

Package ‘BADER’

November 4, 2024

Type Package

Title Bayesian Analysis of Differential Expression in RNA Sequencing Data

Version 1.45.0

Date 2012-12-21

Author Andreas Neudecker, Matthias Katzfuss

Maintainer Andreas Neudecker <a.neudecker@arcor.de>

Description For RNA sequencing count data, BADER fits a Bayesian hierarchical model. The algorithm returns the posterior probability of differential expression for each gene between two groups A and B. The joint posterior distribution of the variables in the model can be returned in the form of posterior samples, which can be used for further down-stream analyses such as gene set enrichment.

License GPL-2

Suggests pasilla (>= 0.2.10)

biocViews ImmunoOncology, Sequencing, RNASeq, DifferentialExpression, Software, SAGE

git_url <https://git.bioconductor.org/packages/BADER>

git_branch devel

git_last_commit 2798fff

git_last_commit_date 2024-10-29

Repository Bioconductor 3.21

Date/Publication 2024-11-04

Contents

| | |
|-----------------|---|
| BADER | 2 |
| Index | 4 |

Description

This function estimates the posterior distribution of various parameters regarding RNA Sequencing data. The most interesting parameter is the probability of differential expression (DE) between two groups A and B. But also estimates for the log mean and the log dispersion parameter of the underlying poisson - log-normal model can be returned.

Usage

```
BADER(x, design, sizeFactors = TRUE, start = NULL, burn = 1000, reps = 10000, printEvery = 100, saveEvery
```

Arguments

| | |
|-------------|---|
| x | m x n matrix: Every column should contain count data for a subject with m genes or tags. |
| design | Factor specifying the samples' treatment groups. The first level of 'design' corresponds to the treatment group A, the second level to treatment group B |
| sizeFactors | boolean: Whether size factors should be estimated (TRUE) or all set to 1 (FALSE) |
| start | list containing start values for MCMC sampler |
| burn | Number of burning in steps |
| reps | Number of repetitions |
| printEvery | After every printEvery iteration the current step is being printed |
| saveEvery | Every saveEvery-th step is saved for inference |
| t0 | Warming up time for Metropolis-Hastings |
| mode | How much data should be returned? Returning all posterior distributions requires large memory. <ul style="list-style-type: none"> • mode = 0: Only posterior means are returned for every parameter • mode = 1: Full posterior distribution for log fold change parameter is returned • mode = 2: Posterior distributions for the following parameters are returned: log fold change log mean and log dispersion |

Value

A list with posterior distributions / posterior means

Author(s)

Andreas Neudecker

Examples

```
set.seed(21)

## log mean expression
muA <- rnorm(100,4,1)
gam <- c(rnorm(10,0,2),rep(0,90))
muB <- muA + gam

## log dispersion
alphaA <- alphaB <- rnorm(100,-2,1)

## count tables for treatment group a and b
kA <- t(matrix(rnbinom(300,mu=exp(muA),size=exp(-alphaA)),nrow=3,byrow=TRUE))
kB <- t(matrix(rnbinom(300,mu=exp(muB),size=exp(-alphaB)),nrow=3,byrow=TRUE))

x <- cbind(kA,kB)
design <- factor(c("A","A","A","B","B","B"))

results <- BADER(x,design,burn=1000, reps=2000)

## Not run:
plot(results$diffProb,xlab="Index",ylab="posterior DE prob.")

## End(Not run)
```

Index

BADER, [2](#)