

Package ‘stochprofML’

October 14, 2022

Title Stochastic Profiling using Maximum Likelihood Estimation

Version 2.0.3

DateNote Previous CRAN version 2.0.2 on 2020-05-14

Maintainer Lisa Amrhein <amrheinlisa@gmail.com>

Depends R (>= 2.0)

Imports MASS, numDeriv

Description New Version of the R package originally accompanying the paper “Parameterizing cell-to-cell regulatory heterogeneities via stochastic transcriptional profiles” by Sameer S Bajikar, Christiane Fuchs, Andreas Roller, Fabian J Theis and Kevin A Janes (PNAS 2014, 111(5), E626-635 <[doi:10.1073/pnas.1311647111](https://doi.org/10.1073/pnas.1311647111)>). In this paper, we measure expression profiles from small heterogeneous populations of cells, where each cell is assumed to be from a mixture of lognormal distributions. We perform maximum likelihood estimation in order to infer the mixture ratio and the parameters of these lognormal distributions from the cumulated expression measurements. The main difference of this new package version to the previous one is that it is now possible to use different n’s, i.e. a dataset where each tissue sample originates from a different number of cells. We used this on pheno-seq data, see: Tirier, S.M., Park, J., Preusser, F. et al. Pheno-seq - linking visual features and gene expression in 3D cell culture systems. Sci Rep 9, 12367 (2019) <[doi:10.1038/s41598-019-48771-4](https://doi.org/10.1038/s41598-019-48771-4)>).

License GPL (>= 2)

LazyData TRUE

RoxygenNote 6.1.1

NeedsCompilation no

Author Lisa Amrhein [aut, cre],
Christiane Fuchs [aut],
Christoph Kurz [ctb] (Author to function comb.summands.R')

Repository CRAN

Date/Publication 2020-06-10 10:00:03 UTC

R topics documented:

stochprofML-package	2
analyze.sod2	4
analyze.toycluster	6
calculate.ci.EXPLN	8
calculate.ci.LNLN	9
calculate.ci.rLNLN	11
comb.summands	12
d.sum.of.mixtures.EXPLN	13
d.sum.of.mixtures.LNLN	15
d.sum.of.mixtures.rLNLN	17
generate.toydata	18
mix.d.sum.of.mixtures.EXPLN	20
mix.d.sum.of.mixtures.LNLN	21
mix.d.sum.of.mixtures.rLNLN	22
penalty.constraint.EXPLN	23
penalty.constraint.LNLN	25
penalty.constraint.rLNLN	26
set.model.functions	27
sod2	28
stochasticProfilingData	28
stochasticProfilingML	29
stochprof.loop	30
stochprof.results.EXPLN	33
stochprof.results.LNLN	34
stochprof.results.rLNLN	36
stochprof.search.EXPLN	37
stochprof.search.LNLN	39
stochprof.search.rLNLN	40
toycluster.EXPLN	42
toycluster.LNLN	43
toycluster.rLNLN	44
Index	46

stochprofML-package *Stochastic Profiling using Maximum Likelihood Estimation*

Description

This package accompanies the paper "Parameterizing cell-to-cell regulatory heterogeneities via stochastic transcriptional profiles" by Sameer S Bajikar, Christiane Fuchs, Andreas Roller, Fabian J Theis and Kevin A Janes (PNAS 2014, 111(5), E626-635). In that work, we measure expression profiles from small heterogeneous populations of cells. Each cell is assumed to be from a mixture of lognormal and exponential distributions (see details). We perform maximum likelihood estimation in order to infer the mixture ratio and the parameters of the lognormal/exponential distributions from the cumulated expression measurements.

Details

Package: stochprofML
Type: Package
Version: 2.0
Date: 2018-05-07
License: GPL (>= 2)

There are three stochastic profiling models: The lognormal-lognormal (LN-LN) model assumes that each cell is from a mixture of one or more lognormal distributions with different log-means but identical log-standard deviations. In the relaxed lognormal-lognormal (rLN-LN) model, the log-standard deviations are not necessarily identical. The exponential-lognormal (EXP-LN) model considers the mixture of zero, one or more lognormal distributions and one exponential distribution.

The user can immediately start with data generation and model estimation using the two functions `stochasticProfilingData` and `stochasticProfilingML`, respectively. These functions prompt the user to input the settings/data. Typical analyses are then performed without the user having to delve into the structure of this package.

When not using one of the above two functions, parameters can be estimated calling `stochprof.loop`, which again utilizes three other functions: `stochprof.search.LNLN`/`stochprof.search.rLNLN`/`stochprof.search.EXPLN` in order to calculate and locally optimize the likelihood function; `stochprof.results.LNLN`/`stochprof.results.rLNLN`/`stochprof.results.EXPLN` for evaluating these results; and `calculate.ci.LNLN`/`calculate.ci.rLNLN`/`calculate.ci.EXPLN` for calculating confidence intervals.

Two essential functions are `r.sum.of.mixtures.LNLN`/`r.sum.of.mixtures.rLNLN`/`r.sum.of.mixtures.EXPLN` and `d.sum.of.mixtures.LNLN`/`d.sum.of.mixtures.rLNLN`/`d.sum.of.mixtures.EXPLN` for the density and random number generation of the distribution assumed for all measurements in the stochastic profiling model.

Version 2.0 shows one significant extension to previous versions: The number of cells n in the input data does no longer need to be the same over all samples.

The package provides four datasets: `sod2`, containing real measurements for one gene, and `toycluster.LNLN`/`toycluster.rLNLN`/`toycluster.EXPLN`, containing artificial data for 12 genes generated with the three stochastic profiling models.

Examples for typical analyses are given below.

Author(s)

Lisa Amrhein, Christiane Fuchs

Maintainer: Lisa Amrhein <amrheinlisa@gmail.com>

References

"Parameterizing cell-to-cell regulatory heterogeneities via stochastic transcriptional profiles" by Sameer S Bajikar*, Christiane Fuchs*, Andreas Roller, Fabian J Theis[^] and Kevin A Janes[^]: PNAS 2014, 111(5), E626-635 (* joint first authors, [^] joint last authors) <doi:10.1073/pnas.1311647111>

"Pheno-seq - linking visual features and gene expression in 3D cell culture systems" by Stephan M. Tirier, Jeongbin Park, Friedrich Preusser, Lisa Amrhein, Zuguang Gu, Simon Steiger, Jan-Philipp Mallm, Teresa Krieger, Marcel Waschow, Bjoern Eismann, Marta Gut, Ivo G. Gut, Karsten Rippe, Matthias Schlesner, Fabian Theis, Christiane Fuchs, Claudia R. Ball, Hanno Glimm, Roland Eils & Christian Conrad: Sci Rep 9, 12367 (2019) <doi:10.1038/s41598-019-48771-4>

Examples

```
## Not run:
# Generate synthetic data. The user is prompted to input all settings.
stochasticProfilingData()

# Estimate a stochastic profiling model. The user is prompted to input the data
# and other parameters.
stochasticProfilingML()

# Generate a synthetic dataset (without measurement error) for one gene
# and estimate the parameters from this data.
generate.toydata()

# Estimate the model parameters for the SOD2 dataset.
analyze.sod2()

# Estimate the model parameters for the 12-gene toycluster.
analyze.toycluster()

## End(Not run)
```

analyze.sod2

Analysis of SOD2 data in stochastic profiling model

Description

Estimation of the model parameters for the SOD2 dataset provided in this package.

Usage

```
analyze.sod2(model = "LN-LN", TY = 2, use.constraints = F)
```

Arguments

model	model for which one wishes to estimate the parameters: "LN-LN", "rLN-LN" or "EXP-LN"
TY	number of types of cells that is assumed in the stochastic model
use.constraints	if TRUE, constraints on the individual population densities are applied; see <code>penalty.constraint.LNLN</code> , <code>penalty.constraint.rLNLN</code> and <code>penalty.constraint.EXPLN</code> for details.

Details

The `sod2` dataset contains real 10-cell samplings from the detoxifying enzyme, SOD2. This function estimates the parameters of the stochastic profiling models for this data. At the end, it graphically represents a histogram of the SOD2 data together with the estimated probability density function.

Value

A list as returned by `stochprof.loop`, i.e. the following components:

<code>mle</code>	maximum likelihood estimate
<code>neg-loglikeli</code>	value of the negative log-likelihood function at maximum likelihood estimate
<code>ci</code>	approximate marginal maximum likelihood confidence intervals for the maximum likelihood estimate
<code>pargrid</code>	matrix containing parameter combinations and according values of the target function
<code>bic</code>	Bayesian information criterion value
<code>adj.bic</code>	adjusted Bayesian information criterion value which takes into account the numbers of parameters that were estimated during the preanalysis of a gene cluster (not applicable here, hence NULL)
<code>pen</code>	penalization for densities not fulfilling required constraints. If <code>use.constraints</code> is FALSE, this has no practical meaning. If <code>use.constraints</code> is TRUE, this value is included in <code>loglikeli</code> .

Author(s)

Lisa Amrhein, Christiane Fuchs

Maintainer: Lisa Amrhein <amrheinlisa@gmail.com>

References

"Parameterizing cell-to-cell regulatory heterogeneities via stochastic transcriptional profiles" by Sameer S Bajikar*, Christiane Fuchs*, Andreas Roller, Fabian J Theis[^] and Kevin A Janes[^]: PNAS 2014, 111(5), E626-635 (* joint first authors, [^] joint last authors) <doi:10.1073/pnas.1311647111>

"Pheno-seq - linking visual features and gene expression in 3D cell culture systems" by Stephan M. Tirier, Jeongbin Park, Friedrich Preusser, Lisa Amrhein, Zuguang Gu, Simon Steiger, Jan-Philipp Mallm, Teresa Krieger, Marcel Waschow, Bjoern Eismann, Marta Gut, Ivo G. Gut, Karsten Rippe, Matthias Schlesner, Fabian Theis, Christiane Fuchs, Claudia R. Ball, Hanno Glimm, Roland Eils & Christian Conrad: Sci Rep 9, 12367 (2019) <doi:10.1038/s41598-019-48771-4>

analyze.toycluster *Analysis of toyclusters in stochastic profiling model*

Description

Estimation of the model parameters for the 12-gene toyclusters provided in this package. This is done in three steps: an optional preanalysis of the single genes, an analysis of three 4-gene subclusters, and finally the analysis of the entire 12-gene cluster.

Usage

```
analyze.toycluster(model = "LN-LN", data.model = "LN-LN", TY = 2,
  preanalyze = T, show.plots = T, use.constraints = F)
```

Arguments

model	model for which one wishes to estimate the parameters: "LN-LN", "rLN-LN" or "EXP-LN"
data.model	model which has generated the 12-gene dataset: "LN-LN", "rLN-LN" or "EXP-LN"
TY	number of types of cells that is assumed in the stochastic model
preanalyze	if TRUE, the single-gene preanalysis as described below is carried out
show.plots	if TRUE, interim results are graphically displayed. This requires the user to confirm each new plot.
use.constraints	if TRUE, constraints on the individual population densities are applied; see <code>penalty.constraint.LNLN</code> , <code>penalty.constraint.rLNLN</code> and <code>penalty.constraint.EXPLN</code> for details.

Details

This function carries out estimation of the model parameters for the `toycluster.LNLN`, `toycluster.rLNLN` or `toycluster.EXPLN` dataset. This contains perfectly observed measurements for 12 genes and 16 tissue samples, assuming 10-cell samplings and two different types of cells. The true underlying parameters are given on the help page for the datasets.

The estimation is performed in three steps:

In an optional preanalysis (carried out if `preanalyze` is TRUE), each gene is considered individually, i.e. for each gene the parameters are estimated (these are p , μ_1 , μ_2 and σ for LN-LN, p , μ_1 , μ_2 , σ_1 and σ_2 for rLN-LN, and p , μ , σ and λ for EXP-LN). This gives a rough idea about the location of the parameters at computationally low cost. This might speed up the analysis of the larger clusters. From the confidence intervals of the single-gene estimates, one can construct appropriate parameter ranges for the following step.

In the main step of the estimation procedure, the 12 genes are divided into three groups of size four. This is because the stochastic profiling model for 12 genes involves 48 (LN-LN and EXP-LN) to 49 (rLN-LN) parameters, which is computationally expensive and sometimes unreliable. Simulation

studies showed that datasets comprising four genes are sufficient to estimate the log-means when there is data from 16 experiments available. For each of these 4-gene clusters, 10 (LN-LN and EXP-LN) or 11 (rLN-LN) parameters are estimated. The three groups result from a hierarchical clustering of the entire dataset. The genes numbers are (7,5,2,8), (1,3,4,10) and (9,6,12,11) for the LN-LN model, (12,9,6,11), (4,10,5,3) and (1,7,8,2) for the rLN-LN model and (11,1,10,9), (3,5,8,7) and (4,2,12,6) for the EXP-LN model.

