Package ‘DeMixT’

October 16, 2019

Title Cell type-specific deconvolution of heterogeneous tumor samples with two or three components using expression data from RNAseq or microarray platforms

Version 1.0.4

Date 2019-08-11

Author Zeya Wang <zw17.rice@gmail.com>, Wenyi Wang <wwang7@@mdanderson.org>

Maintainer Zeya Wang <zw17.rice@gmail.com>, Fan Gao <fgao3@mdanderson.org>

Description DeMixT is a software package that performs deconvolution on transcriptome data from a mixture of two or three components.

LazyData TRUE

Depends R (>= 3.6), parallel, SummarizedExperiment, knitr, KernSmooth

Imports matrixStats, stats

VignetteBuilder knitr

NeedsCompilation yes

biocViews Software, StatisticalMethod, Classification, GeneExpression, Sequencing, Microarray, TissueMicroarray, Coverage

License GPL-3

git_url https://git.bioconductor.org/packages/DeMixT

git_branch RELEASE_3_9

git_last_commit 70e4f40

git_last_commit_date 2019-08-15

Date/Publication 2019-10-15

R topics documented:

`DeMixT` *Deconvolution of heterogeneous tumor samples with two or three components using expression data from RNAseq or microarray platforms*

Description

DeMixT is a software that performs deconvolution on transcriptome data from a mixture of two or three components.

Usage

```
DeMixT(data.Y, data.comp1, data.comp2 = NULL, niter = 10, nbin = 50,
if.filter = TRUE, ngene.selected.for.pi = 250, mean.diff.in.CM = 0.25,
tol = 10^(-5), output.more.info = FALSE,
nthread = parallel::detectCores() - 1)
```

Arguments

`data.Y` A SummarizedExperiment object of expression data from mixed tumor samples. It is a G by Sy matrix where G is the number of genes and Sy is the number of mixed samples. Samples with the same tissue type should be placed together in columns.

`data.comp1` A SummarizedExperiment object of expression data from reference component 1 (e.g., normal). It is a G by $S1$ matrix where G is the number of genes and $S1$ is the number of samples for component 1.

`data.comp2` A SummarizedExperiment object of expression data from additional reference samples. It is a G by $S2$ matrix where G is the number of genes and $S2$ is the number of samples for component 2. Component 2 is needed only for running a three-component model.

`niter` The maximum number of iterations used in the algorithm of iterated conditional modes (ICM, Ref[1]). A larger value better guarantees the convergence in estimation but increases the running time. The default is 10.

`nbin` The number of bins used in numerical integration for computing complete likelihood. A larger value increases accuracy in estimation but increases the running time, especially in a three-component deconvolution problem. The default is 50.

`if.filter` The logical flag indicating whether a predetermined filter rule is used to select genes for proportion estimation. The default is TRUE.

`ngene.selected.for.pi`
The percentage or the number of genes used for proportion estimation. The difference between the expression levels from mixted tumor samples and the known component(s) are evaluated, and the most differentially expressed genes are selected. It is enabled when if.filter = TRUE. The default is 250.

`mean.diff.in.CM`
: Threshold of expression difference in selecting genes in the component merging strategy. We merge three-component to two-component by selecting genes with similar expressions for the two known components. Genes with the mean differences less than the threshold will be selected for component merging. It is used in the three-component setting, and is enabled when if.filter = TRUE. The default is 0.25.

`tol`
: The convergence criterion. The default is 10^(-5).

`output.more.info`
: The logical flag indicating whether to show the estimated proportions in each iteration in the output.

`nthread`
: The number of threads used for deconvolution when OpenMP is availble in the system. The default is the number of whole threads minus one. In our no-OpenMP version, it is set to 1.

Value

`pi`
: Matrix of estimated proportions for each known component. $\pi 1$ corresponds to the proportion estimate for the first known component. $\pi 2$ corresponds to the second known component.

`pi.iter`
: Estimated proportions in each iteration. It is a $numberofiterationXSyX1$ array in two-component setting, and a $number_of_iterationXSyX2$ array in three-component setting. This is enabled only when output.more.info = TRUE.

`ExprT`
: Matrix of deconvolved expression profiles corresponding to T-component in mixed samples for a given subset of genes. Each row corresponds to one gene and each column corresponds to one sample.

`ExprN1`
: Matrix of deconvolved expression profiles corresponding to N1-component in mixed samples for a given subset of genes. Each row corresponds to one gene and each column corresponds to one sample.

