

ODE-constrained mixture modeling

0

Generated by Doxygen 1.8.11

Contents

1	ODEMM Documentation	2
1.1	Introduction	2
1.2	Availability	2
1.3	Installation	2
1.4	Licensing	2
1.5	Models	3
1.5.1	Incorporation of mechanistic description of the mean	3
1.5.2	Incorporation of mechanistic description of the mean and covariance	3
1.6	Distributions	3
2	Examples	3
3	File Index	4
3.1	File List	4
4	File Documentation	5
4.1	collectConditions.m File Reference	5
4.1.1	Detailed Description	5
4.1.2	Function Documentation	6
4.2	computeMixtureProbability.m File Reference	7
4.2.1	Detailed Description	7
4.2.2	Function Documentation	8
4.3	distributions/logn/func_dmudxi_logn_mean.m File Reference	9
4.3.1	Detailed Description	9
4.3.2	Function Documentation	9
4.4	distributions/logn/func_dmudxi_logn_median.m File Reference	10
4.4.1	Detailed Description	10
4.4.2	Function Documentation	10
4.5	distributions/logn/func_dsigma2dxi_logn.m File Reference	11
4.5.1	Detailed Description	11

4.5.2	Function Documentation	11
4.6	distributions/logn/func_dSigmadxi_logn.m File Reference	12
4.6.1	Detailed Description	12
4.6.2	Function Documentation	12
4.7	distributions/logn/func_Sigma_logn.m File Reference	13
4.7.1	Detailed Description	13
4.7.2	Function Documentation	13
4.8	distributions/logn/logoflognpdf.m File Reference	14
4.8.1	Detailed Description	14
4.8.2	Function Documentation	14
4.9	distributions/logn/logofmvnpdf.m File Reference	14
4.9.1	Detailed Description	15
4.9.2	Function Documentation	15
4.10	distributions/norm/func_dmudxi_norm.m File Reference	15
4.10.1	Detailed Description	16
4.10.2	Function Documentation	16
4.11	distributions/norm/func_dsigma2dxi_norm.m File Reference	16
4.11.1	Detailed Description	17
4.11.2	Function Documentation	17
4.12	distributions/norm/func_dSigmadxi_norm.m File Reference	17
4.12.1	Detailed Description	18
4.12.2	Function Documentation	18
4.13	distributions/norm/func_Sigma_norm.m File Reference	18
4.13.1	Detailed Description	18
4.13.2	Function Documentation	18
4.14	distributions/norm/logofnormpdf.m File Reference	19
4.14.1	Detailed Description	19
4.14.2	Function Documentation	19
4.15	generateODEMM.m File Reference	20
4.15.1	Detailed Description	20

4.15.2	Function Documentation	20
4.16	getLognMeanVar.m File Reference	21
4.16.1	Detailed Description	21
4.16.2	Function Documentation	21
4.17	getRREsigmas.m File Reference	22
4.17.1	Detailed Description	22
4.17.2	Function Documentation	22
4.18	getScalingFactors.m File Reference	23
4.18.1	Detailed Description	23
4.18.2	Function Documentation	24
4.19	load_plot_settings.m File Reference	24
4.19.1	Detailed Description	24
4.20	logLikelihood.m File Reference	24
4.20.1	Detailed Description	25
4.20.2	Function Documentation	25
4.21	plotODEMix.m File Reference	27
4.21.1	Detailed Description	27
4.21.2	Function Documentation	27
4.22	printParams.m File Reference	28
4.22.1	Detailed Description	28
4.22.2	Function Documentation	28
4.23	testSigmaPointApp_mod.m File Reference	28
4.23.1	Detailed Description	29
4.23.2	Function Documentation	29

1 ODEMM Documentation

1.1 Introduction

Cellular heterogeneity occurs at multiple levels.

ODEMM combines mixture modeling with mechanistic models for the individual subpopulations.

ODEMM offers

- models that are able to include mechanistic descriptions of the means of individual subpopulations, e.g., by reaction rate equations (RRE) (Hasenauer et al., PloS CB (2014), Loos et al., CM-SB (2016)). For more details see [below](#).
- Hierarchical population models that incorporate means and covariances of the individual subpopulations, e.g., provided by the sigma-point approximation or the moment-closure approximation. For more details see [below](#).

1.2 Availability

ODEMM is a freely available MATLAB (MathWorks) toolbox available at <https://github.com/ICB-DCM/ODEMM/>. It can be retrieved by downloading the zip archive at <https://github.com/ICB-DCM/ODEMM/archive/master.zip> or cloning the git repository via

```
1 git clone git@github.com:ICB-DCM/ODEMM.git
```

1.3 Installation

If the repository was cloned, [install_ODEMM.m](#) needs to be run to add the folders to the MATLAB search path. If the zip archive was downloaded the archive needs to be unzipped before executing [install_ODEMM.m](#)

Toolboxes required for the examples

In principle, every simulation that provides means (and covariances) can be incorporated into ODEMM. For our examples, we used the simulations obtained by AMICI and calibrated the models using the parameter estimation toolbox PESTO.

- AMICI (simulation): <https://github.com/ICB-DCM/AMICI>
- SPToolbox (sigma-point approximation): <https://github.com/ICB-DCM/SPToolbox>
- PESTO (parameter estimation): <https://github.com/ICB-DCM/PESTO>

1.4 Licensing

See LICENSE file in the ODEMM source directory.

1.5 Models

ODEMM implements different kinds of ODE constrained mixture models.

1.5.1 Incorporation of mechanistic description of the mean

If the mean of a subpopulation is described by, e.g., RRE, the variances of the measurements are treated as additional parameters. An example for setting up RRE constrained mixture models is given in `models_RRE()` which can be found in `examples/conversion_reaction/`.

1.5.2 Incorporation of mechanistic description of the mean and covariance

When not only a mechanistic description of the mean, but also of the covariance is provided by the simulation function for the individual subpopulations, a hierarchical population model can be created. In our examples, we assessed two approximations for obtaining the statistical moments of the subpopulations:

- Sigma-point approximation (`/examples/conversion_reaction/models_SP`)
- Moment-closure approximation (`/examples/two_stage_intrinsic`)

1.6 Distributions

For the mixture distribution, ODEMM implements

- multivariate normal distributions
- multivariate log-normal distributions.

The density functions and corresponding functions required for the models can be found in `distributions`.

