Package ‘ISLET’

May 29, 2024

Type Package

Title Individual-Specific cell type referencing Tool

Version 1.6.0

Date 2023-09-15

Description ISLET is a method to conduct signal deconvolution for genomics data. It can estimate the individual-specific and cell-type-specific reference panels, when there are multiple samples observed from each subject. It takes the input of the observed mixture data (feature by sample matrix), and the cell type mixture proportions (sample by cell type matrix), and the sample-to-subject information. It can solve for the reference panel on the individual-basis and conduct test to identify cell-type-specific differential expression (csDE) genes. It also improves estimated cell type mixture proportions by integrating personalized reference panels.

License GPL-2

Depends R(>= 4.2.0), Matrix, parallel, BiocParallel, SummarizedExperiment, BiocGenerics, lme4, nnls

Imports stats, methods, purrr, abind

Suggests BiocStyle, knitr, rmarkdown, htmltools, RUnit, dplyr


biocViews Software, RNASeq, Transcriptomics, Transcription, Sequencing, GeneExpression, DifferentialExpression, DifferentialMethylation

Encoding UTF-8

LazyData false

VignetteBuilder knitr

git_url https://git.bioconductor.org/packages/ISLET
git_branch RELEASE_3_19
git_last_commit b173ae9
git_last_commit_date 2024-04-30

Repository Bioconductor 3.19

Date/Publication 2024-05-29


**Contents**

- caseEst ................................. 2
- ctrlEst .................................. 3
- dataPrep .................................. 4
- dataPrepSlope ............................ 5
- GE600 .................................... 6
- GE600age .................................. 7
- imply ..................................... 8
- implyDataPrep .............................. 9
- isletSolve ................................. 10
- isletTest ................................. 12

**Index**

14

---

**caseEst**  

*ISLET reference estimation results extraction*

**Description**

This function, `caseEst`, extracts the estimated reference panels from the case group. It takes one `outputSol` object as the input, and produces a list of matrices containing the estimated reference panels. The length of the list is equal to the number of cell types. For each matrix, it contains the gene by subject reference panel for that specific cell type.

**Usage**

```r
caseEst(res.sol)
```

**Arguments**

- `res.sol`  
  A `outputSol` object from `isletSolve` step.

**Details**

This is the accessor function to extract the estimated reference panels from `isletSolve` step.

**Value**

A list of matrices containing the estimated reference panels. The length of the list is equal to the number of cell types. For each matrix, it contains the gene by subject reference panel for that specific cell type.
**ctrlEst**

Author(s)

Hao Feng <hxf155@case.edu>

Examples

```r
data(GE600)

study123input <- dataPrep(dat_se=GE600_se)
res.sol <- isletSolve(input=study123input)


```

**Description**

This function, `ctrlEst`, extracts the estimated reference panels from the control group. It takes one `outputSol` objects as the input, and produce a list of matrices containing the estimated reference panels. The length of the list is equal to the number of cell types. For each matrix, it contains the gene by subject reference panel for that specific cell type.

**Usage**

```r
ctrlEst(res.sol)
```

**Arguments**

- `res.sol` A `outputSol` objects from `isletSolve` step.

**Details**

This is the accessor function help to extract the estimated reference panels from `isletSolve` step.

**Value**

A list of matrices containing the estimated reference panels. The length of the list is equal to the number of cell types. For each matrix, it contains the gene by subject reference panel for that specific cell type.

**Author(s)**

Hao Feng <hxf155@case.edu>
dataPrep

Description
This function, dataPrep, is a necessary step to make your data ready and acceptable to ISLET. It takes one SummarizedExperiment objects listed below as the input, and produce a list ready to be feed into ISLET downstream deconvolution (function isletSolve) and/or cell type-specific differentially expressed gene testing (function isletTest).

Usage
dataPrep(dat_se)

Arguments

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>dat_se</td>
<td>A SummarizedExperiment object containing both case group and control. It should contain the observed feature by sample matrix, group label (case or ctrl) for each sample, subject IDs for each sample, and cell type proportions for each sample. The feature by sample matrix is stored in the assay, while the remaining components are stored in the colData slot. In the colData slot, the group label should be stored in the first column, subject IDs should be stored in the second column, and the cell type proportions take the remaining columns. Subject IDs need to be sorted before using the dataPrep function.</td>
</tr>
</tbody>
</table>

Details
This is the initial step for using ISLET, to prepare your data input ready for downstream deconvolution (function isletSolve) and/or differentially expressed gene testing (function isletTest). The input data must follow requirements listed above.

