

ODE-constrained mixture modeling

1.0

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	Conversion Reaction	3
3	Differential protein expression (one stage)	4
4	Influence of extracellular scaffolds on pain signaling	4
5	Subpopulation differences of sensory neurons	4
6	File Index	4
6.1	File List	4
7	File Documentation	10
7.1	collectConditions.m File Reference	10
7.1.1	Detailed Description	11
7.1.2	Function Documentation	11
7.2	computeMixtureProbability.m File Reference	12
7.2.1	Detailed Description	12
7.2.2	Function Documentation	12
7.3	distributions/logn/func_dmudxi_logn_mean.m File Reference	13
7.3.1	Detailed Description	13
7.3.2	Function Documentation	13
7.4	distributions/logn/func_dmudxi_logn_median.m File Reference	14

7.4.1	Detailed Description	14
7.4.2	Function Documentation	14
7.5	distributions/logn/func_dsigma2dxi_logn.m File Reference	15
7.5.1	Detailed Description	15
7.5.2	Function Documentation	15
7.6	distributions/logn/func_dSigmadxi_logn.m File Reference	16
7.6.1	Detailed Description	16
7.6.2	Function Documentation	16
7.7	distributions/logn/func_Sigma_logn.m File Reference	17
7.7.1	Detailed Description	17
7.7.2	Function Documentation	17
7.8	distributions/logn/logoflognpdf.m File Reference	17
7.8.1	Detailed Description	17
7.8.2	Function Documentation	18
7.9	distributions/logn/logofmvnpdf.m File Reference	18
7.9.1	Detailed Description	18
7.10	distributions/norm/func_dmudxi_norm.m File Reference	18
7.10.1	Detailed Description	18
7.10.2	Function Documentation	18
7.11	distributions/norm/func_dsigma2dxi_norm.m File Reference	19
7.11.1	Detailed Description	19
7.11.2	Function Documentation	19
7.12	distributions/norm/func_dSigmadxi_norm.m File Reference	20
7.12.1	Detailed Description	20
7.12.2	Function Documentation	20
7.13	distributions/norm/func_Sigma_norm.m File Reference	21
7.13.1	Detailed Description	21
7.13.2	Function Documentation	21
7.14	distributions/norm/logofnormpdf.m File Reference	21
7.14.1	Detailed Description	22

7.14.2	Function Documentation	22
7.15	examples/conversion_reaction/data/generate_data_cr.m File Reference	22
7.15.1	Detailed Description	22
7.16	examples/conversion_reaction/main_optimization_cr.m File Reference	23
7.16.1	Detailed Description	23
7.17	examples/conversion_reaction/main_sampling_cr.m File Reference	23
7.17.1	Detailed Description	23
7.18	examples/conversion_reaction/main_singlecell_prediction.m File Reference	23
7.18.1	Detailed Description	24
7.18.2	Function Documentation	24
7.19	examples/subpopulation_differences/main_singlecell_prediction.m File Reference	24
7.19.1	Detailed Description	24
7.20	examples/conversion_reaction/model_SP_all.m File Reference	24
7.20.1	Detailed Description	25
7.21	examples/conversion_reaction/model_SP_k1.m File Reference	25
7.21.1	Detailed Description	25
7.22	examples/conversion_reaction/model_SP_k1k2.m File Reference	25
7.22.1	Detailed Description	26
7.23	examples/conversion_reaction/model_SP_k1k3.m File Reference	26
7.23.1	Detailed Description	26
7.24	examples/conversion_reaction/model_SP_k2.m File Reference	26
7.24.1	Detailed Description	27
7.25	examples/conversion_reaction/model_SP_k2k3.m File Reference	27
7.25.1	Detailed Description	27
7.26	examples/conversion_reaction/model_SP_k3.m File Reference	27
7.26.1	Detailed Description	28
7.27	examples/conversion_reaction/models_RRE.m File Reference	28
7.27.1	Detailed Description	28
7.28	examples/conversion_reaction/plot_cr.m File Reference	28
7.28.1	Detailed Description	28