2 Examples

The following examples are included:

- Conversion reaction (`examples/conversion_reaction`)
- Differential protein expression (`examples/differential_protein_expression`)
- Subpopulation differences in NGF-induced Erk1/2 signaling (`examples/subpopulation_differences`)
- Differences mediated by extracellular scaffolds in NGF-induced Erk1/2 signaling (`examples/ECM_differences`)
- Intrinsic noise for a two stage model of gene expression (`examples/two_stage_intrinsic`)

These models require the freely available toolboxes

- AMICI for simulation (<http://icb-dcm.github.io/AMICI/>)
- PESTO for the parameter estimation (<http://icb-dcm.github.io/PESTO/>)
- SPToolbox for the sigma-point approximation (<http://icb-dcm.github.io/SPToolbox/>)

3 File Index

3.1 File List

Here is a list of all documented files with brief descriptions:

collectConditions.m	This function collects all different conditions regarding input/differences between subpopulations/experiments and timepoints	5
computeMixtureProbability.m	Robust calculation of a mixture distribution likelihood	7
generateODEMM.m	This function generates file defining ODE-constrained mixture model	20
getLognMeanVar.m	This function calculations the mean and variances, and the corresponding derivatives of a multivariate log-normal distribution given the parameters and	21
getRREsigmas.m	This function defines the parameters needed for the parametrization of the variances in case Reaction Rate Equations are used for the mechanistic description of the means	22
getScalingFactors.m	Calculates scaling factors for replicates such that the log of the means are as similar as possible	23
getSigmaPointApp_status_mod.m		??
install_ODEMM.m		??
load_plot_settings.m	Functions to set font sizes and colors for the visualization	24
logLikelihood.m	This function evaluates the likelihood function for a given model, data and parameter vector	24
plotODEMix.m	Routine to plot the ODE-constrained mixture model	27
printParams.m	Help function to print parameters names and values	28
testSigmaPointApp_mod.m	Modified version of the testSigmaPointApp_status.m function of the SPToolbox	28
distributions/logn/func_dmudxi_logn_mean.m	This function calculates the derivative of mu of a log-normal distribution for the case of linking the mean of the simulation to the mean of the log-normal distribution	9
distributions/logn/func_dmudxi_logn_median.m	This function calculates the derivative of mu of a log-normal distribution for the case of linking the mean of the simulation to the median of the log-normal distributin	10
distributions/logn/func_dsigma2dxi_logn.m	This function calcuatates the derivative of σ^2 in case of univariate measurements and a log-normal distribution assumption	11

distributions/logn/ func_dSigmadxi_logn.m	
This function maps means and variances to of a log-normal distribution	12
distributions/logn/ func_Sigma_logn.m	
This function maps means and variances to of a log-normal distribution	13
distributions/logn/ logoflognpdf.m	
Modified version of MATLAB function LOGNPDF	14
distributions/logn/ logofmvnpdf.m	
Modified version of MATLAB function MVNPDF	14
distributions/norm/ func_dmudxi_norm.m	
This function calculates the derivative of mu of a normal distribution	15
distributions/norm/ func_dsigma2dxi_norm.m	
This function calculates the derivative of σ^2 in case of univariate measurements and a normal distribution assumption	16
distributions/norm/ func_dSigmadxi_norm.m	
This function maps the means and variances to Sigma of a normal distribution	17
distributions/norm/ func_Sigma_norm.m	
This function maps the means and variances to Sigma of a normal distribution	18
distributions/norm/ logofnormpdf.m	
Modified version of MATLAB function NORMPDF such that the log-density is given back	19

4 File Documentation

4.1 collectConditions.m File Reference

This function collects all different conditions regarding input/differences between subpopulations/experiments and timepoints.

Functions

- `mlhsSubst< mlhsInnerSubst< matlabtypesubstitute, conditions >,mlhsInnerSubst< matlabtypesubstitute, D > > collectConditions` (matlabtypesubstitute D, matlabtypesubstitute M)

This function collects all different conditions regarding input/differences between subpopulations/experiments and timepoints.

4.1.1 Detailed Description

This function collects all different conditions regarding input/differences between subpopulations/experiments and timepoints.

4.1.2 Function Documentation

4.1.2.1 `mlhsSubst< mlhsInnerSubst< matlabtypesubstitute, conditions >,mlhsInnerSubst< matlabtypesubstitute, D > >`
`collectConditions (matlabtypesubstitute D, matlabtypesubstitute M)`

This function collects all different conditions regarding input/differences between subpopulations/experiments and timepoints.

USAGE

`[conditions,D] = collectConditions(D,M)`

Parameters

<i>D</i>	data struct
<i>M</i>	model struct

Return values

<i>conditions</i>	conditions struct
<i>D</i>	updated data struct

Required fields of D:

- *t* -- time vector
- *u* -- vector of stimulations

Required fields of M:

- *n_subpop* -- number of subpopulations
- *u{s,e}* — input vector describing differences between subpopulations and experiments

Generated fields of D:

- *c* -- *n_subpop* x (*n_u* + *n_differences*) matrix linking condition to data

Generated fields of conditions:

- *input* -- (*n_u* + *n_differences*) x 1 input vector
- *time* -- 1 x *n_t* time vector
- *sigma* -- 1 x *n_t* vector of sigmas for condition *c*

Definition at line 17 of file collectConditions.m.

Referenced by logLikelihood(), and plotODEMix().

4.2 computeMixtureProbability.m File Reference

Robust calculation of a mixture distribution likelihood.

Functions

- `mlhsInnerSubst< matlabtypesubstitute, varargin > computeMixtureProbability` (matlabtypesubstitute varargin)

Robust calculation of a mixture distribution likelihood.

4.2.1 Detailed Description

Robust calculation of a mixture distribution likelihood.

4.2.2 Function Documentation

4.2.2.1 `mlhsInnerSubst` < `matlabtypesubstitute`, `varargout` > `computeMixtureProbability` (`matlabtypesubstitute` *varargin*)

Robust calculation of a mixture distribution likelihood.

USAGE

```
[logp,dlogpdx] = computeMixtureProbability(w,q_i,H_i)
[logp] = computeMixtureProbability(w,q_i)
```

Parameters

<i>varargin</i>	<pre>1 computeMixtureProbability (w, q_i, H_i)</pre> <p><i>Required Parameters for varargin:</i></p> <ul style="list-style-type: none"> • w (1 x n_s) vector with weights • q_i ($n \times n_s$) matrix with $\log(p_i)$ for every column • H_i ($n \times n_{xi} \times n_s$) s.th. $d(w_i * p_i)/dx_i = p_i * H_i$
-----------------	---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

Return values

<i>varargout</i>	
<i>logp</i>	1x1 scalar of loglikelihood
<i>dlogpdx_i</i>	$n_{xi} \times 1$ vector of gradient

Definition at line 17 of file computeMixtureProbability.m.

Referenced by logLikelihood().

4.3 distributions/logn/func_dmudxi_logn_mean.m File Reference

This function calculates the derivative of mu of a log-normal distribution for the case of linking the mean of the simulation to the mean of the log-normal distribution.

Functions

- `mlhslInnerSubst < matlabtypesubstitute, dmudxi > func_dmudxi_logn_mean (matlabtypesubstitute t, matlabtypesubstitute x, matlabtypesubstitute dx_dxi, matlabtypesubstitute Sigma, matlabtypesubstitute dSigmadx_i, matlabtypesubstitute xi, matlabtypesubstitute u, matlabtypesubstitute dim)`

This function calculates the derivative of mu of a log-normal distribution for the case of linking the mean of the simulation to the mean of the log-normal distribution.

4.3.1 Detailed Description

This function calculates the derivative of mu of a log-normal distribution for the case of linking the mean of the simulation to the mean of the log-normal distribution.