In the final step, the log-means μ are fixed to the maximum likelihood estimates that resulted from the main step. Then there remain only p , σ and possibly λ to be estimated. These are inferred now.

Throughout the whole estimation process, interim results are printed into the console and, if `show.plots` is TRUE, graphically displayed.

Value

The final result for the chosen 12-gene cluster. That is a list as returned by `stochprof.loop`, i.e. the following components:

<code>mle</code>	maximum likelihood estimate
<code>neg-loglikeli</code>	value of the negative log-likelihood function at maximum likelihood estimate
<code>ci</code>	approximate marginal maximum likelihood confidence intervals for the maximum likelihood estimate
<code>pargrid</code>	matrix containing parameter combinations and according values of the target function
<code>bic</code>	Bayesian information criterion value
<code>adj.bic</code>	adjusted Bayesian information criterion value which takes into account the numbers of parameters that were estimated during the preanalysis of the gene cluster
<code>pen</code>	penalization for densities not fulfilling required constraints. If use.constraints is FALSE, this has no practical meaning. If use.constraints is TRUE, this value is included in <code>loglikeli</code> .

Author(s)

Lisa Amrhein, Christiane Fuchs

Maintainer: Lisa Amrhein <amrheinlisa@gmail.com>

References

"Parameterizing cell-to-cell regulatory heterogeneities via stochastic transcriptional profiles" by Sameer S Bajikar*, Christiane Fuchs*, Andreas Roller, Fabian J Theis[^] and Kevin A Janes[^]: PNAS 2014, 111(5), E626-635 (* joint first authors, [^] joint last authors) <doi:10.1073/pnas.1311647111>

"Pheno-seq - linking visual features and gene expression in 3D cell culture systems" by Stephan M. Tirier, Jeongbin Park, Friedrich Preusser, Lisa Amrhein, Zuguang Gu, Simon Steiger, Jan-Philipp Mallm, Teresa Krieger, Marcel Waschow, Bjoern Eismann, Marta Gut, Ivo G. Gut, Karsten Rippe, Matthias Schlesner, Fabian Theis, Christiane Fuchs, Claudia R. Ball, Hanno Glimm, Roland Eils & Christian Conrad: Sci Rep 9, 12367 (2019) <doi:10.1038/s41598-019-48771-4>

calculate.ci.EXPLN *Maximum likelihood confidence intervals for EXP-LN model*

Description

Calculates approximate marginal maximum likelihood confidence intervals with significance level α for all parameters in the EXP-LN model.

Usage

```
calculate.ci.EXPLN(alpha, parameter, prev.result, dataset, n, TY,
  fix.mu = F, fixed.mu)
```

Arguments

alpha	the significance level
parameter	the maximum likelihood estimate around which the confidence interval is centered; if this value is missing, it is determined from prev.result. This parameter has to be on the original scale, not on the logit-/log-transformed scale as used during the estimation procedure. It has to be $TY \cdot (m+1)$ -dimensional (or m -dimensional, if $TY=1$), even for $fix.mu==T$.
prev.result	a list of previous results as returned by stochprof.results. This variable is only used when 'parameter' is missing.
dataset	a matrix which contains the cumulated expression data over all cells in a tissue sample. Columns represent different genes, rows represent different tissue samples.
n	number of cells taken from each tissue sample. This can also be a vector stating how many cells are in each sample separately.
TY	the number of types of cells that is assumed in the stochastic model
fix.mu	if TRUE, the log-means of the lognormal distributions are kept fixed in the estimation procedure. Otherwise, they are to be estimated.
fixed.mu	a vector containing the values to which the log-means should be fixed if $fix.mu==T$. The order of components is as follows: (mu_type_1_gene_1, mu_type_1_gene_2, ..., mu_type_2_gene_1, mu_type_2_gene_2, ...). The parameter needs to be specified only when $fix.mu==T$.

Details

The intervals are approximate because the function uses the formula

$$[\theta_i \pm q_{(1-\alpha/2)} * \sqrt{H_{ii}}],$$

where θ_i is the estimate of the i .th parameter, $q_{(1-\alpha/2)}$ is the $1-\alpha/2$ quantile of the standard normal distribution, H is the inverse Hessian of the negative log likelihood function evaluated at the maximum likelihood estimate; H_{ii} is the i .th diagonal element of H . This approximation

implicitly assumes that the log likelihood function is unimodal. The confidence interval is first calculated on the transformed, unrestricted parameter space and then back-transformed to the original one.

Value

Approximate marginal maximum likelihood confidence intervals for all parameter components: Each row corresponds to one parameter (in the same order as always used in the stochastic profiling model). The first column contains lower bounds, the second column upper bounds.

Author(s)

Lisa Amrhein, Christiane Fuchs

Maintainer: Lisa Amrhein <amrheinlisa@gmail.com>

References

"Parameterizing cell-to-cell regulatory heterogeneities via stochastic transcriptional profiles" by Sameer S Bajikar*, Christiane Fuchs*, Andreas Roller, Fabian J Theis[^] and Kevin A Janes[^]: PNAS 2014, 111(5), E626-635 (* joint first authors, [^] joint last authors) <doi:10.1073/pnas.1311647111>

"Pheno-seq - linking visual features and gene expression in 3D cell culture systems" by Stephan M. Tirier, Jeongbin Park, Friedrich Preusser, Lisa Amrhein, Zuguang Gu, Simon Steiger, Jan-Philipp Mallm, Teresa Krieger, Marcel Waschow, Bjoern Eismann, Marta Gut, Ivo G. Gut, Karsten Rippe, Matthias Schlesner, Fabian Theis, Christiane Fuchs, Claudia R. Ball, Hanno Glimm, Roland Eils & Christian Conrad: Sci Rep 9, 12367 (2019) <doi:10.1038/s41598-019-48771-4>

calculate.ci.LNLN

Maximum likelihood confidence intervals for LN-LN model

Description

Calculates approximate marginal maximum likelihood confidence intervals with significance level alpha for all parameters in the LN-LN model.

Usage

```
calculate.ci.LNLN(alpha, parameter, prev.result, dataset, n, TY,
  fix.mu = F, fixed.mu)
```

Arguments

alpha	the significance level
parameter	the maximum likelihood estimate around which the confidence interval is centered; if this value is missing, it is determined from prev.result. This parameter has to be on the original scale, not on the logit-/log-transformed scale as used during the estimation procedure. It has to be TY*(m+1)-dimensional, even for fix.mu==T.

prev.result	a list of previous results as returned by stochprof.results. This variable is only used when 'parameter' is missing.
dataset	a matrix which contains the cumulated expression data over all cells in a tissue sample. Columns represent different genes, rows represent different tissue samples.
n	number of cells taken from each tissue sample. This can also be a vector stating how many cells are in each sample separatly.
TY	the number of types of cells that is assumed in the stochastic model
fix.mu	if TRUE, the log-means are kept fixed in the estimation procedure. Otherwise, they are to be estimated.
fixed.mu	a vector containing the values to which the log-means should be fixed if fix.mu==T. The order of components is as follows: (mu_type_1_gene_1, mu_type_1_gene_2, ..., mu_type_2_gene_1, mu_type_2_gene_2, ...). The parameter needs to be specified only when fix.mu==T.

Details

The intervals are approximate because the function uses the formula

$$[\text{theta}_i \pm q_{(1-\alpha/2)} * \text{sqrt}(H_{ii})],$$

where theta_i is the estimate of the i .th parameter, $q_{(1-\alpha/2)}$ is the $1-\alpha/2$ quantile of the standard normal distribution, H is the inverse Hessian of the negative log likelihood function evaluated at the maximum likelihood estimate; H_{ii} is the i .th diagonal element of H . This approximation implicitly assumes that the log likelihood function is unimodal. The confidence interval is first calculated on the transformed, unrestricted parameter space and then back-transformed to the original one.

Value

Approximate marginal maximum likelihood confidence intervals for all parameter components: Each row corresponds to one parameter (in the same order as always used in the stochastic profiling model). The first column contains lower bounds, the second column upper bounds.

Author(s)

Lisa Amrhein, Christiane Fuchs

Maintainer: Lisa Amrhein <amrheinlisa@gmail.com>

References

"Parameterizing cell-to-cell regulatory heterogeneities via stochastic transcriptional profiles" by Sameer S Bajikar*, Christiane Fuchs*, Andreas Roller, Fabian J Theis[^] and Kevin A Janes[^]: PNAS 2014, 111(5), E626-635 (* joint first authors, [^] joint last authors) <doi:10.1073/pnas.1311647111>

"Pheno-seq - linking visual features and gene expression in 3D cell culture systems" by Stephan M. Tirier, Jeongbin Park, Friedrich Preusser, Lisa Amrhein, Zuguang Gu, Simon Steiger, Jan-Philipp Mallm, Teresa Krieger, Marcel Waschow, Bjoern Eismann, Marta Gut, Ivo G. Gut, Karsten Rippe,

Matthias Schlesner, Fabian Theis, Christiane Fuchs, Claudia R. Ball, Hanno Glimm, Roland Eils & Christian Conrad: Sci Rep 9, 12367 (2019) <doi:10.1038/s41598-019-48771-4>

calculate.ci.rLNLN *Maximum likelihood confidence intervals for rLN-LN model*

Description

Calculates approximate marginal maximum likelihood confidence intervals with significance level alpha for all parameters in the rLN-LN model.

Usage

```
calculate.ci.rLNLN(alpha, parameter, prev.result, dataset, n, TY,
  fix.mu = F, fixed.mu)
```

Arguments

alpha	the significance level
parameter	the maximum likelihood estimate around which the confidence interval is centered; if this value is missing, it is determined from prev.result. This parameter has to be on the original scale, not on the logit-/log-transformed scale as used during the estimation procedure. It has to be $((m+1)*TY-1)$ -dimensional, even for fix.mu==T.
prev.result	a list of previous results as returned by stochprof.results. This variable is only used when 'parameter' is missing.
dataset	a matrix which contains the cumulated expression data over all cells in a tissue sample. Columns represent different genes, rows represent different tissue samples.
n	number of cells taken from each tissue sample. This can also be a vector stating how many cells are in each sample separatly.
TY	the number of types of cells that is assumed in the stochastic model
fix.mu	if TRUE, the log-means are kept fixed in the estimation procedure. Otherwise, they are to be estimated.
fixed.mu	a vector containing the values to which the log-means should be fixed if fix.mu==T. The order of components is as follows: (mu_type_1_gene_1, mu_type_1_gene_2, ..., mu_type_2_gene_1, mu_type_2_gene_2, ...). The parameter needs to be specified only when fix.mu==T.

Details

The intervals are approximate because the function uses the formula

$$[\theta_i \pm q_{(1-\alpha/2)} * \sqrt{H_{ii}}],$$

where θ_i is the estimate of the i .th parameter, $q_{(1-\alpha/2)}$ is the $1-\alpha/2$ quantile of the standard normal distribution, H is the inverse Hessian of the negative log likelihood function evaluated at the maximum likelihood estimate; H_{ii} is the i .th diagonal element of H . This approximation implicitly assumes that the log likelihood function is unimodal. The confidence interval is first calculated on the transformed, unrestricted parameter space and then back-transformed to the original one.

Value

Approximate marginal maximum likelihood confidence intervals for all parameter components: Each row corresponds to one parameter (in the same order as always used in the stochastic profiling model). The first column contains lower bounds, the second column upper bounds.

Author(s)

Lisa Amrhein, Christiane Fuchs

Maintainer: Lisa Amrhein <amrheinlisa@gmail.com>

References

"Parameterizing cell-to-cell regulatory heterogeneities via stochastic transcriptional profiles" by Sameer S Bajikar*, Christiane Fuchs*, Andreas Roller, Fabian J Theis[^] and Kevin A Janes[^]: PNAS 2014, 111(5), E626-635 (* joint first authors, [^] joint last authors) <doi:10.1073/pnas.1311647111>

"Pheno-seq - linking visual features and gene expression in 3D cell culture systems" by Stephan M. Tirier, Jeongbin Park, Friedrich Preusser, Lisa Amrhein, Zuguang Gu, Simon Steiger, Jan-Philipp Mallm, Teresa Krieger, Marcel Waschow, Bjoern Eismann, Marta Gut, Ivo G. Gut, Karsten Rippe, Matthias Schlesner, Fabian Theis, Christiane Fuchs, Claudia R. Ball, Hanno Glimm, Roland Eils & Christian Conrad: Sci Rep 9, 12367 (2019) <doi:10.1038/s41598-019-48771-4>

comb.summands

Combinations of fixed number of summands with pre-defined sum.