7.29	examples/conversion_reaction/plot_cr_SIAppendix.m File Reference	28
7.29.1	Detailed Description	29
7.30	examples/conversion_reaction/plot_cr_variabilityReduction.m File Reference	29
7.30.1	Detailed Description	29
7.30.2	Function Documentation	29
7.31	examples/conversion_reaction/run_lppd_cr.m File Reference	29
7.31.1	Detailed Description	30
7.31.2	Function Documentation	30
7.32	examples/conversion_reaction/run_thermIntegration_cr.m File Reference	30
7.32.1	Detailed Description	30
7.32.2	Function Documentation	31
7.33	examples/conversion_reaction/sclogLikelihood_cr.m File Reference	31
7.33.1	Detailed Description	31
7.33.2	Function Documentation	32
7.34	examples/conversion_reaction/simulation/CR_log_syms.m File Reference	32
7.34.1	Detailed Description	33
7.34.2	Function Documentation	33
7.35	examples/conversion_reaction/simulation/CR_syms.m File Reference	33
7.35.1	Detailed Description	33
7.35.2	Function Documentation	33
7.36	examples/conversion_reaction/simulation/generate_simFiles_cr.m File Reference	34
7.36.1	Detailed Description	34
7.37	examples/differential_protein_expression/data/generate_data_oneStage.m File Reference	34
7.37.1	Detailed Description	34
7.38	examples/differential_protein_expression/main_oneStage_SP_1D.m File Reference	34
7.38.1	Detailed Description	35
7.39	examples/differential_protein_expression/main_oneStage_SP_2D.m File Reference	35
7.39.1	Detailed Description	35
7.40	examples/differential_protein_expression/plot_fit_oneStage.m File Reference	35
7.40.1	Detailed Description	35

7.41	examples/differential_protein_expression/plot_uncertainty_oneStage.m File Reference	36
7.41.1	Detailed Description	36
7.42	examples/differential_protein_expression/simulation/generate_simFile_oneStage.m File Reference	36
7.42.1	Detailed Description	36
7.43	examples/differential_protein_expression/simulation/oneStage_syms.m File Reference	36
7.43.1	Detailed Description	36
7.43.2	Function Documentation	36
7.44	examples/ECM_differences/data/load_datasets_PDL_Coll.m File Reference	37
7.44.1	Detailed Description	37
7.45	examples/ECM_differences/generate_ECM_models.m File Reference	37
7.45.1	Detailed Description	37
7.46	examples/ECM_differences/main_ECM_differences.m File Reference	37
7.46.1	Detailed Description	38
7.47	examples/ECM_differences/plot_ECM_fit.m File Reference	38
7.47.1	Detailed Description	38
7.48	examples/ECM_differences/plot_ECM_ranks.m File Reference	38
7.48.1	Detailed Description	38
7.49	examples/ECM_differences/plot_ECM_valid.m File Reference	38
7.49.1	Detailed Description	39
7.50	examples/ECM_differences/run_fittings_ECM.m File Reference	39
7.50.1	Detailed Description	39
7.50.2	Function Documentation	39
7.51	examples/ECM_differences/run_profile_ECM.m File Reference	39
7.51.1	Detailed Description	40
7.52	examples/subpopulation_differences/dephosphorylation/exp_decay.m File Reference	40
7.52.1	Detailed Description	40
7.53	examples/subpopulation_differences/dephosphorylation/import_dephospho_data.m File Reference	40
7.53.1	Detailed Description	40
7.54	examples/subpopulation_differences/dephosphorylation/main_analysis_dephospho.m File Reference	40
7.54.1	Detailed Description	41

