Package ‘DSA’

January 18, 2013

Title DSA: Digital sorting of complex tissues for cell type specific gene expression profiles

Version 0.99.0

Date 2013-1-17

Author Yi Zhong, Ying-Wooi Wan, Kaifang Pang, Lionel M.L. Chow, Zhandong Liu

Description Functions to implement Digital sorting algorithm (DSA) in extracting cell type specific gene expression profiles from mixed tissue samples.

Maintainer Ying-Wooi Wan<yingwoow@bcm.edu>

Dependencies

License GPL2.0

R topics documented:

`cell.gene` *cell-gene key table*

Description

This is the cell-gene key table: mapping between liver, brain, and lung to their respective tissue specific markers genes. This was obtained from Tissue Specific Gene Expression Regulation (TIGER) database and GENENOTE

Usage

```
data('cell.gene')
```

Format

A data matrix of two columns and 237 rows of marker genes.

References

Liu X, Yu X, Zack DJ, Zhu H, Qian J: TiGER: a database for tissue-specific gene expression and regulation. BMC Bioinformatics 2008, 9:271

Yanai I, Benjamin H, Shmoish M, Chalifa-Caspi V, Shklar M, Ophir R, Bar-Even A, Horn-Saban S, Safran M, Domany E, Lancet D, Shmueli O: Genome-wide midrange transcription profiles reveal expression level relationships in human tissue specification. Bioinformatics 2005, 21:650<96>659.

`Deconvolution` *Deconvolution function*

Description

Function to estimate the deconvoluted signals - deconvolve the mixture signals to cell-type specific signals for each gene.

Usage

```
Deconvolution(data, weight, method = "LM", l = 0, u = 2^34)
```

Arguments

`data`	data matrix of the mixture signals, with genes in row, cell type in column, in anti-log scale.
`weight`	weight matrix, with cell type in row, tissue types in column.

`method` methods used in estimating true signals for each tissue type. Default to ’LM’ for linear regression. Other methods included ’LG’ (logistic regression),’QP_LM’ or ’QP_LG’ (quadradic programming with constraint on the estimated parameter on linear/logistic regression) For methods ’LG’ and ’QP_LG’, input data is transformed into log-scaled and returned values are anti-logged.

`l,u` values for the lower- (l) and upper- (u) bound used in setting the vector for values of b0 (bvec for solve.QP) in solving for quadratic programming. Defaults to 0 and 2^34 respectively.

Value

`paraM` a matrix of deconvoluted signals (anti-log scale) with genes in row, tissue types in columns

See Also

`EstimateWeight`, `GSM_QP`, `solve.QP`,

Examples

```
## load package DSA
library(DSA)

## load sample data
data(’mix.signals’)
data(’cell.gene’)

# In this data, thr first three samples contain signal from only one cell
pure <- mix[, 1:3]
mix <- mix[, 4:14]

# Generate a list object set of marker genes for each cell type
unique_cell_type <- unique(cell.gene[,2])
gene_list <- list()
for( i in 1 : length(unique_cell_type)){
gene_list[[i]] <- cell.gene[cell.gene[,2] == unique_cell_type[i],1]
}
names(gene_list) <- unique_cell_type

# Estimate weight matrix
estimated_weight <- EstimateWeight(2^mix, gene_list, method=”LM”)

# Estimate deconvoluted signals
deconv <- as.matrix(Deconvolution(2^mix, t(estimated_weight$weight), method=”QP_LM”, l=min(2^mix), u=max(2^mix))
rownames(deconv) <- rownames(mix)
colnames(deconv) <- unique_cell_type

# check the estimated cell-type specific signals
for(i in 1:3){
print(cor(2^(pure[,i]),deconv[,i]))
}
```

DSA *Interface function to implement DSA*

Description

This is the main interface function to implement digital sorting algorithm for cell type specific gene expression profiles.

