Package 'r.jive'

October 14, 2022

Type Package

Title Perform JIVE Decomposition for Multi-Source Data

Version 2.4

Date 2020-11-11

Author Michael J. O'Connell [aut, cre],
Eric F. Lock [aut],
Adam Kaplan [ctb]

Maintainer Michael J. O'Connell <oconnemj@miamioh.edu>

Description Performs the Joint and Individual Variation Explained (JIVE) decomposition on a list of data sets when the data share a dimension, returning low-rank matrices that capture the joint and individual structure of the data [O'Connell, MJ and Lock, EF (2016) <doi:10.1093/bioinformatics/btw324>]. It provides two methods of rank selection when the rank is unknown, a permutation test and a Bayesian Information Criterion (BIC) selection algorithm. Also included in the package are three plotting functions for visualizing the variance attributed to each data source: a bar plot that shows the percentages of the variability attributable to joint and individual structure, a heatmap that shows the structure of the variability, and principal component plots.

License GPL-3

Imports gplots, abind, graphics, stats

Suggests knitr, rmarkdown

Depends R(>= 2.10.0)

VignetteBuilder knitr

NeedsCompilation no

Repository CRAN

Date/Publication 2020-11-17 08:10:08 UTC

R topics documented:

---

r.jive-package   *Perform JIVE Decompositions for Multi-Source Data*

Description

Performs the Joint and Individual Variation Explained (JIVE) decompositions on a list of data sets when the data share a dimension, returning low-rank matrices that capture the joint and individual structure of the data. It provides two methods of rank selection when the rank is unknown, a permutation test and a Bayesian Information Criterion (BIC) selection algorithm. Also included in the package are three plotting functions for visualizing the variance attributed to each data source: a bar plot that shows the percentages of the variability attributable to joint and individual structure, a heatmap that shows the structure of the variability, and principal component plots.

Details

Package:	r.jive
Type:	Package
Version:	2.4
Date:	2020-11-11
License:	GPL-3

Author(s)

Michael J. O'Connell and Eric F. Lock

Maintainer: Michael J. O'Connell <oconnemj@miamioh.edu>

References

Lock, E. F., Hoadley, K. A., Marron, J. S., & Nobel, A. B. (2013). Joint and individual variation explained (JIVE) for integrated analysis of multiple data types. The Annals of Applied Statistics, 7(1), 523-542.

O'Connell, M. J., & Lock, E.F. (2016). R.JIVE for Exploration of Multi-Source Molecular Data. Bioinformatics advance access: 32(18):2877-2879, 2016.

Examples

```
set.seed(10)
##Load data that were simulated as in Section 2.4 of Lock et al., 2013,
##with rank 1 joint structure, and rank 1 individual structure for each dataset
data(SimData)
# Using default method ("perm")
Results <- jive(SimData)
summary(Results)

# Using BIC rank selection
BIC_result <- jive(SimData, method="bic")
summary(BIC_result)

###Load the permutation results
data(SimResults)
# Visualize results
showVarExplained(Results)
# showVarExplained is also called by the "jive" S3 class default plot method

#show heatmaps
showHeatmaps(Results)

#show PCA plots
showPCA(Results,1,c(1,1))
```

BRCA_Data   *BRCA TCGA Dataset*

Description

These data were obtained from the data freeze for The Cancer Genome Atlas flagship BRCA publication (Cancer Genome Atlas Network, 2013), and processed as described in Lock and Dunson, 2013. Gene expression, methylation, and miRNA data are provided for 348 BRCA tumor samples.

Usage

```
data(BRCA_data)
```

Format

This dataset is a list of three entries for three different molecular sources:

- Data[[1]] (Data$Expression): gene expression matrix for 654 genes (rows) and 348 samples (columns)
- Data[[2]] (Data$Methylation): DNA methylation matrix for 574 cg sites (rows) and 348 samples (columns)
- Data[[3]] (Data$miRNA): miRNA expression matrix for 423 cg sites (rows) and 348 samples (columns).

The 348 columns are shared by the data sources (here, they correspond to tumor samples)

References

Cancer Genome Atlas Network. 2012. "Comprehensive Molecular Portraits of Human Breast Tumours." Nature 490 (7418): 61-70.

Lock, E.F. and Dunson, D.B. 2013. "Bayesian Consensus Clustering." Bioinformatics 29 (20): 2610-16.

