The TwslmSpikeWeight Package

July 23, 2007

Version 1.0.1
Date 2006-01-17
Maintainer Deli Wang <deli.wang@ccc.uab.edu>
Author Deli Wang, Jian Huang
Depends splines
Title Systematic approaches using TW-SLM for incorporating control spots and data quality information to improve normalization of cDNA microarray data.
Description TwslmSpikeWeight is for normalization of cDNA microarray data with the two-way semilinear model(TW-SLM). It incorporates information from control spots and data quality in the TW-SLM to improve normalization of cDNA microarray data. Huber’s and Tukey’s bisquare weight functions are available for robust estimation of the TW-SLM.
License GPL version 2 or newer

R topics documented:

HU ................................................................. 1
HUweightdata .................................................. 3
TwslmSpikeWeight ........................................... 4

Index 10

HU

Human placenta versus the Universal reference hybridization data

Description
This is a data frame consists 5 variables.

Usage
data(HU)
Format

A data frame with 75714 observations on 5 variables:

- [1] HUid: gene identification number
- [2] HUslide: slide number
- [3] HUblock: block number within each slide
- [4] HUint: total intensity $0.5\log_2(RG)$
- [5] HUratio: intensity ratios $\log_2\frac{R}{G}$

where R is background adjusted signal intensity in the Cy5 channel, G is background adjusted signal intensity in the Cy3 channel.

Source

These data is the part of data set generated by human placenta versus the universal reference hybridization by Dr. Bento M. Soares lab in the University of Iowa. The data sets were used by Wang et al. (2006,2007) for the purpose of methodology demonstration.

References


HUweightdata

Spot quality information for Human placenta versus the Universal reference hybridization

Description

This data set consists 10 variables.

Usage

data(HUweightdata)

Format

A list with 75714 observations on 10 variables:

- [1] RF: front mean intensity for the Cy5
- [2] RB: background mean intensity for the Cy5
- [3] SDRF: standard deviation of RF
- [4] SDRB: standard deviation of RB
- [5] GF: front mean intensity for the Cy3
TwslmSpikeWeight

[.6] GB background mean intensity for the Cy3
[.7] SDGF standard deviation of GF
[.8] SDGB standard deviation of GB
[.9] NF number of pixels used for the front intensity
[.10] NB number of pixels used for the background intensity

Source

These data is the part of data set generated by human placenta versus the universal reference hybridization by Dr. Bento M. Soares lab in the University of Iowa. The data sets were used by Wang et al. (2006,2007) for the purpose of methodology demonstration.

References


TwslmSpikeWeight

**Normalization of cDNA microarray data using spike and spot quality information in the two-way semi-linear model (TW-SLM)**

**Description**

By incorporating spike and spot quality information, normalization of cDNA microarray data using the TW-SLM could be improved. Two methods are available for estimation, including robust estimation methods and the least square method. The B-splines is used to estimate nonparametric normalization curves in the model.

**Usage**

TwslmSpikeWeight(sld,blk,geneid,rt,intn,weight,s.sld,s.blk,s.geneid,s.rt,
    s.intn,s.weight,s.constant=0.0,df=12,degree=3,block.norm=FALSE,
    robust=TRUE,robust.name="Tukey", scale.constant=2.5,
    weight.constant=4.685,ibeta=NULL,iscale=NULL,tol=1e-5)

spike(sld,geneid,rt,intn,weight=NULL,s.sld,s.geneid,s.rt,s.intn,
    s.weight=NULL,s.constant=0.0,df=12,degree=3,tol=1e-5)

robust.spike(sld,geneid,rt,intn,weight=NULL,s.sld,s.geneid,s.rt,s.intn,
    s.weight=NULL,s.constant=0.0,df=12,degree=3,
    robust.name="Tukey",scale.constant=2.5,weight.constant=4.685,
    ibeta=NULL,iscale=NULL,tol=1e-5)

