Package 'CORM'

July 2, 2014

Type Package

Version 1.0.2

License GPL (>= 2)

Description We proposed a new model-based clustering method, called the clustering of regression models method(CORM), which groups genes that share a similar relationship to the covariate(s). This method provides a unified approach for a family of clustering procedures and can be applied to data collected with various experimental designs. This package includes the implementation for two such clustering procedures: (1) the Clustering of Linear Models (CLM) method, and (2) the Clustering of Linear Mixed Models (CLMM) method.

Title The Clustering of Regression Models Method

Depends R (>= 2.10.0), cluster, limma

Suggests MASS

URL http://www.r-project.org,http://www.mskcc.org/mskcc/html/60448.cfm

Author Li-Xuan Qin [aut],Jiejun Shi [cre]

Maintainer Jiejun Shi <shi.abraham.2010@gmail.com>

NeedsCompilation no

Repository CRAN

Date/Publication 2014-06-30 10:50:49

R topics documented:

`BreastCancer` *Breast Cancer Data Set*

Description

Zhao et al. (2004) studied gene expression profiles of two types of breast cancer, invasive ductal carcinoma (IDC) and invasive lobular carcinoma (ILC). They analyzed the expression profiles of 38 IDC samples and 21 ILC samples, using cDNA arrays spotted for 42,000 clones. With the significance analysis of microarrays (SAM) method (Efron et al., 2001), they identified a total of 474 clones that were differentially expressed between IDCs and ILCs, representing 354 unique genes. This example data set contains the normalized expression value of these 354 genes for the 59 samples.

Usage

```
data(BreastCancer)
```

Format

1. A list comprised of two components: **normalizedData** and **designMatrix**.
2. **normalizedData** is a matrix containing the normalized breast cancer data, whose row names are gene IDs and column names indicate two subtypes of breast cancer (IDC vs. ILC).
3. **designMatrix** is the covariates matrix used to fit the clustering of linear models (CLM), whose row names are samples and column names are covariates.

References

Zhao et al. (2004). Different gene expression patterns in invasive lobular and ductal carcinomas of the breast. *Molecular Biology of the Cell*,15, 2523-2536

`CORM` *Clustering of Regression Models Method*

Description

The clustering of regression models method models the expression level of each gene with regression and clusters genes based on the regression coefficients.

Details

Package:	CORM
Type:	Package
Version:	1.0.2
Date:	2014-6-22
License:	GPL(>=2)

.

Author(s)

Li-Xuan Qin <qinl@mskcc.org>

References

- Li-Xuan Qin and Steven G. Self (2006). The clustering of regression models method with applications in gene expression data. *Biometrics* 62, 526-533.
- Li-Xuan Qin (2008). An integrative analysis of microRNA and mRNA expression - a case study. *Cancer Informatics* 6, 369-379.
- Li-Xuan Qin, Linda Breeden and Steven G. Self (2014). Finding gene cluster for a replicated time course study. *BMC Res Notes* 7:60.

See Also

`fit.CLM`, `fit.CLMM`, `fit.CLMM.2`

`fit.CLM` *Clustering of Linear Models Method*

Description

Fit a CLM model for cross-sectional data.

Usage

```
fit.CLM(data.y, data.x, n.clst, n.start = 1)
```

Arguments

`data.y`	matrix of gene expression data, data.y[j, i] for sample i and gene j.
`data.x`	matrix of sample covariates, data.x[i, p] for sample i and covariate p.
`n.clst`	an integer, number of clusters .
`n.start`	an integer used to get the starting value for the EM algorithm.

Details

This function implements the Clustering of Linear Models Method of Qin and Self (2006). This method clusters genes based on the estimated regression parameters that model the relation between gene expression and sample covariates.

Value

u.hat	a matrix containing the cluster membership probability for each gene, whose row names are genes and column names are clusters.
theta.hat	a list comprised of four components: *zeta.hat*, *pi.hat*, *sigma2.hat*, *llh*. They are described as below:
zeta.hat	a matrix with the estimated regression parameters with one row for each cluster.
pi.hat	a vector with the relative frequency for each cluster.
sigma2.hat	a vector of variance parameters.
llh	log likelihood for the model.

Author(s)

Li-Xuan Qin <qinl@mskcc.org>

References

- Li-Xuan Qin and Steven G. Self (2006). The clustering of regression models method with applications in gene expression data. *Biometrics* 62, 526-533.
- Li-Xuan Qin (2008). An integrative analysis of microRNA and mRNA expression - a case study. *Cancer Informatics* 6, 369-379.

