

Package ‘PDM’

August 17, 2011

Type Package

Title Partition Decoupling

Version 1.0

Date 2011-04-15

Author Rosemary Braun

Maintainer Rosemary Braun <rosemary.braun@gmail.com>

Description This package carries out the analysis described in Braun,Leibon, Pauls, and Rockmore, Partition Decoupling for Multi-gene Analysis of Gene Expression Profiling Data (2011). Briefly, the method uses iterated spectral clustering and “scrubbing” steps to articulate relationships between samples at multiple scales.

License GPL (>= 2)

LazyLoad yes

Depends methods, graphics, stats, mclust

Suggests

Imports stats, graphics

R topics documented:

PDM-package	2
clusters,PDMlayers-method	2
distanceFn	3
fiedlerVec,PDMlayers-method	4
golub1999	4
layerSpectra-class	5
PDM	6
PDMlayers-class	7
plot	8
spectra,PDMlayers-method	9
spectrum,PDMlayers-method	10
Index	11

PDM-package

Partition Decoupling Method (PDM)

Description

This package carries out the analysis described in Braun, Leibon, Pauls, and Rockmore, Partition Decoupling for Multi-gene Analysis of Gene Expression Profiling Data (2011). Briefly, the method uses iterated spectral clustering and “scrubbing” steps to articulate relationships between samples at multiple scales.

Details

The main workhorse method, `PDM`, carries out the iterated spectral clustering and scrubbing steps, returning a `PDMLayers` object with the clustering output from each layer.

Author(s)

Rosemary Braun <rosemary.braun@gmail.com>

References

Braun, Leibon, Pauls, and Rockmore (2011). Partition Decoupling for Multi-gene Analysis of Gene Expression Profiling Data. URL <http://arxiv.org/abs/1002.3946>

See Also

`PDM` for the main workhorse method.

Examples

```
# To run see the vignett and run the demo,
## Not run:
library(PDM)
vignette("PDM")
demo(PDM)

## End(Not run)
```

clusters,PDMLayers-method
clusters

Description

Extract clusters from a `PDMLayers` object

Usage

```
## S4 method for signature 'PDMLayers'
clusters(object)
```

Arguments

object a [PDMlayers](#) object containing PDM output

Value

A data frame containing the cluster assignment for every sample (rows) in each layer (columns).

Examples

```
data(golub1999)
pdm.golub <- PDM(golub1999$expr, 2)
clusters(pdm.golub)[1:10, ]
table(clusters(pdm.golub)$layer.1, golub1999$pheno)
table(clusters(pdm.golub)$layer.2, golub1999$pheno)
table(paste("l1.", clusters(pdm.golub)$layer.1, ".l2.", clusters(pdm.golub)$layer.2), golub
```

distanceFn

Distance Functions

Description

Functions to compute pairwise distances of columns in the supplied data, scaled to range on [0,1]. Two are given; one may define ones own to pass as an argument to PDM.

Usage

```
euclideanDist(x)
correlationDist(x)
```

Arguments

x A numeric matrix.

Details

`euclideanDist` gives the pairwise Euclidean distance for the columns of `x`, scaled to range on [0,1]. `correlationDist` computes the (Pearson) correlations between all columns in `x` using pairwise-complete observations, then converts the correlations `r` to a cord distance on a unit sphere according to $d = \sin(\arccos(r)/2)$.

Please note that if you intend to supply a user-defined distance function to [PDM](#), it should operate column-wise, and return values scaled to (0,1).

Value

A square matrix of size `ncol(x)` populated by the pairwise distances between each column in `x`.

fiedlerVec, PDMLayers-method
Fiedler Vector

Description

Extract Fiedler vector or complete embedding from a PDMLayers or layerSpectra object

Usage

```
## S4 method for signature 'PDMLayers'
fiedlerVec(object, layer)
## S4 method for signature 'layerSpectra'
fiedlerVec(object)
## S4 method for signature 'PDMLayers'
embedding(object, layer)
## S4 method for signature 'layerSpectra'
embedding(object)
```

Arguments

object A [PDMLayers](#) object containing PDM output, or a [layerSpectra](#) object.
layer Optionally, an integer indicating the desired layer (defaults to last).