4.3.2 Function Documentation

- 4.3.2.1 `mlhslInnerSubst < matlabtypesubstitute, dmudxi > func_dmudxi_logn_mean (matlabtypesubstitute t, matlabtypesubstitute x, matlabtypesubstitute dx_dxi, matlabtypesubstitute Sigma, matlabtypesubstitute dSigmadx_i, matlabtypesubstitute xi, matlabtypesubstitute u, matlabtypesubstitute dim)`

This function calculates the derivative of mu of a log-normal distribution for the case of linking the mean of the simulation to the mean of the log-normal distribution.

Parameters

t	time vector (not used, included for consistency and possible extensions)
x	vector of means and variances (not used, included for consistency and possible extensions)
$dxdxi$	derivatives of means and variances
Σ	(not used, included for consistency and possible extensions)
$d\Sigma dxi$	(not used, included for consistency and possible extensions)
ξ	parameter vector (not used, included for consistency and possible extensions)
u	input (not used, included for consistency and possible extensions)
n_dim	dimension of measurement

Return values

$dmudxi$	derivative of mu of a log-normal distribution
----------	-----------------------------------------------

Definition at line 17 of file func_dmudxi_logn_mean.m.

4.4 distributions/logn/func_dmudxi_logn_median.m File Reference

This function calculates the derivative of mu of a log-normal distribution for the case of linking the mean of the simulation to the median of the log-normal distributin.

Functions

- `mlhsInnerSubst < matlabtypesubstitute, dmudxi > func_dmudxi_logn_median (matlabtypesubstitute t , matlabtypesubstitute x , matlabtypesubstitute $dxdxi$, matlabtypesubstitute Σ , matlabtypesubstitute $d\Sigma dxi$, matlabtypesubstitute ξ , matlabtypesubstitute u , matlabtypesubstitute dim)`

This function calculates the derivative of mu of a log-normal distribution for the case of linking the mean of the simulation to the median of the log-normal distributin.

4.4.1 Detailed Description

This function calculates the derivative of mu of a log-normal distribution for the case of linking the mean of the simulation to the median of the log-normal distributin.

4.4.2 Function Documentation

- 4.4.2.1** `mlhsInnerSubst < matlabtypesubstitute, dmudxi > func_dmudxi_logn_median (matlabtypesubstitute t , matlabtypesubstitute x , matlabtypesubstitute $dxdxi$, matlabtypesubstitute Σ , matlabtypesubstitute $d\Sigma dxi$, matlabtypesubstitute ξ , matlabtypesubstitute u , matlabtypesubstitute dim)`

This function calculates the derivative of mu of a log-normal distribution for the case of linking the mean of the simulation to the median of the log-normal distributin.

Parameters

t	time vector (not used, included for consistency and possible extensions)
-----	--------------------------------------------------------------------------

Parameters

x	vector of means and variances (not used, included for consistency and possible extensions)
$dxdxi$	derivatives of means and variances
Σ	(not used, included for consistency and possible extensions)
$d\Sigma dxi$	(not used, included for consistency and possible extensions)
xi	parameter vector (not used, included for consistency and possible extensions)
u	input (not used, included for consistency and possible extensions)
n_dim	dimension of measurement

Return values

$dmudxi$	derivative of mu of a log-normal distribution
----------	-----------------------------------------------

Definition at line 17 of file func_dmudxi_logn_median.m.

4.5 distributions/logn/func_dsigma2dxi_logn.m File Reference

This function calculates the derivative of σ^2 in case of univariate measurements and a log-normal distribution assumption.

Functions

- `mlhsInnerSubst< matlabtypesubstitute, dsigma2dxi > func_dsigma2dxi_logn` (matlabtypesubstitute t , matlabtypesubstitute x , matlabtypesubstitute $dxdxi$, matlabtypesubstitute xi , matlabtypesubstitute $varargin$)

This function calculates the derivative of σ^2 in case of univariate measurements and a log-normal distribution assumption.

4.5.1 Detailed Description

This function calculates the derivative of σ^2 in case of univariate measurements and a log-normal distribution assumption.

4.5.2 Function Documentation

- 4.5.2.1 `mlhsInnerSubst< matlabtypesubstitute, dsigma2dxi > func_dsigma2dxi_logn` (matlabtypesubstitute t , matlabtypesubstitute x , matlabtypesubstitute $dxdxi$, matlabtypesubstitute xi , matlabtypesubstitute $varargin$)

This function calculates the derivative of σ^2 in case of univariate measurements and a log-normal distribution assumption.

USAGE: `dsigma2dxi = func_dsigma2dxi_logn(t,x,dxdxi,xi)` `dsigma2dxi = func_dsigma2dxi_logn(t,x,dxdxi,xi,noise,dnoisedxi,noise←_model)`

Parameters

t	time vector (not used, included for consistency and possible extensions)
x	vector of means and variances (not used, included for consistency and possible extensions)

Parameters

<i>dxdxi</i>	derivatives of means and variances
<i>xi</i>	parameter vector(not used, included for consistency and possible extensions)
<i>varargin</i>	<pre>1 func_dsigma2dxi_logn (..., noise, dnoisedxi, noise_model)</pre> <p><i>Required Parameters for varargin:</i></p> <ul style="list-style-type: none"> • noise parameter for measurement noise • dnoisedxi derivative of measurement noise • noise_model

Return values

<i>dsigma2dxi</i>	derivative of σ^2 of a log-normal distribution
-------------------	-------------------------------------------------------

Definition at line 17 of file func_dsigma2dxi_logn.m.

4.6 distributions/logn/func_dSigmadxi_logn.m File Reference

This function maps means and variances to of a log-normal distribution.

Functions

- mlhsInnerSubst< matlabtypesubstitute, dSigmadxi > [func_dSigmadxi_logn](#) (matlabtypesubstitute t, matlabtypesubstitute x, matlabtypesubstitute dxdxi, matlabtypesubstitute xi, matlabtypesubstitute n_dim, matlabtypesubstitute varargin)

This function maps means and variances to of a log-normal distribution.

4.6.1 Detailed Description

This function maps means and variances to of a log-normal distribution.

4.6.2 Function Documentation

- #### 4.6.2.1 mlhsInnerSubst< matlabtypesubstitute, dSigmadxi > func_dSigmadxi_logn (matlabtypesubstitute t, matlabtypesubstitute x, matlabtypesubstitute dxdxi, matlabtypesubstitute xi, matlabtypesubstitute n_dim, matlabtypesubstitute varargin)

This function maps means and variances to of a log-normal distribution.

USAGE

```
dSigmadxi = func_dSigmadxi_logn(t,x,dxdxi,xi,n_n_dim) dSigmadxi = func_dSigmadxi_logn(t,x,dxdxi,xi,n_n_dim,noise,dnoisedxi,multiplicative)
```

Definition at line 17 of file func_dSigmadxi_logn.m.

4.7 distributions/logn/func_Sigma_logn.m File Reference

This function maps means and variances to of a log-normal distribution.

Functions

- `mlhsInnerSubst< matlabtypesubstitute, Sigma > func_Sigma_logn` (matlabtypesubstitute `t`, matlabtypesubstitute `x`, matlabtypesubstitute `xi`, matlabtypesubstitute `n_dim`, matlabtypesubstitute `varargin`)

This function maps means and variances to of a log-normal distribution.