See Also

fit.CLM, fit.CLMM, fit.CLMM.2

Examples

```
#Example 1
 #test data
  data(BreastCancer)
  data.y <- BreastCancer$normalizedData
  data.x <- BreastCancer$designMatrix
 #fit the model
  n.clst <- 9
  fit1   <- fit.CLM(data.y, data.x, n.clst)
  fit1.u <- apply(fit1$u.hat, MARGIN=1, FUN=order, decreasing=TRUE)[1,]
 #display the results
  index.IDC <- which(data.x[,2]==0)
  index.ILC <- which(data.x[,2]==1)
  mean.IDC  <- apply(data.y[,index.IDC], MARGIN=1, FUN=mean, na.rm=TRUE)
  mean.ILC  <- apply(data.y[,index.ILC], MARGIN=1, FUN=mean, na.rm=TRUE)

  color  <- rainbow(n.clst)
  par(mai=c(1,1,0.5,0.1),cex.axis=0.8, cex.lab=1,mgp=c(1.5,0.5,0))
  plot((mean.IDC+mean.ILC)/2,
       (mean.IDC-mean.ILC),
       xlab="(IDC mean + ILC mean)/2",
       ylab="IDC mean - ILC mean",
       pch=paste(fit1.u),
```

```
      col=color[fit1.u],
      main=paste("K=",n.clst))

## Not run:
#Example 2
 #test data
  data(miRTargetGenes)
  data.y <- miRTargetGenes$normalizedData
  data.x <- miRTargetGenes$designMatrix
 #fit the model
  n.clst <- 9
  n.start<- 20
  fit2    <- fit.CLM(data.y, data.x, n.clst, n.start)
  fit2.u   <- apply(fit2$u.hat, MARGIN=1, FUN=order, decreasing=TRUE)[1,]
  fit2.u.o <- factor(fit2.u, levels=c(1,5,6,7,4,8,2,9,3), labels=1:9)
  library(limma)
 plot.y  <- lmFit(data.y, data.x)$coef %*% cbind(c(1,0,0,0),c(1,0,1,0),c(1,1,0,0),c(1,1,1,1))
  plot.x   <- 1:4
 #display the results
  color  <- rainbow(n.clst)
  par(mfrow=c(3,4),mai=c(0.35, 0.4, 0.4, 0.2), mgp=c(1.6,0.4,0), tck=-0.01, las=2)
  for(k in 1:n.clst){
   plot(plot.x, plot.y[1,], type="n", xaxt="n", ylim=range(plot.y),
        xlab="", ylab="gene expression")
   axis(1, plot.x, c("Normal \n","Normal \n +miRNA","Tumor \n","Tumor \n +miRNA"),
        las=1, cex.axis=1, mgp=c(1.5,1.2,0))
   title(paste("cluster", k))
   abline(h=0, lty=2)
   for(j in which(fit2.u.o==k)) points(plot.x, plot.y[j,], type="b", col=color[k])
  }

## End(Not run)
```

`fit.CLMM` *Clustering of Linear Mixed Models Method*

Description

Fit a CLMM model for time course data (with or without replicates). If replicated time courses, all replicates should be measured at the same time points. Missing data are allowed.

Usage

```
fit.CLMM(data.y, data.x, data.z, n.clst, n.start = 1)
```

Arguments

`data.y`	a three dimensional array of gene expression data, data.y[j, i, t] for gene j, sample i, and time point t. Missing values should be indicated by "NA". And at least one case not missing in each pair of observations.

data.x	a three dimensional array of fixed effects (common for all genes), data.x[i, t, p] for sample i, time point t, and covariate p.
data.z	a three dimensional array of random effects (common for all genes), data.z[i, t, q] for sample i, time point t, and covariate p.
n.clst	an integer, number of clusters.
n.start	an integer used to get the starting value for the EM algorithm.

Details

This function implements the Clustering of Linear Mixed Models Method of Qin and Self (2006).

Value

u.hat	a matrix containing the cluster membership for the genes.
b.hat	an array containing the estimated random effects.
theta.hat	a list comprised of five components: *zeta.hat*, *pi.hat*, *D.hat*, *sigma2.hat* and *llh*. They are described as below:
zeta.hat	a matrix of the estimated fixed effects with one row for each cluster.
pi.hat	a vector of the relative frequency for each cluster.
D.hat	the estimated random effects variances for the clusters.
sigma2.hat	the estimated measurement error variances for the clusters.
llh	log likelihood for the model.