Value

For `fiedlerVec`, A vector with the Fiedler vector (coordinates along the first nontrivial Laplacian eigenvector) for the given [PDMLayers](#) layer or [layerSpectra](#). For `embedding`, a matrix containing the embedded data after the non-linear dimension reduction (Laplacian eigenmap), with samples in rows and Laplacian eigenmap dimensions in columns.

Examples

```
data(golub1999)
pdm.golub <- PDM(golub1999$expr, 2)
# Fiedler vector from layer 1:
fiedlerVec(pdm.golub, 1)[1:10]
# Fiedler vector from last (here, 2nd) layer:
fiedlerVec(pdm.golub)[1:10]
```

golub1999 *Gene expression data from Golub et al, 1999*

Description

Gene expression and phenotype data from Golub et al, 1999

Usage

```
data(golub1999)
```

Format

A list containing a matrix (`golub1999$expr`) with gene expression values for 999 genes for 38 leukemia patients and a vector (`golub1999$pheno`) of their leukemia phenotypes (`golub1999$pheno`) for exploring the functionality of the package.

References

Molecular classification of cancer: class discovery and class prediction by gene expression monitoring. Golub, TR and Slonim, DK and Tamayo, P and Huard, C and Gaasenbeek, M and Mesirov, JP and Coller, H and Loh, ML and Downing, JR and Caligiuri, MA and Bloomfield, CD and Lander, ES. Science 1999, 286:531-537

Examples

```
data(golub1999)
dim(golub1999$expr)
table(golub1999$pheno)
```

layerSpectra-class *"layerSpectra" class and methods for PDM output*

Description

This class holds PDM layer spectral data, including the embedded data used, the spectrum (eigenvalues), the 0.05 quantile of resampled eigenvalues, and the 0.95 quantile of resampled eigengaps. It is primarily stored for diagnostic plots, rather than for the user.

Slots

embedding: Matrix containing the embedded data, ie, the Laplacian eigenvectors corresponding to the significant eigenvalues, ordered from most to least "coarse"; column [,2] is the Fiedler vector.

spectra: The Laplacian eigenvalues, ordered from smallest (most significant) to largest; index [2] is the Fiedler value, while [1] is the trivial 0-eigenvalue.

refSpectrum: The 0.05 quantile of resampled eigenvalues.

refGaps: The 0.95 quantile of resampled eigengaps.

Methods

spectrum : extracts the spectrum, dropping the trivial 0-eigenvalue.

embedding : extracts the complete embedded data, dropping the trivial first vector.

fiedlerVec :extracts the Fiedler vector from the embedded data.

plot signature(x = "layerSpectra"): Plots the spectrum and spectral gaps against the resampled values. Note, this will generate two plots, so set `par(ask=T)` to be able to see the first if calling `plot` on a `layerSpectra` directly; alternatively, use `plot(PDMLayers, layer)` for the appropriate layer, which will provide the same spectral plots in two panels.

PDM

*PDM***Description**

Function for computing the PDM layers

Usage

```
PDM(data, max.Layers=4, params)
```

Arguments

<code>data</code>	A matrix or data frame containing the samples (columns) to be clustered. Alternatively, a <code>PDMlayers</code> object output from a previous PDM run may be used; in this case, the scrubbed data will be clustered.
<code>max.Layers</code>	Maximum number of layers to return; default is 4. (PDM may return fewer; see details below).
<code>params</code>	A <code>list</code> containing settable parameters for the spectral clustering algorithm; see details below.

