Package 'SAVER'

October 12, 2022

Type Package

Version 1.1.2

Title Single-Cell RNA-Seq Gene Expression Recovery

Description An implementation of a regularized regression prediction and empirical Bayes method to recover the true gene expression profile in noisy and sparse single-cell RNA-seq data. See Huang M, et al (2018) <doi:10.1038/s41592-018-0033-z> for more details.

Maintainer Mo Huang <mohuangx@gmail.com>

License GPL-2

Encoding UTF-8

LazyData true

Depends R (>= 3.0.1)

Imports glmnet, foreach, methods, iterators, doParallel, Matrix

RoxygenNote 7.0.0

URL https://github.com/mohuangx/SAVER

BugReports https://github.com/mohuangx/SAVER/issues

Suggests knitr, rmarkdown

VignetteBuilder knitr

NeedsCompilation no

Author Mo Huang [aut, cre],
Nancy Zhang [aut],
Mingyao Li [aut]

Repository CRAN

Date/Publication 2019-11-13 19:30:03 UTC

R topics documented:

SAVER-package 2
calc.a 3

calc.estimate . . . 3
calc.loglik.a . . . 5
calc.maxcor . . . 6
calc.post . . . 7
combine.saver . . . 7
cor.genes . . . 8
expr.predict . . . 9
get.mu . . . 10
linnarsson . . . 10
linnarsson_saver . . . 11
sample.saver . . . 11
saver . . . 12
saver.fit . . . 14

Index **17**

SAVER-package *SAVER: Single-cell Analysis Via Expression Recovery*

Description

The SAVER package implements SAVER, a gene expression recovery method for single-cell RNA sequencing (scRNA-seq). Borrowing information across all genes and cells, SAVER provides estimates for true expression levels as well as posterior distributions to account for estimation uncertainty. See Huang et al (2018) for more details.

Author(s)

Mo Huang (Maintainer), <mohuangx@gmail.com>

Nancy Zhang, <nzh@wharton.upenn.edu>

Mingyao Li, <mingyao@pennmedicine.upenn.edu>

Source

https://github.com/mohuangx/SAVER

calc.a *Optimizes variance*

Description

Finds the prior parameter that maximizes the marginal likelihood given the prediction.

Usage

```
calc.a(y, mu, sf)

calc.b(y, mu, sf)

calc.k(y, mu, sf)
```

Arguments

y	A vector of observed gene counts.
mu	A vector of predictions from expr.predict.
sf	Vector of normalized size factors.

Details

calc.a returns a prior alpha parameter assuming constant coefficient of variation. calc.b returns a prior beta parameter assuming constant Fano factor. calc.k returns a prior variance parameter assuming constant variance.

Value

A vector with the optimized parameter and the negative log-likelihood.

calc.estimate *Calculate estimate*

Description

Calculates SAVER estimate

Usage

```
calc.estimate(
  x,
  x.est,
  cutoff = 0,
  coefs = NULL,
  sf,
  scale.sf,
  pred.gene.names,
  pred.cells,
  null.model,
  nworkers,
  calc.maxcor,
  estimates.only
)

calc.estimate.mean(x, sf, scale.sf, mu, nworkers, estimates.only)

calc.estimate.null(x, sf, scale.sf, nworkers, estimates.only)
```

Arguments

x	An expression count matrix. The rows correspond to genes and the columns correspond to cells.
x.est	The log-normalized predictor matrix. The rows correspond to cells and the columns correspond to genes.
cutoff	Maximum absolute correlation to determine whether a gene should be predicted.
coefs	Coefficients of a linear fit of log-squared ratio of largest lambda to lambda of lowest cross-validation error. Used to estimate model with lowest cross-validation error.
sf	Normalized size factor.
scale.sf	Scale of size factor.
pred.gene.names	Names of genes to perform regression prediction.
pred.cells	Index of cells to perform regression prediction.
null.model	Whether to use mean gene expression as prediction.
nworkers	Number of cores registered to parallel backend.
calc.maxcor	Whether to calculate maximum absolute correlation.
estimates.only	Only return SAVER estimates. Default is FALSE.
mu	Matrix of prior means

Details

The SAVER method starts by estimating the prior mean and variance for the true expression level for each gene and cell. The prior mean is obtained through predictions from a LASSO Poisson

regression for each gene implemented using the `glmnet` package. Then, the variance is estimated through maximum likelihood assuming constant variance, Fano factor, or coefficient of variation variance structure for each gene. The posterior distribution is calculated and the posterior mean is reported as the SAVER estimate.