7.55	examples/subpopulation_differences/dephosphorylation/plot_subpop_dephoshp_supplement.m	
	File Reference	41
7.55.1	Detailed Description	41
7.56	examples/subpopulation_differences/generate_nosubpop_file.m	File Reference 41
7.56.1	Detailed Description	41
7.57	examples/subpopulation_differences/generate_subpop_files.m	File Reference 41
7.57.1	Detailed Description	41
7.58	examples/subpopulation_differences/generate_subpop_TrkA.m	File Reference 42
7.58.1	Detailed Description	42
7.59	examples/subpopulation_differences/main_subpopulation_analysis.m	File Reference 42
7.59.1	Detailed Description	42
7.60	examples/subpopulation_differences/plot_subpop_fit.m	File Reference 42
7.60.1	Detailed Description	42
7.61	examples/subpopulation_differences/plot_subpop_profiles.m	File Reference 42
7.61.1	Detailed Description	43
7.62	examples/subpopulation_differences/plot_subpop_valid.m	File Reference 43
7.62.1	Detailed Description	43
7.63	examples/subpopulation_differences/run_fitting_subpop.m	File Reference 43
7.63.1	Detailed Description	43
7.63.2	Function Documentation	44
7.64	examples/subpopulation_differences/run_logmarg_subpop.m	File Reference 44
7.64.1	Detailed Description	44
7.65	examples/subpopulation_differences/run_lppd_subpop.m	File Reference 44
7.65.1	Detailed Description	44
7.66	examples/subpopulation_differences/simulation/generate_simFile_ErkSignaling.m	File Reference . 45
7.66.1	Detailed Description	45
7.67	examples/subpopulation_differences/simulation/ODEmodel_syms_sPsET_loglog.m	File Reference 45
7.67.1	Detailed Description	45
7.67.2	Function Documentation	45
7.68	examples/two_stage_exandintrinsic/data/generate_data_twostage.m	File Reference 45
7.68.1	Detailed Description	46

7.69	examples/two_stage_exandintrinsic/main_MA.m File Reference	46
7.69.1	Detailed Description	46
7.70	examples/two_stage_exandintrinsic/main_MA_1subpop.m File Reference	46
7.70.1	Detailed Description	46
7.71	examples/two_stage_exandintrinsic/main_MA_1subpop_extrinsic.m File Reference	47
7.71.1	Detailed Description	47
7.72	examples/two_stage_exandintrinsic/main_RRE.m File Reference	47
7.72.1	Detailed Description	47
7.73	examples/two_stage_exandintrinsic/simulation/compile_simFiles_geneExp.m File Reference	47
7.73.1	Detailed Description	48
7.74	examples/two_stage_exandintrinsic/simulation/MEC_2_LD_2_c_geneExp_extrinsic_syms.m File Reference	48
7.74.1	Detailed Description	48
7.74.2	Function Documentation	48
7.75	examples/two_stage_exandintrinsic/simulation/MEC_2_LD_2_c_geneExp_MA_syms.m File Reference	48
7.75.1	Detailed Description	49
7.75.2	Function Documentation	49
7.76	examples/two_stage_exandintrinsic/simulation/modelDef_geneExp.m File Reference	49
7.76.1	Detailed Description	49
7.77	examples/two_stage_exandintrinsic/simulation/modelDef_geneExp_extrinsic.m File Reference	49
7.77.1	Detailed Description	50
7.78	examples/two_stage_exandintrinsic/simulation/RRE_geneExp_RRE_syms.m File Reference	50
7.78.1	Detailed Description	50
7.78.2	Function Documentation	50
7.79	generateODEMM.m File Reference	50
7.79.1	Detailed Description	51
7.79.2	Function Documentation	51
7.80	getLognMeanVar.m File Reference	52
7.80.1	Detailed Description	53
7.80.2	Function Documentation	53

7.81	getRREsigmas.m File Reference	53
7.81.1	Detailed Description	54
7.81.2	Function Documentation	54
7.82	getScalingFactors.m File Reference	55
7.82.1	Detailed Description	55
7.82.2	Function Documentation	55
7.83	getSigmaPointApp_status_mod.m File Reference	56
7.83.1	Detailed Description	56
7.83.2	Function Documentation	57
7.84	install_ODEMM.m File Reference	57
7.84.1	Detailed Description	57
7.85	load_plot_settings.m File Reference	57
7.85.1	Detailed Description	57
7.86	logLikelihood.m File Reference	57
7.86.1	Detailed Description	58
7.86.2	Function Documentation	58
7.87	plotODEMM.m File Reference	60
7.87.1	Detailed Description	60
7.87.2	Function Documentation	60
7.88	printParams.m File Reference	61
7.88.1	Detailed Description	61
7.88.2	Function Documentation	61
7.89	testSigmaPointApp_mod.m File Reference	62
7.89.1	Detailed Description	62
7.89.2	Function Documentation	62

1 ODEMM Documentation

1.1 Introduction

Cellular heterogeneity occurs at multiple levels. To cover these 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., CMSB (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.