Details

The PDM function carries out iterated spectral clustering and “scrubbing” (projection of data onto centroids and returning the residuals) steps, yielding “layers” of clusters. The procedure is carried out until one of three criteria is met: either `maxLayers` is reached; the scrubbed (residual) data has spectral characteristics indistinguishable from noise (“partition failure”); or the centroids are not independent (“projection failure”).

The parameters for the spectral clustering procedure may be set by supplying them as a list to argument `params`. The settable parameters are:

distanceFn A function (named or anonymous) for calculating the pairwise distances between all columns in `data`. For user-supplied distance functions, note that distances should be scaled to lie on $[0,1]$. Refer to `distanceFn` for examples and two supplied functions. Default is `correlationDist`.

sigma The scaling parameter for the Gaussian transformation. Should be non-negative; 0 yields untransformed data. Default is `sigma=1`.

resample.runs Number of resamplings used to generate the reference distribution of the graph spectra. Default is `resample.runs=40`.

resample.by The reference distribution of the graph spectra may be obtained either through resampling the distances ("`distances`") or the covariates/rows in the original data ("`rows`"); default is "`distances`".

compare How should the threshold for significant eigenvalues be set? "`fiedgap`" compares the spectral gaps ($\lambda_i - \lambda_{i-1}$) to the 0.95 percentile gap of the Fiedler value in the resampled data; "`gaps`" compares the spectral gaps to the corresponding 0.95 percentile of gaps in the resampled data; "`fiedval`" compares the spectra (λ_i) to the 0.05 percentile Fiedler value (λ_1) in the resampled data, and "`vals`" compares the spectra to the 0.05 percentile spectra in the resampled data. Default is "`fiedgap`".

k Numeric number of clusters or the method to automatically compute it: "BIC" fits Gaussian mixture models to the Fiedler vector and returns the number of components with best BIC; "NsigEigVals" takes as the number of clusters the number of significant eigenvalues obtained from the resampling. Default is "BIC".

l Dimensions for the embedding or the method to automatically compute dimensions: "sigEigvals" uses the vectors with significant eigenvalues obtained from the resampling; "k" takes the first \$k\$ vectors, where \$k\$ is the number of clusters (either provided or automatically determined as above); "double.k" takes the first \$2k\$ vectors. If `l` is numeric, the first \$l\$ vectors are used. Default is "sigEigvals".

scrubOnProjectionFailure Boolean: return scrubbed data even if cluster centroids are not independent? Default FALSE.

Value

A `PDMlayers` object with at most `maxLayers` layers containing the clustering output. See `PDMlayers`.

Examples

```
data(golub1999)
pdm.golub <- PDM(golub1999$expr)
pdm.golub
## Not run: plot(pdm.golub,pch=golub1999$pheno)

# force 3 clusters and set sigma=0.5 through params, returning at most 1 layer:
pdm.golub.k3 <- PDM(golub1999$expr,1, params=list(sigma=0.5,k=3))
## Not run: plot(pdm.golub.k3,pch=golub1999$pheno)
```

PDMlayers-class *"PDMlayers" class and methods for PDM output*

Description

This class holds PDM output data, including the clustering results for each layer, the spectra and embedding for each layer, the PDM parameters used, and the current scrubbed data.

Slots

clusters: Data frame of cluster assignments: one column per layer, one row per sample.

params: The parameters fed to PDM for the spectral clustering; see `PDM`.

PDMspectra: A list of `layerSpectra` objects containing the spectral data for each layer.

scrubbed: A numeric matrix containing the "scrubbed" data after the most recent layer (may be NA; see `PDM`).

