approximately horizontal.

Option `file` is used only if device is specified (i.e. non `NULL`).

Value

None. Displays the plots on the chosen `device`.

Note

End-user function.

Author(s)

- "Jean-Eudes Dazard, Ph.D." `<jxd101@case.edu>`
- "Hua Xu, Ph.D." `<huaxu77@gmail.com>`
- "Alberto Santana, MBA." `<ahs4@case.edu>`

Maintainer: "Jean-Eudes Dazard, Ph.D." `<jxd101@case.edu>`

Acknowledgments: This project was partially funded by the National Institutes of Health (P30-CA043703 to J-E.DAZARD).

References

- Dazard J-E., Hua Xu and J. S. Rao (2011). "*R package MVR for Joint Adaptive Mean-Variance Regularization and Variance Stabilization.*" In JSM Proceedings, Section for Statistical Programmers and Analysts. Miami Beach, FL, USA: American Statistical Association IMS - JSM, 3849-3863.
- Dazard J-E. and J. S. Rao (2012). "*Joint Adaptive Mean-Variance Regularization and Variance Stabilization of High Dimensional Data.*" Comput. Statist. Data Anal. 56(7):2317-2333.

See Also

`justvsn` (R package **vsn**) Variance stabilization and calibration for microarray data. `loess` (R package **stats**) Fit a polynomial surface determined by one or more numerical predictors, using local fitting.

Examples

```
## Not run:
    #=================================================
    # Loading the library and its dependencies
    #=================================================
    library("MVR")

    #=================================================
    # MVR package news
    #=================================================
    MVR.news()
```

```
#================================================
# MVR package citation
#================================================
citation("MVR")

#================================================
# Loading of the Synthetic and Real datasets
# (see description of datasets)
#================================================
data("Synthetic", "Real", package="MVR")
?Synthetic
?Real

#================================================
# Mean-Variance Regularization (Real dataset)
# Multi-Group Assumption
# Assuming unequal variance between groups
# Without cluster usage
#================================================
nc.min <- 1
nc.max <- 30
probs <- seq(0, 1, 0.01)
n <- 6
GF <- factor(gl(n = 2, k = n/2, len = n),
             ordered = FALSE,
             labels = c("M", "S"))
mvr.obj <- mvr(data = Real,
               block = GF,
               log = FALSE,
               nc.min = nc.min,
               nc.max = nc.max,
               probs = probs,
               B = 100,
               parallel = FALSE,
               conf = NULL,
               verbose = TRUE)

#================================================
# Summary Stabilization Diagnostic Plots (Real dataset)
# Multi-Group Assumption
# Assuming unequal variance between groups
#================================================
stabilization.diagnostic(obj = mvr.obj,
                         title = "Stabilization Diagnostic Plots
                         (Real - Multi-Group Assumption)",
                         span = 0.75,
                         degree = 2,
                         family = "gaussian",
                         device = NULL,
                         horizontal = FALSE,
                         width = 7,
                         height = 5)

#================================================
# Mean-Variance Regularization (Real dataset)
# Single-Group Assumption
```

```
# Assuming equal variance between groups
# Without cluster usage
#=================================================
nc.min <- 1
nc.max <- 30
probs <- seq(0, 1, 0.01)
n <- 6
mvr.obj <- mvr(data = Real,
               block = rep(1,n),
               log = FALSE,
               nc.min = nc.min,
               nc.max = nc.max,
               probs = probs,
               B = 100,
               parallel = FALSE,
               conf = NULL,
               verbose = TRUE)

#=================================================
# Summary Stabilization Diagnostic Plots (Real dataset)
# Single-Group Assumption
# Assuming equal variance between groups
#=================================================
stabilization.diagnostic(obj = mvr.obj,
                         title = "Stabilization Diagnostic Plots
                         (Real - Single-Group Assumption)",
                         span = 0.75,
                         degree = 2,
                         family = "gaussian",
                         device = NULL,
                         horizontal = FALSE,
                         width = 7,
                         height = 5)

## End(Not run)
```

`Synthetic` *Multi-Groups Synthetic Dataset*

Description

Generation of a synthetic dataset with *n*=10 observations (samples) and $p = 100$ variables, where $nvar = 20$ of them are significantly different between the two sample groups.