4.7.1 Detailed Description

This function maps means and variances to of a log-normal distribution.

4.7.2 Function Documentation

4.7.2.1 `mlhsInnerSubst< matlabtypesubstitute, Sigma > func_Sigma_logn` (matlabtypesubstitute `t`, matlabtypesubstitute `x`, matlabtypesubstitute `xi`, matlabtypesubstitute `n_dim`, matlabtypesubstitute `varargin`)

This function maps means and variances to of a log-normal distribution.

USAGE

`Sigma = func_Sigma_logn(t,x,xi,n_dim,noise,noisemodel)`

Parameters

<code>t</code>	time vector
<code>x</code>	vector including means and variances
<code>xi</code>	(not used, included for consistency and possible extensions)
<code>n_dim</code>	dimension of measurement
<code>varargin</code>	<pre>1 func_Sigma_logn (..., noise, noisemodel)</pre> <p><i>Required Parameters for varargin:</i></p> <ul style="list-style-type: none"> • noise • noisemodel

Return values

<code>Sigma</code>	<code>n_t x n_dim x n_dim</code>
--------------------	----------------------------------

Definition at line 17 of file `func_Sigma_logn.m`.

4.8 distributions/logn/logoflognpdf.m File Reference

Modified version of MATLAB function LOGNPDF.

Functions

- mlhsInnerSubst< matlabtypesubstitute, logy > [logoflognpdf](#) (matlabtypesubstitute x, matlabtypesubstitute mu, matlabtypesubstitute sigma)

Modified version of MATLAB function LOGNPDF.

4.8.1 Detailed Description

Modified version of MATLAB function LOGNPDF.

4.8.2 Function Documentation

4.8.2.1 mlhsInnerSubst< matlabtypesubstitute, logy > logoflognpdf (matlabtypesubstitute x, matlabtypesubstitute mu, matlabtypesubstitute sigma)

Modified version of MATLAB function LOGNPDF.

LOGNPDF Lognormal probability density function (pdf). $Y = \text{LOGNPDF}(X, MU, SIGMA)$ returns values at X of the lognormal pdf with distribution parameters MU and $SIGMA$. MU and $SIGMA$ are the mean and standard deviation, respectively, of the associated normal distribution. The size of Y is the common size of the input arguments. A scalar input functions as a constant matrix of the same size as the other inputs.

Default values for MU and $SIGMA$ are 0 and 1 respectively.

See also

LOGNCDF, LOGNFIT, LOGNINV, LOGNLIKE, LOGNRND, LOGNSTAT.

Definition at line 17 of file logoflognpdf.m.

Referenced by logLikelihood().

4.9 distributions/logn/logofmvnpdf.m File Reference

Modified version of MATLAB function MVNPDF.

Functions

- mlhsInnerSubst< matlabtypesubstitute, y > [logofmvnpdf](#) (matlabtypesubstitute X, matlabtypesubstitute Mu, matlabtypesubstitute Sigma)

Modified version of MATLAB function MVNPDF.

4.9.1 Detailed Description

Modified version of MATLAB function MVNPDF.

4.9.2 Function Documentation

4.9.2.1 `mlhsInnerSubst < matlabtypesubstitute, y > logofmvnpdf (matlabtypesubstitute X, matlabtypesubstitute Mu, matlabtypesubstitute Sigma)`

Modified version of MATLAB function MVNPDF.

MVNPDF Multivariate normal probability density function (pdf). $Y = \text{MVNPDF}(X)$ returns the probability density of the multivariate normal distribution with zero mean and identity covariance matrix, evaluated at each row of X . Rows of the N -by- D matrix X correspond to observations or points, and columns correspond to variables or coordinates. Y is an N -by-1 vector.

$Y = \text{MVNPDF}(X, MU)$ returns the density of the multivariate normal distribution with mean MU and identity covariance matrix, evaluated at each row of X . MU is a 1-by- D vector, or an N -by- D matrix, in which case the density is evaluated for each row of X with the corresponding row of MU . MU can also be a scalar value, which MVNPDF replicates to match the size of X .

$Y = \text{MVNPDF}(X, MU, SIGMA)$ returns the density of the multivariate normal distribution with mean MU and covariance $SIGMA$, evaluated at each row of X . $SIGMA$ is a D -by- D matrix, or an D -by- D -by- N array, in which case the density is evaluated for each row of X with the corresponding page of $SIGMA$, i.e., MVNPDF computes $Y(l)$ using $X(l,:)$ and $SIGMA(:, :, l)$. If the covariance matrix is diagonal, containing variances along the diagonal and zero covariances off the diagonal, $SIGMA$ may also be specified as a 1-by- D matrix or a 1-by- D -by- N array, containing just the diagonal. Pass in the empty matrix for MU to use its default value when you want to only specify $SIGMA$.

If X is a 1-by- D vector, MVNPDF replicates it to match the leading dimension of MU or the trailing dimension of $SIGMA$.

Example

```
mu = [1 -1]; Sigma = [.9 .4; .4 .3];
[X1,X2] = meshgrid(linspace(-1,3,25)<tt>, linspace(-3,1,25)</tt>);
X = [X1(:) X2(:)];
p = mvnpdf(X, mu, Sigma);
surf(X1,X2,reshape(p,25,25));
```

See also

MVTPDF, MVNCDF, MVNRND, NORMPDF.

Definition at line 17 of file logofmvnpdf.m.

Referenced by logLikelihood().

4.10 distributions/norm/func_dmudxi_norm.m File Reference

This function calculates the derivative of mu of a normal distribution.

Functions

- `mlhsInnerSubst< matlabtypesubstitute, dmudxi > func_dmudxi_norm` (matlabtypesubstitute `t`, matlabtypesubstitute `x`, matlabtypesubstitute `dxdxi`, matlabtypesubstitute `Sigma`, matlabtypesubstitute `dSigmadxi`, matlabtypesubstitute `xi`, matlabtypesubstitute `u`, matlabtypesubstitute `n_dim`)

This function calculates the derivative of mu of a normal distribution.

4.10.1 Detailed Description

This function calculates the derivative of mu of a normal distribution.

4.10.2 Function Documentation

- 4.10.2.1 `mlhsInnerSubst< matlabtypesubstitute, dmudxi > func_dmudxi_norm` (matlabtypesubstitute `t`, matlabtypesubstitute `x`, matlabtypesubstitute `dxdxi`, matlabtypesubstitute `Sigma`, matlabtypesubstitute `dSigmadxi`, matlabtypesubstitute `xi`, matlabtypesubstitute `u`, matlabtypesubstitute `n_dim`)

This function calculates the derivative of mu of a normal distribution.

Parameters

<code>t</code>	time vector (not used, included for consistency and possible extensions)
<code>x</code>	vector of means and variances (not used, included for consistency and possible extensions)
<code>dxdxi</code>	derivatives of means and variances
<code>Sigma</code>	(not used, included for consistency and possible extensions)
<code>dSigmadxi</code>	(not used, included for consistency and possible extensions)
<code>xi</code>	parameter vector (not used, included for consistency and possible extensions)
<code>u</code>	input (not used, included for consistency and possible extensions)
<code>n_dim</code>	dimension of measurement

Return values

<code>dmudxi</code>	derivative of mu of a normal distribution
---------------------	-------------------------------------------

Definition at line 17 of file `func_dmudxi_norm.m`.