Author(s)

Li-Xuan Qin <qinl@mskcc.org>

References

- Li-Xuan Qin and Steven G. Self (2006). The clustering of regression models method with applications in gene expression data. *Biometrics* 62, 526-533.
- Li-Xuan Qin, Linda Breeden and Steven G. Self (2014). Finding gene clusters for a replicated time course study. *BMC Res Notes* 7:60.

See Also

fit.CLM, fit.CLMM, fit.CLMM.2

Examples

```
#Example 1
#test data
  data(YeastCellCycle)
  data.y <- YeastCellCycle$normalizedData.sample
  data.x <- YeastCellCycle$designMatrix
#fit the model
  n.clst <- 6
```

```
  fit1 <- fit.CLMM(data.y, data.x, data.x, n.clst)

  fit1.u <- apply(fit1$u.hat, MARGIN=1, FUN=order, decreasing=TRUE)[1,]
  zeta.fitted <- fit1$theta.hat$zeta.hat
  profile <- data.x[1,,] %*% t(zeta.fitted)
#display the profile of each cluster
  n.knots <- 7
  plot.x <- n.knots*(1:dim(data.y)[3]-1)

  par(mfrow=c(2, ceiling((n.clst)/2)),mai=c(0.5,0.5,0.5,0.1),mgp=c(1,0.3,0))
  for(k in 1:n.clst){
  # plot the fitted cluster-specific profiles
    plot(plot.x,profile[,k],type="l",
         ylim=c(-2,2), main=paste("Cluster",k),
         xlab="time (min)", ylab=NA,xaxt="n",lwd=2)
    axis(side=1, at=plot.x[(1:8)*2-1], labels=paste(plot.x[(1:8)*2-1]), cex.axis=0.8)
  # plot the observed profiles for genes in this cluster
    i.plot <- (1:dim(data.y)[1])[fit1.u==k]
    for(j in i.plot) { lines(plot.x, data.y[j,1,], lty=3, lwd=1)}
    text(84,-1.9, paste(length(which(fit1.u==k)),"genes"))
  }

## Not run:
#Example 2
#test data
  data(YeastCellCycle2)
  data.y <- YeastCellCycle2$normalizedData.WT
  data.x <- YeastCellCycle2$designMatrix.WT
#fit the model
  n.clst <- 8
  fit1   <- fit.CLMM(data.y,data.x[,,1:9],data.x[,,1:9],n.clst)
  fit1.u <- apply(fit1$u.hat, MARGIN=1, FUN=order, decreasing=TRUE)[1,]
  zeta.fitted <- fit1$theta.hat$zeta.hat
  profile.WT <- YeastCellCycle2$designMatrix.WT[1,,1:9] %*% t(zeta.fitted)
#display the profile of each cluster
 # remove bad time points for WTs
  n.time  <- 25
  time.WT <- (1:n.time)[-22]
  n.rpl.WT<- length(time.WT)
  n.gene.short<- dim(data.y)[1]
 # gene-specific profile: observed profiles averaged over replicates
  data.WT.mean  <- matrix(0, nrow=n.gene.short, ncol=n.rpl.WT)
  for(j in 1:n.gene.short){
   data.WT.mean[j,] <- (data.y[j,1,]+data.y[j,2,])/2
  }
 # plot observed profiles by cluster
  col.panel=rainbow(8)
  par(mfrow=c(3, 3),mai=c(0.3,0.25,0.2,0.05))
  for(k in 1:n.clst){
  plot(5*(time.WT-1), profile.WT[,k], type="l", col=col.panel[k], ylim=c(-2,2),
       xlab="Time", ylab="Expression Value", main=paste("WT: cluster",k))
  i.plot  <- (1:n.gene.short)[fit1.u==k]
  for(j in i.plot) lines(5*(time.WT-1), data.WT.mean[j,], lty=1)
```

```
  lines(5*(time.WT-1), profile.WT[,k], col=col.panel[k], lwd=2)
  text(125, -1.9, pos=2, paste(length(i.plot)," genes"))
  }

## End(Not run)
```

`fit.CLMM.2` *Clustering of Linear Mixed Models Method*

Description

Fit a CLMM model for replicated time course data allowing for two sets of time points among biological or technical replicates. Missing value are allowed.

