Package ‘PamGeneMixed’

February 15, 2013

Type Package

Title Preprocessing and Modeling Kinase Activity Profiles in PamChip Data

Version 1.02

Date 2012-12-01

Author Pushpike Thalikarathne, Ziv Shkedy and Dan Lin

Maintainer Pushpike Thalikarathne <Pushpike@gmail.com>

Description This package contains the functions that can be used to analyze the PamChip microarray data. The package is mainly useful in fitting semi-parametric mixed effects models while accommodating various types of mean structures and correlation structures present in Kinase activity profiling experiments. The package can be used to visualize, cleaning Pam Chip data. And also it can be used in model fitting, velocity estimating, and testing the estimated velocities between the groups. The PamGeneMixed package is built up as a software implementation for the paper titled The Use of Semi-parametric Mixed Models to Analyze PamChip Peptide Array Data: an Application to an Oncology Experiment which is published in Bioinformatics (2011).

Depends R (>= 2.10), mgcv, gplots, utils, graphics, nlme, methods, stats, Biobase

License GPL-3

LazyLoad yes

Repository CRAN

Date/Publication 2012-12-18 07:12:30

NeedsCompilation no
**AutoPamGeneMix**

**Description**

This is a wrapper function for `PamGeneMix` and can be used to fit peptide-specific semi-parametric mixed effects models to cleaned PamChip microarray data from function `PreProcessAllPeptides`. The structure of the function is more or less the same as `PamGeneMix`. However, it automates the analysis for list of peptides.

Function can handle complex mean structures and complex random effects structures depending on the experimental design of the PamChip arrays. It estimates mean smoothing function and first order derivative, the velocity, of the fitted curve along with point-wise 95% confidence intervals. The estimated group specific velocities are compared and tested at first, end time points as well as for entire velocity profile. However, velocities can be compared at any time point within the available time range. All these tests are performed for each peptide separately.

This function can handle two or more grouping structures in the data. Function returns list of data frames in which p-values are given. And corresponding fitted gamm output objects for each peptide is saved as Rdata file.

For S4 class information, s. `PamChipAllPepResults`.
Usage

AutoPamGeneMix(formula=y~-1+ResTrt+s(time,by=ResTrt,bs="tp",m=3),
Weights=varIdent(form=~1|ResTrt),
Random.structure=list(ArrayNum=~1,CellLineResTrt=~1+time+time2,ID=~1+time+time2),
CleanedPamData,
TestAt=NA,
PathOutPut)

Arguments

formula A GAM formula (see also formula.gam and PamGeneMix). This is like the formula for a glm except that smooth terms (s and te) can be added to the right hand side of the formula. Note that ids for smooths and fixed smoothing parameters are not supported.

Weights In the generalized case, weights with the same meaning as glm weights. An lme type weights argument may only be used in the identity link gaussian case, with no offset (see documentation for lme for details of how to use such an argument)

Random.structure The (optional) random effects structure as specified in a call to lme: only the list form is allowed, to facilitate manipulation of the random effects structure within gamm in order to deal with smooth terms. See example below.

CleanedPamData A data frame with cleaned PamChip array data from PreProcessAllPeptides.

TestAt Time point at which test should be performed, possible value should be 0<t<end time. By default the estimated group specific velocities will be compared and tested at first, end time points as well as for entire velocity profile

PathOutPut Direct path where the output object from PamGeneMix can be saved. If missing objects are saved to current working directory.

Value

A list of objects of class PamChipAllPepResults.

Author(s)

Pushpike Thilakarathne, Ziv Shkedy and Dan Lin

References


See Also

PamGeneMix, PreProcessAllPeptides, PreProcessPam, gamm, lm, gam, glm

Examples

```r
data(PamChipData)

PeptidesNames <- setdiff(colnames(PamChipData),
c("ID", "ResState", "ArrayNum", "CellName", "TreatName","Time"))

#------- only two groups in the data -------------------
temp1 <- PreProcessAllPeptides(pep.names=PeptidesNames[1:5],
PamSig=PamChipData[PamChipData[,c("ResState")]=="R",],PathOutPut="C:/Temp")

# Case I -----------------------------------
# Testing the group specific velocities at t=0,t=end and for entire profile.
Results1 <- AutoPamGeneMix(formula=y~1+ResTrt+s(time,by=ResTrt,bs="tp",m=3),
Weights=varIdent(form=~1|ResTrt),
Random.structure=list(ArrayNum=~1,CellLineResTrt=~1+time+time2,ID=~1+time+time2),
temp1,PathOutPut="C:/FittedGamm")

#---- use show
show(Results1)
# output -
#-------- Results for set of Peptides --------
#Class : PamChipAllPepResults
#No. Peptides : 5
#Total Run time (in seconds) : 133.89
#Number of Groups : 2
#Fitted Gamm Objects are saved as RData Objects in :
# C:/FittedGamm
#-----------------------------------------

#use --
VolcanoPam(Results1,Topp=5,plotting=TRUE)

# Case II-----------------------------------
# Testing the group specific velocities at t=0,t=end, t=30, and for entire profile.
Results2 <- AutoPamGeneMix(formula=y~1+ResTrt+s(time,by=ResTrt,bs="tp",m=3),
Weights=varIdent(form=~1|ResTrt),
Random.structure=list(ArrayNum=~1,CellLineResTrt=~1+time+time2,ID=~1+time+time2),
temp1,PathOutPut="C:/FittedGamm")
```
**CleanedPamData-class**

Random.structure=list(ArrayNum=~1,CellLineResTrt=~1+time+time2,ID=~1+time+time2),
temp1,TestAt=30,PathOutPut="C:/FittedGamm")

#---- use show
#show(Results2)
#VolcanoPam(Results2,Topp=5,plotting=TRUE)

#-------- Four groups in the data ---
# Apply preprocessing steps for each peptide
temp2<-PreProcessAllPeptides(pep.names=PeptidesNames,
PamSig=PamChipData,PathOutPut="C:/Temp")

#-----------------------------------
# Fit the model for each peptide
# This would take approximately 6 hrs to run !!! and depends on your system speed.
ResultsAll<-AutoPamGeneMix(formula=y~-1+ResTrt+s(time,by=ResTrt,bs="tp",m=3),
Weights=varIdent(form=~1|ResTrt),
Random.structure=list(ArrayNum=~1,CellLineResTrt=~1+time+time2,ID=~1+time+time2),
temp2,TestAt=30,PathOutPut="C:/FittedGamm")

---

### Description

Object returned by `PreProcessAllPeptides`.

