# Package ‘RNAseqCovarImpute’

**March 28, 2024**

**Title**  Impute Covariate Data in RNA Sequencing Studies  
**Version**  1.0.2  
**URL**  https://github.com/brennanhilton/RNAseqCovarImpute  
**BugReports**  https://github.com/brennanhilton/RNAseqCovarImpute/issues

**Description**  The RNAseqCovarImpute package implements multiple imputation of missing covariates and differential gene expression analysis by: 1) Randomly binning genes into smaller groups, 2) Creating M imputed datasets separately within each bin, where the imputation predictor matrix includes all covariates and the log counts per million (CPM) for the genes within each bin, 3) Estimating gene expression changes using voom followed by lmFit functions, separately on each M imputed dataset within each gene bin, 4) Unbinning the gene sets and stacking the M sets of model results before applying the squeezeVar function to apply a variance shrinking Bayesian procedure to each M set of model results, 5) Pooling the results with Rubins’ rules to produce combined coefficients, standard errors, and P-values, and 6) Adjusting P-values for multiplicity to account for false discovery rate (FDR).

**License**  GPL-3  
**Encoding**  UTF-8  
**Roxygen**  list(markdown = TRUE)  
**RoxygenNote**  7.2.3  
**VignetteBuilder**  knitr  
**biocViews**  RNASeq, GeneExpression, DifferentialExpression, Sequencing  
**Suggests**  BiocStyle, knitr, rmarkdown, tidyr, stringr, testthat (>= 3.0.0)  
**Depends**  R (>= 4.3.0)  
**Imports**  Biobase, BiocParallel, stats, limma, dplyr, magrittr, rlang, edgeR, foreach, mice  
**Config/testthat/edition**  3
RNAseqCovarImpute-package

Collate 'RNAseqCovarImpute-package.R' 'combine_rubins.R'
   'example_DGE.R' 'example_data.R' 'get_gene_bin_intervals.R'
   'impute_by_gene_bin_helper.R' 'impute_by_gene_bin.R'
   'voom_sx_sy.R' 'lowess_all_gene_bins.R' 'voom_master_lowess.R'
   'limmavoom_imputed_data_list_helper.R'
   'limmavoom_imputed_data_list.R'

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

git_branch RELEASE_3_18

git_last_commit d82f00b

git_last_commit_date 2023-10-26

Repository Bioconductor 3.18

Date/Publication 2024-03-27

Author Brennan Baker [aut, cre] (<https://orcid.org/0000-0001-5459-9141>),
   Sheela Sathyanarayana [aut],
   Adam Szpiro [aut],
   James MacDonald [aut],
   Alison Paquette [aut]

Maintainer Brennan Baker <brennanhilton@gmail.com>

R topics documented:

RNAseqCovarImpute-package .................................................. 2
combine_rubins .............................................................. 3
example_data ................................................................. 5
example_DGE ................................................................. 6
get_gene_bin_intervals ....................................................... 6
impute_by_gene_bin .......................................................... 7
impute_gene_bin_helper ..................................................... 8
limmavoom_imputed_data_list .............................................. 9
limmavoom_imputed_data_list_helper .................................... 10
lowess_all_gene_bins ....................................................... 11
voom_master_lowess ....................................................... 12
voom_sx_sy ................................................................. 12

Index 14
**Description**

The RNAseqCovarImpute package implements multiple imputation of missing covariates and differential gene expression analysis by: 1) Randomly binning genes into smaller groups, 2) Creating M imputed datasets separately within each bin, where the imputation predictor matrix includes all covariates and the log counts per million (CPM) for the genes within each bin, 3) Estimating gene expression changes using voom followed by lmFit functions, separately on each M imputed dataset within each gene bin, 4) Un-binning the gene sets and stacking the M sets of model results before applying the squeezeVar function to apply a variance shrinking Bayesian procedure to each M set of model results, 5) Pooling the results with Rubins’ rules to produce combined coefficients, standard errors, and P-values, and 6) Adjusting P-values for multiplicity to account for false discovery rate (FDR).

**Author(s)**

**Maintainer:** Brennan Baker <brennanhilton@gmail.com> (ORCID)

Authors:

- Sheela Sathyanarayana
- Adam Szpiro
- James MacDonald
- Alison Paquette

**See Also**

Useful links:

- [https://github.com/brennanhilton/RNAseqCovarImpute](https://github.com/brennanhilton/RNAseqCovarImpute)
- Report bugs at [https://github.com/brennanhilton/RNAseqCovarImpute/issues](https://github.com/brennanhilton/RNAseqCovarImpute/issues)

---

**Description**

Combines results from each imputed dataset using Rubin's rules.