Usage

```
fit.CLMM.2(data.y1,data.x1,data.z1,data.y2,data.x2,data.z2,n.clst, n.run = 1)
```

Arguments

`data.y1`	a three dimensional array of gene expression data, data.y1[j, i, t] for gene j, sample i, and time point t. Missing values should be indicated by "NA". And at least one case not missing in each pair of observations.
`data.x1`	a three dimensional array of fixed effects (common for all genes), data.x1[i, t, p] for sample i, time point t, and covariate p.
`data.z1`	a three dimensional array of random effects (common for all genes), data.z1[i, t, q] for sample i, time point t, and covariate p.
`data.y2`	a three dimensional array of gene expression data, data.y2[j, i, t] for gene j, sample i, and time point t. Missing values should be indicated by "NA". And at least one case not missing in each pair of observations.
`data.x2`	a three dimensional array of fixed effects (common for all genes), data.x2[i, t, p] for sample i, time point t, and covariate p.
`data.z2`	a three dimensional array of random effects (common for all genes), data.z2[i, t, q] for sample i, time point t, and covariate p.
`n.clst`	an integer, number of clusters.
`n.run`	an integer used to get the starting value for the EM algorithm.

Details

This function implements the Clustering of Linear Mixed Models Method of Qin and Self (2006).

Value

u.hat	a matrix containing the cluster membership for the genes.
b.hat	an array containing the estimated random effects.
theta.hat	a list comprised of five components: *zeta.hat*, *pi.hat*, *D.hat*, *sigma2.hat* and *llh*. They are described as below:
zeta.hat	a matrix of the estimated fixed effects with one row for each cluster.
pi.hat	a vector of the relative frequency for each cluster.
D.hat	the estimated random effects variances for the clusters.
sigma2.hat	the estimated measurement error variances for the clusters.
llh	log likelihood for the model.

Author(s)

Li-Xuan Qin <qinl@mskcc.org>

References

- Li-Xuan Qin and Steven G. Self (2006). The clustering of regression models method with applications in gene expression data. *Biometrics* 62, 526-533.
- Li-Xuan Qin, Linda Breeden and Steven G. Self (2014). Finding gene clusters for a replicated time course study. *BMC Res Notes* 7:60.

See Also

fit.CLM, fit.CLMM, fit.CLMM.2

Examples

```
## Not run:
#test data
 data(YeastCellCycle2)
 data.y1 <- YeastCellCycle2$normalizedData.WT
 data.x1 <- YeastCellCycle2$designMatrix.WT
 data.y2 <- YeastCellCycle2$normalizedData.SM
 data.x2 <- YeastCellCycle2$designMatrix.SM
 n.clst  <- 8
 fit1 <- fit.CLMM.2(data.y1=data.y1,data.x1=data.x1,data.z1=data.x1,
                    data.y2=data.y2,data.x2=data.x2,data.z2=data.x2,
  n.clst=n.clst)
 fit1.u  <- apply(fit1$u.hat, MARGIN=1, FUN=order, decreasing=TRUE)[1,]
 zeta.fitted <- fit1$theta.hat$zeta.hat
 profile.WT  <- YeastCellCycle2$designMatrix.WT[1,,] %*% t(zeta.fitted)
 profile.SM  <- YeastCellCycle2$designMatrix.SM[1,,] %*% t(zeta.fitted)
# remove bad time points for WTs and SMs
 n.time  <- 25
 time.WT <- (1:n.time)[-22]
 time.SM <- (1:n.time)[-c(6,9,12)]
 n.rpl.WT<- length(time.WT)
```

```
 n.rpl.SM<- length(time.SM)
 n.gene.short<-dim(YeastCellCycle2$normalizedData.WT)[1]
# gene-specific profile: observed profiles averaged over replicates
 data.WT.mean  <- matrix(0, nrow=n.gene.short, ncol=n.rpl.WT)
 data.SM.mean <- matrix(0, nrow=n.gene.short, ncol=n.rpl.SM)
 for(j in 1:n.gene.short){
  data.WT.mean[j,] <- (YeastCellCycle2$normalizedData.WT[j,1,]+
                       YeastCellCycle2$normalizedData.WT[j,2,])/2
  data.SM.mean[j,] <- (YeastCellCycle2$normalizedData.SM[j,1,]+
                       YeastCellCycle2$normalizedData.SM[j,2,])/2
 }
# plot observed profiles by cluster -- wild type
 col.panel=rainbow(8)
 par(mai=c(0.3,0.25,0.2,0.05),mfrow=c(3,3))
 for(k in 1:n.clst){
  plot(5*(time.WT-1), profile.WT[,k], type="l", col=col.panel[k], ylim=c(-2,2),
       xlab="Time", ylab="Expression Value", main=paste("WT: cluster",k))
  i.plot <- (1:n.gene.short)[fit1.u==k]
  for(j in i.plot) lines(5*(time.WT-1), data.WT.mean[j,], lty=1)
  lines(5*(time.WT-1), profile.WT[,k], col=col.panel[k], lwd=2)
  text(125, -1.9, pos=2, paste(length(i.plot)," genes"))
}
# plot observed profiles by cluster -- single mutant
 par(mai=c(0.3,0.25,0.2,0.05),mfrow=c(3,3))
 for(k in 1:n.clst){
  plot(5*(time.SM-1), profile.SM[,k], type="l", col=col.panel[k], ylim=c(-2,2),
       xlab="Time", ylab="Expression Value", main=paste("SM: cluster",k))
  i.plot <- (1:n.gene.short)[fit1.u==k]
  for(j in i.plot) lines(5*(time.SM-1), data.SM.mean[j,], lty=1)
  lines(5*(time.SM-1), profile.SM[,k], col=col.panel[k], lwd=2)
  text(125, -1.9, pos=2, paste(length(i.plot)," genes"))
}
# plot fitted profiles by cluster
 par(mai=c(0.3,0.25,0.2,0.05),mfrow=c(3,3))
 for(k in 1:n.clst){
  plot(5*(time.WT-1), profile.WT[,k], type="l", ylim=c(-2,2),
       xlab="Time", ylab="Expression Value", lwd=2)
  title(paste("Cluster", k))
  lines(5*(time.SM-1), profile.SM[,k], lty=3, lwd=2)
  if(k==1) legend(60, 2, c("WT", "SM"), lty=1:2, cex=0.8)
 }

## End(Not run)
```