### Slots

- **PeptideData**: A list of data frames as many as peptides analyzed.
- **np**: Number of peptides analyzed.
- **nc**: Number of cell lines used.
- **path**: Path where the Cleaned Peptide specific RData have been saved.

### Methods

- **show** Use `show(CleanedPamData-object)` for brief information.

### Author(s)

Pushpike Thilakarathne, Ziv Shkedy and Dan Lin

### See Also

- `PreProcessPam`
ObservedCellProfiles

Description
Object returned by `PreProcessPam`.

Slots
Profiles.Set.ID: A data frame of cleaned cell line data of a peptide.
p: Index of the Peptide.
cel: Index of the cell line used.

Methods
show Use `show(CleanedPeptide-object)` for brief information.

Author(s)
Pushpike Thilakarathne, Ziv Shkedy and Dan Lin

See Also
`PreProcessAllPeptides`

ObservedCellProfiles  Observed Cell Line Specific Profiles

Description
This function visualizes the observed Cell line specific profiles for a given peptide of the PamChip microarray data.

Usage
`ObservedCellProfiles(pepname, DataMat, Trt.Group=c("Treatment","Control"), Res, log.true = FALSE)`

Arguments
pepname Name of the peptide to be used in visualization.
DataMat A data frame of PamChip.
Trt.Group Treatment group for which visualization is required. Default is for both “Treatment” and “Control” groups.
Res Indicator for Responsive (R) or non-responsive (NR) group.
log.true Default FALSE and no log2 transformation is performed for intensities.
**ObservedGroupProfiles**

**Value**

none

**Author(s)**

Pushpike Thilakarathne, Ziv Shkedny and Dan Lin

**See Also**

ObservedRepProfiles, ObservedGroupProfiles

**Examples**

```r
#--------
for (i in 1:148) {
  par(ask=TRUE)
  ObservedCellProfiles(pepname=paste("Pep",i,sep=""), DataMat=PamChipData,
                       Trt.Group=c("Treatment","Control"), Res=c("R","NR"), log.true=FALSE)
  cat(i)
}

ObservedCellProfiles(pepname="Pep58", DataMat=PamChipData,
                      Trt.Group=c("Treatment","Control"), Res=c("R","NR"), log.true=FALSE)

ObservedCellProfiles(pepname="Pep9", DataMat=PamChipData,
                      Trt.Group=c("Treatment","Control"), Res=c("R","NR"), log.true=FALSE)

#----------log2 transformed ------------------------------
ObservedCellProfiles(pepname="Pep9", DataMat=PamChipData,
                      Trt.Group=c("Treatment","Control"), Res=c("R","NR"), log.true=TRUE)

ObservedCellProfiles(pepname="Pep26", DataMat=PamChipData,
                      Trt.Group=c("Treatment","Control"), Res=c("R","NR"), log.true=TRUE)
```

**ObservedGroupProfiles**

**Observed Group Specific Profiles**

**Description**

This function estimate and visualize observed average profiles for each group for a given peptide.
ObservedGroupProfiles

Usage

ObservedGroupProfiles(pepname="Pep5",DataMat,
Trt.Group=c("Treatment","Control"),Res="R",log.true=FALSE)

Arguments

pepname Name of the peptide to be used in visualization.
DataMat A data frame of PamChip.
Trt.Group Treatment group for which visualization is required. Default is for both "Treatment" and "Control" groups.
Res Indicator for Responsive (R) or non-responsive (NR) group.
log.true Default FALSE and no log2 transformation is performed for intensities.

Value

none

Author(s)

Pushpike Thilakarathne, Ziv Shkedy and Dan Lin

See Also

ObservedRepProfiles, ObservedCellProfiles

Examples

data(PamChipData)

ObservedGroupProfiles(pepname="Pep142",DataMat=PamChipData,
Trt.Group=c("Treatment","Control"),Res=c("R","NR"),log.true=FALSE)

ObservedGroupProfiles(pepname="Pep142",DataMat=PamChipData,
Trt.Group=c("Treatment","Control"),Res=c("R","NR"),log.true=TRUE)

ObservedGroupProfiles(pepname="Pep1/zero.noslash8",DataMat=PamChipData,
Trt.Group=c("Treatment","Control"),Res=c("R","NR"),log.true=FALSE)

ObservedGroupProfiles(pepname="Pep1/zero.noslash8",DataMat=PamChipData,
Trt.Group=c("Treatment","Control"),Res=c("R","NR"),log.true=TRUE)
ObservedRepProfiles  

Observed replicate specific profiles

Description

Observed replicate specific profiles for a given cell line and peptide.