O'Connell, M. J., & Lock, E.F. (2016). R.JIVE for Exploration of Multi-Source Molecular Data. Bioinformatics advance access: 10.1093/bioinformatics/btw324.

---

jive   *JIVE Decomposition for Multi-source Data*

Description

Given a list of linked data sets, this algorithm will return low-rank matrices of joint and individual structure. The jive function is a wrapper that centers and scales the data, replaces the missing values using the SVDmiss function if necessary, then proceeds with a specified rank selection method. The jive.iter function performs the joint and individual variation explained (JIVE) decomposition, given ranks and the processed data set. The functions jive.perm and bic.jive perform rank selection using a permutation test and the Bayesian Information Criterion, respectively.

Usage

```
jive(data, rankJ = 1, rankA = rep(1, length(data)), method = "perm",
    dnames = names(data), conv = "default", maxiter = 1000, scale = TRUE, center = TRUE,
      orthIndiv = TRUE, est = TRUE, showProgress=TRUE)

jive.iter(data, rankJ = 1, rankA = rep(1, length(data)), conv = 1e-06,
          maxiter = 1000, orthIndiv = TRUE, showProgress=TRUE)

jive.perm(data, nperms = 100, alpha = 0.05, est = TRUE, conv = 1e-06,
          maxiter = 1000, orthIndiv = TRUE, showProgress=TRUE)

bic.jive(data, n = unlist(lapply(data, ncol)) * unlist(lapply(data, nrow)),
          d = unlist(lapply(data, nrow)), conv = 1e-06, maxiter = 1000,
          orthIndiv = TRUE, showProgress=TRUE)
```

Arguments

`data`	A list of two or more linked data matrices on which to perform the JIVE decomposition. These matrices must have the same column dimension, which is assumed to be common.
`rankJ`	An integer giving the joint rank of the data, if known. If not given, this will be calculated using the chosen method. If the method is "given" then the default is 1.
`rankA`	A vector giving the indvidual ranks of the data, if known. If not given, this will be calculated using the chosen method. If the method is "given" then the default is rep(1, length(data)).
`method`	A string with the method to use for rank selection. Possible options are "given", "perm", and "bic". The default is "perm". If ranks are known, you should use "given".
`dnames`	A vector containing the names of the data sources. Default is names(data).
`conv`	A value indicating the convergence criterion.
`maxiter`	The maximum number of iterations for each instance of the JIVE algorithm.
`scale`	A boolean indicating whether or not the data should be scaled. If TRUE, each data set is divided by its Frobenius norm prior to the JIVE algorithm. Default is TRUE.
`center`	A boolean indicating whether or not the data should be centered. If TRUE, the rows of each data set are mean-centered. Default is TRUE.
`orthIndiv`	A boolean indicating whether or not the algorithm should enforce orthogonality between individual structures. The default is TRUE.
`est`	A boolean indicating whether or not the data should first be compressed via singular value decomposition before running the JIVE algorithm; this will yield identical results, but can improve computational efficiency dramatically for data with more rows than columns. The default is TRUE.
`showProgress`	A boolean indicating whether or not to give output showing the progress of the algorithm. If TRUE, the algorithm will print out updates about the number of iterations the algorithm is taking and the progress of the rank selection method, if applicable. If FALSE, the algorithms will give no printed output when run.
`nperms`	A value indicating the number of permutations for rank estimation. Default is 100.
`alpha`	A value between 0 and 1 giving the quantile to use for rank estimation. Default is .05.
`n`	A vector for the total number of entries in each data source, for use in the BIC calculation. The default is to calculate the total number of entries in each element of data.
`d`	A vector for the total number of variables (rows) in each data source, for use in the BIC calculation. The default is to calculate the number of rows in each element of data.

Details

It is recommended to make all calls to the JIVE functions using the jive() wrapper, as this function does all of the pre-processing of the data (centering, scaling, handling missingness, and reducing the data set to increase computational efficiency). The algorithm will print the number of iterations for each call of the JIVE iteration function.