Description

Returns all combinations of k numbers between 0 and n whose sum equals n .

Usage

```
comb.summands(n, k)
```

Arguments

n	the sum of the k summands
k	the number of summands

Details

Returns all combinations of k numbers (non-negative integers) between 0 and n whose sum equals n . The order of the summands matters, i.e. $2+3=5$ and $3+2=5$ would count as two different combinations.

Value

A matrix with k columns. Each row contains a different combination of k non-negative integers which sum up to n .

Author(s)

Christoph Kurz

References

"Parameterizing cell-to-cell regulatory heterogeneities via stochastic transcriptional profiles" by Sameer S Bajikar*, Christiane Fuchs*, Andreas Roller, Fabian J Theis[^] and Kevin A Janes[^]: PNAS 2014, 111(5), E626-635 (* joint first authors, [^] joint last authors) <doi:10.1073/pnas.1311647111>

"Pheno-seq - linking visual features and gene expression in 3D cell culture systems" by Stephan M. Tirier, Jeongbin Park, Friedrich Preusser, Lisa Amrhein, Zuguang Gu, Simon Steiger, Jan-Philipp Mallm, Teresa Krieger, Marcel Waschow, Bjoern Eismann, Marta Gut, Ivo G. Gut, Karsten Rippe, Matthias Schlesner, Fabian Theis, Christiane Fuchs, Claudia R. Ball, Hanno Glimm, Roland Eils & Christian Conrad: Sci Rep 9, 12367 (2019) <doi:10.1038/s41598-019-48771-4>

d.sum.of.mixtures.EXPLN

Sums of mixtures of zero, one or more lognormal random variables and one exponential random variable

Description

Density and random generation of a sum of i.i.d. random variables, where each random variable is from the following mixture distribution: With probability p_i , it is of type i . For all but the largest i , it is lognormally distributed with log-mean μ_i and log-standard deviation σ_i . Otherwise it is exponentially distributed with rate λ .

Usage

```
d.sum.of.mixtures.EXPLN(y, n, p.vector, mu.vector, sigma.vector, lambda,
  logdens = T)
r.sum.of.mixtures.EXPLN(k, n, p.vector, mu.vector, sigma.vector, lambda, N.matrix)
```

Arguments

<code>y</code>	the argument at which the density is evaluated
<code>k</code>	number of i.i.d. random variables returned by this function (in the considered application: number of tissue samples)
<code>n</code>	the number of random variables entering each sum (in the considered application: number of cells per tissue sample). This can also be a vector stating how many cells are in each sample separately.
<code>p.vector</code>	
<code>mu.vector</code>	vector ($\mu_1, \mu_2, \dots, \mu_{(T-1)}$) containing the log-means for each lognormal type (types 1 to T-1)
<code>sigma.vector</code>	vector ($\sigma_1, \dots, \sigma_{(T-1)}$) containing the log-standard deviations σ for each lognormal type (types 1 to T-1)
<code>lambda</code>	the rate for the exponential type (type T)
<code>logdens</code>	if TRUE, the log of the density is returned
<code>N.matrix</code>	optional. Matrix, that shows the decomposition of samples to be generated. Each row stands for a future sample and the columns show the decomposition of types for the specific sample such that the row sum should be <code>n</code> (either a value, i.e. all the same or a vector).

Details

The lengths of `mu.vector` and `sigma.vector` have to be identical. `p.vector` has to have one component more. Its length automatically determines the number of different types. `lambda` has to be a scalar.

Value

'`d.sum.of.mixtures.EXPLN`' gives the density, and '`r.sum.of.mixtures.EXPLN`' generates random variables.

Author(s)

Lisa Amrhein, Christiane Fuchs

Maintainer: Lisa Amrhein <amrheinlisa@gmail.com>

References

"Parameterizing cell-to-cell regulatory heterogeneities via stochastic transcriptional profiles" by Sameer S Bajikar*, Christiane Fuchs*, Andreas Roller, Fabian J Theis[^] and Kevin A Janes[^]: PNAS 2014, 111(5), E626-635 (* joint first authors, [^] joint last authors) <doi:10.1073/pnas.1311647111>

"Pheno-seq - linking visual features and gene expression in 3D cell culture systems" by Stephan M. Tirier, Jeongbin Park, Friedrich Preusser, Lisa Amrhein, Zuguang Gu, Simon Steiger, Jan-Philipp Mallm, Teresa Krieger, Marcel Waschow, Bjoern Eismann, Marta Gut, Ivo G. Gut, Karsten Rippe, Matthias Schlesner, Fabian Theis, Christiane Fuchs, Claudia R. Ball, Hanno Glimm, Roland Eils & Christian Conrad: Sci Rep 9, 12367 (2019) <doi:10.1038/s41598-019-48771-4>

Examples

```
# generate random variables
p <- c(0.25,0.75)
mu <- 2
sigma <- 0.3
lambda <- 5
set.model.functions("EXP-LN")

r <- r.sum.of.mixtures.EXPLN(10^4,10,p,mu,sigma,lambda)
hist(r,xlab="Sum of mixtures of lognormals",freq=FALSE,breaks=100,ylim=c(0,0.2))

# plot according theoretical density function
x <- seq(round(min(r)),round(max(r)),(round(max(r))-round(min(r)))/500)
y <- d.sum.of.mixtures.EXPLN(x,10,p,mu,sigma,lambda,logdens=FALSE)
lines(x,y,col="blue",lwd=3)
```

d.sum.of.mixtures.LNLN

Sums of mixtures of lognormal random variables

Description

Density and random generation of a sum of i.i.d. random variables, where each random variable is from the following mixture distribution: With probability p_i , it is of type i . In that case, it is lognormally distributed with log-mean μ_i and log-standard deviation σ_i .

Usage

```
d.sum.of.mixtures.LNLN(y, n, p.vector, mu.vector, sigma.vector, logdens = T)
r.sum.of.mixtures.LNLN(k, n, p.vector, mu.vector, sigma.vector, N.matrix)
```

Arguments

y	the argument at which the density is evaluated
k	number of i.i.d. random variables returned by this function (in the considered application: number of tissue samples)
n	the number of random variables entering each sum (in the considered application: number of cells per tissue sample). This can also be a vector stating how many cells are in each sample separately.
p.vector	vector (p_1, p_2, \dots, p_T) containing the probabilities for each type of cell. Its elements have to sum up to one
mu.vector	vector ($\mu_1, \mu_2, \dots, \mu_T$) containing the log-means for each type
sigma.vector	vector ($\sigma_1, \dots, \sigma_T$) containing the log-standard deviations σ for each type
logdens	if TRUE, the log of the density is returned

`N.matrix` optional. Matrix, that shows the decomposition of samples to be generated. Each row stands for a future sample and the columns show the decomposition of types for the specific sample such that the row sum should be `n` (either a value, i.e. all the same or a vector).

Details

The lengths of `p.vector`, `mu.vector` and `sigma.vector` have to be identical. Their lengths automatically determine the number of different types.

Value

'`d.sum.of.mixtures.LNLN`' gives the density, and '`r.sum.of.mixtures.LNLN`' generates random variables.

Author(s)

Lisa Amrhein, Christiane Fuchs

Maintainer: Lisa Amrhein <amrheinlisa@gmail.com>

References

"Parameterizing cell-to-cell regulatory heterogeneities via stochastic transcriptional profiles" by Sameer S Bajikar*, Christiane Fuchs*, Andreas Roller, Fabian J Theis[^] and Kevin A Janes[^]: PNAS 2014, 111(5), E626-635 (* joint first authors, [^] joint last authors) <doi:10.1073/pnas.1311647111>

"Pheno-seq - linking visual features and gene expression in 3D cell culture systems" by Stephan M. Tirier, Jeongbin Park, Friedrich Preusser, Lisa Amrhein, Zuguang Gu, Simon Steiger, Jan-Philipp Mallm, Teresa Krieger, Marcel Waschow, Bjoern Eismann, Marta Gut, Ivo G. Gut, Karsten Rippe, Matthias Schlesner, Fabian Theis, Christiane Fuchs, Claudia R. Ball, Hanno Glimm, Roland Eils & Christian Conrad: Sci Rep 9, 12367 (2019) <doi:10.1038/s41598-019-48771-4>

Examples

```
# generate random variables
p <- c(0.25,0.75)
mu <- c(2,-1)
sigma <- c(0.3,0.1)

set.model.functions("LN-LN")

r <- r.sum.of.mixtures.LNLN(10^4,10,p,mu,sigma)
hist(r,xlab="Sum of mixtures of lognormals",freq=FALSE,breaks=100,ylim=c(0,0.2))

# plot according theoretical density function
x <- seq(round(min(r)),round(max(r)),(round(max(r))-round(min(r)))/500)
y <- d.sum.of.mixtures.LNLN(x,10,p,mu,sigma,logdens=FALSE)
lines(x,y,col="blue",lwd=3)
```

d.sum.of.mixtures.rLNLN

Sums of mixtures of lognormal random variables

Description

Density and random generation of a sum of i.i.d. random variables, where each random variable is from the following mixture distribution: With probability p_i , it is of type i . In that case, it is lognormally distributed with log-mean μ_i and log-standard deviation σ_i .

Usage

```
d.sum.of.mixtures.rLNLN(y, n, p.vector, mu.vector, sigma.vector, logdens = T)
r.sum.of.mixtures.rLNLN(k, n, p.vector, mu.vector, sigma.vector, N.matrix)
```

Arguments

y	the argument at which the density is evaluated
k	number of i.i.d. random variables returned by this function (in the considered application: number of tissue samples)
n	the number of random variables entering each sum (in the considered application: number of cells per tissue sample). This can also be a vector stating how many cells are in each sample separately.
p.vector	vector (p_1, p_2, \dots, p_T) containing the probabilities for each type of cell. Its elements have to sum up to one
mu.vector	vector ($\mu_1, \mu_2, \dots, \mu_T$) containing the log-means for each type
sigma.vector	vector ($\sigma_1, \dots, \sigma_T$) containing the log-standard deviations σ for each type
logdens	if TRUE, the log of the density is returned
N.matrix	optional. Matrix, that shows the decomposition of samples to be generated. Each row stands for a future sample and the columns show the decomposition of types for the specific sample such that the row sum should be n (either a value, i.e. all the same or a vector).

Details

The lengths of p.vector, mu.vector and sigma.vector have to be identical. Their lengths automatically determine the number of different types.

Value

'd.sum.of.mixtures.rLNLN' gives the density, and 'r.sum.of.mixtures.rLNLN' generates random variables.

Author(s)

Lisa Amrhein, Christiane Fuchs

Maintainer: Lisa Amrhein <amrheinlisa@gmail.com>

References

"Parameterizing cell-to-cell regulatory heterogeneities via stochastic transcriptional profiles" by Sameer S Bajikar*, Christiane Fuchs*, Andreas Roller, Fabian J Theis[^] and Kevin A Janes[^]: PNAS 2014, 111(5), E626-635 (* joint first authors, [^] joint last authors) <doi:10.1073/pnas.1311647111>

"Pheno-seq - linking visual features and gene expression in 3D cell culture systems" by Stephan M. Tirier, Jeongbin Park, Friedrich Preusser, Lisa Amrhein, Zuguang Gu, Simon Steiger, Jan-Philipp Mallm, Teresa Krieger, Marcel Waschow, Bjoern Eismann, Marta Gut, Ivo G. Gut, Karsten Rippe, Matthias Schlesner, Fabian Theis, Christiane Fuchs, Claudia R. Ball, Hanno Glimm, Roland Eils & Christian Conrad: Sci Rep 9, 12367 (2019) <doi:10.1038/s41598-019-48771-4>

Examples

```
# generate random variables
p <- c(0.25,0.75)
mu <- c(2,-1)
sigma <- c(0.3,0.1)

set.model.functions("rLN-LN")

r <- r.sum.of.mixtures.rLNLN(10^4,10,p,mu,sigma)
hist(r,xlab="Sum of mixtures of lognormals",freq=FALSE,breaks=100,ylim=c(0,0.2))

# plot according theoretical density function
x <- seq(round(min(r)),round(max(r)),(round(max(r))-round(min(r)))/500)
y <- d.sum.of.mixtures.rLNLN(x,10,p,mu,sigma,logdens=FALSE)
lines(x,y,col="blue",lwd=3)
```

generate.toydata

Generation and analysis of synthetic data in stochastic profiling model

Description

Generation of a dataset of 500 i.i.d measurements as considered in the stochastic profiling model. Afterwards estimation of the model parameters and comparison of the estimates with the true value.