Value
dataPrep returns a S4 object, containing elements ready to serve as the input for downstream deconvolution (function isletSolve) and/or differentially expressed gene testing (function isletTest).

Author(s)
Hao Feng <hxf155@case.edu>
dataPrepSlope

Examples

data(GE600)
ls()
## [1] "GE600_se"

study123input <- dataPrep(dat_se=GE600)

dataPrepSlope

Data preparation function specifically for slope effect testing

Description

This function, dataPrepSlope, is a necessary step to make your data ready and acceptable to ISLET slope effect testing. It takes one SummarizedExperiment objects containing both case and control group as the input, and produce a list ready to be feed into ISLET downstream deconvolution (function isletSolve) and/or differentially expressed gene testing (function isletTest).

Usage

dataPrepSlope(dat_se)

Arguments

dat_se A SummarizedExperiment object containing both case and control group. It should contain the observed feature by sample matrix, group label (case or ctrl) for each sample, subject IDs for each sample, a slope variable for each sample, and cell type proportions for each sample. The feature by sample matrix is stored in the count assay, while the remaining components are stored in the colData slot. In the colData slot, the group label (case or ctrl) take the first column, subject IDs take the second column, the slope variable takes the third column, and the cell type proportions take the remaining columns. Subject IDs need to be sorted before using the dataPrepSlope function.

Details

This is the initial step for using ISLET, to prepare your data input ready for downstream cell-type-specific differentially expressed gene testing (function isletTest) with respect to the slope variable. The input data matrices must follow requirements listed above, and samples/subjects must be ordered and match across matrices.

Value

dataPrepSlope returns a $4$ object, containing elements ready to serve as the input for cell-type-specific differentially expressed gene testing (function isletTest).

Author(s)

Hao Feng <hxf155@case.edu>
Examples

data(GE600age)
ls()
## [1] "GE600age_se"

#(1) Data preparation
study456input <- dataPrepSlope(dat_se=GE600age_se)

#(2) [Downstream] Test for slope effect(i.e. age) difference in csDE testing
#result.test <- isletTest(input=study456input)

GE600 ISLET deconvolution example raw input data

Description

GE600 contains the raw example datasets for ISLET. It has the gene expression values, in the form of RNA-seq raw read counts, for 10 genes by 520 sample, with 83 cases and 89 controls, and multiple repeated measurements (i.e. time points) per subject. Data were combined by case/control status, into one single SummarizedExperiment object. These raw example datasets will need to be converted by the dataPrep function, and then they will be ready for downstream deconvolution (function isletSolve) and/or differentially expressed gene testing (function isletTest).

Usage

data(GE600)

Format

One SummarizedExperiment object containing the following elements:

counts A gene expression value dataset, in the form of RNA-seq raw read counts, of 10 genes by 520 sample, with 83 cases and 89 controls, and multiple repeated measurements (i.e. time points) per subject.

colData Sample meta-data. The first column is the group status (i.e. case/ctrl), the second column is the subject ID, shows the relationship between the samples IDs and their subject IDs. The remaining 6 columns (i.e. column 3-8) are the cell type proportions of all samples by their 6 cell types.

Value

One SummarizedExperiment object.
Examples

data(GE600)
ls()
## [1] "GE600_se"

# show GE600_se details
GE600_se
# An object of class "SummarizedExperiment"

# Then, we can proceed to data preparation step, function 'dataPrep' for ISLET. 
# The rest of the deconvolution/csDE analysis will then follow.

---

**GE600age**

*ISLET example datasets for slope variable testing in csDE*

Description

GE600age contains the example input datasets for ISLET’s slope testing function. It has the gene expression values, in the form of RNA-seq raw read counts, for 10 genes by 520 sample, with 83 cases and 89 controls, and multiple repeated measurements (i.e. time points) per subject. Temporal measures are at different age, with an age variable stored in the metadata. This is the main variable-of-interest in downstream testing. Data were combined by case/control status, into one SummarizedExperiment object. These example datasets will need to be converted by the dataPrep function, and then they will be ready for downstream deconvolution (function isletSolve) and/or differentially expressed gene testing (function isletTest).

Usage

data(GE600age)

Format

One SummarizedExperiment object containing the following elements:

- **counts** A gene expression value dataset, in the form of RNA-seq raw read counts, of 10 genes by 520 sample, with 83 cases and 89 controls, and multiple repeated measurements (i.e. time points) per subject.