Value

Returns an object of class jive.

data	a list containing the centered and scaled data sets, with missing values replaced, if applicable.
joint	a list containing matrices that capture the joint structure of the data.
individual	a list containing matrices that capture the individual structure of the data.
rankJ	a value giving joint rank of the data.
rankA	a vector giving the individual ranks of the data.
method	a string denoting the rank selection method used.
bic.table	if bic rank selection used, a matrix that shows the BIC values for different ranks.
converged	if permutation rank selection used, a boolean stating whether or not the rank selection converged within the maximum number of iterations.
scale	A list of rour elements: $Center and $Scale are booleans stating whether the data were centered or scaled, respectively, $'Center Values' gives the value subtracted from each row and $'Scale Values' gives the multiplicative scale factor for each source.

Author(s)

Michael J. O'Connell and Eric F. Lock

References

Lock, E. F., Hoadley, K. A., Marron, J. S., & Nobel, A. B. (2013). Joint and individual variation explained (JIVE) for integrated analysis of multiple data types. The Annals of Applied Statistics, 7(1), 523-542.

O'Connell, M. J., & Lock, E.F. (2016). R.JIVE for Exploration of Multi-Source Molecular Data. Bioinformatics, 32(18):2877-2879, 2016.

Jere, S., Dauwels, J., Asif, M. T., Vie, N. M., Cichocki, A., and Jaillet, P. (2014). Extracting commuting patterns in railway networks through matrix decompositions. In 13th IEEE International Conference on Control, Automation, Robotics and Vision (ICARCV), pages 541-546. IEEE.

See Also

summary.jive, plot.jive

Examples

```
set.seed(10)
##Load data that were simulated as in Section 2.4 of Lock et al., 2013,
##with rank 1 joint structure, and rank 1 individual structure for each dataset
data(SimData)
# Using default method ("perm")
Results <- jive(SimData)
summary(Results)

# Using BIC rank selection
#We set the maximum number of iterations allowed to 50 to speed up this example.
#In practice we recommend a higher value, such as the default of 1000,
#to ensure that all results converge.
BIC_result <- jive(SimData,method="bic",maxiter=50)
summary(BIC_result)
```

---

jive.predict	*Predict JIVE scores for new data*

Description

Computes joint and individual variation explained (JIVE) scores for new data via iterative least squares, with fixed loadings given by a previous JIVE analysis.

Usage

```
jive.predict(data.new, jive.output)
```

Arguments

data.new	A list of two or more linked data matrices on which to estimate JIVE scores. These matrices must have the same column dimension N, which is assumed to be common.
jive.output	An object of class "jive", with row dimensions matching those for data.new.

Value

joint.scores	r X N matrix of joint scores
individual.scores	
	List where entry [[i]] gives the r_i X N matrix of individual scores for source i
errors	Vector of the proportion of total variance explained over iterations during estimation
joint.load	d X r matrix of joint loadings
indiv.load	List where entry [[i]] gives the d_i X N matrix of individual laodings for source i

Author(s)

Adam Kaplan

References

Kaplan, A. and Lock, E.F. (2017). Prediction with Dimension Reduction of Multiple Molecular Data Sources for Patient Survival. arXiv:1704.02069, 2017.

See Also

[jive](#)

Examples

```
##Load data that were simulated as in Section 2.4 of Lock et al., 2013,
##with rank 1 joint structure, and rank 1 individual structure for each dataset
data(SimData)
##load JIVE results (using default settings) for simulated data
data(SimResults)
#predict JIVE scores for data (treated as "new data" here)
pred.results <- jive.predict(SimData,Results)
##estimated joint structure is pred.results$joint.load %*% pred.results$joint.scores
##estimated individual structure for source i is
##pred.results$indiv.load[[i]] %*% pred.results$indiv.scores[[i]]
```

---

pjsum   *Calculate Number of Free Parameters for BIC Calculation*

Description

Computes the number of free parameters from the individual structure of the data. Used internally to calculate the BIC for the JIVE decomposition.

Usage

```
pjsum(dim, rank)
```

Arguments

dim	A vector containing the number of rows of each data source.
rank	A vector containing the ranks of the individual structure.

Value

Returns the number of free parameters.

Author(s)

Michael J. O'Connell and Eric F. Lock

Examples

```
pjsum(c(25,50), c(1,2))
```

---

plot.jive   *Create Plots for a JIVE Object*

Description

Three types of plots are available. By default (or type="var"), this creates a bar plot showing the percentage of variability attributable to joint structure, individual structure, and residual variance. With type="heat", it will create a series of heatmaps. With type="pca", it will give principal component plots.