Usage

```
generate.toydata(model = "LN-LN")
```

Arguments

model the chosen stochastic profiling model: "LN-LN", "rLN-LN" or "EXP-LN"

Details

This function first generates a dataset of 500 i.i.d. 10-cell samplings as considered in the stochastic profiling models "LN-LN", "rLN-LN" and "EXP-LN". The employed parameters are $TY=2$ (i.e. two different types of cells are assumed) and $p=c(0.2,0.8)$ for all models. Furthermore, $\mu=c(1.5,-1.5)$ and $\sigma=0.2$ for the LN-LN model, $\mu=c(1.5,-1.5)$ and $\sigma=(0.2,0.6)$ for the rLN-LN model, and $\mu=1.5$, $\sigma=0.2$ and $\lambda=0.5$ for the EXP-LN model. The generated data is displayed in a histogram together with the theoretical probability density function. At the end of the estimation procedure, the profile log-likelihood plots are shown. Finally, the true and the estimated probability density functions are compared and the estimation results are printed.

Value

A list as returned by `stochprof.loop`, i.e. the following components:

<code>mle</code>	maximum likelihood estimate
<code>neg-loglikeli</code>	value of the negative log-likelihood function at maximum likelihood estimate
<code>ci</code>	approximate marginal maximum likelihood confidence intervals for the maximum likelihood estimate
<code>pargrid</code>	matrix containing parameter combinations and according values of the target function
<code>bic</code>	Bayesian information criterion value
<code>adj.bic</code>	adjusted Bayesian information criterion value which takes into account the numbers of parameters that were estimated during the preanalysis of a gene cluster. Is only calculated if <code>parameter subgroups</code> is given, otherwise set to <code>NULL</code> .
<code>pen</code>	penalization for densities not fulfilling required constraints. If <code>use.constraints</code> is <code>FALSE</code> , this has no practical meaning. If <code>use.constraints</code> is <code>TRUE</code> , this value is included in <code>loglikeli</code> .

Author(s)

Lisa Amrhein, Christiane Fuchs

Maintainer: Lisa Amrhein <amrheinlisa@gmail.com>

References

"Parameterizing cell-to-cell regulatory heterogeneities via stochastic transcriptional profiles" by Sameer S Bajikar*, Christiane Fuchs*, Andreas Roller, Fabian J Theis[^] and Kevin A Janes[^]: PNAS 2014, 111(5), E626-635 (* joint first authors, [^] joint last authors) <doi:10.1073/pnas.1311647111>

"Pheno-seq - linking visual features and gene expression in 3D cell culture systems" by Stephan M. Tirier, Jeongbin Park, Friedrich Preusser, Lisa Amrhein, Zuguang Gu, Simon Steiger, Jan-Philipp Mallm, Teresa Krieger, Marcel Waschow, Bjoern Eismann, Marta Gut, Ivo G. Gut, Karsten Rippe, Matthias Schlesner, Fabian Theis, Christiane Fuchs, Claudia R. Ball, Hanno Glimm, Roland Eils & Christian Conrad: Sci Rep 9, 12367 (2019) <doi:10.1038/s41598-019-48771-4>

mix.d.sum.of.mixtures.EXPLN

Density of the sum of mixtures of zero, one or more lognormal random variables and one exponential random variable weighted by all possible summands

Description

Density of a sum of i.i.d. random variables, where each random variable is from the following mixture distribution: With probability p_i , it is of type i . For all but the largest i , it is lognormally distributed with log-mean μ_i and log-standard deviation σ_i . Otherwise it is exponentially distributed with rate λ . The density is somehow a "mixed" one, as for all values in n the density of the random variable is calculated and the weighted average is taken to be the density of this specific value.

Usage

mix.d.sum.of.mixtures.EXPLN(y, n.vector, p.vector, mu.vector, sigma.vector, lambda)

Arguments

y	the argument at which the density is evaluated
n.vector	the number of random variables entering each sum (in the considered application: number of cells per tissue sample). This can also be a vector stating how many cells are in each sample separately
p.vector	vector (p_1, p_2, \dots, p_T) containing the probabilities for each type of cell. Its elements have to sum up to one
mu.vector	vector ($\mu_1, \mu_2, \dots, \mu_{(T-1)}$) containing the log-means for each lognormal type (types 1 to T-1)
sigma.vector	vector ($\sigma_1, \dots, \sigma_{(T-1)}$) containing the log-standard deviations σ for each lognormal type (types 1 to T-1)
lambda	the rate for the exponential type (type T)

Details

The lengths of mu.vector and sigma.vector have to be identical. p.vector has to have one component more. Its length automatically determines the number of different types. lambda has to be a scalar.

Value

'mix.d.sum.of.mixtures.EXPLN' gives the density of a random variable originating from one of the tissue samples in the mixed n-vector.

Author(s)

Lisa Amrhein, Christiane Fuchs

Maintainer: Lisa Amrhein <amrheinlisa@gmail.com>

References

"Parameterizing cell-to-cell regulatory heterogeneities via stochastic transcriptional profiles" by Sameer S Bajikar*, Christiane Fuchs*, Andreas Roller, Fabian J Theis[^] and Kevin A Janes[^]: PNAS 2014, 111(5), E626-635 (* joint first authors, [^] joint last authors) <doi:10.1073/pnas.1311647111>

"Pheno-seq - linking visual features and gene expression in 3D cell culture systems" by Stephan M. Tirier, Jeongbin Park, Friedrich Preusser, Lisa Amrhein, Zuguang Gu, Simon Steiger, Jan-Philipp Mallm, Teresa Krieger, Marcel Waschow, Bjoern Eismann, Marta Gut, Ivo G. Gut, Karsten Rippe, Matthias Schlesner, Fabian Theis, Christiane Fuchs, Claudia R. Ball, Hanno Glimm, Roland Eils & Christian Conrad: Sci Rep 9, 12367 (2019) <doi:10.1038/s41598-019-48771-4>

mix.d.sum.of.mixtures.LNLN

Density of the sum of mixtures of lognormal random variables weighted by all possible summands

Description

Density of a random variable that is the sum of i.i.d. random variables, where each random variable is from the following mixture distribution: With probability p_i , it is of type i . In that case, it is lognormally distributed with log-mean μ_i and log-standard deviation σ_i . The density is somehow a "mixed" one, as for all values in n the density of the random variable is calculated and the weighted average is taken to be the density of this specific value.

Usage

mix.d.sum.of.mixtures.LNLN(y, n.vector, p.vector, mu.vector, sigma.vector)

Arguments

y	the argument at which the density is evaluated
n.vector	the number of random variables entering each sum (in the considered application: number of cells per tissue sample). This can also be a vector stating how many cells are in each sample separately
p.vector	vector (p_1, p_2, \dots, p_T) containing the probabilities for each type of cell. Its elements have to sum up to one
mu.vector	vector ($\mu_1, \mu_2, \dots, \mu_T$) containing the log-means for each type
sigma.vector	vector ($\sigma_1, \dots, \sigma_T$) containing the log-standard deviations σ for each type

Details

The lengths of p.vector, mu.vector and sigma.vector have to be identical. Their lengths automatically determine the number of different types.

Value

'mix.d.sum.of.mixtures.LNLN' gives the density of a random variable originating from one of the tissue samples in the mixed n-vector.

Author(s)

Lisa Amrhein, Christiane Fuchs

Maintainer: Lisa Amrhein <amrheinlisa@gmail.com>

References

"Parameterizing cell-to-cell regulatory heterogeneities via stochastic transcriptional profiles" by Sameer S Bajikar*, Christiane Fuchs*, Andreas Roller, Fabian J Theis[^] and Kevin A Janes[^]: PNAS 2014, 111(5), E626-635 (* joint first authors, [^] joint last authors) <doi:10.1073/pnas.1311647111>

"Pheno-seq - linking visual features and gene expression in 3D cell culture systems" by Stephan M. Tirier, Jeongbin Park, Friedrich Preusser, Lisa Amrhein, Zuguang Gu, Simon Steiger, Jan-Philipp Mallm, Teresa Krieger, Marcel Waschow, Bjoern Eismann, Marta Gut, Ivo G. Gut, Karsten Rippe, Matthias Schlesner, Fabian Theis, Christiane Fuchs, Claudia R. Ball, Hanno Glimm, Roland Eils & Christian Conrad: Sci Rep 9, 12367 (2019) <doi:10.1038/s41598-019-48771-4>

mix.d.sum.of.mixtures.rLNLN

Density of the sum of mixtures of lognormal random variables weighted by all possible summands

Description

Density and random generation of a sum of i.i.d. random variables, where each random variable is from the following mixture distribution: With probability p_i , it is of type i . In that case, it is lognormally distributed with log-mean μ_i and log-standard deviation σ_i . The density is somehow a "mixed" one, as for all values in n the density of the random variable is calculated and the weighted average is taken to be the density of this specific value.

Usage

mix.d.sum.of.mixtures.rLNLN(y, n.vector, p.vector, mu.vector, sigma.vector)

Arguments

y	the argument at which the density is evaluated
n.vector	the number of random variables entering each sum (in the considered application: number of cells per tissue sample). This can also be a vector stating how many cells are in each sample separately
p.vector	vector (p1,p2,...pT) containing the probabilities for each type of cell. Its elements have to sum up to one

mu.vector vector ($\mu_1, \mu_2, \dots, \mu_T$) containing the log-means for each type
 sigma.vector vector ($\sigma_1, \dots, \sigma_T$) containing the log-standard deviations sigma for each type

Details

The lengths of p.vector, mu.vector and sigma.vector have to be identical. Their lengths automatically determine the number of different types.

Value

'mix.d.sum.of.mixtures.LNLN' gives the density of a random variable originating from one of the tissue samples in the mixed n-vector.

Author(s)

Lisa Amrhein, Christiane Fuchs
 Maintainer: Lisa Amrhein <amrheinlisa@gmail.com>

References

"Parameterizing cell-to-cell regulatory heterogeneities via stochastic transcriptional profiles" by Sameer S Bajikar*, Christiane Fuchs*, Andreas Roller, Fabian J Theis[^] and Kevin A Janes[^]: PNAS 2014, 111(5), E626-635 (* joint first authors, [^] joint last authors) <doi:10.1073/pnas.1311647111>
 "Pheno-seq - linking visual features and gene expression in 3D cell culture systems" by Stephan M. Tirier, Jeongbin Park, Friedrich Preusser, Lisa Amrhein, Zuguang Gu, Simon Steiger, Jan-Philipp Mallm, Teresa Krieger, Marcel Waschow, Bjoern Eismann, Marta Gut, Ivo G. Gut, Karsten Rippe, Matthias Schlesner, Fabian Theis, Christiane Fuchs, Claudia R. Ball, Hanno Glimm, Roland Eils & Christian Conrad: Sci Rep 9, 12367 (2019) <doi:10.1038/s41598-019-48771-4>

penalty.constraint.EXPLN

Penalization for population densities that do not fulfil certain constraints for the EXP-LN model

Description

In order to force the individual populations to be sufficiently distinct from each other, one can perform penalized optimization. To this end, constraints on the densities are introduced (see details). If the constraints are not fulfilled, a penalization term is added to the negative log-likelihood (which is to be minimized).

Usage

penalty.constraint.EXPLN(dataset, parameter, smoothingpar = 10⁵)

Arguments

dataset	matrix which contains the cumulated expression data over all cells in a tissue sample. Columns represent different genes, rows represent different tissue samples.
parameter	parameter for which the penalization term is calculated. This is a vector containing p , μ , σ and λ .
smoothingpar	weight with which the penalization term is multiplied.

Details

The constraints are as follows: There are TY densities for the TY distinct populations. For each $i=1, \dots, (TY-1)$, one considers the density of population i (the higher regulatory state) and the density of population $i+1$ (the lower regulatory state). The density of the higher regulatory state is constrained to be greater than the density of the lower regulatory state in the domain between the mode of the high state and the largest observation in the dataset.

Introduction of this penalization term does not mean that the constraints will automatically be fulfilled. The parameter estimate will be a trade-off between a maximizer of the unconstrained likelihood function and a minimizer of the penalization function. The higher the parameter `smoothingpar`, the more importance is on fulfilling the constraints.

Value

The population densities are compared on the above described domains. Wherever the constraint is not fulfilled, the difference between the larger and the lower density is calculated. The squares of all such differences are summed up and multiplied with `smoothingpar`. This value is returned.