BlockByBlock.spike(sld,blk,geneid,rt,intn,weight=NULL,s.sld,s.blk,s.geneid,
s.rt, s.intn, s.weight=NULL, s.constant=0.0, df=12, degree=3, robust=TRUE, robust.name="Tukey", scale.constant=2.5, weight.constant=4.685, ibeta=NULL, iscale=NULL, tol=1e-5)

**Arguments**

- **sld**: A vector of array or slide numbers. This argument is required.
- **blk**: A vector of block numbers. This argument is required only for blockwise normalization.
- **geneid**: A vector of gene identification numbers, can be numerical numbers or gene names. This argument is required.
- **rt**: A vector of log2 intensity ratio, i.e. $\log_2(Cy5/Cy3)$. This argument is required.
- **intn**: A vector of average of log two total intensity, i.e. $0.5\log_2(Cy5*Cy3)$. This argument is required.
- **weight**: A vector of weights assigned for non-spike spots. This argument is required for normalization using spot quality information or pre-assigned weights for non-spike spots are provided. Weights must be greater or equal to zero.
- **s.sld**: A vector of array or slide numbers for spike spots. This argument is required.
- **s.blk**: A vector of block numbers where spike spots locate. This argument is required only for blockwise normalization.
- **s.geneid**: A vector of spike identification numbers, can be numerical numbers or spike names. This argument is required.
- **s.rt**: A vector of log2 intensity ratio for spikes, i.e. $\log_2(Cy5/Cy3)$. This argument is required.
- **s.intn**: A vector of averages of log two total intensity for spikes, i.e. $0.5\log_2(Cy5*Cy3)$. This argument is required.
- **s.weight**: A vector of weights assigned for spike spots. This argument is required for normalization using spot quality information or pre-assigned weights for spikes spots are provided. Weights must be greater or equal to zero.
- **s.constant**: A constant which spikes centered after normalization. The default value is zero. This argument is required.
- **df**: The degrees of freedom for B-spline smooth. The default is 12.
- **degree**: The order of polynomials in the B-splines. The default is 3, the cubic spline.
- **block.norm**: A logical value indicating whether blockwise normalization is performed or not. The default is FALSE, which means the default normalization is slide by slide normalization.
- **robust**: A logical value indicating if the robust procedure is incorporated in normalization. The default is TRUE, which means normalization is conducted using a robust method in the TW-SLM. The least square method is used if this argument is FALSE.
- **robust.name**: A name for the robust procedure. The default is "Tukey", which means the location and scale parameters are estimated iteratively with Tukey’s bisquare weight function. Another option is "Huber", which uses Huber’s weight function and the location and scale parameters are estimated iteratively. This option works only if the robust argument is TRUE.
**scale.constant**
a constant chosen for scale estimation in the robust TW-SLM. The default is 2.5.

**weight.constant**
a constant chosen for robust location estimation. The default is 1.345 for Huber’s weight function and 4.685 for Tukey’s weight function.

**ibeta**
a vector for initialization of $\beta$. The default is NULL. The ordinary least square estimators for $\beta$ is one choice of starting values for the robust TW-SLM. Giving this value will speed up convergence.

**iscale**
a value for initialization of the scale parameter in the robust model. The default is NULL. Giving this value will speed up convergence of the algorithm.

**tol**
a convergent criteria for iterative estimation procedure. The default is 1e-5.

**Details**
Normalization is a basic step in the analysis of microarray data. Widely used normalization method for cDNA microarray data was the loess normalization method proposed by Yang et al. (2001). This method requires that at least one of the two underlying biological assumptions, i.e. either (i) a small fraction of genes in the experiment are differentially expressed; or (ii) the up-regulated genes and the down-regulated genes are distributed symmetrically. The TW-SLM is a generalization of the semiparametric regression model. It does not require either of the above two assumptions for normalization of cDNA microarray data. Side information including spike information and spot quality information could be used to improve normalization of cDNA microarray data using TW-SLM.