Usage

```
## S3 method for class 'jive'
plot(x, type="var", ...)
```

Arguments

x	An object of class "jive" to be plotted.
type	A string indicating the type of plot. The default, "var", generates a bar plot of the variance explained, "heat" generates a heatmap, and "pca" generates principal component plots.
...	Additional arguments to pass to the specific plotting functions. See documentation for showVarExplained,showHeatmaps, and showPCA for more details.

Author(s)

Michael J. O'Connell and Eric F. Lock

References

Lock, E. F., Hoadley, K. A., Marron, J. S., & Nobel, A. B. (2013). Joint and individual variation explained (JIVE) for integrated analysis of multiple data types. The Annals of Applied Statistics, 7(1), 523-542.

O'Connell, M. J., & Lock, E.F. (2016). R.JIVE for Exploration of Multi-Source Molecular Data. Bioinformatics advance access: 10.1093/bioinformatics/btw324.

See Also

showVarExplained,showHeatmaps,showPCA

Examples

```
##Load JIVE results (using default settings) for simulated data
##as in Section 2.4 of Lock et al., 2013,
##with rank 1 joint structure, and rank 1 individual structure for each dataset
data(SimResults)
# Visualize results
# Bar plot of variation explained
plot(Results)
# Heatmap
plot(Results,type="heat")
# Principal compontents plots
plot(Results,type="pca",1,c(1,1))
```

---

show.image   *Draw a Heatmap from a Matrix*

Description

Given a matrix, this function draws a heatmap. This function is used internally by the showHeatmaps function.

Usage

```
show.image(Image, ylab = "")
```

Arguments

Image	A matrix for which to create the heatmap.
ylab	A string for the y-label of the plot.

Author(s)

Michael J. O'Connell and Eric F. Lock

References

Lock, E. F., Hoadley, K. A., Marron, J. S., & Nobel, A. B. (2013). Joint and individual variation explained (JIVE) for integrated analysis of multiple data types. The Annals of Applied Statistics, 7(1), 523-542.

O'Connell, M. J., & Lock, E.F. (2016). R.JIVE for Exploration of Multi-Source Molecular Data. Bioinformatics advance access: 10.1093/bioinformatics/btw324.

See Also

showHeatmaps, jive, plot.jive

showHeatmaps	*Heatmaps for JIVE Decompositions*

Description

This function draws heatmaps for the components of a JIVE decomposition.

Usage

```
showHeatmaps(result, order_by = 0, show_all = TRUE)
```

Arguments

result
: An object of class "jive".

order_by
: Specifies how to order the rows and columns of the heatmap. If order_by=-1, the matrices are not re-ordered. If order_by=0, orderings are determined by joint structure. Otherwise, order_by gives the number of the individual structure dataset to determine the ordering. In all cases orderings are determined by complete-linkage hiearchichal clustering of Euclidean distances.

show_all
: Specifies whether to show the full decomposition of the data, JIVE estimates, and noise. If show_all=FALSE, only the matrix (or matrices) that determined the column ordering is shown.

Details

The columns correspond to the shared dimension (for example, a common sample set), and the ordering of the columns is the same for all matrices shown.

Author(s)

Michael J. O'Connell and Eric F. Lock

References

Lock, E. F., Hoadley, K. A., Marron, J. S., & Nobel, A. B. (2013). Joint and individual variation explained (JIVE) for integrated analysis of multiple data types. The Annals of Applied Statistics, 7(1), 523-542.

O'Connell, M. J., & Lock, E.F. (2016). R.JIVE for Exploration of Multi-Source Molecular Data. Bioinformatics advance access: 10.1093/bioinformatics/btw324.

See Also

jive, plot.jive, show.image

Examples

```
##Load JIVE results (using default settings) for simulated data
##as in Section 2.4 of Lock et al., 2013,
##with rank 1 joint structure, and rank 1 individual structure for each dataset
data(SimResults)
# Display the heatmaps (may need to fiddle with plot window dimensions for this to appear well)
showHeatmaps(Results)
# Order by first data set individual structure
showHeatmaps(Results, order_by=1)
# Show only joint structure
showHeatmaps(Results, show_all=FALSE)
```

---

showPCA   *Principal Component Plots for JIVE Decomposition*

Description

Display principal component plots of the joint and individual structure of a data set after a joint and individual variation explained (JIVE) decomposition.