Author(s)

Lisa Amrhein, Christiane Fuchs

Maintainer: Lisa Amrhein <amrheinlisa@gmail.com>

References

"Parameterizing cell-to-cell regulatory heterogeneities via stochastic transcriptional profiles" by Sameer S Bajikar*, Christiane Fuchs*, Andreas Roller, Fabian J Theis[^] and Kevin A Janes[^]: PNAS 2014, 111(5), E626-635 (* joint first authors, [^] joint last authors) <doi:10.1073/pnas.1311647111>

"Pheno-seq - linking visual features and gene expression in 3D cell culture systems" by Stephan M. Tirier, Jeongbin Park, Friedrich Preusser, Lisa Amrhein, Zuguang Gu, Simon Steiger, Jan-Philipp Mallm, Teresa Krieger, Marcel Waschow, Bjoern Eismann, Marta Gut, Ivo G. Gut, Karsten Rippe, Matthias Schlesner, Fabian Theis, Christiane Fuchs, Claudia R. Ball, Hanno Glimm, Roland Eils & Christian Conrad: Sci Rep 9, 12367 (2019) <doi:10.1038/s41598-019-48771-4>

`penalty.constraint.LNLN`*Penalization for population densities that do not fulfil certain constraints for the LN-LN model*

Description

In order to force the individual populations to be sufficiently distinct from each other, one can perform penalized optimization. To this end, constraints on the densities are introduced (see details). If the constraints are not fulfilled, a penalization term is added to the negative log-likelihood (which is to be minimized).

Usage

```
penalty.constraint.LNLN(dataset, parameter, smoothingpar = 10^5)
```

Arguments

<code>dataset</code>	matrix which contains the cumulated expression data over all cells in a tissue sample. Columns represent different genes, rows represent different tissue samples.
<code>parameter</code>	parameter for which the penalization term is calculated. This is a vector containing ρ , μ and σ .
<code>smoothingpar</code>	weight with which the penalization term is multiplied.

Details

The constraints are as follows: There are TY densities for the TY distinct populations. For each $i=1,\dots,(TY-1)$, one considers the density of population i (the higher regulatory state) and the density of population $i+1$ (the lower regulatory state). The density of the higher regulatory state is constrained to be greater than the density of the lower regulatory state in the domain between the mode of the high state and the largest observation in the dataset.

Introduction of this penalization term does not mean that the constraints will automatically be fulfilled. The parameter estimate will be a trade-off between a maximizer of the unconstrained likelihood function and a minimizer of the penalization function. The higher the parameter `smoothingpar`, the more importance is on fulfilling the constraints.

Value

The population densities are compared on the above described domains. Wherever the constraint is not fulfilled, the difference between the larger and the lower density is calculated. The squares of all such differences are summed up and multiplied with `smoothingpar`. This value is returned.

Author(s)

Lisa Amrhein, Christiane Fuchs

Maintainer: Lisa Amrhein <amrheinlisa@gmail.com>

References

- "Parameterizing cell-to-cell regulatory heterogeneities via stochastic transcriptional profiles" by Sameer S Bajikar*, Christiane Fuchs*, Andreas Roller, Fabian J Theis[^] and Kevin A Janes[^]: PNAS 2014, 111(5), E626-635 (* joint first authors, [^] joint last authors) <doi:10.1073/pnas.1311647111>
- "Pheno-seq - linking visual features and gene expression in 3D cell culture systems" by Stephan M. Tirier, Jeongbin Park, Friedrich Preusser, Lisa Amrhein, Zuguang Gu, Simon Steiger, Jan-Philipp Mallm, Teresa Krieger, Marcel Waschow, Bjoern Eismann, Marta Gut, Ivo G. Gut, Karsten Rippe, Matthias Schlesner, Fabian Theis, Christiane Fuchs, Claudia R. Ball, Hanno Glimm, Roland Eils & Christian Conrad: Sci Rep 9, 12367 (2019) <doi:10.1038/s41598-019-48771-4>

penalty.constraint.rLNLN

Penalization for population densities that do not fulfil certain constraints for the rLN-LN model

Description

In order to force the individual populations to be sufficiently distinct from each other, one can perform penalized optimization. To this end, constraints on the densities are introduced (see details). If the constraints are not fulfilled, a penalization term is added to the negative log-likelihood (which is to be minimized).

Usage

```
penalty.constraint.rLNLN(dataset, parameter, smoothingpar = 10^5)
```

Arguments

dataset	matrix which contains the cumulated expression data over all cells in a tissue sample. Columns represent different genes, rows represent different tissue samples.
parameter	parameter for which the penalization term is calculated. This is a vector containing p , μ and σ .
smoothingpar	weight with which the penalization term is multiplied.

Details

The constraints are as follows: There are TY densities for the TY distinct populations. For each $i=1, \dots, (TY-1)$, one considers the density of population i (the higher regulatory state) and the density of population $i+1$ (the lower regulatory state). The density of the higher regulatory state is constrained to be greater than the density of the lower regulatory state in the domain between the mode of the high state and the largest observation in the dataset.

Introduction of this penalization term does not mean that the constraints will automatically be fulfilled. The parameter estimate will be a trade-off between a maximizer of the unconstrained likelihood function and a minimizer of the penalization function. The higher the parameter `smoothingpar`, the more importance is on fulfilling the constraints.

Value

The population densities are compared on the above described domains. Wherever the constraint is not fulfilled, the difference between the larger and the lower density is calculated. The squares of all such differences are summed up and multiplied with `smoothingpar`. This value is returned.

Author(s)

Lisa Amrhein, Christiane Fuchs
 Maintainer: Lisa Amrhein <amrheinlisa@gmail.com>

References

"Parameterizing cell-to-cell regulatory heterogeneities via stochastic transcriptional profiles" by Sameer S Bajikar*, Christiane Fuchs*, Andreas Roller, Fabian J Theis[^] and Kevin A Janes[^]: PNAS 2014, 111(5), E626-635 (* joint first authors, [^] joint last authors) <doi:10.1073/pnas.1311647111>
 "Pheno-seq - linking visual features and gene expression in 3D cell culture systems" by Stephan M. Tirier, Jeongbin Park, Friedrich Preusser, Lisa Amrhein, Zuguang Gu, Simon Steiger, Jan-Philipp Mallm, Teresa Krieger, Marcel Waschow, Bjoern Eismann, Marta Gut, Ivo G. Gut, Karsten Rippe, Matthias Schlesner, Fabian Theis, Christiane Fuchs, Claudia R. Ball, Hanno Glimm, Roland Eils & Christian Conrad: Sci Rep 9, 12367 (2019) <doi:10.1038/s41598-019-48771-4>

set.model.functions *Defines some global model-dependent functions*

Description

Defines which model to use in the following and sets depending on this decision some global model-dependent functions. This function is only needed when using lower level functions of the package such as `d.sum.of.mixtures` which does not have its own model selection input.

Usage

```
set.model.functions(model)
```

Arguments

model	model for which one wishes to estimate the parameters: "LN-LN", "rLN-LN" or "EXP-LN"
-------	--

Value

Loads all model-specific functions and directs the general function to the model specific function, which is the same but with the endings `.LNLN`, `.rLNLN`, and `.EXPLN`.

Author(s)

Lisa Amrhein, Christiane Fuchs
 Maintainer: Lisa Amrhein <amrheinlisa@gmail.com>

sod2

Measurements from the detoxifying enzyme, SOD2

Description

Real 10-cell samplings from the detoxifying enzyme, SOD2. The dataset contains the measurements of SOD2 expression by qPCR in 81 random samplings of 10 ECM-attached cells.

Usage

```
data(sod2)
```

Format

The format is: num [1:81] 0.603 0.873 0.204 1 3.001 ...

Source

"Parameterizing cell-to-cell regulatory heterogeneities via stochastic transcriptional profiles" by Sameer S Bajikar*, Christiane Fuchs*, Andreas Roller, Fabian J Theis[^] and Kevin A Janes[^]: PNAS 2014, 111(5), E626-635 (* joint first authors, [^] joint last authors)

References

"Parameterizing cell-to-cell regulatory heterogeneities via stochastic transcriptional profiles" by Sameer S Bajikar*, Christiane Fuchs*, Andreas Roller, Fabian J Theis[^] and Kevin A Janes[^]: PNAS 2014, 111(5), E626-635 (* joint first authors, [^] joint last authors)

Examples

```
data(sod2)
hist(sod2,breaks=seq(0,7,0.5),col="grey")
```

stochasticProfilingData*User prompt for generation and visualization of synthetic data in stochastic profiling model*

Description

Generation of a dataset of i.i.d measurements as considered in the stochastic profiling model. The user is prompted to input all required settings. He or she hence does not have to delve into the structure of this package.

Usage

```
stochasticProfilingData()
```

Details

This function generates a dataset of i.i.d synthetic measurements as considered in the stochastic profiling model. The user is prompted to input all required settings, e.g. the exact model (LN-LN, rLN-LN or EXP-LN), the number of samples, the number of populations etc. The dataset is returned as a matrix and visualized in histograms together with the data generating probability density function (pdf).

Value

A matrix containing the synthetic data. Columns stand for different genes, rows for samples.

Author(s)

Lisa Amrhein, Christiane Fuchs

Maintainer: Lisa Amrhein <amrheinlisa@gmail.com>

References

"Parameterizing cell-to-cell regulatory heterogeneities via stochastic transcriptional profiles" by Sameer S Bajikar*, Christiane Fuchs*, Andreas Roller, Fabian J Theis[^] and Kevin A Janes[^]: PNAS 2014, 111(5), E626-635 (* joint first authors, [^] joint last authors) <doi:10.1073/pnas.1311647111>

"Pheno-seq - linking visual features and gene expression in 3D cell culture systems" by Stephan M. Tirier, Jeongbin Park, Friedrich Preusser, Lisa Amrhein, Zuguang Gu, Simon Steiger, Jan-Philipp Mallm, Teresa Krieger, Marcel Waschow, Bjoern Eismann, Marta Gut, Ivo G. Gut, Karsten Rippe, Matthias Schlesner, Fabian Theis, Christiane Fuchs, Claudia R. Ball, Hanno Glimm, Roland Eils & Christian Conrad: Sci Rep 9, 12367 (2019) <doi:10.1038/s41598-019-48771-4>

stochasticProfilingML *User prompt for maximum likelihood estimation of stochastic profiling model*

Description

Maximum likelihood estimation for the parameters in the stochastic profiling model. The user is prompted to input the data and all required settings. He or she hence does not have to delve into the structure of this package.

Usage

```
stochasticProfilingML()
```

Details

The function performs maximum likelihood estimation for the parameters in the stochastic profiling model. The user is prompted to input the data and all required settings, i.e. especially the data, but also the exact model (LN-LN, rLN-LN or EXP-LN), the number of populations, the number of cells per sample etc. The data can either be entered manually, or it can be read from a file, or the user enters the name of a variable which contains the data.

Value

A list as returned by `stochprof.loop`, i.e. the following components:

<code>mle</code>	maximum likelihood estimate
<code>neg-loglikeli</code>	value of the negative log-likelihood function at maximum likelihood estimate
<code>ci</code>	approximate marginal maximum likelihood confidence intervals for the maximum likelihood estimate
<code>pargrid</code>	matrix containing parameter combinations and according values of the target function
<code>bic</code>	Bayesian information criterion value
<code>adj.bic</code>	adjusted Bayesian information criterion value which takes into account the numbers of parameters that were estimated during the preanalysis of a gene cluster. Is only calculated if <code>parameter subgroups</code> is given, otherwise set to <code>NULL</code> .
<code>pen</code>	penalization for densities not fulfilling required constraints. If <code>use.constraints</code> is <code>FALSE</code> , this has no practical meaning. If <code>use.constraints</code> is <code>TRUE</code> , this value is included in <code>loglikeli</code> .

Author(s)

Lisa Amrhein, Christiane Fuchs

Maintainer: Lisa Amrhein <amrheinlisa@gmail.com>

References

"Parameterizing cell-to-cell regulatory heterogeneities via stochastic transcriptional profiles" by Sameer S Bajikar*, Christiane Fuchs*, Andreas Roller, Fabian J Theis[^] and Kevin A Janes[^]: PNAS 2014, 111(5), E626-635 (* joint first authors, [^] joint last authors) <doi:10.1073/pnas.1311647111>

"Pheno-seq - linking visual features and gene expression in 3D cell culture systems" by Stephan M. Tirier, Jeongbin Park, Friedrich Preusser, Lisa Amrhein, Zuguang Gu, Simon Steiger, Jan-Philipp Mallm, Teresa Krieger, Marcel Waschow, Bjoern Eismann, Marta Gut, Ivo G. Gut, Karsten Rippe, Matthias Schlesner, Fabian Theis, Christiane Fuchs, Claudia R. Ball, Hanno Glimm, Roland Eils & Christian Conrad: Sci Rep 9, 12367 (2019) <doi:10.1038/s41598-019-48771-4>

<code>stochprof.loop</code>	<i>Maximum likelihood estimation for the parameters in the stochastic profiling model</i>
-----------------------------	---

Description

Maximum likelihood estimation for the parameters in the stochastic profiling model. Because the log-likelihood function is potentially multimodal, no straightforward use of gradient-based approaches for finding globally optimal parameter combinations is possible. To tackle this challenge, this function performs a two-step estimation procedure.