The TW-SLM has the form

$$y_{ij}(s) = \phi_i(x_{ij}(s)) + \beta_j(s) + \epsilon_{ij}(s).$$

where $y_{ij}(s) = \log_2(C_{y5}/C_{y3})$, $\phi_i(x_{ij(s)})$ is the intensity dependent normalization curve for slide $i$, $x_{ij(s)} = 0.5\log_2(C_{y5} \ast C_{y3})$, $\beta_j$ is the relative effect of gene $j$, $\epsilon_{ij(s)}$ is the residual term, for $i = 1, \ldots, n$, where $n$ is the total number of slides, $j = 1, \ldots, J$, where $J$ is the total number of genes in the experiment. Applying the TW-SLM to spike spots specifically and integrate spikes and non-spike genes together in the normalization process. Spikes and non-spike genes share common normalization curves. Spot quality information could also be incorporated in the TW-SLM so that normalization might be improved. Details of integration of spikes and spot quality information in normalization of cDNA microarray data using the TW-SLM can be found in Wang et al. (2007).

The TwslmSpikeWeight package implements the TW-SLM for normalization of cDNA microarray data considering spike spots information and/or spot quality information. Two robust estimation procedures are implemented in the current version of TwslmSpikeWeight: Huber’s method (1981) and Tukey’s method (1986). The spike function implements the ordinary least square estimation method and the robust.spike function implements robust estimation methods. The BlockByBlock.spike function carries out block-wise normalization.

**Value**
An object of a list is returned with components:

**name**
a vector of names of unique genes.

**beta**
an estimated parameters of relative gene expression level for each gene.
fittedvalue  a vector of fitted values in the TW-SLM.
bfit  a vector of fitted values for normalization curves.
slide  a vector of slide number from the input the function. The order is different from the input "sld" vector.
id  a vector of gene ID from the input vector "geneid" with a different order.
ratio  a vector of the log two intensity ratio from the input vector "rt" with a different order.
intensity  a vector of average log two total intensity from the input vector "intn" with a different order.
weight  a vector of weights used in the TW-SLM. These weights could be pre-assigned weights or calculated weights using spot quality information.
rweight  a vector of weights calculated at the last step of convergence in the robust TW-SLM.
spike.slide  a vector of slide number for spikes. The order is different from the input vector "s.sld".
spike.id  a vector of spike identification numbers from the input vector "s.geneid" with a different order.
spike.ratio  a vector of the log two intensity ratio for spikes' intensity with a different order from input "s.rt".
spike.intensity  a vector of average log two total intensity for spikes with a different order from input vector "s.intn".
spike.bfit  a vector of fitted values for spikes after normalization.
spike.weight  a vector of weights for spikes in the TW-SLM. These weights could be pre-assigned weights or calculated weights using spike spot quality information.
rspike.weight  a vector of weights for spikes calculated at the last step of convergence in the robust TW-SLM.
scale  a scale estimator in the two-way semilinear model.

Note

`TwslmSpikeWeight` is the main function to control which normalization method will be used. `spike` is the function for the TW-SLM using the ordinary least squares, `robust.spike` is the function for robust estimation of the TW-SLM, `BlockByBlock.spike` is the function for block-wise normalization using spikes.

Author(s)