Usage

ObservedRepProfiles(pepname, DataMat, cells, Trt.Group = c("Treatment", "Control"), Res = "R", log.true = FALSE, cell.mean = T)

Arguments

pepname      Name of the peptide to be used in visualization.
DataMat      A data frame of PamChip.
cells        Name of the cell that needs to be visualized.
Trt.Group    Treatment group for which visualization is required. Default is for both "Treatment" and "Control" groups.
Res          Indicator for Responsive (R) or non-responsive (NR) group.
log.true     Default FALSE and no log2 transformation is performed for intensities.
cell.mean    A boolean parameter and default is TRUE. If TRUE cell line specific mean curve is drawn.

Value

none.

Author(s)

Pushpike Thilakarathne, Ziv Shkediy and Dan Lin

References

#add reference..

See Also

plot, ObservedCellProfiles, ObservedGroupProfiles
Examples

```r
data(PamChipData)

ObservedRepProfiles(pepname="Pep1",DataMat=PamChipData,
cells="SKBR3",Trt.Group=c("Treatment","Control"),Res="R",log.true=FALSE)

ObservedRepProfiles(pepname="Pep3",DataMat=PamChipData,
cells="SKBR3",Trt.Group=c("Treatment","Control"),Res="R",log.true=FALSE)

ObservedRepProfiles(pepname="Pep130",DataMat=PamChipData,
cells="SUM159",Trt.Group="Treatment",Res="R",log.true=FALSE)

ObservedRepProfiles(pepname="Pep130",DataMat=PamChipData,
cells="SUM159",Trt.Group="Control",Res="R",log.true=FALSE)
```

---

PamChipAllPepResults-class

"PamChipAllPepResults"

---

Description

Object returned by `AutoPamGeneMix`.

Slots

- **AllPepRes**: A list of data frames as many as peptides analyzed.
- **RunTime**: Total run time (in seconds) to analyze the all peptides.
- **NoOfPeptides**: Number of peptides analyzed.
- **TestAt**: Additional time point at which group-specific velocities are compared.
- **n.groups**: Number of groups in the data.
- **path**: Path where the Cleaned Peptide specific RData have been saved.

Methods

- **show** Use `show(PamChipAllPepResults-object)` for brief information.

Author(s)

Pushpike Thilakaratne, Ziv Shkedy and Dan Lin

See Also

`PamGeneMix`, `AutoPamGeneMix`
PamChipData

PamChip Data frame where there are columns for ArrayID, Cell line names, Responsive Statuses, Treatment indicator, Time points, and Replicate ID. And there are columns as many as peptides.

Usage

data(PamChipData)

Format

The format is a data frame which has following structure.

<table>
<thead>
<tr>
<th>ID</th>
<th>ResState</th>
<th>ArrayNum</th>
<th>CellName</th>
<th>TreatName</th>
<th>Time</th>
<th>Pep1</th>
<th>Pep2</th>
<th>Pep3</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>NR</td>
<td>0559C80</td>
<td>MDA231PAR</td>
<td>Control</td>
<td>11</td>
<td>2237</td>
<td>2104</td>
<td>287</td>
</tr>
<tr>
<td>1</td>
<td>NR</td>
<td>0559C80</td>
<td>MDA231PAR</td>
<td>Control</td>
<td>16</td>
<td>3746</td>
<td>3696</td>
<td>391</td>
</tr>
<tr>
<td>1</td>
<td>NR</td>
<td>0559C80</td>
<td>MDA231PAR</td>
<td>Control</td>
<td>21</td>
<td>3979</td>
<td>4051</td>
<td>460</td>
</tr>
<tr>
<td>1</td>
<td>NR</td>
<td>0559C80</td>
<td>MDA231PAR</td>
<td>Control</td>
<td>26</td>
<td>4361</td>
<td>4392</td>
<td>394</td>
</tr>
<tr>
<td>1</td>
<td>NR</td>
<td>0559C80</td>
<td>MDA231PAR</td>
<td>Control</td>
<td>31</td>
<td>4218</td>
<td>4452</td>
<td>393</td>
</tr>
<tr>
<td>7</td>
<td>R</td>
<td>0559C80</td>
<td>N87</td>
<td>Control</td>
<td>36</td>
<td>4679</td>
<td>4855</td>
<td>832</td>
</tr>
<tr>
<td>7</td>
<td>R</td>
<td>0559C80</td>
<td>N87</td>
<td>Control</td>
<td>41</td>
<td>4726</td>
<td>4777</td>
<td>920</td>
</tr>
<tr>
<td>7</td>
<td>R</td>
<td>0559C80</td>
<td>N87</td>
<td>Control</td>
<td>46</td>
<td>5072</td>
<td>5072</td>
<td>952</td>
</tr>
<tr>
<td>7</td>
<td>R</td>
<td>0559C80</td>
<td>N87</td>
<td>Control</td>
<td>51</td>
<td>5147</td>
<td>5344</td>
<td>1039</td>
</tr>
<tr>
<td>7</td>
<td>R</td>
<td>0559C80</td>
<td>N87</td>
<td>Control</td>
<td>56</td>
<td>5091</td>
<td>5204</td>
<td>999</td>
</tr>
<tr>
<td>7</td>
<td>R</td>
<td>0559C80</td>
<td>N87</td>
<td>Control</td>
<td>61</td>
<td>5060</td>
<td>5172</td>
<td>1131</td>
</tr>
</tbody>
</table>

Author(s)

Pushpike Thilakarathne, Ziv Shkedy and Dan Lin

References


See Also

TestPepModelData
Examples

```r
#-- load Data --
data(PamChipData)

#-- Check Data --
PamChipData[1:5,1:10]

#-- Unique Time Points --
UTime<-unique(PamChipData[,c("Time")])
UTime

#-- Responsive and Non-Responsive Cell Lines --
RnRcellines<-unique(PamChipData[,c("CellName","ResState")])
rownames(RnRcellines)<-1:20
RnRcellines

#-- Number of Plates (96 well plates)---
Plates<-unique(PamChipData[,c("ArrayNum")])
Plates
```

---

**PamChipMixed-class**  
"PamChipMixed"

---

**Description**

Object returned by the method **PamGeneMix**.