Usage

```
stochprof.loop(model, dataset, n, TY, genenames = NULL, fix.mu = F,
  fixed.mu, par.range = NULL, prev.result = NULL, loops = 5,
  until.convergence = T, print.output = T, show.plots = T,
  plot.title = "", pdf.file, use.constraints = F, subgroups)
```

Arguments

model	model for which one wishes to estimate the parameters: "LN-LN", "rLN-LN" or "EXP-LN"
dataset	matrix which contains the cumulated expression data over all cells in a tissue sample. Columns represent different genes, rows represent different tissue samples.
n	number of cells taken from each tissue sample. This can also be a vector stating how many cells are in each sample separately.
TY	number of types of cells that is assumed in the stochastic model
genenames	names of the genes in the dataset. For genenames==NULL, the genes will simply be enumerated according to the column numbers in the dataset.
fix.mu	if TRUE, the log-means of the lognormal distributions are kept fixed in the estimation procedure. Otherwise, they are to be estimated.
fixed.mu	vector containing the values to which the log-means should be fixed if fix.mu==T. The order of components is as follows: (mu_type_1_gene_1, mu_type_1_gene_2, ..., mu_type_2_gene_1, mu_type_2_gene_2, ...). This argument needs to be specified only when fix.mu==T.
par.range	range from which the parameter values should be randomly drawn if there is no knowledge from previous iterations of the search algorithm available. This is a matrix with the number of rows being equal to the number of model parameters. The first column contains the lower bound, the second column the upper bound. If par.range==NULL, some rather large range is defined.
prev.result	can contain results from former calls of this function
loops	maximal number of loops carried out in the estimation procedure. Each loops involves various methods to determine the high-likelihood region.
until.convergence	if TRUE, the estimation process is terminated if there had been no improvement concerning the value of the target function between two consecutive loops. Otherwise, the estimation procedure is terminated according to the parameter "loops".
print.output	if TRUE, interim results of the grid search and numerical optimization are printed into the console throughout the estimation procedure
show.plots	if TRUE, profile log-likelihood plots are displayed at the end of the estimation procedure
plot.title	title of each plot if show.plots==T

pdf.file	optional filename. If this is not missing and show.plots==T, the profile log-likelihoods will be plotted into this file.
use.constraints	if TRUE, constraints on the individual population densities are applied; see penalty.constraint.LNLN, penalty.constraint.rLNLN and penalty.constraint.EXPLN for details.
subgroups	list of sets of gene numbers. This parameter should be given only when the present call of stochprof.loop is based on a subanalysis of the subgroups of genes with non-fixed mu. The parameter is used only for calculation of the adjusted BIC which takes into account the number of parameters that had to be estimated during the whole estimation procedure: First, for each of the subclusters, and then for the final analysis.

Details

This function carries out maximum likelihood estimation for the parameters of the stochastic profiling model. Because the log-likelihood function is potentially multimodal, no straightforward use of gradient-based approaches for finding globally optimal parameter combinations is possible. To tackle this challenge, this function performs a two-step estimation procedure: First, it computes the log-likelihood function at randomly drawn parameter combinations to identify high-likelihood regions in parameter space at computationally low cost. Then, it uses the Nelder-Mead algorithm to identify local maxima of the likelihood function. The starting values for this algorithm are randomly drawn from the high-likelihood regions determined in the first step. To further localize the global optimum, the function again performs grid searches of the parameter space, this time around the optimum identified by the Nelder-Mead algorithm. This search creates another space to identify high-likelihood regions, which are then used to seed another Nelder-Mead optimization.

Value

A list with the following components:

mle	maximum likelihood estimate
neg-loglikeli	value of the negative log-likelihood function at maximum likelihood estimate
ci	approximate marginal maximum likelihood confidence intervals for the maximum likelihood estimate
pargrid	matrix containing parameter combinations and according values of the target function
bic	Bayesian information criterion value
adj.bic	adjusted Bayesian information criterion value which takes into account the numbers of parameters that were estimated during the preanalysis of a gene cluster. Is only calculated if parameter subgroups is given, otherwise set to NULL.
pen	penalization for densities not fulfilling required constraints. If use.constraints is FALSE, this has no practical meaning. If use.constraints is TRUE, this value is included in loglikeli.

Author(s)

Lisa Amrhein, Christiane Fuchs

Maintainer: Lisa Amrhein <amrheinlisa@gmail.com>

References

"Parameterizing cell-to-cell regulatory heterogeneities via stochastic transcriptional profiles" by Sameer S Bajikar*, Christiane Fuchs*, Andreas Roller, Fabian J Theis[^] and Kevin A Janes[^]: PNAS 2014, 111(5), E626-635 (* joint first authors, [^] joint last authors) <doi:10.1073/pnas.1311647111>

"Pheno-seq - linking visual features and gene expression in 3D cell culture systems" by Stephan M. Tirier, Jeongbin Park, Friedrich Preusser, Lisa Amrhein, Zuguang Gu, Simon Steiger, Jan-Philipp Mallm, Teresa Krieger, Marcel Waschow, Bjoern Eismann, Marta Gut, Ivo G. Gut, Karsten Rippe, Matthias Schlesner, Fabian Theis, Christiane Fuchs, Claudia R. Ball, Hanno Glimm, Roland Eils & Christian Conrad: Sci Rep 9, 12367 (2019) <doi:10.1038/s41598-019-48771-4>

stochprof.results.EXPLN

Evaluation of results from estimation of EXP-LN model

Description

Evaluates the set of results that are passed to this function. That means, it removes entries where the target function is equal to infinity, it removes double entries, it removes unlikely parameter combinations (if there are too many) etc., and it sorts the data. When `show.plots==T`, the results are graphically displayed.

Usage

```
stochprof.results.EXPLN(prev.result, TY, show.plots = T, plot.title = "",
  pdf.file, fix.mu = F)
```

Arguments

<code>prev.result</code>	contains parameter combinations and the respective value of the target function. It is typically the output of 'stochprof.search.EXPLN'.
<code>TY</code>	number of types of cells assumed in the model
<code>show.plots</code>	if TRUE, the results are plotted. In particular, one plot is produced for each parameter, with the value of the parameter plotted against the value of the target function. This is not exactly the profile log-likelihood function because there is no conditioning on the other parameters being equal to the ML estimate. If the estimation procedure has converged, however, one can recognize the shape of the profile log-likelihood from these plots. A red bar indicates the position of the maximum likelihood estimator.
<code>plot.title</code>	title of each plot if <code>show.plots==T</code>

<code>pdf.file</code>	plots will be written into this file when this argument is not missing. The file has to include the entire path.
<code>fix.mu</code>	if TRUE, the log-mean of the lognormal distributions has been kept fixed. In that case, no plots will be produced for these parameters.

Value

Matrix with sorted and evaluated results. The columns are exactly the same as those in 'prev.result'. The first row contains the best estimate.

Author(s)

Lisa Amrhein, Christiane Fuchs

Maintainer: Lisa Amrhein <amrheinlisa@gmail.com>

References

"Parameterizing cell-to-cell regulatory heterogeneities via stochastic transcriptional profiles" by Sameer S Bajikar*, Christiane Fuchs*, Andreas Roller, Fabian J Theis[^] and Kevin A Janes[^]: PNAS 2014, 111(5), E626-635 (* joint first authors, [^] joint last authors) <doi:10.1073/pnas.1311647111>

"Pheno-seq - linking visual features and gene expression in 3D cell culture systems" by Stephan M. Tirier, Jeongbin Park, Friedrich Preusser, Lisa Amrhein, Zuguang Gu, Simon Steiger, Jan-Philipp Mallm, Teresa Krieger, Marcel Waschow, Bjoern Eismann, Marta Gut, Ivo G. Gut, Karsten Rippe, Matthias Schlesner, Fabian Theis, Christiane Fuchs, Claudia R. Ball, Hanno Glimm, Roland Eils & Christian Conrad: Sci Rep 9, 12367 (2019) <doi:10.1038/s41598-019-48771-4>

stochprof.results.LNLN

Evaluation of results from estimation of LN-LN model

Description

Evaluates the set of results that are passed to this function. That means, it removes entries where the target function is equal to infinity, it removes double entries, it removes unlikely parameter combinations (if there are too many) etc., and it sorts the data. When `show.plots==T`, the results are graphically displayed.

Usage

```
stochprof.results.LNLN(prev.result, TY, show.plots = T, plot.title = "",
  pdf.file, fix.mu = F)
```

Arguments

<code>prev.result</code>	contains parameter combinations and the respective value of the target function. It is typically the output of <code>'stochprof.search.LNLN'</code> .
<code>TY</code>	number of types of cells assumed in the model
<code>show.plots</code>	if TRUE, the results are plotted. In particular, one plot is produced for each parameter, with the value of the parameter plotted against the value of the target function. This is not exactly the profile log-likelihood function because there is no conditioning on the other parameters being equal to the ML estimate. If the estimation procedure has converged, however, one can recognize the shape of the profile log-likelihood from these plots. A red bar indicates the position of the maximum likelihood estimator.
<code>plot.title</code>	title of each plot if <code>show.plots==T</code>
<code>pdf.file</code>	plots will be written into this file when this argument is not missing. The file has to include the entire path.
<code>fix.mu</code>	if TRUE, the log-mean of the lognormal distributions has been kept fixed. In that case, no plots will be produced for these parameters.

Value

Matrix with sorted and evaluated results. The columns are exactly the same as those in `'prev.result'`. The first row contains the best estimate.

Author(s)

Lisa Amrhein, Christiane Fuchs

Maintainer: Lisa Amrhein <amrheinlisa@gmail.com>

References

"Parameterizing cell-to-cell regulatory heterogeneities via stochastic transcriptional profiles" by Sameer S Bajikar*, Christiane Fuchs*, Andreas Roller, Fabian J Theis[^] and Kevin A Janes[^]: PNAS 2014, 111(5), E626-635 (* joint first authors, [^] joint last authors) <doi:10.1073/pnas.1311647111>

"Pheno-seq - linking visual features and gene expression in 3D cell culture systems" by Stephan M. Tirier, Jeongbin Park, Friedrich Preusser, Lisa Amrhein, Zuguang Gu, Simon Steiger, Jan-Philipp Mallm, Teresa Krieger, Marcel Waschow, Bjoern Eismann, Marta Gut, Ivo G. Gut, Karsten Rippe, Matthias Schlesner, Fabian Theis, Christiane Fuchs, Claudia R. Ball, Hanno Glimm, Roland Eils & Christian Conrad: Sci Rep 9, 12367 (2019) <doi:10.1038/s41598-019-48771-4>

 stochprof.results.rLNLN

Evaluation of results from estimation of rLN-LN model

Description

Evaluates the set of results that are passed to this function. That means, it removes entries where the target function is equal to infinity, it removes double entries, it removes unlikely parameter combinations (if there are too many) etc., and it sorts the data. When `show.plots==T`, the results are graphically displayed.

Usage

```
stochprof.results.rLNLN(prev.result, TY, show.plots = T, plot.title = "",
  pdf.file, fix.mu = F)
```

Arguments

<code>prev.result</code>	contains parameter combinations and the respective value of the target function. It is typically the output of 'stochprof.search.rLNLN'.
<code>TY</code>	number of types of cells assumed in the model
<code>show.plots</code>	if TRUE, the results are plotted. In particular, one plot is produced for each parameter, with the value of the parameter plotted against the value of the target function. This is not exactly the profile log-likelihood function because there is no conditioning on the other parameters being equal to the ML estimate. If the estimation procedure has converged, however, one can recognize the shape of the profile log-likelihood from these plots. A red bar indicates the position of the maximum likelihood estimator.
<code>plot.title</code>	title of each plot if <code>show.plots==T</code>
<code>pdf.file</code>	plots will be written into this file when this argument is not missing. The file has to include the entire path.
<code>fix.mu</code>	if TRUE, the log-mean of the lognormal distributions has been kept fixed. In that case, no plots will be produced for these parameters.

Value

Matrix with sorted and evaluated results. The columns are exactly the same as those in 'prev.result'. The first row contains the best estimate.