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>
- PESTO (parameter estimation): <https://github.com/ICB-DCM/PESTO>
- SPToolbox (sigma-point approximation): <https://github.com/ICB-DCM/SPToolbox> (required for the examples incorporating means and covariances)
- CERENA (moment approximation): <https://cerenadevelopers.github.io/CERENA/> (required for the example two_stage_exandintrinsic)

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_exandintrinsic`)

1.6 Distributions

For the mixture distribution, ODEMM implements

- multivariate normal distributions and
- multivariate log-normal distributions.

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

2 Conversion Reaction

This example reproduces the results of the analysis of "Unraveling sources of heterogeneity" of Loos et al., Cell Systems (2018). In this example, models incorporating only the **mean** (`model_RRE.m`), and models incorporating **mean and variance** (e.g., `model_SP_k3.m`) are implemented. The latter allow for cell-to-cell variability of certain parameters of the model and incorporate the sigma-point approximation to obtain the statistical properties of the individual subpopulations.

First, the simulation file needs to be compiled using the MATLAB Toolbox AMICI. This can be done by running `./simulation/generate_simFiles_cr.m`.

All models for this example are optimized within the function `main_optimization_cr.m`.

The Bayesian model selection is performed in `main_sampling_cr.m`.

The prediction of single-cell trajectories is performed in `main_singlecell_prediction.m`.

To reproduce the figures of the paper, `run_plot_cr.m`.

3 Differential protein expression (one stage)

This example reproduces the results of the analysis of "Identification of differential protein expression using multivariate data" of Loos et al., Cell Systems (2018).

First, the simulation file needs to be compiled using the MATLAB Toolbox AMICI. This can be done by running `./simulation/generate_simFile_oneStage.m`.

The script `main_oneStage_SP_1D.m` reproduces the results for model calibration based on the marginal distributions. The script `main_oneStage_SP_2D.m` reproduces the results for model calibration based on the marginal distributions.

The figures of the paper can be reproduced with the scripts `plot_fit_oneStage.m` and `plot_fit_uncertainty.m`.

4 Influence of extracellular scaffolds on pain signaling

This example reproduces the results of the analysis of the "influence of extracellular scaffolds on sensitization signaling" of Loos et al., Cell Systems (2018).

First the simulation file needs to be compiled using the MATLAB Toolbox AMICI. This can be done by running `../subpopulation_differences/simulation/generate_simFile_ErkSignaling.m`.

The main file for the analysis is `main_ECM_differences.m`.

5 Subpopulation differences of sensory neurons

This example reproduces the results of the analysis of "causal differences between subpopulations of cultured sensory neurons" of Loos et al., Cell Systems (2018).

First the simulation file needs to be compiled using the MATLAB Toolbox AMICI. This can be done by running `./simulation/generate_simFile_ErkSignaling.m`.

The main file for the analysis is `main_subpopulation_analysis.m`.