**Slots**

- `res.t1` A vector of length two in which *p-value* for comparing group-specific velocities at t=0 along with test statistic are returned.
- `res.tend` A vector of length two in which *p-value* for comparing group-specific velocities at t=max(t) along with test statistic are returned.
- `res.tany` If test.at is given then a vector of length two in which *p-value* for comparing group-specific velocities at t=tgiven along with test statistic are returned. Otherwise NA is returned.
- `test.at` Time point at which additional test needs to be performed.
- `PValProfile` A vector of size two where first gives *p-value* for comparing group-specific velocities for entire profile and second value gives the Ch-Squared test statistic.
- `t.mesh` Fine grid of time points that has been used to produce smooth curves.
- `End.time` Maximum time point in the Dataset.
- `n.g` Number of groups in the data
- `gnames` Group names.
- `v` Estimated velocity. And it is a vector of size equals to length of the t.mesh times number of groups.
v.sd  Estimated variances at each time point. And it is a vector of size equals to length of the t.mesh times number of groups.
Vall  Full variance covariance matrix of the estimated velocities.
yfitfixef Fitted mean function. Returns as a vector of size equals to length(t.mesh) times n.g
yfit  Estimated BLUP.
gammran3 Fitted gamm object.
PTx  A data frame used to fit the model.

Methods

show Use show(PamChipMixed-object) for brief information.
summary Use summary(PamChipMixed-object) to apply the classic summary() function to the PamChipMixed-object), s. summary,PamChipMixed-method

Author(s)

Pushpike Thilakarathne, Ziv Shkedy and Dan Lin

See Also

PamGeneMix

PamGeneMix Fitting Semi-Parametric Mixed Models to PamChip Data

Description

This is a wrapper function for gamm and can be used to fit semi-parametric mixed effects model to PamChip microarray data. Function can handle complex mean structures and complex random effects structures depending on the experimental design of the PamChip arrays. It estimates mean smoothing function and first order derivative, the velocity, of the fitted curve along with point-wise 95% confidence intervals. The estimated group specific velocities will be compared and tested at first, end time points as well as for entire velocity profile. However, velocities can be compared at any time point within the available time range.

Usage

PamGeneMix(formula,Correlation=NULL,Weights=varIdent(form=~1),
PTx=list(),Random.structure=NULL,
Control.list=list(maxIter=200, msMaxIter=250 ,msMaxEval=1000,apVar=TRUE),
test.at=NA )
Arguments

**formula**  
A GAM formula (see also `formula.gam` and `gam.models`). This is like the formula for a glm except that smooth terms (`s` and `te`) can be added to the right hand side of the formula. Note that ids for smooths and fixed smoothing parameters are not supported.

**Random.structure**  
The (optional) random effects structure as specified in a call to `lme`: only the list form is allowed, to facilitate manipulation of the random effects structure within `gamm` in order to deal with smooth terms. See example below.

**Correlation**  
An optional corStruct object (see `corClasses`) as used to define correlation structures in `lme`. Any grouping factors in the formula for this object are assumed to be nested within any random effect grouping factors, without the need to make this explicit in the formula (this is slightly different to the behaviour of `lme`).

**Weights**  
In the generalized case, weights with the same meaning as `glm` weights. An `lme` type weights argument may only be used in the identity link gaussian case, with no offset (see documentation for `lme` for details of how to use such an argument)

**Control.list**  
A list of fit control parameters for `lme` to replace the defaults returned by `lmeControl`. Note the setting for the number of EM iterations used by `lme`: smooths are set up using custom pdMat classes, which are currently not supported by the EM iteration code. If you supply a list of control values, it is advisable to include niterEM=0, as well, and only increase from 0 if you want to perturb the starting values used in model fitting (usually to worse values!). The optimMethod option is only used if your version of R does not have the nlminb optimizer function.

**PTx**  
A data frame with PamChip array data.

**test.at**  
Time point at which test should be performed, possible value should be 0<t<end time. By default the estimated group specific velocities will be compared and tested at first, end time points as well as for entire velocity profile

Value

A list of objects of class `PamChipMixed`. The results can be visualized using the method `VisualizePamGeneMix`.

Author(s)

Pushpike Thilakarathne, Ziv Shkedy and Dan Lin

References


**See Also**

`VisualizePamGeneMix`, `AutoPamGeneMix`, `testVarCom`, `gamm`, `lme`, `gam`

**Examples**

```r
# Examples of using PamGeneMix
data(TestPepModelData)
head(TestPepModelData)

#---------------- section I -------------------
#In this section, we use only Responsive cell lines data such that we have only two groups: treatment and control for the analysis.
#In this particular model, the cell line-specific random intercept is considered to capture correlation of the intensity measurement over time within the cell line. We assumed cell line-specific random slopes for linear as well as for quadratic time effects to capture different evolution of kinase activity over time. Moreover, we allow these cell lines-specific random structures to be different for each group. We assumed group-specific random structure for cell lines.

PTx<-TestPepModelData

# log2 transform the response
PTx$y<-log2(PTx[,ncol(PTx)])
PTx<-PTx[PTx[,c("ResState")]=="R",]

n.groups<-length(unique(PTx[,c("ResState")]))*length(unique(PTx[,c("TreatName")]))
n.groups

xhlp<-lm(y~1+TreatName,data=PTx,x=TRUE)$x
for (i in 1:n.groups) PTx$ResTrt[xhlp[,i]==1]<-i
PTx$ResTrt<-as.factor(PTx$ResTrt)
levels(PTx$ResTrt)<-colnames(xhlp)

#--unique cell lines
cellLines<-levels(PTx$CellName)
PTx$ResState<-as.factor(PTx$ResState)

#--create interaction between Treatment and cell lines
```
PTx$CellLineResTrt<-0
xhlp<-lm(y~1+CellName:TreatName,data=PTx,x=TRUE)$x
ncols<-ncol(xhlp)
for (i in 1:ncols) PTx$CellLineResTrt[xhlp[,i]==1]<-i
PTx$CellLineResTrt<-as.factor(PTx$CellLineResTrt)
levels(PTx$CellLineResTrt)<-colnames(xhlp)

