

Package ‘GuidedClustering’

August 17, 2011

Type Package

Title A method for the combined analysis of a microarray data set and experimental data

Version 0.9.0

Date 2011-03-15

Author Matthias Maneck, Rainer Spang

Maintainer Matthias Maneck <Matthias.Maneck@klinik.uni-regensburg.de>

Description Guided clustering is a method for the combined analysis of a (clinical) microarray data set and experimental data. The experimental data may be a second microarray data set representing a perturbation experiment containing a perturbed and control group. Alternatively it may be any kind of data that gives gene wise information about the experimental response.

License GPL-3

LazyLoad yes

Depends R (>= 2.12.0), mvtnorm

R topics documented:

GuidedClustering-package	2
getNoiseMatrix	2
getSimulatedData	4
guidedClustering	5
sigmaP	7
Index	9

GuidedClustering-package

A method for the combined analysis of a microarray data set and experimental data

Description

Guided clustering is a method for the combined analysis of a (clinical) microarray data set and experimental data. The experimental data may be a second microarray data set representing a perturbation experiment containing a perturbed and control group. Alternatively it may be any kind of data that gives gene wise information about the experimental response.

Details

Package:	GuidedClustering
Type:	Package
Version:	1.0
Date:	2011-03-15
License:	GPL-3
LazyLoad:	yes
Depends:	R (>= 2.12.0), mvtnorm

Author(s)

Matthias Maneck, Rainer Spang

Maintainer: Who to complain to <yourfault@somewhere.net> Matthias Maneck <Matthias.Maneck@klinik.uni-regensburg.de>

References

Maneck, M., Schrader, A., Kube, D. and Spang, R., Genomic data integration using guided clustering. to be published

getNoiseMatrix

A function that generates a noise matrix for the simulation procedure of GuidedClustering.

Description

Generates matrix of multivariate normal distributed noise using a block structured covariance matrix as proposed by Guo et al. (2007). The functions generates alternating correlated and anticorrelated blocks of genes.

Usage

```
getNoiseMatrix(nr.genes, nr.coreg, p, noise.pat, noise.cell, nr.samples,  
               nr.cell)
```

Arguments

<code>nr.genes</code>	The number of genes (rows) of the resulting noise matrix.
<code>nr.coreg</code>	The number of genes in a block of correlated genes. This determines the block-size in the covariance matrix.
<code>p</code>	The basic correlation between two genes.
<code>noise.pat</code>	The signal to noise ratio for the patient (tumor) samples (columns).
<code>noise.cell</code>	The signal to noise ratio for the cell line (guiding) samples (columns).
<code>nr.samples</code>	The number of patient (tumor) samples.
<code>nr.cell</code>	The number of cell line (guiding) samples in each group. Since two groups are generated (perturbed and control) $2 * \text{nr.cell}$ samples are generated.

Details

The covariance matrix contains $(\text{nr.genes} / \text{nr.coreg})$ blocks of correlated genes whereat correlation and anticorrelation alternates. The 1st block contains correlated genes, the 2nd block anticorrelated genes and so on. The correlation strength is determined by the parameter p . In a consecutive ordering the covariance between a gene i and the other genes k of a correlated block will be $p^{(li-kl)}$, where i and k are gene indices. In a block of anticorrelated genes the covariance will be $(-p)^{(li-kl)}$.

Value

A matrix containing multivariate normal distributed noise.

Author(s)

Matthias Maneck

References

Guo, Y., Hastie, T. and Tibshirani, R. (2007). Regularized linear discriminant analysis and its application in microarrays. *Biostatistics*, 8(1), 86-100.

See Also

[sigmaP](#), [rmvnorm](#)

Examples

```
## Generate noise matrix for 80 patient 20 control and 20 perturbed  
## samples with 1500 genes in block of 300. The signal-to-noise ratio  
## is 1.5 for the patients and 2 for the cell lines  
noise <- getNoiseMatrix(nr.genes=1500, nr.coreg=300, p=0.9, noise.pat=1.5,  
                        noise.cell=2, nr.samples=80, nr.cell=20)
```

getSimulatedData	<i>The function generates simulated data suitable for the Guided Clustering method.</i>
------------------	-----------------------------------------------------------------------------------------

Description

Generates simulated data suitable for the Guided Clustering method.