**Usage**

```r
combine_rubins(
    DGE,
    model_results,
    predictor, 
    covariate = NULL,
    robust = FALSE,
    winsor.tail.p = c(0.05, 0.1)
)
```
4  

**combine_rubins**

**Arguments**

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DGE</td>
<td>A DGEList object.</td>
</tr>
<tr>
<td>model_results</td>
<td>Output from limma::voom_imputed_data.</td>
</tr>
<tr>
<td>predictor</td>
<td>Independent variable of interest, in the form of a linear model contrast. Must be a variable in voom_formula.</td>
</tr>
<tr>
<td>covariate</td>
<td>Arguments passed to limma::squeezeVar. If non-NULL, var.prior will depend on this numeric covariate. Otherwise, var.prior is constant.</td>
</tr>
<tr>
<td>robust</td>
<td>Arguments passed to limma::squeezeVar. Logical, should the estimation of df.prior and var.prior be robustified against outlier sample variances?</td>
</tr>
<tr>
<td>winsor.tail.p</td>
<td>Arguments passed to limma::squeezeVar. Numeric vector of length 1 or 2, giving left and right tail proportions of x to Winsorize. Used only when robust=TRUE.</td>
</tr>
</tbody>
</table>

**Value**

Dataframe with one row per gene containing coefficients standard errors, degrees of freedom, t-statistics, P-Values, and adjusted P-values from the limma-voom pipeline.

| coef_combined | combined logFCs across the multiple imputed datasets using Rubin’s rules |
| SE_P          | pooled standard error across the multiple imputed datasets using Rubin’s rules |
| SE_P_bayes    | pooled standard error across the multiple imputed datasets using Rubin’s rules squeezed to global mean variance trend curve with limma-voom Bayesian procedure |
| df            | limma-voom residual degrees of freedom adjusted for Rubin’s rules |
| df_bayes      | limma-voom residual degrees of freedom adjusted for Rubin’s rules and Bayesian procedure |
| rubins_t      | t-statistic = coef_combined divided by SE_P |
| rubins_t_bayes| t-statistic = coef_combined divided by SE_P_bayes |
| combined_p    | p-value from two-sided t-distribution alpha = 0.05 using rubins_t |
| combined_p_bayes | p-value from two-sided t-distribution alpha = 0.05 using rubins_t_bayes |
| combined_p_adj | false discovery rate (FDR) adjusted combined_p |
| combined_p_adj_bayes | false discovery rate (FDR) adjusted combined_p_bayes |

**Examples**

```r
data(example_data)
data(example_DGE)
intervals <- get_gene_bin_intervals(example_DGE, example_data, n = 10)
gene_bin_impute <- impute_by_gene_bin(example_data, intervals, example_DGE, m = 2)
```
coef_se <- limmavoom_imputed_data_list(
  gene_intervals = intervals,
  DGE = example_DGE,
  imputed_data_list = gene_bin_impute,
  m = 2,
  voom_formula = "~x + y + z + a + b"
)

final_res <- combine_rubins(
  DGE = example_DGE,
  model_results = coef_se,
  predictor = "x"
)

---

**example_data**

*Simulated dataset*

**Description**

The exact code used to generate these data are found in the Example_Data_for_RNAseqCovarImpute vignette. In short, example_data contains 500 rows with data for variables x, y, and z, which are continuous normally distributed, and a and b, which are binary variables. Missigness was simulated for all variables other than x such that a complete case analysis would drop 24.2% of participants.

eventually contains random count data from the Poisson distribution for 500 made up genes, ENS1-ENS500

**Usage**

data(example_data)

**Format**

eventually_data:
  data frame with 500 rows and 5 variables
  x continuous normally distributed
  y continuous normally distributed
  z continuous normally distributed
  a binary
  b binary ...