#------------------ Model I --------------------
M1gamm<-PamGeneMix(formula=y~-1+ResTrt+s(time,by=ResTrt,bs="tp",m=3),
PTx=PTx,Random.structure=list(ArrayNum=~1,CellLineResTrt=~1+time+time2,ID=~1+time+time2),
Control.list=list(maxIter=200, msMaxIter=250 ,msMaxEval=1000,apVar=TRUE))
show(M1gamm)
#output
#-------- Fitted PamGeneMix Model --------
#Class: PamChipMixed
#Note: Model is successfully fitted
#Number of Groups: 2
#P-values for comparing group specific velocities
#Test at t=0 : 3.34599e-12
#Test at t=max(t) : 0.1044876
#Test for entire profile : 0
#-------------------------------

M1gamm30<-PamGeneMix(formula=y~-1+ResTrt+s(time,by=ResTrt,bs="tp",m=3),
PTx=PTx,Random.structure=list(ArrayNum=~1,CellLineResTrt=~1+time+time2,ID=~1+time+time2),
Control.list=list(maxIter=200, msMaxIter=250 ,msMaxEval=1000,apVar=TRUE),test.at=30)
show(M1gamm30)
#output
#-------- Fitted PamGeneMix Model --------
#Class: PamChipMixed
#Note: Model is successfully fitted
#Number of Groups: 2
#P-values for comparing group specific velocities
#Test at t=0 : 3.34599e-12
#Test at t= 30 : 0.12664
#Test at t=max(t) : 0.1044876
#Test for entire profile : 0
#-------------------------------

#------- summary ----------
summary(M1gamm30)

#------------------ Model II --------------------

#In this particular model is more or less the same as M1.
#However, correlation structure for the random effects are now common for
#treatment and control groups. See CellName in Random.structure.
M2gamm <- PamGeneMix(formula = y ~ -1 + ResTrt + s(time, by = ResTrt, bs = "tp", m = 3),
                      PTx = PTx, Random.structure = list(ArrayNum = -1, CellName = -1 + time + time2, ID = -1 + time + time2),
                      Control.list = list(maxIter = 200, msMaxIter = 250, msMaxEval = 1000, apVar = TRUE))

show(M2gamm)

#------- summary -----------
summary(M2gamm)

#---------------- section II -------------------
#In this section, we use all Responisive and Non-responsive cell lines data such that we
#have only two groups: treatment and control for the analysis.

PTx <- TestPepModelData

# log2 transform the response
PTx$y <- log2(PTx[, ncol(PTx)])

# we now create interaction between ResState and Treatment as follows.

n.groups <- length(unique(PTx[, c("ResState")])) * length(unique(PTx[, c("TreatName")]))
PTx$ResTrt <- 0
xhlp <- lm(y ~ -1 + ResState:TreatName, data = PTx, x = TRUE)$x
for (i in 1:n.groups) PTx$ResTrt[xhlp[, i] == 1] <- i
PTx$ResTrt <- as.factor(PTx$ResTrt)
levels(PTx$ResTrt) <- colnames(xhlp)

# unique cell lines
cellLines <- levels(PTx$CellName)
PTx$ResState <- as.factor(PTx$ResState)

# create interaction between Treatment and cell lines
PTx$CellLineResTrt <- 0
xhlp <- lm(y ~ -1 + CellName:TreatName, data = PTx, x = TRUE)$x
ncols <- ncol(xhlp)
for (i in 1:ncols) PTx$CellLineResTrt[xhlp[, i] == 1] <- i
PTx$CellLineResTrt <- as.factor(PTx$CellLineResTrt)
levels(PTx$CellLineResTrt) <- colnames(xhlp)

#------------Model III ------------------
#The more complex model for which we assumed that group-specific smoothing parameter and
#group-specific variance covariance structure. we used thin plate regression splines with third
#order derivative. In this model nested random effects structure is assumed for replicates within
#the cell lines. And for the cell lines - specific random effects are assumed to be realized
#from each group separately.

M3gamm <- PamGeneMix(formula = y ~ -1 + ResTrt + s(time, by = ResTrt, bs = "tp", m = 3),
                      Weights = varIdent(form = ~1 | ResTrt),
                      PTx = PTx, Random.structure = list(ArrayNum = -1, CellLineResTrt = -1 + time + time2, ID = -1 + time + time2),
                      Control.list = list(maxIter = 200, msMaxIter = 250, msMaxEval = 1000, apVar = TRUE))

show(M3gamm)

#------- summary -----------
summary(M3gamm)
PreProcessAllPeptides

Description

This is a wrapper function for PreProcessPam. And autotmates the preprocessing of list of peptides and Cell lines. This function removes peptide specific extreme observations, profiles and negative intensity measurements. Negative intensity measurements within a profile are replaced by half of the positive minimum intensity measurement of that profile. Extreme profiles have been removed via residuals based on one-way ANOVA (Response for the ANOVA is AUC: Average(residuals) plus or minus 2 times SE(residuals)). Extreme observations within a profile were deleted if they differ more than plus or minus 2 times standard deviations of the profile-specific mean.

