Package ‘backpay’

September 5, 2018

Type Package

Title Identify 'pre-defined' expression PAtterns in transcriptomic/proteomic data

Version 1.0

Date 2018-09-05

Author Thierry Chekouo

Maintainer Thierry Chekouo <thierry.chekouotekou@ucalgary.ca>

Description The backpay package implements the BACkPAy (BAyesian mixture model for identifying Clusters of features (e.g. proteins) with similar 'pre-defined' expression PAtterns) algorithm.

SystemRequirements GSL (GNU Scientific Library)

License GPL (>= 2.0)

Imports scatterplot3d,AUC

RoxygenNote 6.1.0

Encoding UTF-8

R topics documented:

backpay-package *Identify 'pre-defined' expression PAtterns in transcriptomic/proteomic data*

Description

The backpay package implements the BACkPAy (BAyesian mixture model for identifying Clusters of features (e.g. proteins) with similar 'pre-defined' expression PAtterns) algorithm.

Author(s)

Thierry Chekouo

Maintainer: Thierry Chekouo <thierry.chekouotekou@ucalgary.ca>

References

Thierry Chekouo et al (2018), *Investigating Protein Patterns in Human Leukemia Cell Line Experiments: A Bayesian Approach for Extremely Small Sample Sizes*, submitted.

See Also

`BAckPay`, `Generate`

BAckPay *BAyesian mixture model for identifying Clusters of features (e.g. proteins) with similar "pre-defined" expression PAtterns.*

Description

Calculate (i) the marginal posterior probabilities of inclusions for each feature, and (ii) estimated false discovery rate for detecting differential features.

Usage

```
BAckPay(data = data, Ind.Var = Ind.Var, Expe.Var = NULL, sample = 10000,
burnin = 1000, a.tau = a.tau, b.tau = b.tau, c.h = 0.5, b.beta = 0.1)
```

Arguments

`data` Continuous ($\log_2$-) expression data of size $p \times n$, with p the number of features (e.g proteins or genes) and n the number of samples

`Ind.Var` Independent explanatory (categorial) variable. We aim to group features based on the change (or no change) in expression between the modalities of this variable. This argument must be of type "factor".

`Expe.Var` Experimental explanatory (categorial) variable. It's the "confounding" variable. The patterns are compared for every modality of Expe.Var. This argument must be of type "factor".

`sample` Total number of MCMC draws. It must be larger than burnin.

`burnin` Number of draws to discard for burn-in

`a.tau` Shape of $\tau_{hl} \sim \text{Gamma}(a_\tau, b_\tau)$, the truncated parameter of the independent variable effects β_{hl}.

`b.tau` Rate hyperparameters of $\tau_{hl} \sim \text{Gamma}(a_\tau, b_\tau)$, the truncated parameter of the independent variable effects β_{hl}. We have $E(\beta_{hl}) \geq a_\tau / b_\tau$. The choice of these hyperparameters is guided by the experimenter, who would need to decide on a threshold for biological significance. Note that $E(\beta_{hl})$ can be interpreted as the $\log_2$ fold change if the feature expression is $\log_2$ transformed.

`c.h` Variance hyperparameter of the random effect a_{jh}, that captures positive correlations between samples of the same type.

`b.beta` Variance hyperparameter of the effects β_{hl}'s. This parameter is involved in capturing positive correlations between features within the same cluster.

Details

The function will return several R objects, which can be assigned to a variable. To see the results, use the “$" operator.

Value

`probDiff` a numeric vector of the probability of differential expression for each feature in the data set.

`q.valueDiff` Estimated “q values" of detecting differential expression.

`rhoMean` Estimated marginal posterior probabilities of cluster memberships, ρ_{jk}, $P(\rho_{jk} = h)$, where $k = 1, ..., T$, $h = 1, ..., H$, $H = 3^{q-1}$ the number of clusters, q the number of modalities of *Ind.Var*, and T is the number of modalities of the experimental variable (*Expe.Var*) . It's an array of size $T \times p \times H$.

`ProbProtGrp` Estimated (joint) marginal posterior probabilities of proteins for patterns or groups (jMPP). It's a matrix of size $q \times p$ where $q = H^T = 3^{(q-1)*T}$ is the total number of patterns or groups. For instance, if *Ind.Var* has two modalities: Resistant and Sensitive, and the experimental variable has also two modalities: Time 0h and 48h. Hence group *Up-Up* is the pattern or group of features that go up from Resistance to Sensitive for both times 0h and 48h.

