Package 'GEDI'

June 23, 2021

Title Gene Expression Data Integration

Version 1.0

Date 2021-06-23

biocViews

Imports DESeq2,
gcrma,
limma,
biomaRt,
WGCNA,
edgeR,
ggplot2,
progress,
reshape2,
rpart,
stats,
graphics,
utils,
BiocParallel,
Biobase,
SummarizedExperiment,
affy,
curl,
Rdpack

Depends R (>= 4.0.0)

Enhances

Author Mathias N. Stokholm [aut, cre] <MNStokholm@gmail.com>,
Maria Belen Rabaglino [aut],
Haja Kadarmideen [aut] <hajak@dtu.dk>

Maintainer Mathias N. Stokholm <MNStokholm@gmail.com>

Description Enables automatic transcriptomic data integration with a pipeline consisting of four functions that together reads, re-annotates and merges the datasets after which the batch effect is removed. Unsupervised and supervised machine learning methods ensure that the batch correction and data integration was successful.

Encoding UTF-8

RdMacros Rdpack

Roxygen list(markdown = TRUE)

RoxygenNote 7.1.1

R topics documented:

`GEDI-package` *Gene Expression Data Integration*

Description

Enables automatic transcriptomic data integration with a pipeline consisting of four functions that together reads, re-annotates and merges the datasets after which the batch effect is removed. Unsupervised and supervised machine learning methods ensure that the batch correction and data integration was successful.

Details

The main functions are:

- `ReadGE` - reads all the gene expression datasets and stores them in a list
- `GEDI` - maps probe or read IDs to Ensembl gene IDs using the `biomaRt` package and integrates the datasets into one data table
- `BatchCorrection` - removes the batch effect from the integrated dataset
- `VerifyGEDI` - verifies the transcriptomic data integration using one of two supervised machine learning models to predict the samples' status in one batch based on the remaining batches' samples

If you want to read an example pipeline, see the README.md file in the package directory.

The code can be read in the github repository: `https://github.com/s184257/GEDI`

Author(s)

Mathias N. Stokholm [aut, cre] <MNStokholm@gmail.com>, Maria Belen Rabaglino [aut], Haja Kadarmideen [aut] <hajak@dtu.dk>

Maintainer: Mathias N. Stokholm <MNStokholm@gmail.com>

`BatchCorrection` *Remove batch effect from integrated gene expression data*

Description

The `BatchCorrection` function removes the batch effect in an integrated gene expression dataset using functions inspired by `sva::ComBat` and `sva::ComBat_seq` from the `sva` package (Leek et al. 2021). This function also allows users to visualise the data before and after the batch correction with principal component analysis (PCA) plots and boxplots to verify the batch correction.

Usage

```
BatchCorrection(Data, batch, visualise = TRUE, status = NULL)
```

Arguments

`Data`	A `data.frame` or `matrix` of integrated gene expression data with multiple batches. Columns contain samples, and rows are genes.
`batch`	Character vector that describes which sample is in what batch. `length(batch) = ncol(Data)`.
`visualise`	(Optional) Logical value. If `TRUE`, the batch correction is verified using PCA plots and boxplots that visualise the data before and after the batch correction.
`status`	(Optional) Character vector that describes the samples' status, e.g. treated or control. `BatchCorrection` only uses the `status` argument to make plots, i.e. when `visualise = TRUE`.

Details

Bioconductor's `sva` package does not remove batch effect in genes with zero variance or zero counts within any batch. Therefore, the `BatchCorrection` function uses modified versions of the `sva::ComBat` and `sva::ComBat_seq` functions to remove the batch effect in all genes. The modified `ComBat_seq` function corrects data for the batch effect like the modified `ComBat` function, but only for RNA-seq count data.

By setting `visualise = TRUE`, the `BatchCorrection` function verifies the batch correction visually. This produces a PCA plot and a boxplot for the data before and after the batch correction. If the dataset contains count data, the `DESeq2::varianceStabilizingTransformation` function transforms the dataset. This transformation is only used for verification and is not permanent.