Deli Wang (deli.wang@ccc.uab.edu) Jian Huang (jian@stat.uiowa.edu)
References


Examples

```r
## The example used in the paper Wang et al. (2007) will be used to
## demonstrate utilization of the developed package for normalization of cDNA
## microarray data by incorporating spike and spot quality information.

## load data used for the calculation
#require(TwslmSpikeWeight)
data(HU,"HUweightdata")

## a function to calculate weights of spots
ratioweight <- function(RF,RB,SDRF,SDRB,GF,GB,SDGF,SDGB,NF,NB){
  weight <- 1/((SDRF^2/NF+SDRB^2/NB)/(RF-RB)^2+(SDGF^2/NF+SDGB^2/NB)/(GF-GB)^2)
  return(weight)
}

## spikes ids
zerospike <- c("01-NBB-f-02","02-NAP-a-01","03-NAP-a-05","04-NAP-a-09","05-NXEg-e-10",
                "06-NXEg-h-07","07-NBB-h-08","08-NBB-f-05","09-NAP-b-02","10-NS8W-d-02",
                "11-NS8W-c-10","12-NAP-c-05","13-NAP-b-06","14-NAP-b-08","15-NAP-d-08",
                "16-NAP-c-02","17-NBB-g-12","18-NAP-d-03","19-NAP-d-06","20-NXEg-h-06",
                "21-NAP-c-08","22-NBB-h-07","23-NBB-e-09","24-NAP-d-09",
                "25-NAP-b-09","26-NBB-f-08","SC1","SC3","SC5")

## calculate weights
allw <- ratioweight(RF=HUweightdata$RF,RB=HUweightdata$RB,SDRF=HUweightdata$SDRF,
                    SDRB=HUweightdata$SDRB,GF=HUweightdata$GF,GB=HUweightdata$GB,SDGF=HUweightdata$SDGF,
                    SDGB=HUweightdata$SDGB,NF=HUweightdata$NF,NB=HUweightdata$NB)
```
SDGB=HUweightdata$SDGB,NF=HUweightdata$NF,NB=HUweightdata$NB)

HUid <- HU$HUid
HUratio <- HU$HUratio
HUint <- HU$HUint
HUslide <- HU$HUslide
HUblock <- HU$HUblock

flag <- match(HUid,zerospike,nomatch=0)>0

## calculate weights

w <- allw[!flag]
huid <- HUid[!flag]
hurt <- HUratio[!flag]
huint <- HUint[!flag]
husld <- HUslide[!flag]
hublk <- HUblock[!flag]

## spikes information

sw <- allw[flag]
shuid <- HUid[flag]
shurt <- HUratio[flag]
shuint <- HUint[flag]
shusld <- HUslide[flag]
shublk <- HUblock[flag]

## spike, robust, weight;

huSRW <- TwsImSpikeWeight(sld=husld,geneid=huid, rt=hurt, intn=huint, weight=w, 
  s.sld=shusld, s.geneid=shuid, s.rt=shurt, s.intn=shuint, s.weight=sw, s.constant=0.0, 
  df=12, degree=3, block.norm=FALSE, robust=TRUE, robust.name="Tukey", scale.constant=2.5, 
  weight.constant=4.685, ibeta=NULL, iscale=NULL, tol=0.0001)

a <- huSRW$slide
b <- huSRW$spike.slide
rr <- huSRW$ratio[a==1]
intt <- huSRW$intensity[a==1]
ff <- huSRW$bfit[a==1]
ii <- order(intt)

plot(intt,rr,xlim=c(0,16))
lines(intt[ii],ff[ii],col="red")
points(huSRW$spike.intensity[b==1],huSRW$spike.ratio[b==1],col="blue")

## zerospike estimates (mean, median, sd)

zerospikemeanR <- aggregate(huSRW$spike.ratio-huSRW$spike.bfit,by=list(huSRW$spike.id),mean)
spikemeanR <- mean(2^zerospikemeanR$x)
spikemedianR <- median(2^zerospikemeanR$x)
spikesdR <- sd(2^zerospikemeanR$x)
Index

*Topic **datasets**
  HU, 1
  HUweightdata, 2
*Topic **nonparametric**
  Tws1mSpikeWeight, 3
*Topic **robust**
  Tws1mSpikeWeight, 3
*Topic **smooth**
  Tws1mSpikeWeight, 3

BlockByBlock.spike
  (Tws1mSpikeWeight), 3

HU, 1
HUweightdata, 2

robust.spike(Tws1mSpikeWeight), 3

spike(Tws1mSpikeWeight), 3

Tws1mSpikeWeight, 3