Value

A list with the following components

`est`	Recovered (normalized) expression
`se`	Standard error of estimates
`maxcor`	Maximum absolute correlation for each gene. 2 if not calculated
`lambda.max`	Smallest value of lambda which gives the null model.
`lambda.min`	Value of lambda from which the prediction model is used
`sd.cv`	Difference in the number of standard deviations in deviance between the model with lowest cross-validation error and the null model
`ct`	Time taken to generate predictions.
`vt`	Time taken to estimate variance.

`calc.loglik.a` *Calculates marginal likelihood*

Description

Calculates the marginal likelihood given the prediction under constant coefficient of variation (a), Fano factor (b), and variance (k).

Usage

```
calc.loglik.a(a, y, mu, sf)

calc.loglik.b(b, y, mu, sf)

calc.loglik.k(k, y, mu, sf)
```

Arguments

`a, b, k`	Prior parameter.
`y`	A vector of observed gene counts.
`mu`	A vector of predictions from `expr.predict`.
`sf`	Vector of normalized size factors.

Details

`calc.loglik.a` returns the shifted negative log-likelihood under constant coefficient of variation. `calc.loglik.b` returns the shifted negative log-likelihood under constant Fano factor. `calc.loglik.k` returns the shifted negative log-likelihood under constant variance.

Value

A shifted negative marginal log-likelihood.

`calc.maxcor` *Calculate maximum correlation*

Description

Calculates the maximum absolute correlation between two matrices along the columns

Usage

```
calc.maxcor(x1, x2)
```

Arguments

`x1`	Named matrix 1
`x2`	Named matrix 2

Details

This function calculates the maximum absolute correlation for each column of `x2` against each column of `x1`. The matrices are named and if the names overlap between `x1` and `x2`, the correlation between the same named entries is set to zero.

Value

A vector of maximum absolute correlations

Examples

```
x1 <- matrix(rnorm(500), 100, 5)
x2 <- x1 + matrix(rnorm(500), 100, 5)
colnames(x1) <- c("A", "B", "C", "D", "E")
colnames(x2) <- c("A", "B", "C", "D", "E")
cor(x1, x2)
calc.maxcor(x1, x2)
```

calc.post *Calculates SAVER posterior*

Description

Given prediction and prior variance, calculates the Gamma posterior distribution parameters for a single gene.

Usage

```
calc.post(y, mu, sf, scale.sf)
```

Arguments

y	A vector of observed gene counts.
mu	A vector of prior means.
sf	Vector of normalized size factors.
scale.sf	Mean of the original size factors.

Details

Let α be the shape parameter and β be the rate parameter of the prior Gamma distribution. Then, the posterior Gamma distribution will be

$$Gamma(y + \alpha, sf + \beta),$$

where y is the observed gene count and sf is the size factor.

Value

A list with the following components

estimate	Recovered (normalized) expression
se	Standard error of expression estimate

combine.saver *Combines SAVER*

Description

Combines SAVER objects

Usage

```
combine.saver(saver.list)
```

Arguments

saver.list	List of SAVER objects

Details

If SAVER was applied to a dataset for chunks of genes (by specifying pred.genes and pred.genes.only = TRUE), this function combines the individual SAVER objects into one SAVER object.

Value

A combined SAVER object

Examples

```
data("linnarsson")

## Not run:
a <- saver(linnarsson, pred.genes = 1:5, pred.genes.only = TRUE)
b <- saver(linnarsson, pred.genes = 6:10, pred.genes.only = TRUE)
ab <- combine.saver(list(a, b))

## End(Not run)
```

---

cor.genes *Calculates gene-to-gene and cell-to-cell SAVER correlation*

Description

Adjusts for SAVER estimation uncertainty by calculating and adjusting gene-to-gene and cell-to-cell correlation matrices

Usage

```
cor.genes(x, cor.mat = NULL)

cor.cells(x, cor.mat = NULL)
```

Arguments

x	A saver object.
cor.mat	If a correlation matrix of the SAVER estimates was already obtained, then it can be provided as an input to avoid recomputation.

Details

The SAVER estimates that are produced have varying levels of uncertainty depending on the gene and the cell. These functions adjust the gene-to-gene and cell-to-cell correlations of the SAVER estimates to reflect the estimation uncertainty.

Value

An adjusted correlation matrix.

Examples

```
data("linnarsson_saver")

gene.cor <- cor.genes(linnarsson_saver)
```

`expr.predict` *Calculates SAVER prediction.*

Description

Uses `cv.glmnet` from the `glmnet` package to return the SAVER prediction.

