Package 'BASiCStan'

November 14, 2024

Title Stan implementation of BASiCS

Version 1.8.0

Date 2024-02-14

Description Provides an interface to infer the parameters of BASiCS using the variational inference (ADVI), Markov chain Monte Carlo (NUTS), and maximum a posteriori (BFGS) inference engines in the Stan programming language. BASiCS is a Bayesian hierarchical model that uses an adaptive Metropolis within Gibbs sampling scheme. Alternative inference methods provided by Stan may be preferable in some situations, for example for particularly large data or posterior distributions with difficult geometries.

License GPL-3

Encoding UTF-8

VignetteBuilder knitr

Roxygen list(markdown = TRUE)

RoxygenNote 7.2.1

Biarch true

Depends R (>= 4.2), BASiCS, rstan (>= 2.18.1)

Imports methods, glmGamPoi, scran, scuttle, stats, utils, SingleCellExperiment, SummarizedExperiment, Rcpp (>= 0.12.0), RcppParallel (>= 5.0.1), rstantools (>= 2.1.1)

LinkingTo BH (>= 1.66.0), Rcpp (>= 0.12.0), RcppEigen (>= 0.3.3.3.0), RcppParallel (>= 5.0.1), rstan (>= 2.18.1), StanHeaders (>= 2.18.0)

SystemRequirements GNU make

biocViews ImmunoOncology, Normalization, Sequencing, RNASeq, Software, GeneExpression, Transcriptomics, SingleCell, DifferentialExpression, Bayesian, CellBiology

URL https://github.com/Alanocallaghan/BASiCStan

BugReports https://github.com/Alanocallaghan/BASiCStan/issues

Suggests testthat (>= 3.0.0), knitr, rmarkdown

Config/testthat/edition 3

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

git_branch RELEASE_3_20

git_last_commit df5d144

git_last_commit_date 2024-10-29

Repository Bioconductor 3.20

Date/Publication 2024-11-14

Author Alan O'Callaghan [aut, cre],
Catalina Vallejos [aut]

Maintainer Alan O'Callaghan <alan.ocallaghan@outlook.com>

Contents

`BASiCStan-package` *The 'BASiCStan' package.*

Description

Provides an interface to infer the parameters of BASiCS using the variational inference (ADVI), Markov chain Monte Carlo (NUTS), and maximum a posteriori (BFGS) inference engines in the Stan programming language. BASiCS is a Bayesian hierarchical model that uses an adaptive Metropolis within Gibbs sampling scheme. Alternative inference methods provided by Stan may be preferable in some situations, for example for particularly large data or posterior distributions with difficult geometries. See also BASiCS_MCMC

References

Stan Development Team (2020). RStan: the R interface to Stan. R package version 2.21.2. https://mc-stan.org

Vallejos, Marioni and Richardson (2015). PLoS Computational Biology. https://doi.org/10.1371/journal.pcbi.1004333.

Vallejos, Richardson and Marioni (2016). Genome Biology. https://doi.org/10.1186/s13059-016-0930-3

Eling et al (2018). Cell Systems. https://doi.org/10.1016/j.cels.2018.06.011

`BASiCStan` *Stan implementation of BASiCS.*

Description

The stan programming language enables the use of MAP, VB, and HMC inference. Only the regression mode featuring a joint prior between mean and overdispersion parameters is implemented

Usage

```
BASiCStan(
  Data,
  Method = c("vb", "sampling", "optimizing"),
  WithSpikes = length(altExpNames(Data)) > 0,
  Regression = TRUE,
  BatchInfo = Data$BatchInfo,
  L = 12,
  PriorMu = c("EmpiricalBayes", "uninformative"),
  NormFactorFun = scran::calculateSumFactors,
  ReturnBASiCS = TRUE,
  Verbose = TRUE,
  ...
)
```

Arguments

`Data`	SingleCellExperiment object
`Method`	Inference method. One of: `"vb"` for Variational Bayes, `"sampling"` for Hamiltonian Monte Carlo, `"optimizing"` for or maximum *a posteriori* estimation.
`WithSpikes`	Do the data contain spike-in genes? See BASiCS for details. When `WithSpikes=FALSE`, the cell-specific scaling normalisation factors are fixed; use `NormFactorFun` to specify how size factors should be generated or extracted.
`Regression`	Use joint prior for mean and overdispersion parameters? Included for compatibility with `BASiCS_MCMC`, but only `TRUE` is supported.
`BatchInfo`	Vector describing which batch each cell is from.
`L`	Number of regression terms (including slope and intercept) to use in joint prior for mu and delta.
`PriorMu`	Type of prior to use for mean expression. Default is "EmpiricalBayes", but "uninformative" is the prior used in Eling et al. and previous work.
`NormFactorFun`	Function that returns cell-specific scaling normalisation factors. See `computeSumFactors` for details on the default.
`ReturnBASiCS`	Should the object be converted into a BASiCS_Chain object?
`Verbose`	Should output of the stan commands be printed to the terminal?
`...`	Passed to vb or sampling.

Value

An object of class `BASiCS_Chain`.

Examples

```
library("BASiCS")
sce <- BASiCS_MockSCE(NGenes = 10, NCells = 10)

fit_spikes <- BASiCStan(sce, tol_rel_obj = 1)

## uses fixed scaling normalisation factors
fit_nospikes <- BASiCStan(sce, WithSpikes = FALSE, tol_rel_obj = 1)
```

`Stan2BASiCS` *Convert Stan fits to* `BASiCS_Chain` *objects.*

Description

Convert Stan fits to `BASiCS_Chain` objects.

Usage

```
Stan2BASiCS(
  x,
  gene_names = attr(x, "gene_names"),
  cell_names = attr(x, "cell_names"),
  size_factors = attr(x, "size_factors")
)
```

Arguments

`x`	A stan object
`gene_names, cell_names`	Gene and cell names. The reason this argument exists is that by default, stan fit parameters are not named. NOTE: this must be the same order as the data supplied to `BASiCStan`.
`size_factors`	Cell-specific scaling normalisation factors, to be stored as part of the chain object when `WithSpikes=FALSE`.

Value

A `BASiCS_Chain` object.

Examples

```
library("BASiCS")
sce <- BASiCS_MockSCE(NGenes = 10, NCells = 10)

fit_spikes <- BASiCStan(sce, ReturnBASiCS = FALSE, tol_rel_obj = 1)
summary(fit_spikes)
```

Index