Usage

```
DSA(mix, cell.gene, weight = NULL, method = "LM", out.cell.file = NULL,
out.weight.file = NULL, log2 = TRUE, l = 0, u = 2^34, meq = 0)
```

Arguments

`mix` matrix of mixture signals (genes in row, cell type in column).

`cell.gene` the cell-gene key table: mapping between cell type and cell specific genes. This table needs to be only two columns: first column with gene symbols, second column the cell type.

`weight` weight matrix (samples in row, cell types in column). Default to 'NULL', where no weight matrix is provided by user, and the weight will be estimated by the function.

`method` methods used in estimating weight and true signals for each tissue type. Default to 'LM' for linear regression. Other methods included 'LG' (logistic regression),'QP_LM' or 'QP_LG' (quadradic programming with constraint on the estimated parameter on linear/logistic regression).

`out.cell.file` file name to store the deconvoluted signals. default to 'NULL' where no file will be created.

`out.weight.file` file name to store the estimated weights. default to 'NULL' where no file will be created.

`log2` flag indicating if the input mixture signals are in log2 scale. Default to 'TRUE'.

`l,u` values for the lower- (l) and upper- (u) bound used in setting the vector for values of b0 (bvec for solve.QP) in solving for quadratic programming. Defaults to 0 and 2^34 respectively.

`meq` default to zero (used to set meq for solve.QP)

Details

This is the mail function (an interface) to implement digital sorting algorithm for cell type specific gene expression profiles, by proving minimum of an data matrix for mixture signal and cell-gene key tables. You may provide specific parameters for weight estimation and deconvolution.

Value

A list object containing two elements:

est.weight	a list of estimated weight ('estimated_weight') and the model's mean square error ('mse'). estimated_weight is a matrix of cell types in row and samples in columns. mse defaults to 'NULL' if method is not 'LM'.
deconv	a matrix of deconvoluted signals: genes in row, cell types in columns.

See Also

EstimateWeight, Deconvolution, GSM_QP, solve.QP,

Examples

```
## load package DSA
library(DSA)

## load sample data
data('mix.signals')
data('cell.gene')
data('weight')

# In this data, thr first three samples contain signal from only one cell
pure <- mix[, 1:3]
mix <- mix[, 4:14]

weight.orig <- weight
weight <- weight[4:14, ]

## Exp 1: Obtain the cell-type specific signals estimated from DSA with default parameters:
## estimated weights and linear model in deconvolution
decov.LM <- DSA(mix, cell.gene)

## Exp 2: Obtain the cell-type specific signals estimated from DSA with estimated
## weight and quadratic programming
dec.QLM <- DSA(mix, cell.gene, method="QP_LM")

# check cell type proportions
t.weight <- as.numeric(weight)/100
e.weight <- as.numeric(t(dec.QLM$est.weight))
limits <- range(t.weight, e.weight)

plot(t.weight, e.weight, xlim=limits, pch=19, ylim=limits)
cor(t.weight, e.weight)
abline(a=0, b=1)

# check the estimated cell-type specific signals
par(mfrow=c(1,3))
for(i in 1:3){
print(cor(2^(pure[,i]),2^(dec.QLM$deconv[,i])))
}
```

EstimateWeight *EstimateWeight function*

Description

Function to estimate the weight matrix. Based on the set of marker genes for each cell type, this function estimate the cell-specific proportions (weight) for each sample.

Usage

```
EstimateWeight(mix_ob, gene_list, method = "LM", l = 0, u = 2^34)
```

Arguments

`data` data matrix of the mixture signals, with genes in row, cell type in column, in anti-log scale.

`mix_ob` data matrix of the mixture signals, with genes in row, cell type in column, in anti-log scale.

`gene_list` list of the length in the number of tissue types. Each list element contains gene symbols representing the tissue type.