Usage

```
expr.predict(
  x,
  y,
  pred.cells = 1:length(y),
  seed = NULL,
  lambda.max = NULL,
  lambda.min = NULL
)
```

Arguments

`x`	A log-normalized expression count matrix of genes to be used in the prediction.
`y`	A normalized expression count vector of the gene to be predicted.
`pred.cells`	Index of cells to use for prediction. Default is to use all cells.
`seed`	Sets the seed for reproducible results.
`lambda.max`	Maximum value of lambda which gives null model.
`lambda.min`	Value of lambda from which the prediction model is used

Details

The SAVER method starts with predicting the prior mean for each cell for a specific gene. The prediction is performed using the observed normalized gene count as the response and the normalized gene counts of other genes as predictors. `cv.glmnet` from the `glmnet` package is used to fit the LASSO Poisson regression. The model with the lowest cross-validation error is chosen and the fitted response values are returned and used as the SAVER prediction.

Value

A vector of predicted gene expression.

get.mu *Output prior predictions*

Description

Outputs prior predictions

Usage

```
get.mu(x, saver.obj)
```

Arguments

x	Original count matrix.
saver.obj	SAVER output.

Details

This function outputs prior mean predictions μ used in fitting the SAVER model.

Value

A matrix of prior mean predictions

Examples

```
data("linnarsson")
data("linnarsson_saver")

mu <- get.mu(linnarsson, linnarsson_saver)
```

linnarsson *Mouse brain single-cell RNA-seq dataset*

Description

3,529 genes and 200 cells from a mouse brain scRNA-seq dataset.

Usage

```
linnarsson
```

Format

An object of class `matrix` with 3529 rows and 200 columns.

References

Zeisel, A., Munoz-Manchado, A. B., Codeluppi, S., Lonnerberg, P., La Manno, G., Jureus, A., ... Linnarsson, S. (2015). Cell types in the mouse cortex and hippocampus revealed by single-cell RNA-seq. *Science, 347*(6226), 1138-1142.

`linnarsson_saver` *SAVER recovered mouse brain single-cell RNA-seq dataset*

Description

Output of running ‘saver‘ on the ‘linnarsson‘ dataset.

Usage

```
linnarsson_saver
```

Format

An object of class `saver` of length 3.

References

Zeisel, A., Munoz-Manchado, A. B., Codeluppi, S., Lonnerberg, P., La Manno, G., Jureus, A., ... Linnarsson, S. (2015). Cell types in the mouse cortex and hippocampus revealed by single-cell RNA-seq. *Science, 347*(6226), 1138-1142.

`sample.saver` *Samples from SAVER*

Description

Samples from the posterior distribution output by SAVER.

Usage

```
sample.saver(x, rep = 1, efficiency.known = FALSE, seed = NULL)
```

Arguments

`x`	A `saver` object.
`rep`	Number of sampled datasets. Default is 1.
`efficiency.known`	Whether the efficiency is known. Default is `FALSE`.
`seed`	seed used in `set.seed`.

Details

The SAVER method outputs a posterior distribution, which we can sample from for downstream analysis. The posterior distribution accounts for uncertainty in the SAVER estimation procedure. If the efficiency is known, negative binomial sampling is performed; otherwise, gamma sampling is performed.

Value

A matrix of expression values sampled from SAVER posterior. If `rep` > 1, a list of matrices is returned

Examples

```
data("linnarsson_saver")

samp1 <- sample.saver(linnarsson_saver, seed = 50)
```

`saver` *Runs SAVER*

Description

Recovers expression using the SAVER method.

Usage

```
saver(
  x,
  do.fast = TRUE,
  ncores = 1,
  size.factor = NULL,
  npred = NULL,
  pred.cells = NULL,
  pred.genes = NULL,
  pred.genes.only = FALSE,
  null.model = FALSE,
  mu = NULL,
  estimates.only = FALSE
)
```

Arguments

`x`	An expression count matrix. The rows correspond to genes and the columns correspond to cells. Can be sparse.
`do.fast`	Approximates the prediction step. Default is TRUE.

ncores	Number of cores to use. Default is 1.
size.factor	Vector of cell size normalization factors. If x is already normalized or normalization is not desired, use size.factor = 1. Default uses mean library size normalization.
npred	Number of genes for regression prediction. Selects the top npred genes in terms of mean expression for regression prediction. Default is all genes.
pred.cells	Indices of cells to perform regression prediction. Default is all cells.
pred.genes	Indices of specific genes to perform regression prediction. Overrides npred. Default is all genes.
pred.genes.only	Return expression levels of only pred.genes. Default is FALSE (returns expression levels of all genes).
null.model	Whether to use mean gene expression as prediction.
mu	Matrix of prior means.
estimates.only	Only return SAVER estimates. Default is FALSE.