The PCA plot is a scatterplot where the x- and y-axis are the first two principal components. Initially, the samples aggregate in batches. After batch correction, the samples should no longer aggregate in batches, but instead, the batches should lie on top of each other. Suppose it was possible to distinguish between samples based on some other variable within each batch before the batch effect was removed. In that case, it must still be possible to differentiate the samples after the batch correction.

The boxplots show the distribution of gene expression values for all samples. After the batch correction, the distribution should be identical across all batches.

Value

A `data.frame` with the same dimensions as the `Data` argument.

References

Leek JT, Johnson WE, Parker HS, Fertig EJ, Jaffe AE, Zhang Y, Storey JD, Torres LC (2021). *sva: Surrogate Variable Analysis*. R package available at https://bioconductor.org/packages/release/bioc/html/sva.html, version 3.40.0.

See Also

ReadGE GEDI VerifyGEDI

Examples

```
## Not run:
### From example in README.md
## Continuing example from GEDI()'s example, see help("GEDI")
# Each dataset is a batch
# Each sample has a status of 'positive' or 'negative'
# Batch and Status variables is seen below:
summary(as.factor(batch))
#> B1 B2 B3
#>  8  6  6
summary(as.factor(status))
#> control treated
#>      10      10

# The batch effect is removed
cData <- BatchCorrection(dat, batch, status = status)
#> Found3batches
#> Adjusting for0covariate(s) or covariate level(s)
#> Standardizing Data across genes
#> Fitting L/S model and finding priors
#> Finding parametric adjustments
#> Adjusting the Data

# To see the resulting figure, check the README.md file in the
# package GitHub repository: github.com/s184257/GEDI

## End(Not run)
```

BM_attributes *List valid BioMart attributes*

Description

BM_attributes allows users to overview all valid BioMart attributes for a given species using the biomaRt::listAttributes function from the biomaRt package (Durinck et al. 2005; Durinck et al. 2009). By searching through the available BioMart attributes, the user can find the BioMart attribute corresponding to the type of reporter IDs used in a dataset.

Usage

```
BM_attributes(species)
```

Arguments

species	A species covered by BioMart Ensembl. The species must be written with the first letter of the genus name followed by the specific name, e.g. *Homo sapiens* is "hsapiens", and *Mus musculus* is "mmusculus". Use listSpecies to find valid species inputs.

Value

A table with names and descriptions for all BioMart attributes for a given species

References

Durinck S, Moreau Y, Kasprzyk A, Davis S, De Moor B, Brazma A, Huber W (2005). "BioMart and Bioconductor: A powerful link between biological databases and microarray data analysis." *Bioinformatics*, **21**(16), 3439–3440. ISSN 13674803, doi: 10.1093/bioinformatics/bti525, https://pubmed.ncbi.nlm.nih.gov/16082012/.

Durinck S, Spellman PT, Birney E, Huber W (2009). "Mapping identifiers for the integration of genomic datasets with the R/ Bioconductor package biomaRt." *Nature Protocols*, **4**(8), 1184–1191. ISSN 17542189, doi: 10.1038/nprot.2009.97, https://pubmed.ncbi.nlm.nih.gov/19617889/.

See Also

GEDI listSpecies biomaRt::listDatasets listAttributes

Examples

```
## For species: Bos taurus (cow)
# Find all BioMart Attributes for Bos taurus
# if Bos taurus is a valid species in BioMart
species <- "btaurus"
if (species %in% listSpecies()){
    BM_attributes(species = species)
}
```

GEDI *Re-annotate and integrate gene expression datasets*

Description

GEDI allows users to integrate transcriptomic datasets into one data table containing samples from all the datasets. The first step is to map the reporter IDs to Ensembl gene IDs. The second step is to collapse rows mapped to the same Ensembl gene ID. The third step is to merge the datasets. Optionally, GEDI can fill in missing data using a linear regression model.