Methods

show `signature(object = "PDMlayers")`: shows a summary of the `PDMlayers` object, including the number of layers, number of samples, cluster assignments for first ten samples, and slot names.

clusters: extracts a data frame with the cluster assignments from each layer: one column per layer, one row per sample.

spectra: extracts the `layerSpectra` object containing the spectral data for the specified layer. If no layer is specified, the last layer is used.

spectrum: extracts the spectrum (eigenvalues; 0-eigenvalue dropped) for the specified layer. If no layer is specified, the last layer is used.

fiedlerVec: extracts the Fiedler vector for the specified layer. If no layer is specified, the last layer is used.

embedding: extracts the complete dimension-reduced Laplacian eigenmap embedding for the specified layer. If no layer is specified, the last layer is used.

plot: Plots the spectrum, spectral gaps, and fiedler vector for the y -th layer. See `plot` for details.

Examples

```
data(golub1999)
pdm.golub <- PDM(golub1999$expr)
pdm.golub
```

plot

Plot PDMlayers output

Description

Plot spectra, Fiedler vector, and clustering for PDM layers

Usage

```
## S4 method for signature 'PDMlayers,numeric'
plot(x,y,Nvals=40,main,...)
## S4 method for signature 'PDMlayers'
plot(x,Nvals=40,main,...)
```

Arguments

<code>x</code>	A <code>PDMlayers</code> object output from <code>PDM</code> .
<code>y</code>	Optionally, the number of the layer to plot. If <code>y</code> is not specified, each layer will be plotted.
<code>Nvals</code>	Most number of eigenvalues/eigengaps to plot? Default is maximum of 40.
<code>main</code>	An optional character string for the main title.
<code>...</code>	Additional arguments to <code>plot</code> ; see details.

Details

This function attempts to produce a three-panel plot for a PDM layer:

Top panel: Spectrum (eigenvalues) plotted as circles (up to the `Nvals`'th one and omitting the trivial 0-valued first eigenvalue); 0.05 quantile of the resampled null distributions for the eigenvalues plotted as green dashes; 0.05 quantile of the resampled null distributions for the Fiedler value (first non-0 eigenvalue) given as a cyan line. Fill of circles denotes significance WRT the null Fiedler value; color denotes significance WRT the null spectrum.

Middle panel: Spectral gaps plotted as circles (up to the `Nvals`'th one, and omitting the trivial first one); 0.95 quantile of the resampled null distributions for the eigengaps plotted as green dashes; 0.95 quantile of the resampled null distributions for the Fiedler gap given as a cyan line. Fill of circles denotes significance WRT the null Fiedler gap; color denotes significance WRT the null eigengaps.

Bottom panel: Fiedler vector values (y-axis) vs. sample index for each sample, along with the Fiedler vector distribution shown as a marginal plot on the right. Color of the samples indicates the cluster assignment for that sample in the layer shown. The marginal plot is useful as a diagnostic of the number a clusters; the number of components that best fit this distribution is taken as the number of clusters in PDM.

The `...` argument may be used to supply additional arguments to the plot for the **bottom left** Fiedler vector plot, eg, `pch`; see [plot](#) for more information. However, note that you may not override the colors with `col` in this function! (To plot the spectrum or Fiedler vector in a more flexibly user-defined way, use [fiedlerVec](#) to obtain the Fiedler vector and [spectra/spectrum](#) to extract the spectra.)

The current graphics panel after the plot is finished will be the one on the bottom left, permitting the addition of lines or labels as desired by the user.

Examples

```
data(golub1999)
pdm.golub <- PDM(golub1999$expr)
## Not run:
# default plots for two layers
plot(pdm.golub,main="Golub1999 PDM")
# plot of first layer, with pch given by sample phenotype
# A = AML, B = B-cell ALL, T = T-cell ALL
plot(pdm.golub,1,main="Golub1999 PDM",pch=golub$pheno)
# drop vertical lines separating phenotypes
# first find the indices where the phenotypes change:
separateAt <- which(golub1999$pheno[-length(golub1999$pheno)]!=golub1999$pheno[-1])
abline(v=separateAt+0.5,col="grey")

## End(Not run)
```

spectra,PDMlayers-method
spectra

Description

Extract spectral graph data from a `PDMlayers` object

Usage

```
## S4 method for signature 'PDMlayers'
spectra(object, layer)
```

Arguments

`object` a `PDMlayers` object containing PDM output.
`layer` an integer indicating the desired layer (defaults to last).