Author(s)

Lisa Amrhein, Christiane Fuchs

Maintainer: Lisa Amrhein <amrheinlisa@gmail.com>

References

"Parameterizing cell-to-cell regulatory heterogeneities via stochastic transcriptional profiles" by Sameer S Bajikar*, Christiane Fuchs*, Andreas Roller, Fabian J Theis[^] and Kevin A Janes[^]: PNAS 2014, 111(5), E626-635 (* joint first authors, [^] joint last authors) <doi:10.1073/pnas.1311647111>

"Pheno-seq - linking visual features and gene expression in 3D cell culture systems" by Stephan M. Tirier, Jeongbin Park, Friedrich Preusser, Lisa Amrhein, Zuguang Gu, Simon Steiger, Jan-Philipp Mallm, Teresa Krieger, Marcel Waschow, Bjoern Eismann, Marta Gut, Ivo G. Gut, Karsten Rippe, Matthias Schlesner, Fabian Theis, Christiane Fuchs, Claudia R. Ball, Hanno Glimm, Roland Eils & Christian Conrad: Sci Rep 9, 12367 (2019) <doi:10.1038/s41598-019-48771-4>

stochprof.search.EXPLN

Calculation of the log likelihood function of the EXP-LN model

Description

Calculates the log likelihood function of the parameters of the EXP-LN model for a given dataset at certain parameter values.

Usage

```
stochprof.search.EXPLN(dataset, n, TY, method = "grid", M = 10,
  par.range = NULL, prev.result = NULL, fix.mu = F, fixed.mu,
  genenames = NULL, print.output = F, use.constraints = F)
```

Arguments

dataset	matrix which contains the cumulated expression data over all cells in a tissue sample. Columns represent genes, rows represent tissue samples.
n	number of cells taken from each tissue sample. This can also be a vector stating how many cells are in each sample separately
TY	number of types of cells that is assumed in the stochastic model
method	determines whether a grid search or the Nelder-Mead algorithm should be applied: If method=="grid", the log likelihood function is simply evaluated at certain parameter values that are randomly drawn. If method=="optim", a Nelder-Mead search starts at a randomly drawn set of parameter values in order to find a local maximum. The resulting locally optimal parameter is stored in the results matrix as one row.
M	number of randomly drawn parameter combinations
par.range	range from which the parameter values should be randomly drawn. This is a matrix with the number of rows being equal to the number of model parameters. The first columns contains the lower bound, the second column the upper bound. If par.range==NULL, some rather large range is defined.
prev.result	can contain results from former calls of this function

<code>fix.mu</code>	if TRUE, the log-means are kept fixed in the estimation procedure. Otherwise, they are to be estimated.
<code>fixed.mu</code>	vector containing the values to which the log-means should be fixed if <code>fix.mu==T</code> . The order of components is as follows: (<code>mu_type_1_gene_1</code> , <code>mu_type_1_gene_2</code> , ..., <code>mu_type_2_gene_1</code> , <code>mu_type_2_gene_2</code> , ...). This argument needs to be specified only when <code>fix.mu==T</code> .
<code>genenames</code>	names of the genes in the dataset. For <code>genenames==NULL</code> , the genes will simply be enumerated according to the column numbers in the dataset.
<code>print.output</code>	if TRUE, interim results of the grid search and numerical optimization are printed into the console throughout the estimation procedure
<code>use.constraints</code>	if TRUE, constraints on the individual population densities are applied; see <code>penalty.constraint.EXPLN</code> for details.

Details

The values at which the target function is calculated are randomly drawn from some range specified by `"par.range"`. If `method=="grid"`, the target function is simply evaluated at such a randomly drawn parameter vector. If `method=="optim"`, this randomly drawn vector is passed to the Nelder-Mead algorithm as a starting value in order to search for a local maximum around it.

Value

A matrix with the following entries: Each row corresponds to one parameter combination. All columns but the last one contain the parameter values at which the log likelihood function has been computed. The column names are the parameter names. The last column ("`target`") is the negative log likelihood function computed at the respective parameter vector. For numerical reasons, this target value is set to the minimum of 10^7 and the actual value.

Author(s)

Lisa Amrhein, Christiane Fuchs

Maintainer: Lisa Amrhein <amrheinlisa@gmail.com>

References

"Parameterizing cell-to-cell regulatory heterogeneities via stochastic transcriptional profiles" by Sameer S Bajikar*, Christiane Fuchs*, Andreas Roller, Fabian J Theis[^] and Kevin A Janes[^]: PNAS 2014, 111(5), E626-635 (* joint first authors, [^] joint last authors) <doi:10.1073/pnas.1311647111>

"Pheno-seq - linking visual features and gene expression in 3D cell culture systems" by Stephan M. Tirier, Jeongbin Park, Friedrich Preusser, Lisa Amrhein, Zuguang Gu, Simon Steiger, Jan-Philipp Mallm, Teresa Krieger, Marcel Waschow, Bjoern Eismann, Marta Gut, Ivo G. Gut, Karsten Rippe, Matthias Schlesner, Fabian Theis, Christiane Fuchs, Claudia R. Ball, Hanno Glimm, Roland Eils & Christian Conrad: Sci Rep 9, 12367 (2019) <doi:10.1038/s41598-019-48771-4>

stochprof.search.LNLN *Calculation of the log likelihood function of the LN-LN model*

Description

Calculates the log likelihood function of the parameters of the LN-LN model for a given dataset at certain parameter values.

Usage

```
stochprof.search.LNLN(dataset, n, TY, method = "grid", M = 10,
  par.range = NULL, prev.result = NULL, fix.mu = F, fixed.mu,
  genenames = NULL, print.output = F, use.constraints = F)
```

Arguments

dataset	matrix which contains the cumulated expression data over all cells in a tissue sample. Columns represent genes, rows represent tissue samples.
n	number of cells taken from each tissue sample. This can also be a vector stating how many cells are in each sample separatly
TY	number of types of cells that is assumed in the stochastic model
method	determines whether a grid search or the Nelder-Mead algorithm should be applied: If method=="grid", the log likelihood function is simply evaluated at certain parameter values that are randomly drawn. If method=="optim", a Nelder-Mead search starts at a randomly drawn set of parameter values in order to find a local maximum. The resulting locally optimal parameter is stored in the results matrix as one row.
M	number of randomly drawn parameter combinations
par.range	range from which the parameter values should be randomly drawn. This is a matrix with the number of rows being equal to the number of model parameters. The first columns contains the lower bound, the second column the upper bound. If par.range==NULL, some rather large range is defined.
prev.result	can contain results from former calls of this function
fix.mu	if TRUE, the log-means are kept fixed in the estimation procedure. Otherwise, they are to be estimated.
fixed.mu	vector containing the values to which the log-means should be fixed if fix.mu==T. The order of components is as follows: (mu_type_1_gene_1, mu_type_1_gene_2, ..., mu_type_2_gene_1, mu_type_2_gene_2, ...). This argument needs to be specified only when fix.mu==T.
genenames	names of the genes in the dataset. For genenames==NULL, the genes will simply be enumerated according to the column numbers in the dataset.
print.output	if TRUE, interim results of the grid search and numerical optimization are printed into the console throughout the estimation procedure

`use.constraints`

if TRUE, constraints on the individual population densities are applied; see `penalty.constraint.LNLN` for details.

Details

The values at which the target function is calculated are randomly drawn from some range specified by `"par.range"`. If `method=="grid"`, the target function is simply evaluated at such a randomly drawn parameter vector. If `method=="optim"`, this randomly drawn vector is passed to the Nelder-Mead algorithm as a starting value in order to search for a local maximum around it.

Value

A matrix with the following entries: Each row corresponds to one parameter combination. All columns but the last one contain the parameter values at which the log likelihood function has been computed. The column names are the parameter names. The last column ("`target`") is the negative log likelihood function computed at the respective parameter vector. For numerical reasons, this target value is set to the minimum of 10^7 and the actual value.

Author(s)

Lisa Amrhein, Christiane Fuchs

Maintainer: Lisa Amrhein <amrheinlisa@gmail.com>

References

"Parameterizing cell-to-cell regulatory heterogeneities via stochastic transcriptional profiles" by Sameer S Bajikar*, Christiane Fuchs*, Andreas Roller, Fabian J Theis[^] and Kevin A Janes[^]: PNAS 2014, 111(5), E626-635 (* joint first authors, [^] joint last authors) <doi:10.1073/pnas.1311647111>

"Pheno-seq - linking visual features and gene expression in 3D cell culture systems" by Stephan M. Tirier, Jeongbin Park, Friedrich Preusser, Lisa Amrhein, Zuguang Gu, Simon Steiger, Jan-Philipp Mallm, Teresa Krieger, Marcel Waschow, Bjoern Eismann, Marta Gut, Ivo G. Gut, Karsten Rippe, Matthias Schlesner, Fabian Theis, Christiane Fuchs, Claudia R. Ball, Hanno Glimm, Roland Eils & Christian Conrad: Sci Rep 9, 12367 (2019) <doi:10.1038/s41598-019-48771-4>

stochprof.search.rLNLN

Calculation of the log likelihood function of the rLN-LN model

Description

Calculates the log likelihood function of the parameters of the rLN-LN model for a given dataset at certain parameter values.

Usage

```
stochprof.search.rLNLN(dataset, n, TY, method = "grid", M = 10,
  par.range = NULL, prev.result = NULL, fix.mu = F, fixed.mu,
  genenames = NULL, print.output = F, use.constraints = F)
```


Arguments

<code>dataset</code>	matrix which contains the cumulated expression data over all cells in a tissue sample. Columns represent genes, rows represent tissue samples.
<code>n</code>	number of cells taken from each tissue sample. This can also be a vector stating how many cells are in each sample separately
<code>TY</code>	number of types of cells that is assumed in the stochastic model
<code>method</code>	determines whether a grid search or the Nelder-Mead algorithm should be applied: If <code>method=="grid"</code> , the log likelihood function is simply evaluated at certain parameter values that are randomly drawn. If <code>method=="optim"</code> , a Nelder-Mead search starts at a randomly drawn set of parameter values in order to find a local maximum. The resulting locally optimal parameter is stored in the results matrix as one row.
<code>M</code>	number of randomly drawn parameter combinations
<code>par.range</code>	range from which the parameter values should be randomly drawn. This is a matrix with the number of rows being equal to the number of model parameters. The first columns contains the lower bound, the second column the upper bound. If <code>par.range==NULL</code> , some rather large range is defined.
<code>prev.result</code>	can contain results from former calls of this function
<code>fix.mu</code>	if TRUE, the log-means are kept fixed in the estimation procedure. Otherwise, they are to be estimated.
<code>fixed.mu</code>	vector containing the values to which the log-means should be fixed if <code>fix.mu==T</code> . The order of components is as follows: (<code>mu_type_1_gene_1</code> , <code>mu_type_1_gene_2</code> , ..., <code>mu_type_2_gene_1</code> , <code>mu_type_2_gene_2</code> , ...). This argument needs to be specified only when <code>fix.mu==T</code> .
<code>genenames</code>	names of the genes in the dataset. For <code>genenames==NULL</code> , the genes will simply be enumerated according to the column numbers in the dataset.
<code>print.output</code>	if TRUE, interim results of the grid search and numerical optimization are printed into the console throughout the estimation procedure
<code>use.constraints</code>	if TRUE, constraints on the individual population densities are applied; see <code>penalty.constraint.rLNLN</code> for details.

Details

The values at which the target function is calculated are randomly drawn from some range specified by `"par.range"`. If `method=="grid"`, the target function is simply evaluated at such a randomly drawn parameter vector. If `method=="optim"`, this randomly drawn vector is passed to the Nelder-Mead algorithm as a starting value in order to search for a local maximum around it.

Value

A matrix with the following entries: Each row corresponds to one parameter combination. All columns but the last one contain the parameter values at which the log likelihood function has been computed. The column names are the parameter names. The last column ("target") is the negative

log likelihood function computed at the respective parameter vector. For numerical reasons, this target value is set to the minimum of 10^7 and the actual value.

Author(s)

Lisa Amrhein, Christiane Fuchs
 Maintainer: Lisa Amrhein <amrheinlisa@gmail.com>

References

"Parameterizing cell-to-cell regulatory heterogeneities via stochastic transcriptional profiles" by Sameer S Bajikar*, Christiane Fuchs*, Andreas Roller, Fabian J Theis[^] and Kevin A Janes[^]: PNAS 2014, 111(5), E626-635 (* joint first authors, [^] joint last authors) <doi:10.1073/pnas.1311647111>
 "Pheno-seq - linking visual features and gene expression in 3D cell culture systems" by Stephan M. Tirier, Jeongbin Park, Friedrich Preusser, Lisa Amrhein, Zuguang Gu, Simon Steiger, Jan-Philipp Mallm, Teresa Krieger, Marcel Waschow, Bjoern Eismann, Marta Gut, Ivo G. Gut, Karsten Rippe, Matthias Schlesner, Fabian Theis, Christiane Fuchs, Claudia R. Ball, Hanno Glimm, Roland Eils & Christian Conrad: Sci Rep 9, 12367 (2019) <doi:10.1038/s41598-019-48771-4>

toycluster.EXPLN *Synthetic data from the EXP-LN model*

Description

A matrix containing synthetic measurements from the stochastic profiling EXP-LN model. There is data for 12 genes (columns) and 16 tissue samples (rows). Each measurement is the sum of 10 i.i.d. random variables from a mixture of one lognormal and one exponential distribution.