This is a balanced design with two sample groups ($G = 2$), under unequal sample group variance.

Usage

```
Synthetic
```

Format

A numeric matrix containing $n = 10$ observations (samples) by rows and $p = 100$ variables by columns, named $v_1, ..., v_p$. Samples are balanced ($n_1 = 5$,$n_2 = 5$) between the two groups (G_1, G_2). Compressed Rda data file.

Author(s)

- "Jean-Eudes Dazard, Ph.D." <jxd101@case.edu>
- "Hua Xu, Ph.D." <huaxu77@gmail.com>
- "Alberto Santana, MBA." <ahs4@case.edu>

Maintainer: "Jean-Eudes Dazard, Ph.D." <jxd101@case.edu>

Acknowledgments: This project was partially funded by the National Institutes of Health (P30-CA043703 to J-E.DAZARD).

Source

See model #2 in Dazard et al., 2011, 2012.

References

- Dazard J-E., Hua Xu and J. S. Rao (2011). "*R package MVR for Joint Adaptive Mean-Variance Regularization and Variance Stabilization.*" In JSM Proceedings, Section for Statistical Programmers and Analysts. Miami Beach, FL, USA: American Statistical Association IMS - JSM, 3849-3863.
- Dazard J-E. and J. S. Rao (2012). "*Joint Adaptive Mean-Variance Regularization and Variance Stabilization of High Dimensional Data.*" Comput. Statist. Data Anal. 56(7):2317-2333.

---

target.diagnostic *Function for Plotting Summary Target Moments Diagnostic Plots*

Description

Plot comparative distribution densities of means and standard deviations of the data before and after Mean-Variance Regularization to check for location shifts between observed first and second moments and their expected target values under a target centered homoscedastic model.

Plot comparative QQ scatterplots to look at departures between observed distributions of first and second moments of the MVR-transformed data and their theoretical distributions assuming independence and normality of all the variables.

Usage

```
target.diagnostic(obj,
                  title = "Target Moments Diagnostic Plots",
                  device = NULL,
                  file = "Target Moments Diagnostic Plots",
                  path = getwd(),
                  horizontal = FALSE,
                  width = 8.5,
                  height = 6.5, ...)
```

Arguments

obj	Object of class "mvr" returned by mvr.
title	Title of the plot. Defaults to "Target Moments Diagnostic Plots".
device	Graphic display device in {NULL, "PS", "PDF"}. Defaults to NULL (standard output screen). Currently implemented graphic display devices are "PS" (Postscript) or "PDF" (Portable Document Format).
file	File name for output graphic. Defaults to "Target Moments Diagnostic Plots".
path	Absolute path (without final (back)slash separator). Defaults to working directory path.
horizontal	Logical scalar. Orientation of the printed image. Defaults to FALSE, that is potrait orientation.
width	Numeric scalar. Width of the graphics region in inches. Defaults to 8.5.
height	Numeric scalar. Height of the graphics region in inches. Defaults to 6.5.
...	Generic arguments passed to other plotting functions.

Details

The plots of the density distribution of means and standard deviations checks that the distributions of means and standard deviations of the MVR-transformed data have correct target first moments, i.e. with mean ~ 0 and mean ~ 1. The expected target mean and standard deviation are shown in red (before and) after MVR-transformation. Caption shows the p-values from the parametric two-sample two-sided t-tests for the equality of parameters to their expectations (assuming normality since usually sample sizes are large : $p \gg 1$, or a relative robustness to moderate violations of the normality assumption).

In the general case, the variables are not normally distributed and not even independent and identically distributed before and after MVR-transformation. Therefore, the distributions of untransformed first and second moments usually differ from their respective theoretical null distributions, i.e., from $N(0, \frac{1}{n})$ for the means and from $\sqrt{\frac{\chi^2_{n-G}}{n-G}}$ for the standard deviations, where G denotes the number of sample groups (see *Dazard, J-E. and J. S. Rao (2012)* for more details). Also, the observed distributions of transformed first and second moments are unknown. This is reflected in the QQ plots, where theoretical and empirical quantiles do not necessarily align with each other. Caption shows the p-values from the nonparametric two-sample two-sided Kolmogorov-Smirnov tests of the null hypothesis that a parameter distribution differs from its theoretical distribution. Each black dot represents a variable. The red solid line depicts the interquartile line, which passes through the first and third quartiles.