- **colData** Sample meta-data. The first column is the case/ctrl group status, the second column is the subject ID, shows the relationship between the samples IDs and their subject IDs. The third column is the age variable for each sample, which is the main variable in downstream testing. The remaining 6 columns (i.e. column 4-9) are the cell type proportions of all samples by their 6 cell types.

Value

One SummarizedExperiment object.
imply

Examples

data(GE600age)
ls()
## [1] "GE600age_se"

#show GE600age_se details
GE600age_se
#An object of class "SummarizedExperiment"

#Then, we can proceed to data preparation step, function 'dataPrep' for ISLET.
##The rest of the csDE testing on age(slope) effect will then follow.

Description

This core function, imply, plays a central role in the imply algorithm. It takes the output from data preparation function implyDataPrep, and utilizes linear mixed-effects models to solve for individual-specific and cell-type-specific reference panels. It has parallel computing implemented to enhance computational efficiency.

Usage

imply(dat123)

Arguments

dat123 The list object output from data preparation function implyDataPrep.

Details

imply is a two-step algorithm to enhance cell deconvolution results by employing subject-specific and cell-type-specific (personalized) reference panels.

In step I, personalized reference panels are generated for each subject using linear mixed-effects models. These personalized references are tailored to individual subjects.

In step II, these personalized reference panels replace the population-level signature matrix, typically used in traditional reference-based deconvolution methods. This substitution enables a personalized deconvolution process across all subjects, by employing the non-negative least squares algorithm.

The function returns a list containing personalized reference panels and improved cell deconvolution results.
A list with the estimated components of the `imply` algorithm:

- **p.ref**: The estimated subject-specific and cell-type-specific reference panels. It is an array of dimension $G$ by $K$ by $N$, where $G$ is the total number of genetic features, $K$ is the total number of cell types, and $N$ is the total number of subjects.

- **imply.prop**: The improved cell deconvolution results based on personalized reference panels from `p.ref`. It is a data.frame of $T$ by $K$, where $T$ is the total number of samples across all subjects. Each subject can have a different number of repeatedly measured samples.

**Author(s)**

Guanqun Meng <gxm324@case.edu>

**Examples**

```r
data(GE600)
ls()
## [1] "GE600_se"

(1) Data preparation
dat123 <- implyDataPrep(sim_se=GE600_se)

(2) improved and personalized cell deconvolution
result <- imply(dat123)
str(result)
#List of 2
# $ p.ref : num [1:10, 1:6, 1:172] 0 0 0.952 13.438 19.007 ... 
# ..- attr(*, "dimnames")=List of 3
# .. ..$ : chr [1:10] "gene1" "gene2" "gene3" "gene4" ...
# .. ..$ : chr [1:6] "Bcells" "Tcells_CD4" "Tcells_CD8" "NKcells" ...
# .. ..$ : chr [1:172] "210298" "223361" "228055" "229203" ...
# $ imply.prop: 'data.frame': 520 obs. of 6 variables:
# ..$ Bcells : num [1:520] 0.4806 0.1912 0.0843 0.2177 0 ...
# ..$ Tcells_CD4: num [1:520] 0.0547 0.0271 0 0 0.343 ...
# ..$ Tcells_CD8: num [1:520] 0.2129 0.0507 0.2584 ...
# ..$ NKcells : num [1:520] 0.1114 0.2487 0.153 0 0.0543 ...
# ..$ Mono : num [1:520] 0.0547 0.0271 0 0 0.343 ...
# ..$ Others : num [1:520] 0.353 0.533 0.55 0.732 0.344 ...
```

**Description**

This `implyDataPrep` function serves a necessary step in preparing your data for `imply`. It takes a single input, a SummarizedExperiment object, and generates a formatted S4 object that can be used as input for personalized deconvolution (`imply`).
Usage

\texttt{implyDataPrep(sim\_se)}

Arguments

\texttt{sim\_se} \hspace{1cm} A SummarizedExperiment object should include both case and control groups, in that order. It should contain the observed feature-by-sample matrix, with corresponding group labels (either "case" or "ctrl"), subject IDs, and cell type proportions for each sample (solved by CIBERSORT). The feature-by-sample matrix is stored in the ‘assay’ slot, while the remaining components are stored in the ‘colData’ slot. Within the ‘colData’ slot, ensure that the group label is stored as ‘group’ in the first column, subject IDs are stored as ‘subject\_ID’ in the second column, and the cell type proportions occupy the remaining columns. Please note that subject IDs must be sorted before utilizing the \texttt{implyDataPrep} function.