4.11 distributions/norm/func_dsigma2dxi_norm.m File Reference

This function calculates the derivative of σ^2 in case of univariate measurements and a normal distribution assumption.

Functions

- `mlhsInnerSubst< matlabtypesubstitute, dsigma2dxi > func_dsigma2dxi_norm` (matlabtypesubstitute `t`, matlabtypesubstitute `x`, matlabtypesubstitute `dxdxi`, matlabtypesubstitute `xi`, matlabtypesubstitute `varargin`)

This function calculates the derivative of σ^2 in case of univariate measurements and a normal distribution assumption.

4.11.1 Detailed Description

This function calculates the derivative of σ^2 in case of univariate measurements and a normal distribution assumption.

4.11.2 Function Documentation

4.11.2.1 `mlhsInnerSubst< matlabtypesubstitute, dSigma2dxi > func_dSigma2dxi_norm (matlabtypesubstitute t, matlabtypesubstitute x, matlabtypesubstitute dxDxi, matlabtypesubstitute xi, matlabtypesubstitute varargin)`

This function calculates the derivative of σ^2 in case of univariate measurements and a normal distribution assumption.

USAGE: `dSigma2dxi = func_dSigma2dxi_norm(t,x,dxDxi,xi)` `dSigma2dxi = func_dSigma2dxi_norm(t,x,dxDxi,xi,noise,dnoisedxi,noise_model)`

Parameters

<i>t</i>	time vector (not used, included for consistency and possible extensions)
<i>x</i>	vector of means and variances (not used, included for consistency and possible extensions)
<i>dxDxi</i>	derivatives of means and variances
<i>xi</i>	parameter vector(not used, included for consistency and possible extensions)
<i>varargin</i>	<pre>1 func_dSigma2dxi_norm (..., noise, dnoisedxi, noise_model)</pre> <p><i>Required Parameters for varargin:</i></p> <ul style="list-style-type: none"> • noise parameter for measurement noise • dnoisedxi derivative of measurement noise • noise_model

Return values

<i>dSigma2dxi</i>	derivative of σ^2 of a normal distribution
-------------------	---------------------------------------------------

Definition at line 17 of file `func_dSigma2dxi_norm.m`.

4.12 distributions/norm/func_dSigmaDxi_norm.m File Reference

This function maps the means and variances to Sigma of a normal distribution.

Functions

- `mlhsInnerSubst< matlabtypesubstitute, dSigmaDxi > func_dSigmaDxi_norm` (matlabtypesubstitute t, matlabtypesubstitute x, matlabtypesubstitute dxDxi, matlabtypesubstitute xi, matlabtypesubstitute n_dim, matlabtypesubstitute varargin)

This function maps the means and variances to Sigma of a normal distribution.

4.12.1 Detailed Description

This function maps the means and variances to Sigma of a normal distribution.

4.12.2 Function Documentation

4.12.2.1 `mlhsInnerSubst< matlabtypesubstitute, dSigmadxi > func_dSigmadxi_norm (matlabtypesubstitute t, matlabtypesubstitute x, matlabtypesubstitute dx dx_i, matlabtypesubstitute xi, matlabtypesubstitute n_dim, matlabtypesubstitute varargin)`

This function maps the means and variances to Sigma of a normal distribution.

USAGE

```
dSigmadxi = func_dSigmadxi_norm(t,x,dx dx_i,xi,n_n_dim) dSigmadxi = func_dSigmadxi_norm(t,x,dx dx_i,xi,n_↔
n_dim,noise,dnoisedxi,multiplicative)
```

Definition at line 17 of file `func_dSigmadxi_norm.m`.

4.13 distributions/norm/func_Sigma_norm.m File Reference

This function maps the means and variances to Sigma of a normal distribution.

Functions

- `mlhsInnerSubst< matlabtypesubstitute, Sigma > func_Sigma_norm` (`matlabtypesubstitute t`, `matlabtypesubstitute x`, `matlabtypesubstitute xi`, `matlabtypesubstitute n_dim`, `matlabtypesubstitute varargin`)
This function maps the means and variances to Sigma of a normal distribution.

4.13.1 Detailed Description

This function maps the means and variances to Sigma of a normal distribution.

4.13.2 Function Documentation

4.13.2.1 `mlhsInnerSubst< matlabtypesubstitute, Sigma > func_Sigma_norm (matlabtypesubstitute t, matlabtypesubstitute x, matlabtypesubstitute xi, matlabtypesubstitute n_dim, matlabtypesubstitute varargin)`

This function maps the means and variances to Sigma of a normal distribution.

USAGE

```
Sigma = func_Sigma_norm(t,x,xi,n_dim,noise,noisemodel)
```

Parameters

<i>t</i>	time vector
<i>x</i>	vector including means and variances
<i>xi</i>	(not used, included for consistency and possible extensions)
<i>n_dim</i>	dimension of measurement
<i>varargin</i>	<pre>1 func_Sigma_norm (..., noise, noisemodel)</pre> <p><i>Required Parameters for varargin:</i></p> <ul style="list-style-type: none"> • noise • noisemodel

Return values

<i>Sigma</i>	$n_t \times n_dim \times n_dim$
--------------	------------------------------------

Definition at line 17 of file func_Sigma_norm.m.

4.14 distributions/norm/logofnormpdf.m File Reference

Modified version of MATLAB function NORMPDF such that the log-density is given back.

Functions

- `mlhsInnerSubst< matlabtypesubstitute, logy > logofnormpdf (matlabtypesubstitute x, matlabtypesubstitute mu, matlabtypesubstitute sigma)`

Modified version of MATLAB function NORMPDF such that the log-density is given back.

4.14.1 Detailed Description

Modified version of MATLAB function NORMPDF such that the log-density is given back.

4.14.2 Function Documentation

4.14.2.1 `mlhsInnerSubst< matlabtypesubstitute, logy > logofnormpdf (matlabtypesubstitute x, matlabtypesubstitute mu, matlabtypesubstitute sigma)`

Modified version of MATLAB function NORMPDF such that the log-density is given back.

NORMPDF Normal probability density function (pdf). `Y = NORMPDF(X,MU,SIGMA)` returns the pdf of the normal distribution with mean MU and standard deviation SIGMA, evaluated at the values in X. The size of Y is the common size of the input arguments. A scalar input functions as a constant matrix of the same size as the other inputs.

Default values for MU and SIGMA are 0 and 1 respectively.

See also

NORMCDF, NORMFIT, NORMINV, NORMLIKE, NORMRND, NORMSTAT.

Definition at line 17 of file logofnormpdf.m.

Referenced by `logLikelihood()`.

4.15 generateODEMM.m File Reference

This function generates file defining ODE-constrained mixture model.