Usage

```
showPCA(result, n_joint = 0, n_indiv = rep(0, length(result$data)),
        Colors = "black", pch=1)
```

Arguments

result	An object of class "jive".
n_joint	The number of joint components to plot.
n_indiv	The vector of the number of individual components to plot for each data set.
Colors	The colors of the data points in the plot. Can be a vector specifying a different color for each sample.
pch	Character to use for plotting. Can be a vector specifying a different character for each sample.

Details

This shows the patterns in the column space that maximize variability of joint or individual structure, analogous to principal components. A multi-panel figure with aligned scatterplots for each pair of principal components, across both joint and individual structure, will be generated. Plotted points correspond to shared columns (e.g., samples).

Author(s)

Michael J. O'Connell and Eric F. Lock

References

Lock, E. F., Hoadley, K. A., Marron, J. S., & Nobel, A. B. (2013). Joint and individual variation explained (JIVE) for integrated analysis of multiple data types. The Annals of Applied Statistics, 7(1), 523-542.

See Also

jive, plot.jive

Examples

```
##Load JIVE results (using default settings) for simulated data
##as in Section 2.4 of Lock et al., 2013,
##with rank 1 joint structure, and rank 1 individual structure for each dataset
data(SimResults)
# Visualize results
# Plot the three components, 1 joint and 1 individual from each source
showPCA(Results,1,c(1,1))
###This displays three scatterplots:
#the first joint principal component vs. the first principal component individual to source 1,
#the first joint component vs.  the first component individual to source 2, and
#the first component individual to source 1 vs. the first component individual to source 2.
```

---

showVarExplained   *Display Variance Explained*

Description

Creates a bar plot displaying the variance explained from a joint and individual variation explained (JIVE) decomposition. Shows the percentage of variance attributed to each of joint structure, individual structure, and residual variance.

Usage

```
showVarExplained(result, col = c("grey20", "grey43", "grey65"))
```

Arguments

result	An object of class "jive".
col	A vector for the colors of the bars in the plot.

Author(s)

Michael J. O'Connell and Eric F. Lock

References

Lock, E. F., Hoadley, K. A., Marron, J. S., & Nobel, A. B. (2013). Joint and individual variation explained (JIVE) for integrated analysis of multiple data types. The Annals of Applied Statistics, 7(1), 523-542.

O'Connell, M. J., & Lock, E.F. (2016). R.JIVE for Exploration of Multi-Source Molecular Data. Bioinformatics advance access: 10.1093/bioinformatics/btw324.

See Also

plot.jive

Examples

```
##Load JIVE results (using default settings) for simulated data
##as in Section 2.4 of Lock et al., 2013,
##with rank 1 joint structure, and rank 1 individual structure for each dataset
data(SimResults)
# Visualize results
showVarExplained(Results)
# showVarExplained is also called by the "jive" S3 class default plot method
plot(Results)
```

SimData   *Simulated Dataset*

Description

These data were simulated as described in Section 2.4 of Lock et al., 2013. There are two simulated sources, with rank 1 joint structure and rank 1 structure individual to each source.

Usage

```
data(SimData)
```

Format

This dataset is a list of two entries:

- Data[[1]] (Data$Data1): 50 variables (rows) and 100 samples (columns)
- Data[[2]] (Data$Data2): 50 variables (rows) and 100 samples (columns)

The 100 columns are shared by the sources.

References

Lock, EF, KA Hoadley, JS Marron, and AB Nobel. 2013. "Joint and Individual Variation Explained (JIVE) for Integrated Analysis of Multiple Data Types." The Annals of Applied Statistics 7 (1): 523-42.

O'Connell, M. J., & Lock, E.F. (2016). R.JIVE for Exploration of Multi-Source Molecular Data. Bioinformatics advance access: 10.1093/bioinformatics/btw324.

SimResults   *JIVE Results for Simulated Dataset*

Description

JIVE results for the simulated data SimData, which were simulated as described in Section 2.4 of Lock et al., 2013. There are two simulated sources, with rank 1 joint structure and rank 1 structure individual to each source. These results are obtained by running JIVE with permutation testing to select the ranks, and other defaults.