Usage

```
data(toycluster.EXPLN)
```

Format

The format is: num [1:16, 1:12] 3.77 4.87 5.05 4.45 5.35 ... - attr(*, "dimnames")=List of 2 ..\$: chr [1:16] "V1" "V2" "V3" "V4"\$: chr [1:12] "gene 1" "gene 2" "gene 3" "gene 4" ...

Details

The true underlying parameters are:

TY = 2, i.e. there are two types of cells

p = (0.225, 0.775), that is the probability for cell type I and II, respectively

mu = (0.1223, 0.2705, 2.1457, 2.2899, 1.6791, 1.1558, 2.4035, 0.1998, 0.9648, 0.0411, 1.4798, 1.4206), that is the log-mean for cell type I for genes 1 to 12

sigma = 0.225, that is the log-standard deviation for type I

lambda = (5.5522, 31.5412, 21.2097, 6.1446, 49.0361, 10.9487, 29.7759, 43.8547, 35.7143, 6.5736, 24.8089, 24.7922), that is the exponential rate for cell type II for genes 1 to 12

Source

"Parameterizing cell-to-cell regulatory heterogeneities via stochastic transcriptional profiles" by Sameer S Bajikar*, Christiane Fuchs*, Andreas Roller, Fabian J Theis[^] and Kevin A Janes[^]: PNAS 2014, 111(5), E626-635 (* joint first authors, [^] joint last authors)

References

"Parameterizing cell-to-cell regulatory heterogeneities via stochastic transcriptional profiles" by Sameer S Bajikar*, Christiane Fuchs*, Andreas Roller, Fabian J Theis[^] and Kevin A Janes[^]: PNAS 2014, 111(5), E626-635 (* joint first authors, [^] joint last authors)

Examples

```
data(toycluster.EXPLN)
par(mfrow=c(3,4))
for (i in 1:ncol(toycluster.EXPLN)) {
  hist(toycluster.EXPLN[,i],xlab="synthetic data from EXP-LN model",
       main=colnames(toycluster.EXPLN)[i],col="grey")
}
par(mfrow=c(1,1))
```

toycluster.LNLN

Synthetic data from the LN-LN model

Description

A matrix containing synthetic measurements from the stochastic profiling LN-LN model. There is data for 12 genes (columns) and 16 tissue samples (rows). Each measurement is the sum of 10 i.i.d. random variables from a mixture of lognormal distributions.

Usage

```
data(toycluster.LNLN)
```

Format

The format is: num [1:16, 1:12] 0.789 4.698 4.643 8.734 12.458 ... - attr(*, "dimnames")=List of 2 ..\$: chr [1:16] "V1" "V2" "V3" "V4"\$: chr [1:12] "gene 1" "gene 2" "gene 3" "gene 4" ...

Details

The true underlying parameters are:

TY = 2, i.e. there are two types of cells

p = (0.225, 0.775), that is the probability for cell type I and II, respectively

mu1 = (1.8853, 2.2758, 0.4748, 0.2658, 1.5745, 2.3938, 1.7389, 2.2148, 0.2104, 2.1032, 0.0638, 1.8109), that is the log-mean for cell type I for genes 1 to 12

$\mu_2 = (-2.6637, -0.6590, -1.6308, -2.0753, -1.5786, -0.8131, -2.4872, -3.4486, -3.4865, -2.1848, -1.3868, -2.8238)$, that is the log-mean for cell type II for genes 1 to 12

$\sigma = 0.225$, that is the log-standard deviation for both cell types

Source

"Parameterizing cell-to-cell regulatory heterogeneities via stochastic transcriptional profiles" by Sameer S Bajikar*, Christiane Fuchs*, Andreas Roller, Fabian J Theis[^] and Kevin A Janes[^]: PNAS 2014, 111(5), E626-635 (* joint first authors, [^] joint last authors)

References

"Parameterizing cell-to-cell regulatory heterogeneities via stochastic transcriptional profiles" by Sameer S Bajikar*, Christiane Fuchs*, Andreas Roller, Fabian J Theis[^] and Kevin A Janes[^]: PNAS 2014, 111(5), E626-635 (* joint first authors, [^] joint last authors)

Examples

```
data(toycluster.LNLN)
par(mfrow=c(3,4))
for (i in 1:ncol(toycluster.LNLN)) {
  hist(toycluster.LNLN[,i],xlab="synthetic data from LN-LN model",
       main=colnames(toycluster.LNLN)[i],col="grey")
}
par(mfrow=c(1,1))
```

toycluster.rLNLN

Synthetic data from the rLN-LN model

Description

A matrix containing synthetic measurements from the stochastic profiling rLN-LN model. There is data for 12 genes (columns) and 16 tissue samples (rows). Each measurement is the sum of 10 i.i.d. random variables from a mixture of lognormal distributions.

Usage

```
data(toycluster.rLNLN)
```

Format

The format is: num [1:16, 1:12] 3.46 2.34 3.98 3.42 3.43 ... - attr(*, "dimnames")=List of 2 ..\$: chr [1:16] "V1" "V2" "V3" "V4"\$: chr [1:12] "gene 1" "gene 2" "gene 3" "gene 4" ...

Details

The true underlying parameters are:

$TY = 2$, i.e. there are two types of cells

$p = (0.225, 0.775)$, that is the probability for cell type I and II, respectively

$\mu_1 = (0.1287, 1.6249, 1.0075, 0.5521, 0.1200, 1.1661, 1.4261, 1.8238, 2.4261, 1.2568, 0.9342, 1.8876)$, that is the log-mean for cell type I for genes 1 to 12

$\mu_2 = (-2.2181, -1.6432, -0.9966, -3.1968, -1.9852, -1.0545, -2.3596, -3.0939, -1.3195, -3.2041, -1.2185, -1.3895)$, that is the log-mean for cell type II for genes 1 to 12

$\sigma = (0.225, 0.625)$, that are the log-standard deviations for the two cell types

Source

"Parameterizing cell-to-cell regulatory heterogeneities via stochastic transcriptional profiles" by Sameer S Bajikar*, Christiane Fuchs*, Andreas Roller, Fabian J Theis[^] and Kevin A Janes[^]: PNAS 2014, 111(5), E626-635 (* joint first authors, [^] joint last authors)

References

"Parameterizing cell-to-cell regulatory heterogeneities via stochastic transcriptional profiles" by Sameer S Bajikar*, Christiane Fuchs*, Andreas Roller, Fabian J Theis[^] and Kevin A Janes[^]: PNAS 2014, 111(5), E626-635 (* joint first authors, [^] joint last authors)

Examples

```
data(toycluster.rLNLN)
par(mfrow=c(3,4))
for (i in 1:ncol(toycluster.rLNLN)) {
  hist(toycluster.rLNLN[,i],xlab="synthetic data from rLN-LN model",
      main=colnames(toycluster.rLNLN)[i],col="grey")
}
par(mfrow=c(1,1))
```

Index

- * **SOD2**
 - analyze.sod2, 4
 - sod2, 28
- * **combination of type numbers**
 - comb.summands, 12
- * **constraints**
 - penalty.constraint.EXPLN, 23
 - penalty.constraint.LNLN, 25
 - penalty.constraint.rLNLN, 26
- * **data generation**
 - generate.toydata, 18
 - stochasticProfilingData, 28
- * **datasets**
 - sod2, 28
 - toycluster.EXPLN, 42
 - toycluster.LNLN, 43
 - toycluster.rLNLN, 44
- * **maximum likelihood confidence interval**
 - calculate.ci.EXPLN, 8
 - calculate.ci.LNLN, 9
 - calculate.ci.rLNLN, 11
- * **maximum likelihood estimation**
 - stochasticProfilingML, 29
 - stochprof.loop, 30
 - stochprof.search.EXPLN, 37
 - stochprof.search.LNLN, 39
 - stochprof.search.rLNLN, 40
- * **mixture of lognormals**
 - d.sum.of.mixtures.EXPLN, 13
 - d.sum.of.mixtures.LNLN, 15
 - d.sum.of.mixtures.rLNLN, 17
 - mix.d.sum.of.mixtures.EXPLN, 20
 - mix.d.sum.of.mixtures.LNLN, 21
 - mix.d.sum.of.mixtures.rLNLN, 22
- * **model dependent function definition**
 - set.model.functions, 27
- * **model selection**
 - set.model.functions, 27
- * **package**
 - stochprofML-package, 2
- * **penalization**
 - penalty.constraint.EXPLN, 23
 - penalty.constraint.LNLN, 25
 - penalty.constraint.rLNLN, 26
- * **probability density function**
 - d.sum.of.mixtures.EXPLN, 13
 - d.sum.of.mixtures.LNLN, 15
 - d.sum.of.mixtures.rLNLN, 17
 - mix.d.sum.of.mixtures.EXPLN, 20
 - mix.d.sum.of.mixtures.LNLN, 21
 - mix.d.sum.of.mixtures.rLNLN, 22
- * **random number generator**
 - d.sum.of.mixtures.EXPLN, 13
 - d.sum.of.mixtures.LNLN, 15
 - d.sum.of.mixtures.rLNLN, 17
- * **stochastic profiling**
 - analyze.sod2, 4
 - analyze.toycluster, 6
 - calculate.ci.EXPLN, 8
 - calculate.ci.LNLN, 9
 - calculate.ci.rLNLN, 11
 - d.sum.of.mixtures.EXPLN, 13
 - d.sum.of.mixtures.LNLN, 15
 - d.sum.of.mixtures.rLNLN, 17
 - generate.toydata, 18
 - mix.d.sum.of.mixtures.EXPLN, 20
 - mix.d.sum.of.mixtures.LNLN, 21
 - mix.d.sum.of.mixtures.rLNLN, 22
 - penalty.constraint.EXPLN, 23
 - penalty.constraint.LNLN, 25
 - penalty.constraint.rLNLN, 26
 - set.model.functions, 27
 - sod2, 28
 - stochasticProfilingData, 28
 - stochasticProfilingML, 29
 - stochprof.loop, 30
 - stochprof.results.EXPLN, 33
 - stochprof.results.LNLN, 34

- stochprof.results.rLNLN, 36
- stochprof.search.EXPLN, 37
- stochprof.search.LNLN, 39
- stochprof.search.rLNLN, 40
- stochprofML-package, 2
- toycluster.EXPLN, 42
- toycluster.LNLN, 43
- toycluster.rLNLN, 44
- * **sum of lognormals**
 - d.sum.of.mixtures.EXPLN, 13
 - d.sum.of.mixtures.LNLN, 15
 - d.sum.of.mixtures.rLNLN, 17
 - mix.d.sum.of.mixtures.EXPLN, 20
 - mix.d.sum.of.mixtures.LNLN, 21
 - mix.d.sum.of.mixtures.rLNLN, 22
- * **synthetic data**
 - analyze.toycluster, 6
 - generate.toydata, 18
 - stochasticProfilingData, 28
 - toycluster.EXPLN, 42
 - toycluster.LNLN, 43
 - toycluster.rLNLN, 44
- * **user prompt**
 - stochasticProfilingData, 28
 - stochasticProfilingML, 29
- analyze.sod2, 4
- analyze.toycluster, 6
- calculate.ci.EXPLN, 8
- calculate.ci.LNLN, 9
- calculate.ci.rLNLN, 11
- comb.summands, 12
- d.sum.of.mixtures.EXPLN, 13
- d.sum.of.mixtures.LNLN, 15
- d.sum.of.mixtures.rLNLN, 17
- generate.toydata, 18
- mix.d.sum.of.mixtures.EXPLN, 20
- mix.d.sum.of.mixtures.LNLN, 21
- mix.d.sum.of.mixtures.rLNLN, 22
- penalty.constraint.EXPLN, 23
- penalty.constraint.LNLN, 25
- penalty.constraint.rLNLN, 26
- r.sum.of.mixtures.EXPLN
 - (d.sum.of.mixtures.EXPLN), 13
- r.sum.of.mixtures.LNLN
 - (d.sum.of.mixtures.LNLN), 15
- r.sum.of.mixtures.rLNLN
 - (d.sum.of.mixtures.rLNLN), 17
- set.model.functions, 27
- sod2, 28
- stochasticProfilingData, 28
- stochasticProfilingML, 29
- stochprof.loop, 30
- stochprof.results.EXPLN, 33
- stochprof.results.LNLN, 34
- stochprof.results.rLNLN, 36
- stochprof.search.EXPLN, 37
- stochprof.search.LNLN, 39
- stochprof.search.rLNLN, 40
- stochprofML (stochprofML-package), 2
- stochprofML-package, 2
- toycluster.EXPLN, 42
- toycluster.LNLN, 43
- toycluster.rLNLN, 44