Package ‘EDDA’

October 7, 2014

Type Package

Title Experimental Design in Differential Abundance analysis

Version 1.0.1

Date 2014-07-1

Author Li Juntao, Luo Huaien, Chia Kuan Hui Burton, Niranjan Nagarajan

Maintainer Li Juntao<`lij9@gis.a-star.edu.sg`>, Luo Huaien<`luoh2@gis.a-star.edu.sg`>, Niranjan Nagarajan <`nagarajann@gis.a-star.edu.sg`>

Description
EDDA can aid in the design of a range of common experiments such as RNA-seq, Nanostring assays, RIP-seq and Metagenomic sequencing, and enables researchers to comprehensively investigate the impact of experimental decisions on the ability to detect differential abundance.

License GPL (>= 2)

Depends Rcpp (>= 0.10.4),parallel,methods,ROCR,DESeq,baySeq,snow,edgeR

Imports graphics, stats, utils, parallel, methods, ROCR, DESeq,baySeq, snow, edgeR

LinkingTo Rcpp

biocViews Sequencing, ExperimentalDesign, Normalization, RNASeq,ChIPSeq

R topics documented:

`EDDA-package` *Experimental Design in Differential Abundance analysis*

Description

EDDA aids in the design of a range of common experiments including RNA-seq, Nanostring assays, RIP-seq and Metagenomic sequencing, and enables researchers to comprehensively investigate the impact of experimental decisions on the ability to detect differential abundance.

Details

Package:	EDDA
Type:	Package
Version:	0.99.2
Date:	2014-02-12
License:	GPL (>= 2)

generateData() testDATs() computeAUC() plotROC() plotPRC()

Author(s)

Li Juntao, Luo Huaien, Chia Kuan Hui Burton, Niranjan Nagarajan

Maintainer: Li Juntao<lij9@gis.a-star.edu.sg>, Luo Huaien<luoh2@gis.a-star.edu.sg>, Niranjan Nagarajan <nagarajann@gis.a-star.edu.sg>

References

Luo Huaien, Li Juntao,Chia Kuan Hui Burton, Shyam Prabhakar, Paul Robson, Niranjan Nagarajan, The importance of study design for detecting differentially abundant features in high-throughput experiments, under review.

Examples

```
data <- generateData(EntityCount=500)
test.obj <- testDATs(data,DE.methods=c("DESeq","edgeR"),nor.methods="default")
auc.obj  <- computeAUC(test.obj)
plotROC(auc.obj)
plotPRC(auc.obj)
```

computeAUC *compute AUC values.*

Description

compute AUC values for each test.

Usage

```
computeAUC(obj,cutoff=1,numCores=10,
DE.methods=c("Cuffdiff","DESeq","baySeq","edgeR","MetaStats","NOISeq"),
nor.methods=c("default","Mode","UQN","NDE"))
```

Arguments

`obj`	Object from testDATs().
`cutoff`	cutoff for ROC curve. Default is 1.
`numCores`	Number of cores for parallelization. Default is 10.
`DE.methods`	Method list for differential abundance tests. Methods currently available include "Cuffdiff","DESeq", "baySeq","edgeR","MetaStats","NOISeq".
`nor.methods`	Normalization method list. Methods currently available include "default"(default normalization for each DE method), "Mode"(Mode normalization),"UQN"(Upper quartile normalization),"NDE"(non-differential expression normalization).

Author(s)

Li Juntao, and Luo Huaien

References

Luo Huaien, Li Juntao,Chia Kuan Hui Burton, Shyam Prabhakar, Paul Robson, Niranjan Nagarajan, The importance of study design for detecting differentially abundant features in high-throughput experiments, under review.

Examples

```
data <- generateData(EntityCount=200)
test.obj <- testDATs(data,DE.methods="DESeq",nor.methods="default")
auc.obj  <- computeAUC(test.obj)
```

`generateData` *generate count data*

Description

Simulate count data using different models and settings.