Option file is used only if device is specified (i.e. non NULL).

Value

None. Displays the plots on the chosen device.

Note

End-user function.

Author(s)

- "Jean-Eudes Dazard, Ph.D." <jxd101@case.edu>
- "Hua Xu, Ph.D." <huaxu77@gmail.com>

- "Alberto Santana, MBA." <ahs4@case.edu>

Maintainer: "Jean-Eudes Dazard, Ph.D." <jxd101@case.edu>

Acknowledgments: This project was partially funded by the National Institutes of Health (P30-CA043703 to J-E.DAZARD).

References

- Dazard J-E., Hua Xu and J. S. Rao (2011). "*R package MVR for Joint Adaptive Mean-Variance Regularization and Variance Stabilization.*" In JSM Proceedings, Section for Statistical Programmers and Analysts. Miami Beach, FL, USA: American Statistical Association IMS - JSM, 3849-3863.
- Dazard J-E. and J. S. Rao (2012). "*Joint Adaptive Mean-Variance Regularization and Variance Stabilization of High Dimensional Data.*" Comput. Statist. Data Anal. 56(7):2317-2333.

See Also

`justvsn` (R package **vsn**) Variance stabilization and calibration for microarray data. `loess` (R package **stats**) Fit a polynomial surface determined by one or more numerical predictors, using local fitting.

Examples

```
## Not run:
    #=====================================================
    # Loading the library and its dependencies
    #=====================================================
    library("MVR")
    library("RColorBrewer")

    #=====================================================
    # MVR package news
    #=====================================================
    MVR.news()

    #=================================================
    # MVR package citation
    #=================================================
    citation("MVR")

    #=====================================================
    # Loading of the Synthetic and Real datasets
    # (see description of datasets)
    #=====================================================
    data("Synthetic", "Real", package="MVR")
    ?Synthetic
    ?Real

    #=====================================================
    # Mean-Variance Regularization (Real dataset)
    # Multi-Group Assumption
    # Assuming unequal variance between groups
    # Without cluster usage
    #=====================================================
    nc.min <- 1
    nc.max <- 30
```

```
probs <- seq(0, 1, 0.01)
n <- 6
GF <- factor(gl(n = 2, k = n/2, len = n),
             ordered = FALSE,
             labels = c("M", "S"))
mvr.obj <- mvr(data = Real,
               block = GF,
               log = FALSE,
               nc.min = nc.min,
               nc.max = nc.max,
               probs = probs,
               B = 100,
               parallel = FALSE,
               conf = NULL,
               verbose = TRUE)

#=================================================
# Summary Target Moments Diagnostic Plots (Real dataset)
# Multi-Group Assumption
# Assuming unequal variance between groups
#=================================================
target.diagnostic(obj = mvr.obj,
                  title = "Target Moments Diagnostic Plots
                  (Real - Multi-Group Assumption)",
                  device = NULL,
                  horizontal = FALSE,
                  width = 8.5,
                  height = 6.5)

#=================================================
# Mean-Variance Regularization (Real dataset)
# Single-Group Assumption
# Assuming equal variance between groups
# Without cluster usage
#=================================================
nc.min <- 1
nc.max <- 30
probs <- seq(0, 1, 0.01)
n <- 6
mvr.obj <- mvr(data = Real,
               block = rep(1,n),
               log = FALSE,
               nc.min = nc.min,
               nc.max = nc.max,
               probs = probs,
               B = 100,
               parallel = FALSE,
               conf = NULL,
               verbose = TRUE)

#=================================================
# Summary Target Moments Diagnostic Plots (Real dataset)
# Single-Group Assumption
# Assuming equal variance between groups
#=================================================
target.diagnostic(obj = mvr.obj,
                  title = "Target Moments Diagnostic Plots
```

```
                        (Real - Single-Group Assumption)",
                        device = NULL,
                        horizontal = FALSE,
                        width = 8.5,
                        height = 6.5)

## End(Not run)
```

Index