Package 'IsoBayes'

November 12, 2026

Type Package

Title IsoBayes: Single Isoform protein inference Method via Bayesian Analyses

Version 1.10.1

Description IsoBayes is a Bayesian method to perform inference on single protein isoforms.
Our approach infers the presence/absence of protein isoforms, and also estimates their abundance;
additionally, it provides a measure of the uncertainty of these estimates, via:
i) the posterior probability that a protein isoform is present in the sample;
ii) a posterior credible interval of its abundance.
IsoBayes inputs liquid cromatography mass spectrometry (MS) data,
and can work with both PSM counts, and intensities.
When available, trascript isoform abundances (i.e., TPMs) are also incorporated:
TPMs are used to formulate an informative prior for the respective protein isoform relative abundance.
We further identify isoforms where the relative abundance of proteins and transcripts significantly differ.
We use a two-
layer latent variable approach to model two sources of uncertainty typical of MS data:
i) peptides may be erroneously detected (even when absent);
ii) many peptides are compatible with multiple protein isoforms.
In the first layer, we sample the presence/absence of each peptide based on its estimated probability
of being mistakenly detected, also known as PEP (i.e., posterior error probability).
In the second layer, for peptides that were estimated as being present,
we allocate their abundance across the protein isoforms they map to.
These two steps allow us to recover the presence and abundance of each protein isoform.

biocViews StatisticalMethod, Bayesian, Proteomics, MassSpectrometry, AlternativeSplicing, Sequencing, RNASeq, GeneExpression, Genetics, Visualization, Software

License GPL-3

Depends R (>= 4.3.0)

Imports methods, Rcpp, data.table, glue, stats, doParallel, parallel, doRNG, foreach, iterators, ggplot2, HDInterval, SummarizedExperiment, S4Vectors

LinkingTo Rcpp, RcppArmadillo

Suggests knitr, rmarkdown, testthat, BiocStyle

SystemRequirements C++17

VignetteBuilder knitr

ByteCompile true

URL https://github.com/SimoneTiberi/IsoBayes

BugReports https://github.com/SimoneTiberi/IsoBayes/issues

Config/roxygen2/version 8.0.0

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

git_branch RELEASE_3_23

git_last_commit db83dbe

git_last_commit_date 2026-07-21

Repository Bioconductor 3.23

Date/Publication 2026-11-11

Author Jordy Bollon [aut],
Simone Tiberi [aut, cre] (ORCID:
<https://orcid.org/0000-0002-3054-9964>)

Maintainer Simone Tiberi <simone.tiberi@unibo.it>

Contents

IsoBayes-package 2
generate_SE 3
inference 5
input_data 7
plot_relative_abundances 8
plot_traceplot 9

Index **11**

`IsoBayes-package` *IsoBayes: Single Isoform protein inference Method via Bayesian Analyses*

Description

IsoBayes is a Bayesian method to perform inference on single protein isoforms. Our approach infers the presence/absence of protein isoforms, and also estimates their abundance; additionally, it provides a measure of the uncertainty of these estimates, via: i) the posterior probability that a protein isoform is present in the sample; ii) a posterior credible interval of its abundance. IsoBayes inputs liquid cromatography mass spectrometry (MS) data, and can work with both PSM counts, and intensities. When available, trascript isoform abundances (i.e., TPMs) are also incorporated: TPMs are used to formulate an informative prior for the respective protein isoform relative abundance. We further identify isoforms where the relative abundance of proteins and transcripts significantly differ. We use a two-layer latent variable approach to model two sources of uncertainty typical of MS data: i) peptides may be erroneously detected (even when absent); ii) many peptides are compatible with multiple protein isoforms. In the first layer, we sample the presence/absence of each peptide based on its estimated probability of being mistakenly detected, also known as PEP (i.e., posterior error probability). In the second layer, for peptides that were estimated as being present, we allocate their abundance across the protein isoforms they map to. These two steps allow us to recover the presence and abundance of each protein isoform.

Details

The DESCRIPTION file: This package was not yet installed at build time.
Questions relative to IsoBayes should be reported as a new issue at https://github.com/SimoneTiberi/IsoBayes/issues.

To access the vignettes, type: `browseVignettes("IsoBayes")`.

Index: This package was not yet installed at build time.

Author(s)

Jordy Bollon <jordy.bollon@iit.it>, Simone Tiberi <simone.tiberi@unibo.it>

---

generate_SE *Generate SummarizedExperiment object*

Description

`generate_SE` converts the input files, required to run IsoBayes, into a `SummarizedExperiment` object. This object should then be passed to `input_data` function.

Usage

```
generate_SE(
  path_to_peptides_psm = NULL,
  path_to_peptides_intensities = NULL,
  input_type = NULL,
  abundance_type = NULL,
  PEP = TRUE,
  FDR_thd = 0.01
)
```

Arguments

`path_to_peptides_psm`
: a character string indicating the path to one of the following files:
i) the psmtsv file from *MetaMorpheus* tool with PSM counts,
ii) the idXML file from *OpenMS* toolkit, or
iii) a `data.frame` or a path to a tsv file, formatted as explained in the "Input user-provided data" Section of the vignettes (only when input_type = "other").