Usage

PreProcessAllPeptides(pep.names,PamSig,PathOutPut)

Arguments

pep.names List of peptide names to be considered.
PreProcessAllPeptides

PamSig       A PamChipData data frame.
PathOutPut   Preprocessed RData files are created for each peptide and by default save in the
current working directory. Otherwise path to which files can be saved should be
given.

Value
A list of objects of class CleanedPamData

Author(s)
Pushpike Thilakarathne, Ziv Shkedy and Dan Lin

See Also
PreProcessPam

Examples

data(PamChipData)

#-- Peptides Names --
PeptidesNames<-setdiff(colnames(PamChipData),
c("ID", "ResState", "ArrayNum", "CellName", "TreatName","Time"))

#--preprocess five peptides ----
temp<-PreProcessAllPeptides(pep.names=PeptidesNames[1:5],PamSig=PamChipData)

show(temp)
#-------- PreProcessed List of Peptides --------
#Class : CleanedPamData
#No. Peptides : 5
#No. Cell lines: 2/zero.noslash
#Columns names for a certain peptide:
# ID ResState ArrayNum CellName TreatName Time Pep1
#Unique Time points: /zero.noslash 5 1/zero.noslash 15 2/zero.noslash 25 3/zero.noslash 35 4/zero.noslash 45 5/zero.noslash 55 6/zero.noslash
#PreProcessed Data frames are saved as RData Objects in : C:/Users/

#--preprocess three peptides only for responsive cell lines ----
temp2<-PreProcessAllPeptides(pep.names=PeptidesNames[1:3],
PamSig=PamChipData[PamChipData[,c("ResState")]=="R",])

#--preprocess three peptides only for NON-responsive cell lines ----
temp3<-PreProcessAllPeptides(pep.names=PeptidesNames[1:3],
PamSig=PamChipData[PamChipData[,c("ResState")]=="NR",])
PreProcessPam

Pre-processing Cell Line specific data for a given Peptide

Description

This function removes Cell Line specific extreme observations, profiles and negative intensity measurements of a given peptide. Negative intensity measurements within a profile are replaced by half of the positive minimum intensity measurement of that profile. Extreme profiles have been removed via residuals based on one-way ANOVA (Response for the ANOVA is AUC: Average(residuals) plus or minus 2 times SE(residuals)). Extreme observations within a profile were deleted if they differ more than plus or minus 2 times standard deviations of the profile-specific mean.

Usage

PreProcessPam(p=108, cel =4, d=2, PamS, plotting=TRUE )

Arguments

<table>
<thead>
<tr>
<th>p</th>
<th>Peptide to be considered. And should be supplied as numeric value that corresponds to the peptide index.</th>
</tr>
</thead>
<tbody>
<tr>
<td>cel</td>
<td>Cell Line index.</td>
</tr>
<tr>
<td>d</td>
<td>Flexibility for the extreme observations to be removed. Eg. 2 or 3 times standard deviations. Default is 2.</td>
</tr>
<tr>
<td>PamS</td>
<td>A PamChipData data frame.</td>
</tr>
<tr>
<td>plotting</td>
<td>A Boolean parameter. Default is TRUE and observed and filtered data are visualized for a given cell line of a peptide.</td>
</tr>
</tbody>
</table>

Value

A list of objects of class CleanedPeptide.

Author(s)

Pushpike Thilakarathne, Ziv Shkedy and Dan Lin

See Also

PreProcessAllPeptides

Examples

data(PamChipData)

#--------
PreProcessPam( p =20, cel =4,  d =2, PamS=PamChipData, plotting=TRUE )
summary

Summarize Fitted Gamm Object for a given Peptide

Description

This method principally does nothing more than applying the pre-implemented `summary()` function to the slot `gammran3` of an object of class `PamChipMixed`. One then obtains estimated group specific velocities at t=0, t=max(t), t=any, and t=end. And also fitted lme and gam objects are shown.

Arguments

object An object of class `PamChipMixed`.

Value

No return.

Author(s)

Pushpke Thilakaratne, Ziv Shkedey and Dan Lin

See Also

`VisualizePamGeneMix, gamm`
TestPepModelData  
*PamChip Peptide Specific Cleaned Data*

**Description**

This is a cleaned PamChip Peptide specific dataset where there are columns for ArrayID, Cell line names, Responsive Statuses, Treatment indicator, Time points, and Replicate ID. Intensity measurements for the peptide is given in the last column.

**Usage**

```r
data(TestPepModelData)
```

**Format**

The format is a data frame which has following structure.

<table>
<thead>
<tr>
<th>ID</th>
<th>ResState</th>
<th>ArrayNum</th>
<th>CellName</th>
<th>TreatName</th>
<th>Time</th>
<th>Peptide</th>
</tr>
</thead>
<tbody>
<tr>
<td>6</td>
<td>R</td>
<td>0559C80</td>
<td>H3255</td>
<td>Control</td>
<td>11</td>
<td>2706</td>
</tr>
<tr>
<td>6</td>
<td>R</td>
<td>0559C80</td>
<td>H3255</td>
<td>Control</td>
<td>16</td>
<td>4442</td>
</tr>
<tr>
<td>6</td>
<td>R</td>
<td>0559C80</td>
<td>H3255</td>
<td>Control</td>
<td>21</td>
<td>5045</td>
</tr>
<tr>
<td>6</td>
<td>R</td>
<td>0559C80</td>
<td>H3255</td>
<td>Control</td>
<td>26</td>
<td>5491</td>
</tr>
<tr>
<td>6</td>
<td>R</td>
<td>0559C80</td>
<td>H3255</td>
<td>Control</td>
<td>31</td>
<td>5950</td>
</tr>
<tr>
<td>6</td>
<td>R</td>
<td>0559C80</td>
<td>H3255</td>
<td>Control</td>
<td>36</td>
<td>6124</td>
</tr>
</tbody>
</table>