`miRTargetGenes` *miR-let-7f Targets Data Set*

Description

Lu et al. (2005) profiled both miRNA expression and mRNA expression in multiple human cancer types. The miRNA profiles reflected the developmental lineage and differentiation state of the tu-

mours. One of the profiled miRNA is let-7f. Its expression is highly correlated (Pearson correlation < -0.458) with the expression of 178 genes in tumors from 5 cancer types.

Usage

```
data(miRTargetGenes)
```

Format

A list comprised of two components: **normalizedData** and **designMatrix**.

1. **normalizedData** is a matrix containing the normalized data of the miR-let-7f targets, whose row names are gene IDs and column names indicate normal and tumor samples of 5 types of cancer.
2. **designMatrix** is the covariates matrix used to fit the clustering of linear models (CLM), whose row names are samples and column names are covariates.

References

Lu et al. (2005). MicroRNA expression profiles classify human cancers. *Nature*, 435, 834-838

YeastCellCycle *Yeast Cell Cycle Data Set*

Description

This data set contains a subset of yeast cell cycle data taken from Spellman et al. (1998) (See the reference below). Spellman et al. (1998) monitored the genome-wide mRNA levels for 6108 yeast genes at 7-minute intervals for 119 minutes. A total of 256 genes were identified to oscillate significantly in at least two data sets. This example data set contains the log ratios of these 256 genes at the first 16 time points(from 0 min to 105 min).

Usage

```
data(YeastCellCycle)
```

Format

A list comprised of three components: **normalizedData**, **normalizedData.sample** and **designMatrix**.

1. **normalizedData** is a three dimensional array containing the normalized expression data of the 256 genes during yeast cell-cycle.
2. **normalizedData.sample** is a randomly selected sample from **normalizedData**. It only contains 64 genes.
3. **designMatrix** is a three dimensional array used to fit the clustering of linear mixed models (CLMM).

Source

http://genome-www.stanford.edu/cellcycle/data/rawdata/

References

Spellman et al. (1998). Comprehensive identification of cell cycle-regulated genes of the yeast Saccharomyces cerevisiae by microarray hybridization. *Molecular Biology of the Cell*,9, 3273-3297

`YeastCellCycle2` *Yeast Cell Cycle Data Sets of wild type and single mutant*

Description

This data set is taken from a yeast cell cycle study in Qin et al. (2014) (See the reference below). It contains the normalized data of 256 cell cycle dependent genes in wild type(WT) and single mutant(SM) yeast, each mesured over time with two technical replicates.

Usage

```
data(YeastCellCycle2)
```

Format

A list comprised of four components: **normalizedData.WT**, **normalizedData.SM**, **designMatrix.WT**, and **designMatrix.SM**.

1. **normalizedData.WT** and **normalizedData.SM** are three-dimensional arrays containing the normalized expression data for 256 genes in WT and SM yeast, respectively. Missing values are indicated by "NA".
2. **designMatrix.WT** and **designMatrix.SM** are three-dimensional arrays used to fit the clustering of linear mixed models (CLMM).

References

Li-Xuan Qin, Linda Breeden and Steven G. Self (2014). Finding gene clusters for a replicated time course study. *BMC Res Notes* 7:60.

Index