6 File Index

6.1 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

10

`computeMixtureProbability.m`

Robust calculation of a mixture distribution likelihood

12

generateODEMM.m	This function generates a file that defines the ODE-constrained mixture model	50
getLognMeanVar.m	This function calculations the mean and variances, and the corresponding derivatives of a multivariate log-normal distribution given the parameters μ and Σ	52
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	53
getScalingFactors.m	Calculates scaling factors for replicates such that the distance between the means in log-space are minimal	55
getSigmaPointApp_status_mod.m	Modified version of the getSigmaPointApp.m function of the SPToolbox	56
install_ODEMM.m	Script that adds the required paths to the MATLAB search path	57
load_plot_settings.m	Functions to set font sizes and colors for the visualization	57
logLikelihood.m	This function evaluates the likelihood function for a given model, data and parameter vector	57
plotODEMM.m	Routine to plot the ODE-constrained mixture model	60
printParams.m	Help function to print parameters names and values	61
testSigmaPointApp_mod.m	Modified version of the testSigmaPointApp_status.m function of the SPToolbox	62
distributions/logn/func_dmudxi_logn_mean.m	This function calculates the derivative of μ of the (multivariate) log-normal distribution for the case of linking the mean of the observables to the mean of the (multivariate) log-normal distribution	13
distributions/logn/func_dmudxi_logn_median.m	This function calculates the derivative of μ of the (multivariate) log-normal distribution for the case of linking the mean of the observables to the median of the (multivariate) log-normal distribution. This should only be used for the case of using only a mechanistic description of the means of the observables	14
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	15
distributions/logn/func_dSigmadxi_logn.m	This function maps the means and covariances of the observables for a subpopulation to Σ of the multivariate log-normal distribution	16
distributions/logn/func_Sigma_logn.m	This function maps the means and covariances to Σ of a (multivariate) log-normal distribution	17
distributions/logn/logoflognpdf.m	Modified version of MATLAB function LOGNPDF such that the log-density is returned	17

distributions/logn/ logofmvnpdf.m	Modified version of MATLAB function MVNPDF such that the log-density is returned	18
distributions/norm/ func_dmudxi_norm.m	This function calculates the derivative of μ of the (multivariate) normal distribution	18
distributions/norm/ func_dsigma2dxi_norm.m	This function calculates the derivative of σ^2 in case of univariate measurements and a normal distribution assumption	19
distributions/norm/ func_dSigmadxi_norm.m	This function maps the means and variances to Σ of the multivariate normal distribution	20
distributions/norm/ func_Sigma_norm.m	This function maps the means and variances to Σ of the multivariate normal distribution	21
distributions/norm/ logofnormpdf.m	Modified version of MATLAB function NORMPDF such that the log-density is given back	21
examples/conversion_reaction/ main_optimization_cr.m	This is the main script for the first example, the conversion reaction. All Reaction Rate Equation (RRE) models are generated and their parameters are estimated by calling models_RRE.m . All SP models are estimated, for which e.g. model_SP_k1.m indicates that k1 has an additional cell-to-cell variability modeled by the Sigma point approximation	23
examples/conversion_reaction/ main_sampling_cr.m	This is the main script for the Bayesian model selection for the conversion reaction	23
examples/conversion_reaction/ main_singlecell_prediction.m	This script shows the analysis of the single-cell trajectory prediction. First the single-cell parameter based on one single-cell trajectory are sampled and the predicted results are visualized. Then the parameters are optimized for 100 single-cell trajectories. This analysis corresponds to Loos et al., Cell Systems (2018), Figure 3E-G	23
examples/conversion_reaction/ model_SP_all.m	This function generates model and estimates parameters for the model accounting for means and variances (using Sigma Points (SP)). All kinetic parameters are assumed to vary between individual cells: k1: inter- and intra-subpopulation variable k2: cell-to-cell variable k3: cell-to-cell variable	24
examples/conversion_reaction/ model_SP_k1.m	This function generates the model and estimates the parameters in case of accounting for means and variances (using Sigma Points (SP)) The parameter are assumed to be: k1: inter- and intra-subpopulation variable k2: homogeneous k3: homogeneous	25
examples/conversion_reaction/ model_SP_k1k2.m	This function generates the model and estimates the parameters in case of accounting for means and variances (using Sigma Points (SP)) The parameter are assumed to be: k1: inter- and intra-subpopulation variable k2: cell-to-cell variable k3: homogeneous	25

examples/conversion_reaction/model_SP_k1k3.m

This function generates the model and estimates the parameters in case of accounting for means and variances (using Sigma Points (SP)). The parameter are assumed to be:

k1: inter- and intra-subpopulation variable

k2: homogeneous

k3: cell-to-cell variable

26

examples/conversion_reaction/model_SP_k2.m

This function generates the model and estimates the parameters in case of accounting for means and variances (using Sigma Points (SP)). The parameter are assumed to be:

k1: subpopulation variable

k2: cell-to-cell variable

k3: homogeneous

26

examples/conversion_reaction/model_SP_k2k3.m

This function generates the model and estimates the parameters in case of accounting for means and variances (using Sigma Points (SP)). The parameter are assumed to be:

k1: subpopulation variable

k2: cell-to-cell variable

k3: cell-to-cell variable

27

examples/conversion_reaction/model_SP_k3.m

This function generates the model and estimates the parameters in case of accounting for means and variances (using Sigma Points (SP)). The parameter are assumed to be:

k1: subpopulation variable

k2: homogenous

k3: cell-to-cell variable

This model is the ground truth

27

examples/conversion_reaction/models_RRE.m

This function generates Reaction Rate Equation model and estimates the parameters for the models with varying number of parameters for the variances

28

examples/conversion_reaction/plot_cr.m

This script visualizes the results for the conversion reaction

28

examples/conversion_reaction/plot_cr_SIAppendix.m

This script generates the figures for the supplement for the conversion reaction example

28

examples/conversion_reaction/plot_cr_variabilityReduction.m

Clear all close all clc load_plot_settings

29

examples/conversion_reaction/run_lppd_cr.m

This function runs the sampling for the log pointwise predictive density for the conversion reaction reaction example

29

examples/conversion_reaction/run_thermIntegration_cr.m

This function runs thermodynamic integration for the conversion reaction example

30

examples/conversion_reaction/sclogLikelihood_cr.m

This function provides the likelihood for the optimization of the single cell trajectory in the conversion reaction example

31

examples/conversion_reaction/data/generate_data_cr.m

This script generates the artificial data of a conversion process

22

examples/conversion_reaction/simulation/CR_log_syms.m

This function defines the model of the conversion process with the logarithm of the output, which are used for the sigma point approximation

32

examples/conversion_reaction/simulation/ CR_syms.m	
This function defines the model of the conversion process with the linear output, which is used for the Reaction Rate Equations	33
examples/conversion_reaction/simulation/ generate_simFiles_cr.m	
This script generates the functions for the conversion reaction using the toolbox AMICI	34
examples/differential_protein_expression/ main_oneStage_SP_1D.m	
This script generates the model for the 1-dimensional measurements of the differential gene expression example using the sigma point approximation. The parameters are estimated and also the profile likelihoods are calculated using the toolbox PESTO	34
examples/differential_protein_expression/ main_oneStage_SP_2D.m	
This script generates the model for the 2-dimensional measurements of the differential gene expression example using the sigma point approximation. The parameters are estimated and also the profile likelihoods are calculated using the toolbox PESTO	35
examples/differential_protein_expression/ plot_fit_oneStage.m	
This script visualizes the model fit for the two-dimensional data	35
examples/differential_protein_expression/ plot_uncertainty_oneStage.m	
This script visualizes the confidence intervals for the one- and the two-dimensional models	36
examples/differential_protein_expression/data/ generate_data_oneStage.m	
This script generates the artificial data of the differential gene protein expression	34
examples/differential_protein_expression/simulation/ generate_simFile_oneStage.m	
Compilation of one stage simulation file	36
examples/differential_protein_expression/simulation/ oneStage_syms.m	
This function defines the model of the conversion process with the logarithm of the output, which are used for the sigma point approximation	36
examples/ECM_differences/ generate_ECM_models.m	
This function generates the 128 hierarchical models accounting for all possible differences between the extracellular scaffolds. The underlying model for each scaffolds assumes inter- and intra-subpopulation variability of TrkA activity and cell-to-cell variability of relative Erk1/2 levels	37
examples/ECM_differences/ main_ECM_differences.m	
Main script for the analysis of the influence of extracellular scaffolds on pain signaling in sensory neurons	37
examples/ECM_differences/ plot_ECM_fit.m	
Visualization script for the results of the influence of extracellular scaffolds on pain signaling in sensory neurons	38
examples/ECM_differences/ plot_ECM_ranks.m	
Visualization script for ranking of the models and differences	38
examples/ECM_differences/ plot_ECM_valid.m	
Visualization script for the validation of the TrkA and Erk differences	38
examples/ECM_differences/ run_fittings_ECM.m	
This function performs optimization for the all model accounting for differences between PDL and Coll	39
examples/ECM_differences/ run_profile_ECM.m	
This function calculates the profile likelihoods for the final model accounting for differences in TrkA activity, Erk levels and Erk inactivation	39