**Author(s)**

Pushpike Thilakarathne, Ziv Shkedy and Dan Lin

**See Also**

`PamChipData`

**Examples**

```r
#-- load Data --
data(TestPepModelData)

#-- Check Data --
TestPepModelData[1:5,]

#-- Unique Time Points --
UTime<-unique(TestPepModelData[,c("Time")])
UTime

#-- Responsive and Non-Responsive Cell Lines --
RnRcelllines<-unique(TestPepModelData[,c("CellName","ResState")])
```
testVarCom

rownames(RnRcellines)<-1:20
RnRcellines

#-- Number of Plates (96 well plates)---
Plates<-unique(TestPepModelData[,c("ArrayNum")])
Plates

testVarCom

Wald Test to check variances are equal or not

Description

Testing whether the group specific variances are the same or not.

Usage

testVarCom(ObjPamFit)

Arguments

ObjPamFit       Output object from the PamGeneMix.

Value

A list          A list with p-value(s) of the test. And estimated variances and their standard errors for each group. If there are no groups then NA will be returned.

Author(s)

Pushpike Thilakarathne, Ziv Shkedy and Dan Lin

References


See Also

PamGeneMix
Examples

data(TestPepModelData)

PTx<-TestPepModelData

# log2 transform the response
PTx$y<-log2(PTx[,ncol(PTx)])

# we now create interaction between ResState and Treatment as follows.

n.groups<-length(unique(PTx[,c("ResState")]))*length(unique(PTx[,c("TreatName")]))
PTx$ResTrt<-0
xhlp<-lm(y~-1+ResState:TreatName,data=PTx,x=TRUE)$x
for (i in 1:n.groups) PTx$ResTrt[xhlp[,i]==1]<-i
PTx$ResTrt<-as.factor(PTx$ResTrt)
levels(PTx$ResTrt)<-colnames(xhlp)

# unique cell lines

cellLines<-levels(PTx$CellName)
PTx$ResState<-as.factor(PTx$ResState)

# create interaction between Treatment and cell lines

PTx$CellLineResTrt<-0
xhlp<-lm(y~-1+CellName:TreatName,data=PTx,x=TRUE)$x
ncols<-ncol(xhlp)
for (i in 1:ncols) PTx$CellLineResTrt[xhlp[,i]==1]<-i
PTx$CellLineResTrt<-as.factor(PTx$CellLineResTrt)
levels(PTx$CellLineResTrt)<-colnames(xhlp)

M3gamm<-PamGeneMix(formula=y~-1+ResTrt+s(time,by=ResTrt,bs="tp",m=3),Weights=varIdent(form=~1|ResTrt) ,
PTx=PTx,Random.structure=list(ArrayNum=~-1,CellLineResTrt=~1+time+time2,ID=~1+time+time2),
Control.list=list(maxIter=200, msMaxIter=250 ,msMaxEval=1000,apVar=TRUE))

# we can now use testVarCom function to test the hypothesis that
# H: Var(group_R_i) - Var(group_R_j) = 0 and Var(group_NR_i) - Var(group_NR_j) = 0.
# That is testing for treatment effects conditional on the responsive statues

testVarCom(M3gamm)

VisualizePamGeneMix

VisualizePamGeneMix Output Objects

Description

This function visualizes following plots using the fitted PamGeneMix object.
VisualizePamGeneMix

Usage

VisualizePamGeneMix(PamObject, plot.type=c("velocity","velocityCI","smooth.fit","cellline","replicate","SubSpeVelocity"), name.cell=NULL)

Arguments

PamObject Object out from the PamGeneMix.
plot.type Required plot types:
1. velocity: Produce first order derivatives for each group in single plot.
2. velocityCI: Produce first order derivative along with the 95% confidence intervals for each group separately.
3. smooth.fit: Produce smoothing fit for each group severalty with observed profiles as background.
4. cellline: Cell line specific fitted smoothing curves for each group separately.
5. replicate: Replicate specific smoothing fit for a given cell line. Cell line name should be given.
6. SubSpeVelocity: Replicate specific first order derivatives of the fitted smoothing curves for each group separately.
name.cell Name of the cell line that needs to be visualized and this is only applicable when replicate specific smoothing fit needs to be visualized.

Value

No value returns

Author(s)

Pushpike Thilakarathne, Ziv Shkedy and Dan Lin

See Also

ObservedRepProfiles, ObservedGroupProfiles, ObservedCellProfiles

Examples

#--------

data(TestPepModelData)
head(TestPepModelData)

PTx<-TestPepModelData

# log2 transform the response
PTx$y<-log2(PTx[,ncol(PTx)])

PTx<-PTx[PTx[,c("ResState")]=="R",]
VolcanoPam

n.groups<-length(unique(PTx[,c("ResState")]))*length(unique(PTx[,c("TreatName")]))

# n.groups

xhlp<-lm(y~-1+TreatName,data=PTx,x=TRUE)$x

for (i in 1:n.groups) PTx$ResTrt[xhlp[,i]==1]<-i

PTx$ResTrt<-as.factor(PTx$ResTrt)

levels(PTx$ResTrt)<-colnames(xhlp)

# unique cell lines

cellLines<-levels(PTx$CellName)

PTx$ResState<-as.factor(PTx$ResState)

# create interaction between Treatment and cell lines

PTx$CellLineResTrt<-0

xhlp<-lm(y~-1+CellName:TreatName,data=PTx,x=TRUE)$x

ncols<-ncol(xhlp)

for (i in 1:ncols) PTx$CellLineResTrt[xhlp[,i]==1]<-i

PTx$CellLineResTrt<-as.factor(PTx$CellLineResTrt)

levels(PTx$CellLineResTrt)<-colnames(xhlp)