`method` methods used in estimating true signals for each tissue type. Default to 'LM' for linear regression. Other methods included 'QP_LM' (quadradic programming with constraint on the estimated parameter on linear regression)

`l,u` values for the lower- (l) and upper- (u) bound used in setting the vector for values of b0 (bvec for solve.QP) in solving for quadratic programming. Defaults to 0 and 2^34 respectively.

Value

A list object containing two elements:

`weight` matrix of estimated weight, with cell type in row, tissue types in column.

`mse` means square error of the fitted linear model. mse is 'NULL' if method is 'QP_LM'.

See Also

`GSM_QP`, `solve.QP`

Examples

```
## load package DSA
library(DSA)

## load sample data
data('mix.signals')
data('cell.gene')
data(’weight’)

# In this data, thr first three samples contain signal from only one cell
pure <- mix[, 1:3]
mix <- mix[, 4:14]

weight.orig <- weight
weight <- weight[4:14, ]

# Generate a list object set of marker genes for each cell type
unique_cell_type <- unique(cell.gene[,2])
gene_list <- list()
for( i in 1 : length(unique_cell_type)){
gene_list[[i]] <- cell.gene[cell.gene[,2] == unique_cell_type[i],1]
}
names(gene_list) <- unique_cell_type

# Estimate weight matrix
estimated_weight <- EstimateWeight(2^mix, gene_list, method=”LM”)

t.weight <- as.numeric(weight)/100
e.weight <- as.numeric(t(estimated_weight$weight))
limits <- range(t.weight, e.weight)

plot(t.weight, e.weight, xlim=limits, pch=19, ylim=limits)
cor(t.weight, e.weight)
abline(a=0, b=1)
```

GSM_QP *GSM_QP function*

Description

Function to estimate the true signals for each tissue type with quadratic programming.

Usage

```
GSM_QP(ob, weight, l = 0, u = 2^34, meq =0)
```

Arguments

`ob` data matrix of the mixture signals (in anti-log scale), with genes in row, cell type in column.

`weight` weight matrix, with cell type in column, tissue types in row.

`l,u` values for the lower- (l) and upper- (u) bound used in setting the vector for values of b0 (bvec for solve.QP) in solving for quadratic programming. Defaults to 0 and 2^34 respectively.

`meq` default to zero (used to set meq for solve.QP)

Details

This functions depends on the solve.QP function from quadprog package. Its major job is to set up values for the needed parameters call solve.QP to obtain the solution.

Value

`sol`	estimated true signals for each gene (in row) in each cell type (in column).

Examples

```
## load package DSA
library(DSA)

## load sample data
data('mix.signals')
data('cell.gene')
data('weight')

# In this data, thr first three samples contain signal from only one cell
pure <- mix[, 1:3]
mix <- mix[, 4:14]

weight <- weight[4:14, ]

data <- as.matrix(2^mix)
estimate_weight <- as.matrix(weight)
estimate_weight <- estimate_weight/rowSums(estimate_weight)

# Obtain deconvolution
paraM <- GSM_QP(data, estimate_weight, l = min(data), u = max(data) , meq =0)

# check the estimated cell-type specific signals
for(i in 1:3){
print(cor(2^(pure[,i]),paraM[,i]))
}
```

`mix.signals` *mixture expression signals data matrix*

Description

matrix of mixture signals (genes in row, cell type in column) derived from the public microarray data GSE19830. The triplicates were averaged and log2-transformed.

Usage

```
data('mix.signals')
```

Format

A matrix of 31099 rows (probes) and 14 columns (samples/tissues)

References

Shen-Orr SS, Tibshirani R, Khatri P, Bodian DL et al. Cell type-specific gene expression differences in complex tissues. Nat Methods 2010 Apr;7(4):287-9. PMID: 20208531

`weight` *weight data matrix*

Description

This is a sample matrix for the weight proportions for three cell types: lung, liver, brain. The proportions of each specific cell type is in columns and samples in rows.

Usage

```
data('weight')
```

Format

A data matrix with 3 columns and 14 rows.

Index