Usage

```
generateData(SimulModel="Full", SampleVar="medium",
ControlRep=5, CaseRep=ControlRep, EntityCount=1000, FC="Norm(2,1)",
perDiffAbund=0.1, upPDA=perDiffAbund/2, downPDA=perDiffAbund/2,
numDataPoints=100, AbundProfile = "HBR", modelFile = NULL, minAbund=10,varLibsizes=0.1,
inputCount=NULL,inputLabel=NULL,SimulType="auto")
```

Arguments

`SimulModel`	Simulation model used. Default is "Full". SimulModel="NegBinomial" is negtive binomial model which generates data using negtive binomial distribution. SimulModel="Multinom" is multinomial model which generates data mimicking the multinomial sampling process. SimulModel="Full" is a model which combines "NegBinomial" and "Multinom". SimulModel="ModelFree" uses model free approach to generate data by sub-sampling counts from modelFile (if modelFile != NULL) or from input File (if modelFile == NULL)
`SampleVar`	Sample variation: Default is "medium". It could be "low", "medium" and "high" or a real number.
`ControlRep`	Number of replicates for control group. Default is 5.
`CaseRep`	Number of replicates for case group. Default is same as ControlRep.
`EntityCount`	Entity count. Default is 1000.
`FC`	Fold change type. It can be "Norm(mu,sigma)", "logNorm(mu,sigma)", "log2Norm(mu,sigma)" or "Unif(a,b)". mu,sigma and a,b need be predefined. Default is "Norm(2,1)".
`perDiffAbund`	Percentage of differential abundance. Default is 0.1.
`upPDA`	Percentage of up-regulated differential abundance. Default is perDiffAbund/2.
`downPDA`	Percentage of down-regulated differential abundance. Default is perDiffAbund/2.
`numDataPoints`	Number of data points. Default is 100.
`AbundProfile`	AbundProfile for average abundance profile. It can be either the different profiles used in the paper ("HBR", "BP" and "Wu") or it can be location of the abundance profile. Default is "HBR".
`modelFile`	Sample data file for model free approach. Default is NULL. If modelFile = NULL, Model Free approach will subsample from Input file. If modelFile = "SingleCell", Model Free approach will subsample from the available single cell RNA-seq data. if modelFile is the name of a count file, this count file will be used as sample file for sub-sampling.

`minAbund`	Minimum abundance cutoff. Default is 10.
`varLibsizes`	Variability between library sizes. Default is 0.1.
`inputCount`	Input count file. Default is NULL. If not NULL, it learns the parameters (modelFile, SampleVar, perDiffAbund, upPDA, and downPDA) from count data.
`inputLabel`	Label of input count file. The label should be sequence of 0 or 1. Default is NULL.
`SimulType`	Simulation type. It is used only when user has pilot data. Default is "auto". SimulType = "auto", all the parameters (EntityCount, ControlRep, CaseRep, numDataPoint and others) are learned from user's pilot data. SimulType = "auto1", ControlRep, CaseRep, numDataPoint are specified by user input; while EntityCount and all others are learned from user's pilot data. SimulType = "auto2", EntityCount, ControlRep, CaseRep, numDataPoint are specified by user input; while all others are learned from user's pilot data.

Value

`count`	Count matrix.
`DiffAbundList`	Differential abundance list.
`dataLabel`	Data label.

Author(s)

Li Juntao, Luo Huaien, Chia Kuan Hui Burton, Niranjan Nagarajan

References

Luo Huaien, Li Juntao,Chia Kuan Hui Burton, Shyam Prabhakar, Paul Robson, Niranjan Nagarajan, The importance of study design for detecting differentially abundant features in high-throughput experiments, under review.

Examples

```
# generate data with all default options.
data <- generateData()
dim(data$count)
dim(data$DiffAbundList)
data$dataLabel

# generate data with input count.
x <- matrix(rnbinom(1000*15,size=1,mu=10), nrow=1000, ncol=15);
x.lable=c(rep(0,10),rep(1,5))
x[1:50,11:15] <- x[1:50,11:15]*10
x.name=paste("g",1:1000,sep="");
write.table(cbind(x.name,x),"count.txt",row.names =FALSE, sep =\t)

data <- generateData(inputCount="count.txt",inputLabel=x.lable)
dim(data$count)
dim(data$DiffAbundList)
data$dataLabel
```

```
# or generate data with input count and redefined parameters.
data <- generateData(inputCount="count.txt",inputLabel=x.lable,
                     ControlRep=10,CaseRep=10,EntityCount=3000,SimulType="auto2")
dim(data$count)
dim(data$DiffAbundList)
data$dataLabel
```

`HBRmodel` *Average abundance for RNA-seq data from Human Brain Reference.*

Description

RNA-seq datasets.