**Value**

Tibble with 500 rows of data for variables x, y, and z

**Examples**

data(example_data)
**example_DGE**  
*Simulated counts in DGE list*

**Description**

The exact code used to generate these data are found in the Example_Data_for_RNAseqCovarImpute vignette. In short, example_data contains 500 rows with data for variables x, y, and z, which are continuous normally distributed, and a and b, which are binary variables. Missigness was simulated for all variables other than x such that a complete case analysis would drop 24.2% of participants. example_DGE contains random count data from the Poisson distribution for 500 made up genes, ENS1-ENS500

**Usage**

```
data(example_DGE)
```

**Format**

```
example_DGE:
A DGElist with 500 genes and 500 samples
```

**Value**

DGElist for 500 made up genes, ENS1-ENS500

**Examples**

```
data(example_DGE)
```

---

**get_gene_bin_intervals**

**Description**

Creates gene bins. Input DGE list, sample data, and 'n' number of individuals per genes. By default, number of bins and genes per bin are set so that each bin has approximately 1 gene per 10 individuals in the data.

**Usage**

```
get_gene_bin_intervals(DGE, data, n = 10)
```
Arguments

- **DGE**: A DGEList object.
- **data**: Sample data with one row per sample. Sample row order should match the column order in the DGEList.
- **n**: Genes per bin are set so that each bin has approximately 1 gene per n individuals in the data.

Value

Data frame with one row per gene bin. Columns indicate the start and end positions and the number of genes of each bin.

Examples

```r
data(example_data)
data(example_DGE)
intervals <- get_gene_bin_intervals(example_DGE, example_data, n = 10)
gene_bin_impute <- impute_by_gene_bin(example_data, intervals, example_DGE, m = 2)

coef_se <- limmavoom_imputed_data_list(
gene_intervals = intervals,
DGE = example_DGE,
imputed_data_list = gene_bin_impute,
m = 2,
voom_formula = "-x + y + z + a + b"
)

final_res <- combine_rubins(
DGE = example_DGE,
model_results = coef_se,
predictor = "x"
)
```

Description

Loops through DGE list using the gene bin intervals from the "get_gene_bin_intervals" function and makes imputed datasets. For instance, if n = 100 and intervals contains 200 gene bin intervals, output will be a list of 200 sets of 100 imputed datasets. Each of the 200 sets are imputed using only the genes in one gene bin.

Usage

```
impute_by_gene_bin(data, intervals, DGE, m, maxit = 10, BPPARAM = bpparam())
```
Arguments

- **data**: Sample data with one row per sample. Sample row order should match the column order in the DGEList.
- **intervals**: Output from `get_gene_bin_intervals` function. A dataframe where each row contains the start (first column) and end (second column) values for each gene bin interval.
- **DGE**: A DGEList object.
- **m**: Number of imputed data sets.
- **maxit**: Used by `mice` function.
- **BPPARAM**: A BiocParallelParam object

Value

A list of sets of `n` imputed datasets, one per gene bin.

Examples

```r
data(example_data)
data(example_DGE)
intervals <- get_gene_bin_intervals(example_DGE, example_data, n = 10)
gene_bin_impute <- impute_by_gene_bin(example_data, intervals, example_DGE, m = 2)

coef_se <- limmavoom_imputed_data_list(gene_intervals = intervals, DGE = example_DGE, imputed_data_list = gene_bin_impute, m = 2, voom_formula = "-x + y + z + a + b")

final_res <- combine_rubins(DGE = example_DGE, model_results = coef_se, predictor = "x")
```

**Description**

Loops through DGE list using the gene bin intervals from the "get_gene_bin_intervals" function and makes imputed datasets. For instance, if `n = 100` and intervals contains 200 gene bin intervals, output will be a list of 200 sets of 100 imputed datasets. Each of the 200 sets are imputed using only the genes in one gene bin.
Usage

impute_gene_bin_helper(i, intervals, cpm_all, data, m, maxit)

Arguments

intervals  Output from get_gene_bin_intervals function. A dataframe where each row contains the start (first col) and end (second col) values for each gene bin interval.
data  Sample data with one row per sample. Sample row order should match the col order in the DGEList.
m  Number of imputed data sets.
maxit  Used by mice function.
DGE  A DGEList object.
param  Arguments passed to BiocParallel::bpparam()

Value

A list of sets of n imputed datasets, one per gene bin.

Description

Loops through the imputed data list (output from "impute_by_gene_bin" function) and runs limmavoom RNA seq analysis.