Usage

```
GEDI(datasets, attributes, species = "hsapiens", BioMart = TRUE,
     collapseMethod = "MaxMean", predict.missing = FALSE,
     predict.threshold = 0.8, path=NULL)
```

Arguments

datasets	A list of transcriptomic datasets. The ReadGE function can create this list. The GEDI function integrates all datasets in the list.
attributes	Character vector of BioMart attributes that correspond to each datasets reporter IDs. The BM_attributes function can find all available BioMart attributes for a given species.
species	(Optional) Character that refers to a species covered by BioMart Ensembl. The species must be written with the first letter of the genus name followed by the specific name, e.g. *Homo sapiens* is "hsapiens", and *Mus musculus* is "mmusculus". The listSpecies function can find all valid species inputs.
BioMart	(Optional) Logical value. When TRUE (default), the reporter IDs are mapped to Ensembl gene IDs using the biomaRt package. If FALSE, the GEDI function maps reporter IDs to Ensembl IDs using a tab-separated annotation table named "annot.txt" for each dataset. The reporter IDs must be in the first column of the table. The second column holds Ensembl gene IDs. The second columns name must correspond to the BioMart attribute, e.g. "ensembl_gene_id".
collapseMethod	(Optional) Character that determines which method the WGCNA::collapseRows function uses to collapse two rows mapped to the same Ensembl gene ID. Valid options are: MaxMean (default), MinMean, absMaxMean, absMinMean, Average, maxRowVariance and ME. See WGCNA::collapseRows documentation for more info.
predict.missing	(Optional) Logical value. When TRUE, GEDI predicts missing expression values for a gene if the percentage of known values is higher than the predict.threshold. This does not currently work for count data.
predict.threshold	(Optional) Numeric argument between 0 and 1. This argument is ignored if predict.missing = FALSE.
path	(Optional) (Optional) Path to the directory where all data folders are located. The data folders store annotation files if there are any. The current working directory is the default. The names of the data folders are names(datasets).

Details

The GEDI function maps reporter IDs of the datasets to Ensembl gene IDs using the biomaRt package (Durinck et al. 2005; Durinck et al. 2009). However, if the BioMart attribute for a dataset is invalid or missing, an annotation table called "annot.txt" is needed; see the BioMart argument. If the second column does not contain Ensembl gene IDs, then it must contain another type of reporter ID with a known BioMart attribute. The annotation file must be in the same subdirectory as the corresponding dataset. The subdirectory's name must correspond to the dataset's name; see names(datasets). The annotation file is also used if the user chooses not to use BioMart.

Multiple reporter IDs can map to the same Ensembl gene ID. Therefore, these reporter IDs' expression values must be collapsed using the WGCNA::collapseRows function. The method used to collapse the rows can be specified with the collapseMethod argument. After this, the datasets are merged.

If predict.missing = TRUE, then the missing expression values are predicted for all Ensembl gene IDs where the percentage of known expression values exceeds the predict.threshold argument. GEDI uses a linear regression model to predict the missing expression values. The linear regression model uses only one feature; the gene most correlated with the gene with missing values. Only one feature is chosen to get a quick estimate. However, it can still be time-consuming if many rows have missing values.

Value

A Data.frame containing columns from all datasets. All rows are mapped to Ensembl gene IDs. Rows with missing data have been removed from the output.

References

Durinck S, Moreau Y, Kasprzyk A, Davis S, De Moor B, Brazma A, Huber W (2005). "BioMart and Bioconductor: A powerful link between biological databases and microarray data analysis." *Bioinformatics*, **21**(16), 3439–3440. ISSN 13674803, doi: 10.1093/bioinformatics/bti525, https://pubmed.ncbi.nlm.nih.gov/16082012/.

Durinck S, Spellman PT, Birney E, Huber W (2009). "Mapping identifiers for the integration of genomic datasets with the R/ Bioconductor package biomaRt." *Nature Protocols*, **4**(8), 1184–1191. ISSN 17542189, doi: 10.1038/nprot.2009.97, https://pubmed.ncbi.nlm.nih.gov/19617889/.