Usage

```
data(HBRmodel)
```

Format

Data frames with 17597 observations on the following 2 variables.

`GeneName` a character vector

`Count` a numeric vector

References

Au, K.F., Jiang, H., Lin, L., Xing, Y. & Wong, W.H. Detection of splice junctions from paired end RNA-seq data by SpliceMap. Nucleic Acids Res 38, 4570-4578.

`plotPRC` *plot precision-recall curves*

Description

plot precision-recall curves for each test.

Usage

```
plotPRC(obj,DE.methods=c("Cuffdiff","DESeq","baySeq","edgeR","MetaStats","NOISeq"),
nor.methods=c("default","Mode","UQN","NDE"),
plot_type = "o",plot_pch = 20,plot_lwd = 1.75,plot_cex = 1)
```

Arguments

`obj`	Object from testDATs().
`DE.methods`	Method list for differential expression tests. Methods currently available include "Cuffdiff","DESeq","baySeq","edgeR","MetaStats","NOISeq".
`nor.methods`	Normalization method list. Methods currently available include "default"(default normalization for each DE method),"Mode"(Mode normalization),"UQN"(Upper quartile normalization),"NDE"(non-differential expression normalization).
`plot_type`	type option in plot.
`plot_pch`	pch option in plot.
`plot_lwd`	lwd option in plot.
`plot_cex`	cex option in plot.

Author(s)

Li Juntao and Chia Kuan Hui Burton

References

Luo Huaien, Li Juntao,Chia Kuan Hui Burton, Shyam Prabhakar, Paul Robson, Niranjan Nagarajan, The importance of study design for detecting differentially abundant features in high-throughput experiments, under review.

Examples

```
data <- generateData(EntityCount=500)
test.obj <- testDATs(data,DE.methods=c("DESeq","edgeR"),nor.methods="default")
auc.obj  <- computeAUC(test.obj)
plotPRC(auc.obj)
```

`plotROC` *plot Receiver Operating Characteristic curve*

Description

plot Receiver Operating Characteristic curve for each test.

Usage

```
plotROC(obj,DE.methods=c("Cuffdiff","DESeq","baySeq","edgeR","MetaStats","NOISeq"),
nor.methods=c("default","Mode","UQN","NDE"),
plot_type = "o",plot_pch = 20,plot_lwd = 1.75,plot_cex = 1)
```

Arguments

`obj`	Object from testDATs().
`DE.methods`	Method list for differential expression tests. Methods currently available include "Cuffdiff","DESeq","baySeq","edgeR","MetaStats","NOISeq".
`nor.methods`	Normalization method list. Methods currently available include "default"(default normalization for each DE method),"Mode"(Mode normalization),"UQN"(Upper quartile normalization),"NDE"(non-differential expression normalization).
`plot_type`	type option in plot.
`plot_pch`	pch option in plot.
`plot_lwd`	lwd option in plot.
`plot_cex`	cex option in plot.

Author(s)

Li Juntao and Chia Kuan Hui Burton

References

Luo Huaien, Li Juntao,Chia Kuan Hui Burton, Shyam Prabhakar, Paul Robson, Niranjan Nagarajan, The importance of study design for detecting differentially abundant features in high-throughput experiments, under review.

Examples

```
data <- generateData(EntityCount=500)
test.obj <- testDATs(data,DE.methods=c("DESeq","edgeR"),nor.methods="default")
auc.obj  <- computeAUC(test.obj)
plotROC(auc.obj)
```

`SingleCell` *Single-cell RNA-seq data for model free simulation*

Description

Single-cell RNA-seq.

Usage

```
data(SingleCell)
```

Format

Data frames with 51516 rows and 96 columns.

Details This is single-cell RNA-seq data from which counts were generated when using RNA-seq model free approach.