Details

The SAVER method starts by estimating the prior mean and variance for the true expression level for each gene and cell. The prior mean is obtained through predictions from a LASSO Poisson regression for each gene implemented using the glmnet package. Then, the variance is estimated through maximum likelihood assuming constant variance, Fano factor, or coefficient of variation variance structure for each gene. The posterior distribution is calculated and the posterior mean is reported as the SAVER estimate.

Value

If 'estimates.only = TRUE', then a matrix of SAVER estimates.

If 'estimates.only = FALSE', a list with the following components

estimate	Recovered (normalized) expression.
se	Standard error of estimates.
info	Information about dataset.

The info element is a list with the following components:

size.factor	Size factor used for normalization.
maxcor	Maximum absolute correlation for each gene. 2 if not calculated
lambda.max	Smallest value of lambda which gives the null model.
lambda.min	Value of lambda from which the prediction model is used
sd.cv	Difference in the number of standard deviations in deviance between the model with lowest cross-validation error and the null model
pred.time	Time taken to generate predictions.
var.time	Time taken to estimate variance.

maxcor	Maximum absolute correlation cutoff used to determine if a gene should be predicted.
lambda.coefs	Coefficients for estimating lambda with lowest cross-validation error.
total.time	Total time for SAVER estimation.

Examples

```
data("linnarsson")

## Not run:
system.time(linnarsson_saver <- saver(linnarsson, ncores = 12))

## End(Not run)

# predictions for top 5 highly expressed genes
## Not run:
saver2 <- saver(linnarsson, npred = 5)

## End(Not run)

# predictions for certain genes
## Not run:
genes <- c("Thy1", "Mbp", "Stim2", "Psmc6", "Rps19")
genes.ind <- which(rownames(linnarsson)
saver3 <- saver(linnarsson, pred.genes = genes.ind)

## End(Not run)

# return only certain genes
## Not run:
saver4 <- saver(linnarsson, pred.genes = genes.ind, pred.genes.only = TRUE)

## End(Not run)
```

---

saver.fit *Fits SAVER*

Description

Fits SAVER object

Usage

```
saver.fit(
  x,
  x.est,
  do.fast,
  ncores,
```

```
  sf,
  scale.sf,
  pred.genes,
  pred.cells,
  null.model,
  ngenes = nrow(x),
  ncells = ncol(x),
  gene.names = rownames(x),
  cell.names = colnames(x),
  estimates.only
)

saver.fit.mean(
  x,
  ncores,
  sf,
  scale.sf,
  mu,
  ngenes = nrow(x),
  ncells = ncol(x),
  gene.names = rownames(x),
  cell.names = colnames(x),
  estimates.only
)

saver.fit.null(
  x,
  ncores,
  sf,
  scale.sf,
  ngenes = nrow(x),
  ncells = ncol(x),
  gene.names = rownames(x),
  cell.names = colnames(x),
  estimates.only
)
```

Arguments

x	An expression count matrix. The rows correspond to genes and the columns correspond to cells.
x.est	The log-normalized predictor matrix. The rows correspond to cells and the columns correspond to genes.
do.fast	Approximates the prediction step. Default is TRUE.
ncores	Number of cores to use. Default is 1.
sf	Normalized size factor.
scale.sf	Scale of size factor.

pred.genes	Index of genes to perform regression prediction.
pred.cells	Index of cells to perform regression prediction.
null.model	Whether to use mean gene expression as prediction.
ngenes	Number of genes.
ncells	Number of cells.
gene.names	Name of genes.
cell.names	Name of cells.
estimates.only	Only return SAVER estimates. Default is FALSE.
mu	Matrix of prior means.

Details

The SAVER method starts by estimating the prior mean and variance for the true expression level for each gene and cell. The prior mean is obtained through predictions from a Lasso Poisson regression for each gene implemented using the `glmnet` package. Then, the variance is estimated through maximum likelihood assuming constant variance, Fano factor, or coefficient of variation variance structure for each gene. The posterior distribution is calculated and the posterior mean is reported as the SAVER estimate.

Value

A list with the following components

estimate	Recovered (normalized) expression
se	Standard error of estimates
info	Information about fit

Index

∗ **datasets**
linnarsson, 10
linnarsson_saver, 11

calc.a, 3
calc.b *(*calc.a*)*, 3
calc.estimate, 3
calc.k *(*calc.a*)*, 3
calc.loglik.a, 5
calc.loglik.b *(*calc.loglik.a*)*, 5
calc.loglik.k *(*calc.loglik.a*)*, 5
calc.maxcor, 6
calc.post, 7
combine.saver, 7
cor.cells *(*cor.genes*)*, 8
cor.genes, 8

expr.predict, *3*, *5*, 9

get.mu, 10

linnarsson, 10
linnarsson_saver, 11

sample.saver, 11
saver, 12
SAVER-package, 2
saver.fit, 14