See Also

`Generate`

Examples

```
library(backpay)
##---- Simulate data
Gen=Generate(NbrModCov=2,NbrGps=3,p=8000,nbrDuplic=1,seed=1)
data=Gen$data
IndVar=as.factor(Gen$Ind.Var);
ExpVar=as.factor(Gen$Expe.Var);
Result=BAckPay(data=data,Ind.Var=IndVar,Expe.Var=ExpVar, sample=10000,burnin=5000, a.tau=8,b.tau=10);
round(Result$probDiff[1:10],digits=4)
dim(Result$rhoMean)
round(head(Result$rhoMean[1,,]),digits=2)
dim(Result$ProbProtGrp)
print(round(head(Result$ProbProtGrp[,1:10]),digits=2))
```

```
library(AUC)
KnownRho=Gen$RhoKnown ## true clustering memberships
H=3^(length(unique(IndVar))-1);
rhodiffTrue=1-apply(KnownRho[,,(H+1)/2],2,prod)
AUC=auc(roc(Result$probDiff, as.factor(rhodiffTrue)))
AUC
```

clust *Internal function: List possible clusters along with the signs of the effect of each covariate.*

Description

Provide all possible clusters

Usage

```
clust(VV=c(1,0,-1), nbrcov=nbrcov)
```

Arguments

VV	Possible signs of the effects in each cluster.
nbrcov	Number of covariates included in the model. It's NbrModCov-1 where NbrModCov is the number of modalities of the independent variable.

Value

It gives a matrix of dimension $H \times 3$, where $H = 3^{nbrcov}$.

Examples

```
clust(nbrcov=2)
```

DataOrga *Internal Function: Organize the data to plot in 3D.*

Description

It returns averaged data over the replicates.

Usage

```
  DataOrga(Proba=Proba,thres=thres,data=data, Expe.Var =Expe.Var,varcovlist=varcovlist)
```

Arguments

`Proba` Joint marginal probabilities (see paper) to plot features

`thres` Threshold used on joint marginal probabilities (see paper) to plot features

`data` Continuous expression data of size $p \times n$, with p the number of features (e.g proteins or genes) and n the number of samples.

`Expe.Var` Experimental explanatory (categorial) variable. It's the "confounding" variable. The patterns are compared for every modality of Expe.Var.

`varcovlist` Independent variable for each modality of the experimental variable. This is obtained using function *datNorm*.

Value

`DataAveraged` Data averaged over the replicates.

`names` Feature names of the new data set.

See Also

BAckPay

---

datNorm *Internal function: Mean-centering data for each modality of the experimental variable*

Description

This function mean-centers features for each modality of the experimental variable.

Usage

```
datNorm(data=data,Expe.Var=NULL,Ind.Var=Ind.Var)
```

Arguments

`data` Continuous ($\log_2$-) expression data of size $p \times n$, with p the number of features (e.g proteins or genes) and n the number of samples

`Ind.Var` Independent explanatory (categorial) variable. We aim to group features based on the change (or no change) in expression between the modalities of this variable.

`Expe.Var` Experimental explanatory (categorial) variable. It's the "confounding" variable. The patterns are compared for every modality of Expe.Var.

Value

`y` Mean-centered data matrix $p \times n$.

`covlist` Independent variable for each modality of the experimental variable.

`Generate` *Generate independent features (e.g proteins or genes) with* NbrModCov *and* NbrGps *modalities of the independent and experimental variables respectively (see the reference for more details).*

Description

Generate simulated data as esplained in the reference.

Usage

```
Generate(NbrModCov = 2, NbrGps = 3, p = p, nbrDuplic = 2, bmin = 0.5, bmax = 1,
smin = 0.2, smax = 0.2, seed = seed)
```

Arguments

`NbrModCov` This is the number of modalities of the independent variable (q =NbrModCov)

`NbrGps` This is the number of modalities of the experimental variable (T =NbrGps)

`p` number of features (e.g. proteins or genes).

`nbrDuplic` number of duplicates for each combination *Ind.Var/Expe.Var*. For instance, if *nbrDuplic=1*, only one sample is available for each combination *Ind.Var/Expe.Var*.

`bmin` Minimum of the coefficients effects β, which are generated as Uniform(bmin,bmax).

`bmax` Maximum of the coefficients effects β, which are generated as Uniform(bmin,bmax).

`smin` Minimum and maximum of the error variances of proteins in cluster, σ, which are generated as Uniform(smin,smax).

`smax` Maximum of the error variances of proteins in cluster, σ, which are generated as Uniform(smin,smax).

`seed` seed number for generating random numbers.

Value

`data` Expression data of size $p \times n$, with p the number of features (e.g proteins or genes) and n the number of samples ($n = nbrDuplic * NbrModCov * NbrGps$).

`Ind.Var` A vector of size n with modalities $0, 1, ..., NbrModCov - 1$.

`Expe.Var` A vector of size n with modalities $0, 1, ..., NbrGps - 1$.

`KnownRho` A binary array of dimension $T \times p \times H$, of known (true) cluster memberships where KnownRho$[j, k, h] = 1$ if $\rho_{jk} = h$ and 0 otherwise.