examples/ECM_differences/data/load_datasets_PDL_Coll.m	
Script for loading the datasets of extracellular scaffolds poly-D-lysine (PDL) and Collagen I (Coll)	37
examples/subpopulation_differences/generate_nosubpop_file.m	
Generate file for model without any subpopulations	41
examples/subpopulation_differences/generate_subpop_files.m	
Generation of all models accounting for subpopulation differences	41
examples/subpopulation_differences/generate_subpop_TrkA.m	
Generation of final model including differences in TrkA levels	42
examples/subpopulation_differences/main_singlecell_prediction.m	
Script for the single cell prediction for neurons cultured on poly-D-lysine	24
examples/subpopulation_differences/main_subpopulation_analysis.m	
Script for the analysis of subpopulation differences in sensory neurons cultured on poly-D-lysine	42
examples/subpopulation_differences/plot_subpop_fit.m	
Visualization script for the fits of cells on poly-D-lysine	42
examples/subpopulation_differences/plot_subpop_profiles.m	
Visualization script for the profile likelihoods	42
examples/subpopulation_differences/plot_subpop_ranks.m	??
examples/subpopulation_differences/plot_subpop_valid.m	
Visualization script for the validation Figure 5D-F of Loos et al., Cell Systems (2018)	43
examples/subpopulation_differences/run_fitting_subpop.m	
This function optimizes the model of NGF-induced Erk1/2 signaling accounting for the differences in subpopulations denoted by icomb	43
examples/subpopulation_differences/run_logmarg_subpop.m	
This function calculates the log marginals for the model of NGF-induced Erk1/2 signaling accounting for the differences in subpopulations denoted by icomb	44
examples/subpopulation_differences/run_lppd_subpop.m	
This function calculates the log pointwise predictive density for the model of NGF-induced Erk1/2 signaling accounting for the differences in subpopulations denoted by icomb	44
examples/subpopulation_differences/dephosphorylation/exp_decay.m	
Likelihood function for the fit of the dephosphorylation rate	40
examples/subpopulation_differences/dephosphorylation/import_dephospho_data.m	
Script for the import of the dephosphorylation data	40
examples/subpopulation_differences/dephosphorylation/main_analysis_dephospho.m	
Main script for the analysis of the dephosphorylation data	40
examples/subpopulation_differences/dephosphorylation/plot_subpop_dephospho_supplement.m	
Visualization script for the dephosphorylation analysis	41
examples/subpopulation_differences/simulation/generate_simFile_ErkSignaling.m	
This script generates the functions for the NGF-induced Erk1/2 signaling using the toolbox AMICI	45
examples/subpopulation_differences/simulation/ODEmodel_syms_sPsET_loglog.m	
This function provides the model of NGF-induced Erk1/2 signaling. It is compiled using AMICI	45