Usage

```
data(SimResults)
```

Format

Results: an object of class 'jive'.

References

Lock, EF, KA Hoadley, JS Marron, and AB Nobel. 2013. "Joint and Individual Variation Explained (JIVE) for Integrated Analysis of Multiple Data Types." The Annals of Applied Statistics 7 (1): 523-42.

O'Connell, M. J., & Lock, E.F. (2016). R.JIVE for Exploration of Multi-Source Molecular Data. Bioinformatics advance access: 10.1093/bioinformatics/btw324.

---

summary.jive   *Summarize a JIVE Decomposition*

Description

Provides a summary of JIVE output. Displays the method used for rank selection, the chosen ranks, and a table of the proportion of variance attributable to joint structure, individual structure, and residual variance. print.jive only displays the variance table.

Usage

```
## S3 method for class 'jive'
summary(object, ...)

## S3 method for class 'jive'
print(x, ...)
```

Arguments

object	An object of class "jive".
x	An object of class "jive".
...	Additional arguments.

Value

Returns a list.

Method	a string containing the method used for rank selection.
Ranks	the method used for rank selection.
Variance	the method used for rank selection.

Author(s)

Michael J. O'Connell and Eric F. Lock

References

Lock, E. F., Hoadley, K. A., Marron, J. S., & Nobel, A. B. (2013). Joint and individual variation explained (JIVE) for integrated analysis of multiple data types. The Annals of Applied Statistics, 7(1), 523-542.

O'Connell, M. J., & Lock, E.F. (2016). R.JIVE for Exploration of Multi-Source Molecular Data. Bioinformatics advance access: 10.1093/bioinformatics/btw324.

See Also

jive

Examples

```
##Load JIVE results (using default settings) for simulated data
##as in Section 2.4 of Lock et al., 2013,
##with rank 1 joint structure, and rank 1 individual structure for each dataset
data(SimResults)

# Summary method
summary(Results)

# Print method
Results
```

---

svdmiss   *Missing Data SVD*

Description

This function and description borrowed from R package SpatioTemporal (no longer on CRAN),
from authors Paul D. Sampson and Johan Lindstrom. It completes a data matrix using iterative
svd as described in Fuentes et. al. (2006). The function iterates between computing the singular
value decomposition (svd) for the matrix and replacing the missing values by linear regression of
the columns onto the first ncomp svd components. As initial replacement for the missing values
regression on the column averages are used. The function will fail if entire rows and/or columns are
missing from the data matrix.

Usage

```
SVDmiss(X, niter = 25, ncomp = min(4, dim(X)[2]), conv.reldiff = 0.001)
```

Arguments

X	Data matrix, with missing values marked by NA.
niter	Maximum number of iterations to run before exiting, Inf will run until the conv.reldiff criteria is met.
ncomp	Number of SVD components to use in the reconstruction (>0).
conv.reldiff	Assume the iterative procedure has converged when the relative difference between two consecutive iterations is less than conv.reldiff.

Value

A list with the following components:

Xfill	The completed data matrix with missing values replaced by fitting the data to the ncomp most important svd components
svd	The result of svd on the completed data matrix, i.e. svd(Xfill)
status	A vector of status variables: diff, the absolute difference between the two last iterations; rel.diff, the relative difference; n.iter, the number of iterations; and max.iter, the requested maximum number of iterations.

---

svdwrapper   *Wrapper Function to Perform SVD*

Description

Performs SVD on a data matrix using the base svd() function in R, with a workaround to avoid
LAPACK errors. If an SVD of the data matrix gives an error, an SVD of its transpose will be
performed. Used internally when computing the JIVE decomposition. Credit to Art Owen: https:
//stat.ethz.ch/pipermail/r-help/2007-October/143508.html.

Usage

```
svdwrapper(x, nu, nv, verbose=F )
```

Arguments

x	a numeric matrix whos SVD decomposition is to be computed.
nu	the number of left singular vectors to be computed.
nv	the number of right singular vectors to be computed.
verbose	logical. Print error message if needed.

Value

An svd object, as returned by svd(x,nu=nu,nv=nv).

Author(s)

Michael J. O'Connell and Eric F. Lock

Examples

```
x<-matrix(rnorm(100),nrow=10,ncol=10)
SVD = svdwrapper(x,nu=1,nv=1)
```

Index