See Also

ReadGE BatchCorrection VerifyGEDI

Examples

```
## Not run:
### From example in README.md
## Continuing example from ReadGE()'s example, see help("ReadGE")
# Biomart attributes defined for the three datasets
# The species is Bos taurus
attr <- c("ensembl_gene_id", affy_bovine, NA)

# There are no attribute for the third dataset
# An annotation file (annot.txt) are used
# It is located in the datafolder for the dataset

dat <- GEDI(datasets, attributes = attr, BioMart = TRUE,
           species = "btaurus", path = PATH_TO_DATA_FOLDERS)
           #> Connecting to BioMart...
           #> Invalid BioMart attribute, NA in agilent_data
           #> Reading annotation file ('annot.txt') for agilent_data
dim(dat)
#> [1] 8328   20

# The integrated dataset has 20 samples with expression values
# for 8328 genes

## End(Not run)
```

---

listSpecies *List valid species for the biomaRt package*

Description

listSpecies allows the user to overview all available species for BioMart Ensembl using the biomaRt::listDatasets function from the biomaRt package (Durinck et al. 2005; Durinck et al. 2009). listSpecies makes it easy to find the correct species input for the GEDI function.

Usage

```
listSpecies()
```

Value

A table with names and descriptions of all the valid species

References

Durinck S, Moreau Y, Kasprzyk A, Davis S, De Moor B, Brazma A, Huber W (2005). "BioMart and Bioconductor: A powerful link between biological databases and microarray data analysis." *Bioinformatics*, **21**(16), 3439–3440. ISSN 13674803, doi: 10.1093/bioinformatics/bti525, https://pubmed.ncbi.nlm.nih.gov/16082012/.

Durinck S, Spellman PT, Birney E, Huber W (2009). "Mapping identifiers for the integration of genomic datasets with the R/ Bioconductor package biomaRt." *Nature Protocols*, **4**(8), 1184–1191. ISSN 17542189, doi: 10.1038/nprot.2009.97, https://pubmed.ncbi.nlm.nih.gov/19617889/.

See Also

GEDI BM_attributes biomaRt::listDatasets biomaRt::listAttributes

Examples

```
## Get an overview of all valid species in BioMart
tab <- listSpecies()

## Check if Mus musculus is coverd by BioMart
"mmusculus" %in% listSpecies()$species
```

ReadGE *Read multiple gene expression datasets*

Description

ReadGE allows users to read multiple transcriptomic datasets and store them in a list of data.frames. A subdirectory for each dataset contains the data files. The function can read Affymetrix, Agilent and RNA sequencing data.

Usage

```
ReadGE(dataFolders, sources, path = NULL, verbose = TRUE)
```

Arguments

dataFolders	Character vector with the names of the subdirectories containing gene expression datasets.
sources	Character vector with the data source for each dataset. Valid sources are "affy" or "affymetrix", "agilent" and "RNAseq".
path	(Optional) Logical value. If TRUE, the function prints descriptive messages about the progress.

verbose (Optional) Logical value. If TRUE, the function prints descriptive messages about the progress.

Details

ReadGE reads datasets with different methods depending on the source. Sample files for Agilent and Affymetrix data are read in alphabetical order. Samples are presented in the same order on Gene Expression Omnibus (GEO); https://www.ncbi.nlm.nih.gov/geo/.

"affy"/"affymetrix" corresponds to Affymetrix data. ReadGE reads Affymetrix data using the affy::ReadAffy function and converts it to an expression data table using the gcrma::gcrma function. Affymetrix data files have the .CEL or .CEL.gz file extension.

"agilent" corresponds to Agilent data. ReadGE reads Agilent data using the limma::read.maimages function. Agilent sample files have the file extension .txt or .txt.gz. The filenames "targets.txt" and "annot.txt" are ignored.

"RNAseq" corresponds to RNA sequencing data. ReadGE reads RNA-seq data using the DESeq2 package. "RawCounts.txt" is a tab-separated table containing RNA-seq count data. This file must contain raw counts and not transformed or normalised data. ReadGE transforms the raw count data using the DESeq2::varianceStabilizingTransformation function so the RNA sequencing data can be integrated with microarray data. ReadGE does not transform the counts if all datasets are "RNAseq".