Functions

- `mlhsInnerSubst< matlabtypesubstitute, varargout > generateODEMM` (matlabtypesubstitute *D*, matlabtypesubstitute *M*, matlabtypesubstitute *parameters*, matlabtypesubstitute *conditions*, matlabtypesubstitute *varargin*)

This function generates file defining ODE-constrained mixture model.

- `mlhsInnerSubst< matlabtypesubstitute, retstr > mtoc_subst_generateODEMM_m_tsbust_cotm_↔ replace_xi_x_u` (matlabtypesubstitute *symexpr*)
- `mlhsInnerSubst< matlabtypesubstitute, str_dzdx > mtoc_subst_generateODEMM_m_tsbust_cotm_↔ getStrDerivative2Terms` (matlabtypesubstitute *sym_expr*, matlabtypesubstitute *x*, matlabtypesubstitute *dxdxi*, matlabtypesubstitute *xi*)
- `mlhsInnerSubst< matlabtypesubstitute, str_dzdx > mtoc_subst_generateODEMM_m_tsbust_cotm_↔ getStrDerivative3Terms` (matlabtypesubstitute *deriv_name*, matlabtypesubstitute *sym_expr*, matlabtypesubstitute *s*, matlabtypesubstitute *e*, matlabtypesubstitute *x*, matlabtypesubstitute *dxdxi*, matlabtypesubstitute *sigma*, matlabtypesubstitute *dsigmadxi*, matlabtypesubstitute *xi*)

4.15.1 Detailed Description

This function generates file defining ODE-constrained mixture model.

4.15.2 Function Documentation

- 4.15.2.1 `mlhsInnerSubst< matlabtypesubstitute, varargout > generateODEMM` (matlabtypesubstitute *D*, matlabtypesubstitute *M*, matlabtypesubstitute *parameters*, matlabtypesubstitute *conditions*, matlabtypesubstitute *varargin*)

This function generates file defining ODE-constrained mixture model.

USAGE

`M = generateODEMM(D,M,parameters,conditions,options)`

Parameters

<i>D</i>	data struct
<i>M</i>	model struct
<i>parameters</i>	parameters struct
<i>conditions</i>	conditions struct obtained by collectConditions.m
<i>varargin</i>	<pre>1 generateODEMM (..., options)</pre> <p><i>Required Parameters for varargin:</i></p> <ul style="list-style-type: none"> • options

Required fields of M:

Required fields of parameters:

Definition at line 17 of file generateODEMM.m.

4.16 getLognMeanVar.m File Reference

This function calculations the mean and variances, and the corresponding derivatives of a multivariate log-normal distribution given the parameters and .

Functions

- `mlhsSubst< mlhsInnerSubst< matlabtypesubstitute, Zout >,mlhsInnerSubst< matlabtypesubstitute, varargout > > getLognMeanVar (matlabtypesubstitute Z, matlabtypesubstitute n_dim, matlabtypesubstitute varargin)`

This function calculations the mean and variances, and the corresponding derivatives of a multivariate log-normal distribution given the parameters and .

4.16.1 Detailed Description

This function calculations the mean and variances, and the corresponding derivatives of a multivariate log-normal distribution given the parameters and .

4.16.2 Function Documentation

4.16.2.1 `mlhsSubst< mlhsInnerSubst< matlabtypesubstitute, Zout >,mlhsInnerSubst< matlabtypesubstitute, varargout > > getLognMeanVar (matlabtypesubstitute Z, matlabtypesubstitute n_dim, matlabtypesubstitute varargin)`

This function calculations the mean and variances, and the corresponding derivatives of a multivariate log-normal distribution given the parameters and .

USAGE

`[Zout] = getLognMeanVar(Z,n_dim)` `[Zout,dZdthetaout] = getLognMeanVar(Z,n_dim,dZdtheta)`

Parameters

<i>Z</i>	(n_dim + n_dim(n_dim+1)/2) x n_t vector with and
<i>n_dim</i>	dimension of the multivariate log-normal distribution
<i>varargin</i>	<div> 1 getLognMeanVar (..., dZdtheta) </div> <p><i>Required Parameters for varargin:</i></p> <ul style="list-style-type: none"> • dZdtheta (n_dim + n_dim(n_dim+1)/2) x n_theta x n_t
Generated by Doxygen	

Return values

<i>Zout</i>	(n_dim + n_dim(n_dim+1)/2) x n_t vector with mean and Variance
<i>varargout</i>	
<i>dZdthetaout</i>	

Definition at line 17 of file getLognMeanVar.m.

Referenced by logLikelihood(), and plotODEMix().

4.17 getRREsigmas.m File Reference

This function defines the parameters needed for the parametrization of the variances in case Reaction Rate Equations are used for the mechanistic description of the means.

Functions

- `mlhsSubst< mlhsInnerSubst< matlabtypesubstitute, parameters >,mlhsInnerSubst< matlabtypesubstitute, conditions >,mlhsInnerSubst< matlabtypesubstitute, varargout > > getRREsigmas (matlabtypesubstitute parameters, matlabtypesubstitute conditions, matlabtypesubstitute varargin)`

This function defines the parameters needed for the parametrization of the variances in case Reaction Rate Equations are used for the mechanistic description of the means.

4.17.1 Detailed Description

This function defines the parameters needed for the parametrization of the variances in case Reaction Rate Equations are used for the mechanistic description of the means.

4.17.2 Function Documentation

- #### 4.17.2.1
- `mlhsSubst< mlhsInnerSubst< matlabtypesubstitute, parameters >,mlhsInnerSubst< matlabtypesubstitute, conditions >,mlhsInnerSubst< matlabtypesubstitute, varargout > > getRREsigmas (matlabtypesubstitute parameters, matlabtypesubstitute conditions, matlabtypesubstitute varargin)`

This function defines the parameters needed for the parametrization of the variances in case Reaction Rate Equations are used for the mechanistic description of the means.

USAGE

```
[parameters,conditions] = getRREsigmas(parameters,conditions)
[parameters,conditions,D] = getRREsigmas(parameters,conditions,options,D,M)
```

Parameters

<i>parameters</i>	parameters struct
<i>conditions</i>	conditions struct (see ...)
<i>varargin</i>	<pre>1 getRREsigmas (..., options, D, M)</pre>
	<p><i>Required Parameters for varargin:</i></p> <ul style="list-style-type: none"> • options • D data struct (see logLikelihood.m)

Return values

<i>parameters</i>	updated parameters struct
<i>conditions</i>	updated conditions struct
<i>varargout</i>	
<i>D</i>	updated data struct

Required fields of parameters:

Optional fields of options:

- `sigmas` --
 = condition-dependent: (default) assign sigma for every time point
 = time-dependent: one sigma for every subpopulation and time point
 = only-one: only one sigma for everything
 = subpopulation-specific: for every subpopulation one sigma
 boundaries
 - `.min`
 - `.max`

Generated fields of parameters:

- `names` -- names for sigma parameters are added

Generated fields of D:

- `sigma` -- (if `n_dim = 1`)
- `Sigma` -- (if `n_dim = 2`)

Definition at line 17 of file `getRREsigmas.m`.

4.18 getScalingFactors.m File Reference

Calculates scaling factors for replicates such that the log of the means are as similar as possible.