References

Thierry Chekouo et al (2018), *Investigating Protein Patterns in Human Leukemia Cell Line Experiments: A Bayesian Approach for Extremely Small Sample Sizes*, *submitted.*

Examples

```
Gen=Generate(NbrModCov=2,NbrGps=3,p=8000,nbrDuplic=1,seed=1)
dim(Gen$data)
round(head(Gen$data),digits=2)
IndVar=as.factor(Gen$Ind.Var);
ExpVar=as.factor(Gen$Expe.Var);
IndVar
ExpVar
```

`Nameclust` *Internal function: Define the names of the obtained cluster patterns from BAckPAy.*

Description

List all the cluster pattern names (or groups) with respect to the number of modalities of both the independent and experimental variable.

Usage

```
Nameclust(NbrModCov, NbrGps)
```

Arguments

`NbrModCov`	Number of modalities of the independent variable.
`NbrGps`	Number of modalities of the experimental variable (T =NbrGps)

Value

`namecl` Cluster names with respect of the independent variable. For instance, if the independent variable has 3 modalities, then cluster *UpDown* contains features that are up from modality 1 from 2, and down from 2 to 3.

`namegroup` Pattern (or groups) names obtained with combinations of both the independent and experimental variables. If both variables have 2 modalities, then the group pattern *DownFlat-UpUp* contains features that are DownFlat and UpUp for the first and second modality of the experimental variable respectively.

Examples

```
Res=Nameclust(NbrModCov=3, NbrGps=2)
#List of cluster names with 3 modalities from the indep. variable
Res$namecl
#List of all cluster pattern (or groups) names with 3 and 2 modalities
#from the indep. and experimental variables respectively.
Res$namegroup
```

`PlotThreeD` *Plot 3D of patterns/groups*

Description

In addition to the 3D plot, the function also returns (i) the list of the top features with their respective joint marginal probabilities, and (ii) the expression data used to make the plot.

Usage

```
PlotThreeD(data=data,Ind.Var = Ind.Var, Expe.Var = NULL,ProbProtGrp=ProbProtGrp,
patternname=patternname,thres=thres)
```

Arguments

`data` Continuous ($\log_2$-) expression data of size $p \times n$, with p the number of features (e.g proteins or genes) and n the number of samples. It should be the same dataset used in the function *BAckPAy*.

`Ind.Var` Independent explanatory (categorial) variable. We aim to group features based on the change (or no change) in expression between the modalities of this variable. It should be the same *Ind.Var* used in the function *BAckPAy*.

`Expe.Var` Experimental explanatory (categorial) variable. It's the "confounding" variable. The patterns are compared for every modality of Expe.Var. It should be the same *Expe.Var* used in the function *BAckPAy*.

`ProbProtGrp` Estimated (joint) marginal posterior probabilities of proteins for patterns or groups (jMPP). It's a matrix of size $q \times p$ where $q = H^T$ is the total number of patterns or groups. This is obtained from the BAckPay function.

`patternname` Pattern name to plot

`thres` Threshold used on joint marginal probabilities (see paper) to plot features

Value

`ProbSort` a numeric vector of the higher joint marginal posterior probabilities obtained with a threshold of *thres*.

`data` data used to make the plot; centered data of features belonging to pattern *patternname*

See Also

`BAckPay`

Examples

```
library(backpay)
##---- Simulate data
Gen=Generate(NbrModCov=2,NbrGps=3,p=2000,nbrDuplic=1,seed=1)
data=Gen$data
IndVar=as.factor(Gen$Ind.Var);
ExpVar=as.factor(Gen$Expe.Var);
Result=BAckPay(data=data,Ind.Var=IndVar,Expe.Var=ExpVar, sample=10000,burnin=5000, a.tau=8,b.tau=10);
```

```
round(Result$probDiff[1:10],digits=4)
dim(Result$rhoMean)
round(head(Result$rhoMean[1,,]),digits=2)
Names=Nameclust(NbrModCov=2, NbrGps=3)
Names$namegroup[1]
PlotResult= PlotThreeD(data=data,Ind.Var = IndVar, Expe.Var = ExpVar,ProbProtGrp=Result$ProbProtGrp,
patternname=Names$namegroup[1],thres=0.5)
```

ThreeDplot1 *Internal function: Preparing to represent data in 3D*

Description

Intermediate function to represent patterns.

Usage

```
ThreeDplot1(meanF=meanF,nbrMark=nbrMark,NbrGps=NbrGps,NbrModCov=NbrModCov,
LabelsLegend=LabelsLegend,LabelConf=LabelConf,patternName=patternName,miny=miny,
maxy=maxy,thres=thres,maxprob=maxprob)
```

Arguments

meanF	Average expression data over the number of replicates for each modality of the experimental variable.
nbrMark	Number of features to plot
NbrGps	This is the number of modalities of the experimental variable ($T =$NbrGps)
NbrModCov	This is the number of modalities of the independent variable ($q =$NbrModCov)
LabelsLegend	Label names of the independent variable
LabelConf	Label names of the experimental variable
patternName	Pattern name to plot
miny	Minimium value of *meanF*
maxy	Maximum value of *meanF*
thres	Threshold used on joint marginal probabilities (see reference) to plot features
maxprob	The maximum of joint marginal probabilities (see reference) for pattern name *patternName*.

Index