Value

A list of all transcriptomic datasets that were successfully read where names(datasets) = dataFolders.

See Also

GEDI BatchCorrection VerifyGEDI

Examples

```
## Not run:
### From example in README.md
## Three transcriptomic datasets are read
library(GEDI)
# Names of folders with data
dataFolders <- c("RNAseq_data", "affy_data", "agilent_data")
# Which type of data is each dataset
sources <- c("RNAseq", "affy", "agilent")

# Read the data
datasets <- ReadGE(dataFolders, sources,
                   path = "PATH_TO_DATA_FOLDERS")
#> Reading RNAseq_data...
#> Reading affy_data...
#> Reading agilent_data...

## End(Not run)
```

VerifyGEDI *Verify transcriptomic data integration*

Description

VerifyGEDI allows users to verify if the GEDI function successfully integrated the transcriptomic datasets. Before using this function, the BatchCorrection function must remove the integrated dataset's batch effect. The transcriptomic data integration is verified using a supervised machine learning model to predict the samples' status in one batch based on the remaining batches' samples. The forward stepwise selection (FSS) algorithm picks which genes to use as features in the chosen model.

Usage

```
VerifyGEDI(X, y, batch, model = "logistic", stop.at.score = 1,
           verbose = TRUE)
```

Arguments

X	A data.frame or matrix of integrated gene expression data with batch effect removed. Columns contain samples, and rows are genes.
y	A character vector containing a status variable for the samples. For example, healthy/sick or treated/control. y does not have to be binary. y must satisfy: length(y) = ncol(X).
batch	Character vector that describes which sample is in what batch. batch must satisfy: length(batch)=ncol(Data).
model	(Optional) A character describing which model VerifyGEDI uses to predict the status y in each sample in every batch. This function currently supports two models; a logistic regression model ("logistic") and a decision tree classifier ("tree"). "logistic" only works if Y contains two classes.
stop.at.score	(Optional) The threshold is a numeric value between 0 and 1. This value determines at what score to force the FSS algorithm to terminate. The score is how many per cent of the samples are correctly classified. The default value is 1, meaning that the FSS algorithm will end when the model makes 0 errors.
verbose	(Optional) Logical value. If TRUE, the function prints descriptive messages about the progress.

Details

There are more genes than samples in gene expression datasets in almost all cases. Therefore, the VerifyGEDI function must select some of the genes as features for the classifier. VerifyGEDI uses the Forward stepwise selection (FSS) algorithm to do this.

In each iteration of the FSS algorithm, the chosen model is cross-validated across all the batches. The model uses one batch as test data and the remaining batches as training data. This is done for all the batches. The algorithm uses the lowest test score across all the batches to evaluate which gene is the best new feature for the model. The FSS algorithm will continue to do this and add features to the model until the model no longer improves or that the test score exceeds the threshold determined by the stop.at.score argument.

If the FSS algorithm selects two or more features, VerifyGEDI constructs a scatterplot using the first two features as axes. The scatterplot visualises how well the model can separate the samples based on the y variable.

Value

A list with the following elements:

$Feature.idx Indices of rows in X used as features in the model.

$Feature.names Names of rows in X used as features in the model. These would be Ensembl gene IDs if X was created using the GEDI function.

$TrainScores Numeric vector. How the model scores on the training data for each batch.

$TestScores Numeric vector. Describes how the model scores on each batch when training on the remaining batches.

See Also

ReadGE GEDI BatchCorrection

Examples

```
## Not run:
### From README.md example
## Continuing example from BatchCorrection()'s example,
## see help("BatchCorrection")

# The gene expression data integration is verified:
res <- VerifyGEDI(X = cData, y = status, batch = batch, model = "logistic")
#> Number of features:  2
#> ENSBTAG00000003530, ENSBTAG00000000476
#> Worst score across batches:  1

# With an accuracy of 100%, it can be concluded
# that the data integration was successful

## End(Not run)
```

Index