Functions

- `mlhsInnerSubst < matlabtypesubstitute, s > getScalingFactors` (matlabtypesubstitute varargin)
Calculates scaling factors for replicates such that the log of the means are as similar as possible.

4.18.1 Detailed Description

Calculates scaling factors for replicates such that the log of the means are as similar as possible.

4.18.2 Function Documentation

4.18.2.1 `mlhsInnerSubst` < `matlabtypesubstitute`, `s` > `getScalingFactors` (`matlabtypesubstitute` *varargin*)

Calculates scaling factors for replicates such that the log of the means are as similar as possible.

USAGE

```
s = getScalingFactors(ExpC_1, ExpC_2)
```

Required fields of ExpC name: string specifying the conditions .time: time point of measurement .stimulus: stimulus for measurement .replicate(j) .name ... string specifying the replicate/replicate .measurands ... {name of measurand 1,name of measurand 2,...,name of measurand m} .data {n_D1 x m matrix under condition 1, n_D2 x m matrix under condition 2,...,n_Dnc x m matrix under condition n_c} (One row represents one observed cell with the data in the order of the measurands. The different rows provide measurement data for different cells.)

Parameters

<i>varargin</i>	ExpC
-----------------	------

Return values

s	1 x n_r vector including scaling factor for every replicate
---	-------------------------------------------------------------

Definition at line 17 of file `getScalingFactors.m`.

4.19 `load_plot_settings.m` File Reference

Functions to set font sizes and colors for the visualization.

Functions

- `noret::substitute` [load_plot_settings](#) ()
Functions to set font sizes and colors for the visualization.

4.19.1 Detailed Description

Functions to set font sizes and colors for the visualization.

4.20 `logLikelihood.m` File Reference

This function evaluates the likelihood function for a given model, data and parameter vector.

Functions

- mlhsInnerSubst< matlabtypesubstitute, varargout > [logLikelihood](#) (matlabtypesubstitute xi, matlabtypesubstitute M, matlabtypesubstitute D, matlabtypesubstitute varargin)

This function evaluates the likelihood function for a given model, data and parameter vector.

4.20.1 Detailed Description

This function evaluates the likelihood function for a given model, data and parameter vector.

4.20.2 Function Documentation

- 4.20.2.1 mlhsInnerSubst< matlabtypesubstitute, varargout > [logLikelihood](#) (matlabtypesubstitute *xi*, matlabtypesubstitute *M*, matlabtypesubstitute *D*, matlabtypesubstitute *varargin*)

This function evaluates the likelihood function for a given model, data and parameter vector.

USAGE

```
[...] = logLikelihood(xi,M,D,options,conditions,l)
[...] = logLikelihood(xi,M,D,options,conditions)
[...] = logLikelihood(xi,M,D,options)
[logL] = logLikelihood(...)
[logL, dlogL] = logLikelihood(...)
```

Parameters

<i>xi</i>	parameter values
<i>M</i>	model struct
<i>D</i>	data struct
<i>varargin</i>	<ul style="list-style-type: none"> • options: options struct • conditions: generated by function collectConditions.m • l: indices for which of the data the likelihood function should be evaluated

Return values

<i>logL</i>	log-likelihood value
<i>dlogL</i>	gradient of log-likelihood function

Required fields of M:

- *n_subpop* -- number of subpopulations
- *model* -- simulation file with input (T,theta,u) (e.g. generated by [amiwrap](#)), the first output needs to be the status of the simulation, the 4th the simulation output and the 6th the sensitivities (n_t x n_obs x n_theta)

- `mean_ind` -- indices of output for mean
- `var_ind` -- indices of output for variances (empty if using RREs)
- `theta` -- parameters needed for simulation depend on `xi` and `u` the following fields of `M` are generated by `generate_ODEMM`
- `distribution{s,e}` — distribution assumption
 = 'norm': normal distribution assumption
 = 'logn_median': lognormal distribution assumption, mean of simulation linked to median of distribution
 = 'logn_mean': lognormal distribution assumption, mean of simulation linked to mean of distribution
 The following fields are automatically added by [generateODEMM.m](#)
- `dthetadxi` -- gradient of `theta`
- `mu{s,e}` — specification of mixture parameter `mu` for subpopulation `s` and experiment `e`
- `dmudxi{s,e}` — gradient of `mu`
- `sigma{s,e}` — specification of mixture parameter `sigma` (`M.Sigma` in multivariate case (covariance matrix))
- `dsigmadx{i}{s,e}` — gradient of `sigma` (`M.dSigma` in multivariate case)
- `w{s,e}` — specification of weights
- `dwdx{i}{s,e}` — gradient of weights
- `scaling{r,e}` — scaling parameter of replicate `r` in experiment `e`
- `dscalingdxi{r,e}` — gradient of scaling
- `offset{r,e}` — offset
- `doffsetdxi{r,e}` — gradient of offset

Required fields of `D`:

- `n_dim` -- dimension of the measurements (if `n_dim > 2`, toolbox needs to be extended at some points!!)
- `t` -- 1 x `n_t` vector of timepoints
- `u` -- `n_maxu` x `n_u` vector of inputs with `n_maxu`: maximal number of inputs simulatenously used
- `y` -- `n_u` x `n_t` x `n_cells` x `n_dim` data matrix (only needed if replicates are merged and already scaled), `dim` is the dimension of the measurement
- `c` -- `n_subpop` x (`n_u` + `n_differences`) corresponding condition (automatically added by calling `collectCondition.m`)
- `replicate{r}.y` — `n_u` x `n_t` x `n_cells` x `n_dim` data matrix of replicate `r` in experiment `e` (only needed if individual replicates should be fitted)

Optional fields of options:

- `use_robust` -- robust calculation of mixture probability
 = `true`: uses reformulation (default)
 = `false`: classical calculation (not recommended)
- `simulate_musigma` -- true if simulation directly provides ...
- `negLogLikelihood` -- true if negative log-likelihood required
- `replicates` -- true if replicates are fitted individually

Definition at line 17 of file `logLikelihood.m`.

References `collectConditions()`, `computeMixtureProbability()`, `getLognMeanVar()`, `logoflognpdf()`, `logofmvnpdf()`, and `logofnormpdf()`.

4.21 plotODEMix.m File Reference

Routine to plot the ODE-constrained mixture model.