`ExprN2`
: Matrix of deconvolved expression profiles corresponding to N2-component in mixed samples for a given subset of genes in a three-component setting. Each row corresponds to one gene and each column corresponds to one sample.

`Mu`
: Estimated μ of log2-normal distribution for both known ($MuN1, MuN2$) and unknown component (MuT).

`Sigma`
: Estimated σ of log2-normal distribution for both known ($SigmaN1, SigmaN2$) and unknown component ($SigmaT$).

`gene.name`
: The names of genes used in estimating the proportions. If no gene names are rpovided in the original data set, the genes will be automatically indexed. This is enabled only when output.more.info = TRUE.

Author(s)

Zeya Wang, Wenyi Wang

References

J. Besag. "On the statistical analysis of dirty pictures". In: Journal of the Royal Statistical Society. Series B (Methodological) (1986), pp. 259-302.

See Also

http://bioinformatics.mdanderson.org/main/DeMixT

Examples

```
# Example 1: simulated two-component data
data(test.data1.y)
data(test.data1.comp1)
res <- DeMixT(data.Y = test.data1.y, data.comp1 = test.data1.comp1,
if.filter = FALSE, output.more.info = TRUE)
res$pi
head(res$ExprT, 3)
head(res$ExprN1, 3)
head(res$Mu, 3)
head(res$Sigma, 3)
res$pi.iter
res$gene.name

# Example 2: simulated three-component data
# It takes about 15 minutes to finish running
# data(test.data2.y)
# data(test.data2.comp1)
# data(test.data2.comp2)
# res <- DeMixT(data.Y = test.data2.y, data.comp1 = test.data2.comp1,
#               data.comp2 = test.data2.comp2, if.filter = FALSE)

# Example 3: three-component mixed cell line data applying

# component merging strategy
# It takes about 1.5 hours to finish running
# data(test.data3.y)
# data(test.data3.comp1)
# data(test.data3.comp2)
# res <- DeMixT(data.Y = test.data3.y, data.comp1 = test.data3.comp1,
#               data.comp2 = test.data3.comp2, if.filter = TRUE)

# Example: convert a matrix into the SummarizedExperiment format
# library(SummarizedExperiment)
# example <- matrix(c(1, 2, 3, 4, 5, 6), nrow = 2, ncol = 3, byrow = TRUE)
# example.se <- SummarizedExperiment(assays = list(counts = example))
```

DeMixT_S1 *Estimates the proportions of mixed samples for each mixing component*

Description

This function is designed to estimate the proportions of all mixed samples for each mixing component with or without component merging.

Usage

```
DeMixT_S1(data.Y, data.comp1, data.comp2 = NULL, niter = 10, nbin = 50,
if.filter = FALSE, ngene.selected.for.pi = 250,
mean.diff.in.CM = 0.25, tol = 10^(-5),
nthread = parallel::detectCores() - 1)
```

Arguments

`data.Y` A SummarizedExperiment object of expression data from mixed tumor samples. It is a G by Sy matrix where G is the number of genes and Sy is the number of mixed samples. Samples with the same tissue type should be placed together in columns.

`data.comp1` A SummarizedExperiment object of expression data from reference component 1 (e.g., normal). It is a G by $S1$ matrix where G is the number of genes and $S1$ is the number of samples for component 1.

`data.comp2` A SummarizedExperiment object of expression data from additional reference samples. It is a G by $S2$ matrix where G is the number of genes and $S2$ is the number of samples for component 2. Component 2 is needed only for running a three-component model.

`niter` The maximum number of iterations used in the algorithm of iterated conditional modes (ICM, Ref[1]). A larger value better guarantees the convergence in estimation but increases the running time. The default is 10.

`nbin` The number of bins used in numerical integration for computing complete likelihood. A larger value increases accuracy in estimation but increases the running time, especially in a three-component deconvolution problem. The default is 50.

`if.filter` The logical flag indicating whether a predetermined filter rule is used to select genes for proportion estimation. The default is TRUE.

`ngene.selected.for.pi`
The percentage or the number of genes used for proportion estimation. The difference between the expression levels from mixted tumor samples and the known component(s) are evaluated, and the most differentially expressed genes are selected. It is enabled when if.filter = TRUE. The default is 250.

`mean.diff.in.CM`
Threshold of expression difference in selecting genes in the component merging strategy. We merge three-component to two-component by selecting genes with similar expressions for the two known components. Genes with the mean differences less than the threshold will be selected for component merging. It is used in the three-component setting, and is enabled when if.filter = TRUE. The default is 0.25.

`tol` The convergence criterion. The default is 10^(-5).