`path_to_peptides_intensities`
: (optional) a character string indicating the path to the psmtsv file from *MetaMorpheus* with intensity values. Required if 'abundance_type' equals to "intensities" and input_type equals to "metamorpheus".

`input_type`
: a character string indicating the tool used to obtain the peptides file: "metamorpheus", "openMS" or "other".

`abundance_type`
: a character string indicating the type of input: "psm" or "intensities".

`PEP`
: logical; if TRUE (default), the algorithm will account for the probability that peptides are erroneously detected. If FALSE, PEP is ignored. We suggest using PEP with a weak FDR threshold of 0.1 (default parameters options). This is because peptides with FDR > 0.1 are usually unreliable, and associated to high error probabilities (e.g., PEP > 0.9).

`FDR_thd`
: a numeric value indicating the False Discovery Rate threshold to be used to discard unreliable peptides.

Value

A `SummarizedExperiment` object.

Author(s)

Jordy Bollon <jordy.bollon@iit.it> and Simone Tiberi <simone.tiberi@unibo.it>

See Also

input_data

Examples

```
# Load internal data to the package:
data_dir = system.file("extdata", package = "IsoBayes")

# Define the path to the AllPeptides.psmtsv file returned by *MetaMorpheus* tool
path_to_peptides_psm = paste0(data_dir, "/AllPeptides.psmtsv")

# Generate a SummarizedExperiment object
SE = generate_SE(path_to_peptides_psm = path_to_peptides_psm,
                 abundance_type = "psm",
                 input_type = "metamorpheus"
                 )
```

```
# For more examples see the vignettes:
# browseVignettes("IsoBayes")
```

`inference` *Run our two-layer latent variable Bayesian model*

Description

`inference` runs our two-layer latent variable Bayesian model, taking as input the data created by `input_data`.

Usage

```
inference(
  loaded_data,
  map_iso_gene = NULL,
  n_cores = 1,
  K = 2000,
  burn_in = 1000,
  thin = 1,
  traceplot = FALSE
)
```

Arguments

`loaded_data`	`list` of `data.frame` objects, returned by `input_data`.
`map_iso_gene`	(optional) a character string (indicating the path to a csv file with 2 columns), or a data.frame with 2 columns. In both cases, the 1st column must contain the isoform name/id, while the 2nd column has the gene name/id. This argument is required to return protein isoform relative abundances, normalized within each gene (i.e., adding to 1 within a gene), to plot results via `plot_relative_abundances`, and to return protein abundances aggregated by gene with HPD credible interval.
`n_cores`	the number of cores to use during algorithm execution. We suggest increasing the number of threads for large datasets only.
`K`	the number of MCMC iterations. Minimum 2000.
`burn_in`	the number of initial iterations to discard. Minimum 1000.
`thin`	thinning value to apply to the final MCMC chain. Useful for decreasing the memory (RAM) usage.
`traceplot`	a logical value indicating whether to return the posterior chain of the relative abundances of each protein isoform (i.e., "PI"). If TRUE, the posterior chains are stored in 'MCMC' object, and can be plotted via 'plot_traceplot' function.

Value

A `list` of three `data.frame` objects: 'isoform_results', and (only if 'map_iso_gene' is provided) 'normalized_isoform_results' (relative abundances normalized within each gene) and 'gene_abundance'. For more information about the results stored in the three `data.frame` objects, see the vignettes: #browseVignettes("IsoBayes")

Author(s)