Value

A `layerSpectra` object containing the spectral data for the layer-th layer.

spectrum, PDMLayers-method
Graph Spectrum

Description

Extract graph spectrum from a PDMLayers or layerSpectra object

Usage

```
## S4 method for signature 'PDMLayers'
spectrum(object, layer)
## S4 method for signature 'layerSpectra'
spectrum(object)
```

Arguments

`object` A `PDMLayers` object containing PDM output, or a `layerSpectra` object.
`layer` Optionally, an integer indicating the desired layer (defaults to last).

Value

A vector with the spectrum (eigenvalues) for the given `PDMLayers` layer or `layerSpectra`, ordered from smallest (most significant) to largest. The trivial 0-valued eigenvector is dropped.

Examples

```
data(golub1999)
pdm.golub <- PDM(golub1999$expr, 2)
# first 10 eignevals in layer 1
spectrum(pdm.golub, 1)[1:10]
# first 10 eignevals in last (here, 2nd) layer
spectrum(pdm.golub)[1:10]
```

Index

- *Topic **array**
 - distanceFn, 3
 - fiedlerVec, PDMLayers-method, 4
 - spectra, PDMLayers-method, 9
 - spectrum, PDMLayers-method, 10
- *Topic **classes**
 - layerSpectra-class, 5
 - PDMLayers-class, 7
- *Topic **cluster**
 - clusters, PDMLayers-method, 2
 - PDM, 6
- *Topic **datasets**
 - golub1999, 4
- *Topic **graphics**
 - plot, 8
- *Topic **package**
 - PDM-package, 2
- clusters, 7
- clusters
 - (clusters, PDMLayers-method), 2
- clusters, PDMLayers-method, 2
- correlationDist, 6
- correlationDist (distanceFn), 3
- distanceFn, 3, 6
- embedding, 5, 8
- embedding
 - (fiedlerVec, PDMLayers-method), 4
- embedding, layerSpectra-method
 - (fiedlerVec, PDMLayers-method), 4
- embedding, PDMLayers-method
 - (fiedlerVec, PDMLayers-method), 4
- euclideanDist (distanceFn), 3
- fiedlerVec, 5, 8, 9
- fiedlerVec
 - (fiedlerVec, PDMLayers-method), 4
- fiedlerVec, layerSpectra-method
 - (fiedlerVec, PDMLayers-method), 4
- fiedlerVec, PDMLayers-method, 4
- golub1999, 4
- layerSpectra, 4, 7, 8, 10
- layerSpectra-class, 5
- list, 6
- PDM, 2, 3, 6, 7, 8
- PDM, data.frame-method (PDM), 6
- PDM, matrix-method (PDM), 6
- PDM, PDMLayers-method (PDM), 6
- PDM-package, 2
- PDMLayers, 2–10
- PDMLayers (PDMLayers-class), 7
- PDMLayers-class, 7
- plot, 5, 8, 8, 9
- plot, layerSpectra, ANY-method
 - (layerSpectra-class), 5
- plot, layerSpectra-method
 - (layerSpectra-class), 5
- plot, PDMLayers, ANY-method (plot), 8
- plot, PDMLayers, numeric-method
 - (plot), 8
- plot, PDMLayers-method (plot), 8
- show, PDMLayers-method
 - (PDMLayers-class), 7
- spectra, 8, 9
- spectra
 - (spectra, PDMLayers-method), 9
- spectra, PDMLayers-method, 9
- spectrum, 5, 8, 9
- spectrum
 - (spectrum, PDMLayers-method), 10
- spectrum, layerSpectra-method
 - (spectrum, PDMLayers-method), 10
- spectrum, PDMLayers-method, 10