`nthread` The number of threads used for deconvolution when OpenMP is availble in the system. The default is the number of whole threads minus one. In our no-OpenMP version, it is set to 1.

Value

`pi` Matrix of estimated proportions for each known component. $\pi1$ corresponds to the proportion estimate for the first known component. $\pi2$ corresponds to the second known component.

`pi.iter` Estimated proportions in each iteration. It is a $numberofiterationXSyX1$ array in two-component setting, and a $number_of_iterationXSyX2$ array in three-component setting. This is enabled only when output.iter = TRUE.

`gene.name` The names of genes used in estimating the proportions. If no gene names are rpovided in the original data set, the genes will be automatically indexed.

Author(s)

Zeya Wang, Wenyi Wang

References

J. Besag. "On the statistical analysis of dirty pictures". In: Journal of the Royal Statistical Society. Series B (Methodological) (1986), pp. 259-302.

See Also

http://bioinformatics.mdanderson.org/main/DeMixT

Examples

```
# Example 1: estimate proportions for simulated two-component data
data(test.data1.y)
data(test.data1.comp1)
res <- DeMixT_S1(data.Y = test.data1.y,
data.comp1 = test.data1.comp1, if.filter = FALSE)

# Example 2: estimate proportions for simulated three-component data
# This example takes 10 minutes to finish running
# data(test.data2.y)
# data(test.data2.comp1)
# data(test.data2.comp2)
# res <- DeMixT_S1(data.Y = test.data2.y, data.comp1 = test.data2.comp1,
#                  data.comp2 = test.data2.comp2, if.filter = FALSE)

# Example 3: estimate proportions for simulated three-component
# mixed cell line data
# This example takes 1 hour to finish running
# data(test.data2.y)
# data(test.data3.comp1)
# data(test.data3.comp2)
# res <- DeMixT_S1(data.Y = test.data3.y, data.comp1 = test.data3.comp1,
#                  data.comp2 = test.data3.comp2)
```

DeMixT_S2 *Deconvolves expressions of each individual sample for unknown component*

Description

This function is designed to estimate the deconvolved expressions of individual mixed tumor samples for unknown component for each gene.

Usage

```
DeMixT_S2(data.Y, data.comp1, data.comp2 = NULL, givenpi, nbin = 50,
nthread = parallel::detectCores() - 1)
```

Arguments

`data.Y` A SummarizedExperiment object of expression data from mixed tumor samples. It is a G by Sy matrix where G is the number of genes and Sy is the number of mixed samples. Samples with the same tissue type should be placed together in columns.

`data.comp1` A SummarizedExperiment object of expression data from reference component 1 (e.g., normal). It is a G by $S1$ matrix where G is the number of genes and $S1$ is the number of samples for component 1.

`data.comp2` A SummarizedExperiment object of expression data from additional reference samples. It is a G by $S2$ matrix where G is the number of genes and $S2$ is the number of samples for component 2. Component 2 is needed only for running a three-component model.

`givenpi` A vector of proportions for all mixed tumor samples. In two-component analysis, it gives the proportions of the known reference component, and in three-component analysis, it gives the proportions for the two known components.

`nbin` The number of bins used in numerical integration for computing complete likelihood. A larger value increases accuracy in estimation but increases the running time, especially in a three-component deconvolution problem. The default is 50.

`nthread` The number of threads used for deconvolution when OpenMP is availble in the system. The default is the number of whole threads minus one. In our no-OpenMP version, it is set to 1.

Value

`decovExprT` Matrix of deconvolved expression profiles corresponding to T-component in mixed samples for a given subset of genes. Each row corresponds to one gene and each column corresponds to one sample.

`decovExprN1` Matrix of deconvolved expression profiles corresponding to N1-component in mixed samples for a given subset of genes. Each row corresponds to one gene and each column corresponds to one sample.

`decovExprN2` Matrix of deconvolved expression profiles corresponding to N2-component in mixed samples for a given subset of genes in a three-component setting. Each row corresponds to one gene and each column corresponds to one sample.

`decovMu` Estimated μ of log2-normal distribution for both known ($MuN1, MuN2$) and unknown component (MuT).

`decovSigma` Estimated σ of log2-normal distribution for both known ($SigmaN1$, $SigmaN2$) and unknown component ($SigmaT$).

Author(s)

Zeya Wang, Wenyi Wang

References

J. Besag. "On the statistical analysis of dirty pictures". In: Journal of the Royal Statistical Society. Series B (Methodological) (1986), pp. 259-302.