Usage

```
getSimulatedData(snr.pat, snr.cell, overlap, block.size, nr.samples,
                 nr.cell, nr.genes, nr.coreg, p)
```

Arguments

snr.pat	The signal-to-noise ratio for the patient (tumor) samples.
snr.cell	The signal-to-noise ratio for the cell line (guiding) samples.
overlap	A numeric vector that specifies for each simulated effect cluster the number of genes that are differentially expressed between the cell lines.
block.size	A numeric vector specifying the number of genes per cluster in the patient data and differentially expressed in the guiding data.
nr.samples	The number of samples in tumor data.
nr.cell	Number of samples in each group (control, perturbed) of the cell line data.
nr.genes	Total number of genes in the dataset.
nr.coreg	Number of genes in a block of coregulated genes.
p	Basic correlation between genes within a block of coregulated genes.

Details

We simulate artificial data that mimics the application of guided clustering in the context of pathway activation prediction via guiding by perturbation experiments. The data consist of an artificial clinical (tumor) data set T and a guiding data set G consisting control and perturbed samples. Both data sets are generated by adding a signal component and a noise component: $T_{ij} = F_{ij} + w_T * e_{ij}$ $G_{ij} = I_{ij} + w_G * e_{ij}$ The F_{ij} and I_{ij} simulate the target signals, while the noise components e_{ij} simulate measurements fluctuations and biological variability not related to the target signal. The noise component is simulated for both data sets using a multivariate normal distribution with a block structured covariance matrix following Guo et al. (2007). The tuning parameters w_T and w_G are used to calibrate the signal-to-noise ratio. For T_{ij} we generate signals in clusters E_1, E_2, \dots, E_k representing different biological activities in a sample. The signal F_{ij} is constructed in analogy to the additive model. For each gene in a cluster we draw a random number a_i uniformly from the intervals $[0, 1]$, which represents the strength with which the gene responds to pathway activation. Moreover, for every sample we draw a uniformly distributed random number b_j from $[-1, 1]$ which represents the strength of the pathway activation in this sample. F_{ij} is then set to $a_i + b_j$. Note that the clusters mimic different biological activities. Hence, for a fixed sample the index b_j is constant through genes from the same cluster but not for genes from different clusters. For genes that do not fall in any of the clusters F_{ij} is set to zero. The simulation of the guiding data G_{ij} includes a set B_d responding genes. The number of genes in B_d is equal to the number of clusters times the block.size. These genes are simulated differently for control and perturbation samples. For each of them we draw a random number c_i uniformly from $[0, 1]$ and set $I_{ij} = -c_i$

for control samples and $I_{ij} = c_i$ for perturbation samples. For the remaining genes, we set I_{ij} to zero. The size of the intersection of the clusters E_i with the set of responding genes is specified by the user. Signal-to-noise ratios are not constant over genes but we can use the tuning parameters w_T and w_G to calibrate the max signal-to-noise ratio: $R = \max_i(\text{rmsd}(F_i) / \text{rmsd}(e_i))$ where the maximum is taken over all genes and $\text{rmsd}(x) = \sqrt{1/n \sum (x_j - \bar{x})^2}$, and $j=1..n$ where n is the number of samples in T and \bar{x} is the mean of x .

Value

<code>n.matrix</code>	The noise matrix e_{ij} .
<code>e.matrix</code>	The matrix of simulated effects b_i .
<code>o.matrix</code>	The matrix of gene offsets a_i .
<code>expr</code>	The simulated expression matrix $\text{expr} = n.\text{matrix} + e.\text{matrix} + o.\text{matrix}$.
<code>gene.order</code>	A order so that the genes can sorted overlaps first, then non overlap but cell line induced (B_d), then non overlap but with simulated effect, then other genes.
<code>correct.spit</code>	A matrix of the simulated effects b_i .

Author(s)

Matthias Maneck

References

Guo, Y., Hastie, T. and Tibshirani, R. (2007). Regularized linear discriminant analysis and its application in microarrays. *Biostatistics*, 8(1), 86-100.

See Also

[getNoiseMatrix](#), [sigmaP](#), [rmvnorm](#)

Examples

```
## Simulate a data set consisting of 80 patient, 20 control and
## 20 perturbed samples and 1500 genes. The data set contains 3
## clusters of size 200, that overlap with different overlaps
## with the set of responding genes in the guiding data.
simulated.data <- getSimulatedData(snr.pat=1.5, snr.cell=2,
                                   overlap=c(200,100,50), block.size=200,
                                   nr.samples=80, nr.cell=20, nr.genes=1500,
                                   nr.cores=100, p=0.9)
```

<code>guidedClustering</code>	<i>A function for the combined analysis of a microarray data set and experimental data.</i>
-------------------------------	---------------------------------------------------------------------------------------------

Description

Guided clustering is a method for the combined analysis of a (clinical) microarray data set and experimental data. The experimental data may be a second microarray data set representing a perturbation experiment containing a perturbed and control group. Alternatively it may be any kind of data that gives gene wise information about the experimental response.