Details

This is the initial step for preparing your input data for the imply algorithm, making it ready for personalized deconvolution using the \texttt{imply} function. Ensure that your input data adheres to the requirements listed below.

Value

\texttt{implyDataPrep} returns an S4 object containing elements that are prepared to serve as input for downstream deconvolution using the \texttt{imply} function.

Author(s)

Guanqun Meng <gxm324@case.edu>

Examples

\begin{verbatim}
data(GE600)
l3()
## [1] "GE600\_se"

dat123 <- implyDataPrep(sim\_se=GE600\_se)
\end{verbatim}

\texttt{isletSolve} \hspace{1cm} \textit{Solving individual-specific and cell-type-specific reference panels}

Description

This function, \texttt{isletSolve}, is a core function of ISLET. It takes the output from data preparation function \texttt{dataPrep}, and solve for individual-specific and cell-type-specific reference panels. It has parallel computing implemented to speed up the EM algorithm application.
Usage

isletSolve(input, BPPARAM=bpparam())

Arguments

input                  The list object output from data preparation function dataPrep.
BPPARAM              An instance of BiocParallelParam class, e.g. MulticoreParam, SnowParam, SerialParam, to facilitate parallel computing. If using Unix, MulticoreParam is recommended. Customized options within BiocParallelParam class is allowed. If not specified, the default back-end is retrieve.

Details

For both case group and control group, the deconvolution result is a list of length $K$, where $K$ is the number of cell types. For each of the $K$ elements, it is a matrix of dimension $G$ by $N$. It stores the deconvoluted feature ($G$) by subject ($N$) values, for each of the $K$ elements.

Value

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>case.ind.ref</td>
<td>A list of length $K$, where $K$ is the number of cell types. For each of the $K$ elements in this list, it is a feature by subject matrix containing all the feature values (i.e. gene expression values), for case group. It is one of the main products the individual-specific and cell-type-specific solve algorithm.</td>
</tr>
<tr>
<td>ctrl.ind.ref</td>
<td>A list of length $K$, where $K$ is the number of cell types. For each of the $K$ elements in this list, it is a feature by subject matrix containing all the feature values (i.e. gene expression values), for control group. It is one of the main products the individual-specific and cell-type-specific solve algorithm.</td>
</tr>
<tr>
<td>mLLK</td>
<td>A scalar. The log-likelihood from the current model. It can be useful for testing purpose such as Likelihood Ratio Test.</td>
</tr>
</tbody>
</table>

Author(s)

Hao Feng <hxf155@case.edu>

Examples

data(GE600)
lS()
## [1] "GE600_se"

#(1) Data preparation
study123input <- dataPrep(dat_se=GE600_se)

#(2) Individual-specific and cell-type-specific deconvolution
result <- isletSolve(input=study123input)
isletTest

Testing for cell-type-specific Differential Expression (csDE) genes

Description

This function, isletTest, can take the output from data preparation function dataPrep, and test for csDE genes. It uses Likelihood Ratio Test (LRT), iterating all cell types. The output is a matrix of p-values from LRT. It has parallel computing implemented to speed up the EM algorithm application.

Usage

isletTest(input, BPPARAM=bpparam() )

Arguments

input The list object output from data preparation function dataPrep.
BPPARAM An instance of BiocParallelParam class, e.g. MulticoreParam, SnowParam, SerialParam, to facilitate parallel computing. If using Unix, MulticoreParam is recommended. Customized options within BiocParallelParam class is allowed. If not specified, the default back-end is retrieve.

Details

This function implement a LRT, and run individually for each cell type, and then aggregate the results together into a matrix.

Value

A p-value matrix, in the dimension of feature by cell type. Each element is the LRT p-value, by contrasting case group and control group, for one feature in one cell type.

Author(s)

Hao Feng <hxf155@case.edu>

Examples

data(GE600)
ls()
## [1] "GE600_se"

#1 Data preparation
study123input <- dataPrep(dat_se=GE600_se)

#2 [optional for csDE genes testing] Individual-specific and cell-type-specific deconvolution
#result.solve <- isletSolve(input=study123input)
#(3) Test for csDE genes
result.test <- isletTest(input=study123input)
Index

* datasets
  GE600, 6
  GE600age, 7

caseEst, 2
ctrlEst, 3

dataPrep, 4
dataPrepSlope, 5

GE600, 6
GE600_se (GE600), 6
GE600age, 7
GE600age_se (GE600age), 7

imply, 8
implyDataPrep, 9
isletSolve, 10
isletTest, 12