See Also

http://bioinformatics.mdanderson.org/main/DeMix:Overview

Examples

```
# Example 1: two-component deconvolution given proportions
data(test.data1.truth)
data(test.data1.y)
data(test.data1.comp1)
givenpi <- c(t(as.matrix(test.data1.truth[-2,])))
res <- DeMixT_S2(data.Y = test.data1.y,
data.comp1 = test.data1.comp1, givenpi = givenpi)

# Example 2: three-component deconvolution given proportions
# This example takes 10 minutes to finish running
#  data(test.data2.truth)
#  data(test.data2.y)
#  data(test.data2.comp1)
#  data(test.data2.comp2)
#  givenpi <- c(t(test.data2.truth[-3,]))
#  res <- DeMixT_S2(data.Y = test.data2.y, data.comp1 = test.data2.comp1,
#                   data.comp2 = test.data2.comp2, givenpi = givenpi)
```

Optimum_KernelC — *Kernel function for optimizing parameters and hidden variables in DeMixT*

Description

This function is invoked by DeMixT_S1 and DeMixT_S2 to finish parameter estimation and expression deconvolution.

Usage

```
Optimum_KernelC(inputdata, groupid, nhavepi, givenpi, givenpiT,
niter, ninteg, tol,
sg0 = 0.5^2, mu0 = 0.0, nthread = 1)
```

Arguments

`inputdata` A matrix of expression data (e.g gene expressions) from reference (e.g. normal) and mixed samples (e.g. mixed tumor samples). It is a GxS matrix where G is the number of genes and S is the number of samples including reference and mixed samples. Samples with the same tissue type should be placed together in columns (e.g. cbind(normal samples, mixed tumor samples)

`groupid` A vector of indicators to denote if the corresponding samples are reference samples or mixed tumor samples. DeMixT is able to deconvolve mixed tumor samples with at most three components. We use 1 and 2 to denote the samples referencing the first and the second known component in mixed tumor samples. We use 3 to indicate mixed tumor samples prepared to be deconvolved. For example, in two-component deconvolution, we have c(1,1,...,3,3) and in three-component deconvolution, we have c(1,1,...,2,2,...,3,3).

`nhavepi` If it is set to 0, then deconvolution is performed without any given proportions; if set to 1, deconvolution with given proportions for the first and the second known component is run; if set to 2, deconvolution is run with given tumor proportions. This option helps to do deconvolution in different settings. Because in estimation of component-specific proportions, we just use a subset of genes ; so when it is required to deconvolve another subset of genes, we just easily plug back our estimated proportions by setting this option to 1. In our two-step estimation strategy in a three-component setting, this option is set to 2 to implement the second step.

`givenpi` S_{T_N}-Vector of proportions. Given the number of mixed tumor samples is $S_T(S_T < S)$, S_{T_N} is set to $2 * S_T$ in a three-component setting and S_T in a two-component setting. When nhavepi is 1, it is fixed with the given proportions for the first and the second known component of mixed tumor samples, or just for one known component when there is just one type of reference tissues. It has the form of Vector $\pi_{N_1}^1$,$pi_{N_1}^2$,...,$\pi_{N_1}^{S_T}$, $\pi_{N_2}^1$,$\pi_{N_2}^2$,...,$\pi_{N_2}^{S_T}$.

`givenpiT` S_T-Vector of proportions. When nhavepi is set to 2, givenpiT is fixed with given proportions for unknown component of mixed tumor samples. This option is used when we adopt a two-step estimation strategy in deconvolution. It has the form of Vector π_T^1, π_T^2, $\cdots$, $\pi_T^{S_T}$. If option is not 2, this vector can be given with any element.

`niter` The number of iterations used in the algorithm of iterated conditional modes. A larger value can better guarantee the convergence in estimation but increase the computation time.

`ninteg` The number of bins used in numerical integration for computing complete likelihood. A larger value can increase accuracy in estimation but also increase the running time. Especially in three-component deconvolution, the increase of number of bins can greatly lengthen the running time.

`tol` The convergence criterion. The default is 10^(-5).

`nthread` The number of threads used for deconvolution when OpenMP is availble in the system. The default is the number of whole threads minus one. In our no-OpenMP version, it is set to 1.

`sg0` Initial value for σ. The default is 0.5^2.

`mu0` Initial value for μ. The default is 0.

Value

`pi` Matrix of estimated proportions for each known component. The first row corresponds to the proportion estimate of each sample for the first known component (groupid = 1) and the second row corresponds to that for the second known component (groupid = 2)

`decovExpr` A matrix of deconvolved expression profiles corresponding to unknown (e.g tumor) component in mixed samples for a given subset of genes. Each row corresponds to one gene and each column corresponds to one sample.