Usage

```
guidedClustering(data.t, data.g, sigma, ncluster, label.g = NULL,
                 weight = seq(0, 1, 0.1), output.path = NULL)
```

Arguments

<code>data.t</code>	A numeric expression matrix of tumor data with genes as rows and samples as columns.
<code>data.g</code>	A numeric expression matrix of guiding data with genes as rows and samples as columns, or a numeric vector representing the response to any type of perturbation.
<code>sigma</code>	A numeric vector (might have length 1) specifying the global smoothing strength.
<code>ncluster</code>	Number of clusters to extract.
<code>label.g</code>	A numeric vector specifying the groups of <code>data.g</code> (only necessary if <code>data.g</code> is a matrix), 0 - control, 1 - perturbed group.
<code>weight</code>	A numeric vector specifying the weightings between <code>data.t</code> and <code>data.g</code> evaluated by the algorithm, values must be between 0 and 1.
<code>output.path</code>	The directory where weighting plots are stored. If set to NULL no plots are produced.

Details

The function analyses a microarray gene expression data set together with guiding data that represents a gene wise response to any kind of perturbation. Details may be found in the paper referenced below.

Value

<code>effects</code>	A numeric array of estimated PAIs, 1st dimension represents different settings of sigma, 2nd dimension represents the clusters and 3rd dimension represents the samples.
<code>cluster</code>	A list of list of genes 1st dimension represents different settings of sigma, 2nd dimension represents the clusters.
<code>genes</code>	A list of lists of lists of genes, 1st dimension represents different settings of sigma, 2nd dimension represents the clusters, 3rd dimension represents different weighting settings.
<code>activation.g</code>	A numeric vector of perturbation strength per gene calculated and used by the method.
<code>sign.g</code>	A numeric vector of 1s and -1s indicating whether genes were multiplied by -1 or not.

Author(s)

Matthias Maneck

References

Maneck, M., Schrader, A., Kube, D. and Spang, R., Genomic data integration using guided clustering. to be published

Examples

```
## Simulate a data set consisting of 80 patient, 20 control and
## 20 perturbet samples and 1500 genes. The data set contains 3
## clusters of size 200, that overlap with different overlaps
## with the set of responding genes in the guiding data.
simulated.data <- getSimulatedData(snr.pat=1.5, snr.cell=2,
                                   overlap=c(200,100,50), block.size=200,
                                   nr.samples=80, nr.cell=20, nr.genes=1500,
                                   nr.coreg=100, p=0.9)

## run the guiding cluster algorithm to extract the 5 clusters
## with different setting for the scaling parameter sigma
result <- guidedClustering(data.t=simulated.data$expr[,seq(80)],
                           data.g=simulated.data$expr[,seq(81,120)],
                           label.g=rep(c(0,1),c(20,20)), sigma=c(0.23, 0.20),
                           ncluster=3)

## calculated correlation between the 3 simulated and estimated
## effects
apply(simulated.data$correct.split[,1:80], 1, function(x)
      max(cor(x, t(result$effects[1,,]))) )

## plot perturbation response of cluster genes vs non-cluster
## genes
cluster.label <- factor(rep(0, nrow(simulated.data$expr)),
                        levels=seq(0,3))
for (i in seq(3))
  cluster.label[result$cluster[[2]][[i]]] <- i
x11(type="Xlib")
boxplot(result$activation~cluster.label)
```

sigmaP

A function that generates the covariance matrix for a block of coregulated genes.

Description

Generates a covariance matrix for a block of coregulated genes.

Usage

```
sigmaP(p, nr.genes)
```

Arguments

p	The basic correlation between a pair of genes.
nr.genes	The number of genes in the covariance matrix.

Details

Generates a covariance matrix as described in Guo et al. (2007).

Value

The covariance matrix.

Author(s)

Matthias Maneck

References

Guo, Y., Hastie, T. and Tibshirani, R. (2007). Regularized linear discriminant analysis and its application in microarrays. *Biostatistics*, 8(1), 86-100.

Examples

```
## generate a covariance matrix for 100 genes with basic  
## correlation of 0.9  
cov.matrix <- sigmaP(p=0.9, nr.genes=100)
```

Index

*Topic **cluster**

guidedClustering, [5](#)

*Topic **datagen**

getNoiseMatrix, [2](#)

getSimulatedData, [4](#)

sigmaP, [7](#)

*Topic **package**

GuidedClustering-package, [2](#)

getNoiseMatrix, [2](#), [5](#)

getSimulatedData, [4](#)

GuidedClustering
(*GuidedClustering-package*),
[2](#)

guidedClustering, [5](#)

GuidedClustering-package, [2](#)

rmvnorm, [3](#), [5](#)

sigmaP, [3](#), [5](#), [7](#)