#--------------- the model -----------------

M1gamm<-PamGeneMix(formula=y~-1+ResTrt+s(time,by=ResTrt,bs="tp",m=3),
                        PTx=PTx,Random.structure=list(ArrayNum=~-1,CellLineResTrt=~-1+time+time2,ID=~-1+time+time2),
                        Control.list=list(maxIter=2,msMaxIter=25,msMaxEval=1000,apVar=TRUE))

#----------------------------------------------------------------------

VisualizePamGeneMix(M1gamm,plot.type="smooth.fit")

VisualizePamGeneMix(M1gamm,plot.type="cellline")

VisualizePamGeneMix(M1gamm,plot.type="velocityCI")

VisualizePamGeneMix(M1gamm,plot.type="replicate",name.cell= "A431" )

VisualizePamGeneMix(M1gamm,plot.type="replicate",name.cell="DU145" )

VisualizePamGeneMix(M1gamm, plot.type="SubSpeVelocity")

VolcanoPam | Visualize Output Object from AutoPamGeneMix

Description

This function visualizes Volcano plots of p-values returns from the AutoPamGeneMix. It also returns Top K Peptides according to the test at t=0, t=end, t=any time point, and t=entire profile. Corre-
responding p-values are also returned.

Usage

VolcanoPam(ObjectAutoPam,Topp=5,plotting=TRUE)

Arguments

ObjectAutoPam  Object out from the AutoPamGeneMix.
Topp            Number of top K peptides to be considered. Default is five.
plotting       A boolean parameter indicating whether the plots to be shown.

Value

PP              A data frame with p-values.
TopList         Top K peptides at time points where the tests are performed.

Author(s)

Pushpike Thilakarathne, Ziv Shkedy and Dan Lin

See Also

AutoPamGeneMix

Examples

#--------
data(PamChipData)

PeptidesNames<-setdiff(colnames(PamChipData),
  c("ID", "ResState", "ArrayNum", "CellName", "TreatName", "Time"))

#------- Four groups in the data ---
# Apply preprocessing steps for each peptide
temp2<-PreProcessAllPeptides(pep.names=PeptidesNames,PamSig=PamChipData,PathOutPut="C:/Temp")

#------------------------
# Fit the model for each peptide : This will take a while !!
ResultsAll<-AutoPamGeneMix(formula=y~-1+ResTrt+s(time,by=ResTrt,bs="tp",m=3),
Weights=varIdent(form=~1|ResTrt),
Random.structure=list(ArrayNum=~1,CellLineResTrt=~1+time+time2,ID=~1+time+time2),
temp2,TestAt=30,PathOutPut="C:/FittedGamm")

show(ResultsAll)
VolcanoPam(ResultsAll,Topp=1,plotting=FALSE)

VolcanoPam(ResultsAll,Topp=5,plotting=TRUE)
Index

*Topic Mixed Model, Smoothing, Thin Plate regression
  PamChipAllPepResults-class, 10
*Topic Mixed models, Smoothing, Generalized Additive Models, Thin-plate regression splines
  AutoPamGeneMix, 2
  PamChipMixed-class, 12
  PamGeneMix, 13
*Topic Plot
  ObservedRepProfiles, 9
*Topic Semi-Parametric Mixed Models
  summary, 21
*Topic Wald, likelihood-ratio test
  testVarCom, 23
*Topic anova, AUC
  CleanedPamData-class, 5
  CleanedPeptide-class, 6
*Topic anova, datasets
  PreProcessAllPeptides, 18
  PreProcessPam, 20
*Topic datasets
  PamChipData, 11
  TestPepModelData, 22
*Topic plot, smoothing, non-linear, velocity, confidence intervals
  VisualizePamGeneMix, 24
*Topic plot
  ObservedCellProfiles, 6
  ObservedGroupProfiles, 7
  VolcanoPam, 26
AutoPamGeneMix, 2, 10, 15, 26, 27

CleanedPamData, 19
CleanedPamData (CleanedPamData-class), 5
CleanedPamData-class, 5
CleanedPeptide, 20

CleanedPeptide (CleanedPeptide-class), 6
CleanedPeptide-class, 6
corClasses, 14

formula.gam, 3, 14
gam, 4, 15, 21
gam.models, 14
gamm, 3, 4, 13–15, 21
glm, 3, 4, 14
lme, 3, 4, 14, 15, 21
lmeControl, 14

ObservedCellProfiles, 6, 8, 9, 25
ObservedGroupProfiles, 7, 7, 9, 25
ObservedRepProfiles, 7, 8, 9, 25

PamChipAllPepResults, 2, 3
PamChipAllPepResults (PamChipAllPepResults-class), 10
PamChipAllPepResults-class, 10
PamChipData, 11, 22
PamChipMixed, 14, 21
PamChipMixed (PamChipMixed-class), 12
PamChipMixed-class, 12
PamChipMixed-method (PamChipMixed-class), 12
PamGeneMix, 2–4, 10, 12, 13, 13, 23, 25
plot, 9
PreProcessAllPeptides, 2–6, 18, 20
PreProcessPam, 4–6, 18, 19, 20

s, 3, 14
show, CleanedPamData-method
  (CleanedPamData-class), 5
show, CleanedPeptide-method
  (CleanedPeptide-class), 6
show,PamChipAllPepResults-method
  (PamChipAllPepResults-class), 10
show,PamChipMixed-method
  (PamChipMixed-class), 12
summary, 21
summary,PamChipMixed-method(summary), 21

te, 3, 14
TestPepModelData, 11, 22
testVarCom, 15, 23

VisualizePamGeneMix, 14, 15, 21, 24
VolcanoPam, 26