Usage

limmavoom_imputed_data_list(
  gene_intervals,
  DGE,
  imputed_data_list,
  m,
  voom_formula,
  BPPARAM = bpparam()
)

Arguments

gene_intervals  Output from get_gene_bin_intervals function. A dataframe where each row contains the start (first col) and end (second col) values for each gene bin interval.
DGE  A DGEList object.
imputed_data_list  Output from impute_by_gene_bin.
limmavoom_imputed_data_list_helper

m Number of imputed data sets.
voom_formula Formula for design matrix.
BPPARAM A BiocParallelParam object

Value

A dataframe with coefficient, standard error, sigma, and residual degrees of freedom values from limma-voom gene expression analysis. One row per gene and one set of values per imputed dataset.

Examples

```r
data(example_data)
data(example_DGE)
intervals <- get_gene_bin_intervals(example_DGE, example_data, n = 10)
gene_bin_impute <- impute_by_gene_bin(example_data, intervals, example_DGE, m = 2)

coef_se <- limmavoom_imputed_data_list(
gene_intervals = intervals, DGE = example_DGE, imputed_data_list = gene_bin_impute, m = 2, voom_formula = "~x + y + z + a + b"
)

final_res <- combine_rubins(
    DGE = example_DGE, model_results = coef_se, predictor = "x"
)
```

Description

Loops through the imputed data list (output from “impute_by_gene_bin” function) and runs limma-voom RNA seq analysis.

Usage

```r
limmavoom_imputed_data_list_helper(
gene_bin, gene_intervals, DGE,
```
Arguments

gene_intervals  Output from get_gene_bin_intervals function. A dataframe where each row contains the start (first col) and end (second col) values for each gene bin interval.

DGE  A DGEList object.

imputed_data_list  Output from impute_by_gene_bin.

m  Number of imputed data sets.

voom_formula  Formula for design matrix.

Value

A dataframe with coefficient, standard error, sigma, and residual degrees of freedom values from limma-voom gene expression analysis. One row per gene and one set of values per imputed dataset.

Description

Loops through all bins and all M imputations, prepares DGE and design to run voom_sx_sy, which fits gene-wise linear models and extracts log count size (sx) and sqrt residual standard deviations (sy) to make the lowess curve.

Usage

lowess_all_gene_bins(gene_intervals, DGE, imputed_data_list, m, voom_formula)

Value

All sx and sy values for lowess function across all M imputations.
Description

Modified voom function used by limma_voom-imputed_data_list function. Allows input of bins of outcome genes while still accounting for the total library size of all outcome genes, as the total library size is needed to calculate log-cpm values. Also allows use of external sx and sy to create lowess curve. Here, sx and sy should come from all gene bins across all M imputations. Adapted from limma::voom. Code from limma covered by License: GPL (>=2)

Usage

voom_master_lowess(
  counts, 
  design = NULL, 
  lib.size = NULL, 
  normalize.method = "none", 
  block = NULL, 
  correlation = NULL, 
  weights = NULL, 
  span = 0.5, 
  plot = FALSE, 
  save.plot = FALSE, 
  lib.size.all, 
  sx, 
  sy 
)

Value

Same as limma::voom.

Description

Modified voom function used by limma_voom-imputed_data_list function. Allows input of bins of outcome genes while still accounting for the total library size of all outcome genes, as the total library size is needed to calculate log-cpm values. Returns just the sx and sy values needed for lowess curve. Adapted from limma::voom. Code from limma covered by License: GPL (>=2)
voom_sx_sy

Usage

  voom_sx_sy(
    counts,
    design = NULL,
    lib.size = NULL,
    normalize.method = "none",
    block = NULL,
    correlation = NULL,
    weights = NULL,
    span = 0.5,
    plot = FALSE,
    save.plot = FALSE,
    lib.size.all
  )

Value

  Tibble with one col for sx and one for sy for lowess function.
Index

* datasets
  example_data, 5
  example_DGE, 6

* internal
  impute_gene_bin_helper, 8
  limmavoom_imputed_data_list_helper, 10
  lowess_all_gene_bins, 11
  RNAseqCovarImpute-package, 2
  voom_master_lowess, 12
  voom_sx_sy, 12

combine_rubins, 3

element_data, 5
example_DGE, 6
get_gene_bin_intervals, 6
impute_by_gene_bin, 7
impute_gene_bin_helper, 8

limmavoom_imputed_data_list, 9
limmavoom_imputed_data_list_helper, 10
lowess_all_gene_bins, 11

RNAseqCovarImpute
  (RNAseqCovarImpute-package), 2
RNaseqCovarImpute-package, 2

voom_master_lowess, 12
voom_sx_sy, 12