`decovMu` Estimated μ of log2-normal distribution for tumor component.

`decovSigma` Estimated σ of log2-normal distribution for tumor component

`pi1` An $S_T x I$ Matrix of estimated proportions for each iteration $i \in \{1, \cdots, I\}$ for the first known component

`pi2` An $S_T x I$ Matrix of estimated proportions for each iteration $i \in \{1, \cdots, I\}$ for the second known component

Author(s)

Zeya Wang, Wenyi Wang

See Also

http://bioinformatics.mdanderson.org/main/DeMixT

Examples

```
# Example 1: simulated two-component data
data.comp1 <- SummarizedExperiment::assays(test.data1.comp1)[[1]]
data.Y <- SummarizedExperiment::assays(test.data1.y)[[1]]
inputdata <- cbind(data.comp1, data.Y)
groupid <- c(rep(1, ncol(data.comp1)), rep(3, ncol(data.Y)))
Optimum_KernelC(inputdata, groupid, nhavepi = 0,
givenpi = rep(0, 2 * ncol(data.y)),
givenpiT = rep(0, ncol(data.y)),
niter = 10, ninteg = 30, tol = 10^(-4))
```

`test.data1.comp1` *simulated two-component test data*

Description

simulated two-component test data used in function DeMixT

Usage

```
test.data1.comp1
```

Format

A SummarizedExperiment object of expression data from reference component 1

Examples

```
data(test.data1.comp1)
test.data1.comp1
```

`test.data1.truth` *simulated two-component test data*

Description

simulated two-component test data used in function DeMixT

Usage

```
test.data1.truth
```

Format

A SummarizedExperiment object of true proportions, i.e., π_1 and $1 - \pi_1$

Examples

```
data(test.data1.truth)
test.data1.truth
```

`test.data1.y` *simulated two-component test data*

Description

simulated two-component test data used in function DeMixT

Usage

```
test.data1.y
```

Format

A SummarizedExperiment object of expression data from mixed tumor samples

Examples

```
data(test.data1.y)
test.data1.y
```

`test.data2.comp1` *simulated three-component test data*

Description

simulated three-component test data used in function DeMixT

Usage

```
test.data2.comp1
```

Format

A SummarizedExperiment object of expression data from reference component 1

Examples

```
data(test.data2.comp1)
test.data2.comp1
```

`test.data2.comp2` *simulated three-component test data*

Description

simulated three-component test data used in function DeMixT

Usage

```
test.data2.comp2
```

Format

A SummarizedExperiment object of expression data from reference component 2

Examples

```
data(test.data2.comp2)
test.data2.comp2
```

`test.data2.truth` *simulated three-component test data*

Description

simulated three-component test data used in function DeMixT

Usage

```
test.data2.truth
```

Format

A SummarizedExperiment object of true proportions, i.e., π_1 and $1 - \pi_1$

Examples

```
data(test.data2.truth)
test.data2.truth
```

`test.data2.y` *simulated three-component test data*

Description

simulated three-component test data used in function DeMixT

Usage

```
test.data2.y
```

Format

A SummarizedExperiment object of expression data from mixed tumor samples

Examples

```
data(test.data2.y)
test.data2.y
```

`test.data3.comp1` *three-component mixed cell line test data*

Description

three-component mixed cell line test data used in function DeMixT

Usage

```
test.data3.comp1
```

Format

A SummarizedExperiment object of expression data from reference component 1

Examples

```
data(test.data3.comp1)
test.data3.comp1
```

`test.data3.comp2` *three-component mixed cell line test data*

Description

three-component mixed cell line test data used in function DeMixT

Usage

```
test.data3.comp2
```

Format

A SummarizedExperiment object of expression data from reference component 2

Examples

```
data(test.data3.comp2)
test.data3.comp2
```

`test.data3.truth` *three-component mixed cell line test data*

Description

three-component mixed cell line test data used in function DeMixT

Usage

```
test.data3.truth
```

Format

A SummarizedExperiment object of true proportions, i.e., π_1 and $1 - \pi_1$

Examples

```
data(test.data3.truth)
test.data3.truth
```

test.data3.y *three-component mixed cell line test data*

Description

three-component mixed cell line test data used in function DeMixT

Usage

```
test.data3.y
```

Format

three-component mixed cell line test data used in function DeMixT

Examples

```
data(test.data3.y)
test.data3.y
```

Index