References

In-house data.

`testDATs` *Run differential abundance testings*

Description

Perform differential abundance testing on simulated count data.

Usage

```
testDATs(data, numCores=10, minCountsThreshold=0,
DE.methods=c("Cuffdiff","DESeq","baySeq","edgeR","MetaStats","NOISeq"),
nor.methods=c("default","Mode","UQN","NDE"),method.list=NULL)
```

Arguments

`data`	Data object from generateData() function or predifined data object similar to the output of generateData().
`numCores`	Number of cores for parallelization. Default is 10.
`minCountsThreshold`	Minimum counts threshold for filtering. Default is 0.
`DE.methods`	Method list for differential expression tests. Methods currently available include "Cuffdiff","DESeq","baySeq","edgeR","MetaStats","NOISeq".
`nor.methods`	Normalization method list. Methods currently available include "default" (default normalization for each DE method),"Mode"(Mode normalization), "UQN"(Upper quartile normalization), "NDE"(non-differential expression normalization).
`method.list`	The method list for the combination of DE.methods and nor.methods. Default is NULL.

Value

`data`	Data object from generateData() function.
`filterCounts`	filtered count data.
`Cuffdiff`	Result form Cuffdiff with default normalization.
`Cuffdiff_uqn`	Result form Cuffdiff with Upper quartile normalization normalization.
`Cuffdiff_Mode`	Result form Cuffdiff with Mode normalization.
`Cuffdiff_nde`	Result form Cuffdiff with non-differential expression normalization.
`DESeq`	Result form DESeq with default normalization.
`DESeq_uqn`	Result form DESeq with Upper quartile normalization normalization.
`DESeq_Mode`	Result form DESeq with Mode normalization.
`DESeq_nde`	Result form DESeq with non-differential expression normalization.

`baySeq`	Result form baySeq with default normalization.
`baySeq_uqn`	Result form baySeq with Upper quartile normalization normalization.
`baySeq_Mode`	Result form baySeq with Mode normalization.
`baySeq_nde`	Result form baySeq with non-differential expression normalization.
`edgeR`	Result form edgeR with default normalization.
`edgeR_uqn`	Result form edgeR with Upper quartile normalization normalization.
`edgeR_Mode`	Result form edgeR with Mode normalization.
`edgeR_nde`	Result form edgeR with non-differential expression normalization.
`MetaStats`	Result form MetaStats with default normalization.
`MetaStats_uqn`	Result form MetaStats with Upper quartile normalization normalization.
`MetaStats_Mode`	Result form MetaStats with Mode normalization.
`MetaStats_nde`	Result form MetaStats with non-differential expression normalization.
`NOISeq`	Result form NOISeq with default normalization.
`NOISeq_uqn`	Result form NOISeq with Upper quartile normalization normalization.
`NOISeq_Mode`	Result form NOISeq with Mode normalization.
`NOISeq_nde`	Result form NOISeq with non-differential expression normalization.

Author(s)

Li Juntao, Luo Huaien, Chia Kuan Hui Burton, Niranjan Nagarajan

References

Luo Huaien, Li Juntao,Chia Kuan Hui Burton, Shyam Prabhakar, Paul Robson, Niranjan Nagarajan, The importance of study design for detecting differentially abundant features in high-throughput experiments, under review.

Examples

```
data <- generateData(EntityCount=100)
test.obj <- testDATs(data,nor.methods="default")
test.obj <- testDATs(data,DE.methods="DESeq")


# test data with input count.
x <- matrix(rnbinom(1000*15,size=1,mu=10), nrow=1000, ncol=15);
x[1:50,11:15] <- x[1:50,11:15]*10
x.name=paste("g",1:1000,sep="");
write.table(cbind(x.name,x),"count.txt",row.names =FALSE, sep =\t)

x <- read.table("count.txt",head=TRUE,sep=\t)
x.count <- x[,2:16]
x.lable=c(rep(0,10),rep(1,5))
row.names(x.count) <- x[,1]
data <- list(count=x.count,dataLabel=x.lable)
test.obj <- testDATs(data,DE.methods=c("DESeq","edgeR"),nor.methods="default")
```

Index