Functions

- `mlhsInnerSubst< matlabtypesubstitute, varargin > plotODEMix` (matlabtypesubstitute varargin)
Routine to plot the ODE-constrained mixture model.
- `mlhsInnerSubst< matlabtypesubstitute, str_dose > mtoc_subst_plotODEMix_m_tsbus_cotm_getStr↵
Dose` (matlabtypesubstitute D, matlabtypesubstitute e, matlabtypesubstitute d)
- `noret::substitute mtoc_subst_plotODEMix_m_tsbus_cotm_evalModel` (matlabtypesubstitute xi, matlabtypesubstitute M, matlabtypesubstitute D, matlabtypesubstitute e, matlabtypesubstitute r, matlabtypesubstitute d, matlabtypesubstitute X_c, matlabtypesubstitute options, matlabtypesubstitute conditions)
- `mlhsSubst< mlhsInnerSubst< matlabtypesubstitute, lim >,mlhsInnerSubst< matlabtypesubstitute, hist↵
>,mlhsInnerSubst< matlabtypesubstitute, grids > > mtoc_subst_plotODEMix_m_tsbus_cotm_set↵
YminmaxHists` (matlabtypesubstitute D, matlabtypesubstitute e, matlabtypesubstitute d, matlabtypesubstitute options, matlabtypesubstitute ind)
- `noret::substitute mtoc_subst_plotODEMix_m_tsbus_cotm_evalPdf` (matlabtypesubstitute M, matlabtypesubstitute D, matlabtypesubstitute e, matlabtypesubstitute d, matlabtypesubstitute k, matlabtypesubstitute options, matlabtypesubstitute legendflag, matlabtypesubstitute ind, matlabtypesubstitute lim, matlabtypesubstitute hist↵
s, matlabtypesubstitute grids, matlabtypesubstitute plotModel, matlabtypesubstitute plotData)

4.21.1 Detailed Description

Routine to plot the ODE-constrained mixture model.

4.21.2 Function Documentation

4.21.2.1 `mlhsInnerSubst< matlabtypesubstitute, varargin > plotODEMix (matlabtypesubstitute varargin)`

Routine to plot the ODE-constrained mixture model.

USAGE

```
[...] = plotODEMix(D,M,xi)
[...] = plotODEMix(D,M,xi,l,options)
[...] = plotODEMix(D,M,xi,l,options,tu_ind)
[fh] = plotODEMix(...)
[fh,fhm] = plotODEMix(...)
```

Parameters

<i>varargin</i>	
-----------------	--

Definition at line 17 of file plotODEMix.m.

References `collectConditions()`, and `getLognMeanVar()`.

4.22 printParams.m File Reference

Help function to print parameters names and values.

Functions

- `noret::substitute printParams` (matlabtypesubstitute parameters, matlabtypesubstitute varargin)
Help function to print parameters names and values.

4.22.1 Detailed Description

Help function to print parameters names and values.

4.22.2 Function Documentation

4.22.2.1 `noret::substitute printParams (matlabtypesubstitute parameters, matlabtypesubstitute varargin)`

Help function to print parameters names and values.

USAGE

`[] = printParams(parameters,xi)`

Parameters

<i>parameters</i>	parameters struct
<i>varargin</i>	<pre>1 printParams (..., xi)</pre> <p><i>Required Parameters for varargin:</i></p> <ul style="list-style-type: none"> • xi parameter values printed along with parameter name

Required fields of parameters:

- `parameters -- name = {name1,...}` names of parameters

Definition at line 17 of file printParams.m.

4.23 testSigmaPointApp_mod.m File Reference

Modified version of the testSigmaPointApp_status.m function of the SPToolbox.

Functions

- `mlhsInnerSubst< matlabtypesubstitute, SP > testSigmaPointApp_mod` (matlabtypesubstitute varargin)
Modified version of the testSigmaPointApp_status.m function of the SPToolbox.

4.23.1 Detailed Description

Modified version of the testSigmaPointApp_status.m function of the SPToolbox.

4.23.2 Function Documentation

4.23.2.1 mlhsInnerSubst< matlabtypesubstitute, SP > testSigmaPointApp_mod (matlabtypesubstitute *varargin*)

Modified version of the testSigmaPointApp_status.m function of the SPToolbox.

Generated fields of SP:

Definition at line 17 of file testSigmaPointApp_mod.m.

Index

collectConditions
 collectConditions.m, 6
collectConditions.m, 5
 collectConditions, 6
computeMixtureProbability
 computeMixtureProbability.m, 8
computeMixtureProbability.m, 7
 computeMixtureProbability, 8

distributions/logn/func_Sigma_logn.m, 13
distributions/logn/func_dSigmadxi_logn.m, 12
distributions/logn/func_dmudxi_logn_mean.m, 9
distributions/logn/func_dmudxi_logn_median.m, 10
distributions/logn/func_dsigma2dxi_logn.m, 11
distributions/logn/logoflognpdf.m, 14
distributions/logn/logofmvnpdf.m, 14
distributions/norm/func_Sigma_norm.m, 18
distributions/norm/func_dSigmadxi_norm.m, 17
distributions/norm/func_dmudxi_norm.m, 15
distributions/norm/func_dsigma2dxi_norm.m, 16
distributions/norm/logofnormpdf.m, 19

func_Sigma_logn
 func_Sigma_logn.m, 13
func_Sigma_logn.m
 func_Sigma_logn, 13
func_Sigma_norm
 func_Sigma_norm.m, 18
func_Sigma_norm.m
 func_Sigma_norm, 18
func_dSigmadxi_logn
 func_dSigmadxi_logn.m, 12
func_dSigmadxi_logn.m
 func_dSigmadxi_logn, 12
func_dSigmadxi_norm
 func_dSigmadxi_norm.m, 18
func_dSigmadxi_norm.m
 func_dSigmadxi_norm, 18
func_dmudxi_logn_mean
 func_dmudxi_logn_mean.m, 9
func_dmudxi_logn_mean.m
 func_dmudxi_logn_mean, 9
func_dmudxi_logn_median
 func_dmudxi_logn_median.m, 10
func_dmudxi_logn_median.m
 func_dmudxi_logn_median, 10
func_dmudxi_norm
 func_dmudxi_norm.m, 16
func_dmudxi_norm.m
 func_dmudxi_norm, 16
func_dsigma2dxi_logn
 func_dsigma2dxi_logn.m, 11
func_dsigma2dxi_logn.m
 func_dsigma2dxi_logn, 11
func_dsigma2dxi_norm
 func_dsigma2dxi_norm.m, 17

func_dsigma2dxi_norm.m
 func_dsigma2dxi_norm, 17

generateODEMM.m, 20
 generateODEMM, 20
generateODEMM
 generateODEMM.m, 20
getLognMeanVar
 getLognMeanVar.m, 21
getLognMeanVar.m, 21
 getLognMeanVar, 21
getRREsigmas
 getRREsigmas.m, 22
getRREsigmas.m, 22
 getRREsigmas, 22
getScalingFactors
 getScalingFactors.m, 24
getScalingFactors.m, 23
 getScalingFactors, 24

load_plot_settings.m, 24
logLikelihood
 logLikelihood.m, 25
logLikelihood.m, 24
 logLikelihood, 25
logoflognpdf
 logoflognpdf.m, 14
logoflognpdf.m
 logoflognpdf, 14
logofmvnpdf
 logofmvnpdf.m, 15
logofmvnpdf.m
 logofmvnpdf, 15
logofnormpdf
 logofnormpdf.m, 19
logofnormpdf.m
 logofnormpdf, 19

plotODEMix
 plotODEMix.m, 27
plotODEMix.m, 27
 plotODEMix, 27
printParams
 printParams.m, 28
printParams.m, 28
 printParams, 28

testSigmaPointApp_mod
 testSigmaPointApp_mod.m, 29
testSigmaPointApp_mod.m, 28
 testSigmaPointApp_mod, 29