Jordy Bollon <jordy.bollon@iit.it> and Simone Tiberi <simone.tiberi@unibo.it>

See Also

`input_data` and `plot_relative_abundances`

Examples

```
# Load internal data to the package:
data_dir = system.file("extdata", package = "IsoBayes")

# Define the path to the AllPeptides.psmtsv file returned by MetaMorpheus tool
path_to_peptides_psm = paste0(data_dir, "/AllPeptides.psmtsv")

# Generate a SummarizedExperiment object
SE = generate_SE(path_to_peptides_psm = path_to_peptides_psm,
                 abundance_type = "psm",
                 input_type = "metamorpheus"
                 )
# Define the path to the jurkat_isoform_kallisto.tsv with mRNA relative abundance
tpm_path = paste0(data_dir, "/jurkat_isoform_kallisto.tsv")

# Load and process SE object
data_loaded = input_data(SE, path_to_tpm = tpm_path)

# Define the path to the map_iso_gene.csv file.
# Alternatively a data.frame can be used (see documentation).
path_to_map_iso_gene = paste0(data_dir, "/map_iso_gene.csv")

# Run the algorithm
set.seed(169612)
results = inference(data_loaded, map_iso_gene = path_to_map_iso_gene, traceplot = TRUE)

# Results is a list of 3 data.frames:
names(results)

# Main results:
head(results$isoform_results)

# Results normalized within genes
# (relative abunances add to 1 within each gene):
# useful to study alternative splicing within genes:
head(results$normalized_isoform_results)
```

```
# Gene abundance
head(results$gene_abundance)

# results normalized within genes (total abundance of each gene),
# useful to study alternative splicing within genes:
head(results$normalized_isoform_results)

# Plotting results, normalizing within genes
# (relative abundances add to 1 within each gene):
plot_relative_abundances(results,
    gene_id = "TUBB",
    normalize_gene = TRUE)

# Plotting results, NOT normalized
# (relative abundances add to 1 across all isoforms in the dataset):
plot_relative_abundances(results,
    gene_id = "TUBB",
    normalize_gene = FALSE)

# Visualize MCMC chain for isoforms "TUBB-205", "TUBB-206", and  "TUBB-208"
# To visualize traceplots, set "traceplot" to TRUE when running "inference" function
plot_traceplot(results, "TUBB-205")
plot_traceplot(results, "TUBB-206")
plot_traceplot(results, "TUBB-208")

# For more examples see the vignettes:
# browseVignettes("IsoBayes")
```

`input_data` *Load and process input data*

Description

`input_data` reads and processes a `SummarizedExperiment` object collecting input data and metadata required to run IsoBayes model.

Usage

```
input_data(SE, path_to_tpm = NULL)
```

Arguments

`SE`	a `SummarizedExperiment` object created by `generate_SE` function. Alternatively, this object can be created by the user, following the structure specified in the "Input user-provided data" Section of the vignettes
`path_to_tpm`	(optional) a `data.frame` object or a character string indicating the path to a tsv file with mRNA isoform TPMs. The tsv file must have 1 row per isoform, and 2 columns: i) 'isoname': a character string indicating the isoform name; ii) 'tpm': a numeric variable indicating the Transcripts Per Million (TPM) count. Column names must be 'isoname' and 'tpm'.

Value

A `list` of `data.frame` objects, with the data needed to run `inference` function.

Author(s)

Jordy Bollon <jordy.bollon@iit.it> and Simone Tiberi <simone.tiberi@unibo.it>

See Also

`generate_SE`, `inference`

Examples

```
# Load internal data to the package:
data_dir = system.file("extdata", package = "IsoBayes")

# Define the path to the AllPeptides.psmtsv file returned by *MetaMorpheus* tool
path_to_peptides_psm = paste0(data_dir, "/AllPeptides.psmtsv")

# Generate a SummarizedExperiment object
SE = generate_SE(path_to_peptides_psm = path_to_peptides_psm,
                 abundance_type = "psm",
                 input_type = "metamorpheus"
                 )
# Load and process SE object
data_loaded = input_data(SE)

# For more examples see the vignettes:
# browseVignettes("IsoBayes")
```

---

plot_relative_abundances

Plot isoform results

Description

`plot_relative_abundances` plots protein isoforms results, obtained by `inference`, for a specific gene, together with transcripts abundances if available.

Usage

```
plot_relative_abundances(
  res_inference,
  gene_id,
  plot_CI = TRUE,
  normalize_gene = TRUE
)
```

Arguments

res_inference	list of two data.frame objects returned by inference.
gene_id	a character string indicating the gene to be plotted.
plot_CI	logical; if TRUE (default), plot 0.95 level Credibility Intervals for each isoform.
normalize_gene	logical; if TRUE (default), plot isoform relative abundances, normalized within the specified gene (they add to 1 within a gene).

Value

A ggplot object, showing isoform relative abundances for a specific gene.

Author(s)

Jordy Bollon <jordy.bollon@iit.it> and Simone Tiberi <simone.tiberi@unibo.it>

See Also

inference

Examples

```
# see the example of inference function:
help(inference)
```

---

plot_traceplot *Traceplot of the (thinned) posterior chain of the relative abundance of each protein isoform (i.e., pi).*

Description

plot_traceplot plots the traceplot of the (thinned) posterior chain of the relative abundance of each protein isoform (i.e., pi). The vertical grey dashed line indicates the burn-in (the iterations on the left side of the burn-in are discarded in posterior analyses).

Usage

```
plot_traceplot(results, protein_id)
```

Arguments

results	a list of data.frame objects, computed via inference.
protein_id	a character, indicating the protein isoform to plot.

Value

A gtable object.

Author(s)

Simone Tiberi <simone.tiberi@unibo.it>

See Also

inference

Examples

```
# see the example of inference function:
help(inference)
```

Index

∗ **package**
IsoBayes-package, 2

data.frame, *9*

generate_SE, 3, *7, 8*

inference, 5, *8–10*
input_data, *3–6*, 7
IsoBayes *(*IsoBayes-package*)*, 2
IsoBayes-package, 2

plot_relative_abundances, *5, 6*, 8
plot_traceplot, 9