examples/two_stage_exandintrinsic/ main_MA.m	
This script generates the model incorporating the moment approximation, estimates the parameters and calculates the profile likelihoods	46
examples/two_stage_exandintrinsic/ main_MA_1subpop.m	
This script generates the model incorporating the moment approximation, estimates the parameters and calculates the profile likelihoods	46
examples/two_stage_exandintrinsic/ main_MA_1subpop_extrinsic.m	
This script generates the model incorporating the moment approximation, estimates the parameters and calculates the profile likelihoods	47
examples/two_stage_exandintrinsic/ main_RRE.m	
This script generates the model incorporating only the means by reaction rate equations, estimates the parameters and calculates the profile likelihoods	47
examples/two_stage_exandintrinsic/data/ generate_data_twostage.m	
This script generates the data for the two stage gene expression example using the Stochastic Simulation Algorithm (SSA) implemented in the toolbox CERENA	45
examples/two_stage_exandintrinsic/simulation/ compile_simFiles_geneExp.m	
This script generate the simulation files for the example of two stage gene expression including intrinsic and extrinsic noise. This requires the MATLAB Toolbox CERENA. The <code>_syms.m</code> files are generate using CERENA and the model definition files modelDef_geneExp.m for the reaction rate equations and the moment approximation, and modelDef_geneExp_extrinsic.m for the moment approximation with additional extrinsic noise	47
examples/two_stage_exandintrinsic/simulation/ MEC_2_LD_2_c_geneExp_extrinsic_syms.m	
This function provides the model definition for the moment approximation with additional extrinsic noise of the two stage gene expression. It is generated by the toolbox CERENA	48
examples/two_stage_exandintrinsic/simulation/ MEC_2_LD_2_c_geneExp_MA_syms.m	
This function provides the model definition for the moment approximation of the two stage gene expression. It is generated by the toolbox CERENA	48
examples/two_stage_exandintrinsic/simulation/ modelDef_geneExp.m	
Model definition of the reaction network for the two stage gene expression	49
examples/two_stage_exandintrinsic/simulation/ modelDef_geneExp_extrinsic.m	
Model definition of the reaction network for the two stage gene expression with additional extrinsic noise	49
examples/two_stage_exandintrinsic/simulation/ RRE_geneExp_RRE_syms.m	
This function provides the model definition for the reaction rate equations of the two stage gene expression. It is generated by the toolbox CERENA	50

7 File Documentation

7.1 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.

7.1.1 Detailed Description

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

7.1.2 Function Documentation

- 7.1.2.1 `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 capturing 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 `generate_ECM_models()`, `generate_nosubpop_file()`, `generate_subpop_files()`, `generate_subpop_↵_TrkA()`, `logLikelihood()`, `main_MA()`, `main_MA_1subpop()`, `main_MA_1subpop_extrinsic()`, `main_oneStage_S↵P_1D()`, `main_oneStage_SP_2D()`, `main_RRE()`, `main_subpopulation_analysis()`, `model_SP_all()`, `model_SP_↵k1()`, `model_SP_k1k2()`, `model_SP_k1k3()`, `model_SP_k2()`, `model_SP_k2k3()`, `model_SP_k3()`, `models_RRE()`, `plot_ECM_fit()`, `plot_subpop_fit()`, `plotODEMM()`, `run_fitting_subpop()`, `run_fittings_ECM()`, `run_logmarg_subpop()`, `run_lppd_subpop()`, and `run_profile_ECM()`.

7.2 computeMixtureProbability.m File Reference

Robust calculation of a mixture distribution likelihood.

Functions

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

Robust calculation of a mixture distribution likelihood.

7.2.1 Detailed Description

Robust calculation of a mixture distribution likelihood.

7.2.2 Function Documentation

7.2.2.1 `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 w_s • 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>logp</i>	$n \times 1$ scalar of loglikelihood
<i>dlogpdx_i</i>	$n \times n_{xi}$ vector of gradient

Definition at line 17 of file computeMixtureProbability.m.

Referenced by logLikelihood(), run_lppd_cr(), and run_lppd_subpop().

7.3 distributions/logn/func_dmudxi_logn_mean.m File Reference

This function calculates the derivative of μ of the (multivariate) log-normal distribution for the case of linking the mean of the observables to the mean of the (multivariate) log-normal distribution.

Functions

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

This function calculates the derivative of μ of the (multivariate) log-normal distribution for the case of linking the mean of the observables to the mean of the (multivariate) log-normal distribution.

7.3.1 Detailed Description

This function calculates the derivative of μ of the (multivariate) log-normal distribution for the case of linking the mean of the observables to the mean of the (multivariate) log-normal distribution.

7.3.2 Function Documentation

- 7.3.2.1 `mlhsInnerSubst < matlabtypesubstitute, dmudxi > func_dmudxi_logn_mean` (`matlabtypesubstitute t`, `matlabtypesubstitute x`, `matlabtypesubstitute dxdxi`, `matlabtypesubstitute Sigma`, `matlabtypesubstitute dSigmadxi`, `matlabtypesubstitute xi`, `matlabtypesubstitute u`, `matlabtypesubstitute dim`)

This function calculates the derivative of μ of the (multivariate) log-normal distribution for the case of linking the mean of the observables to the mean of the (multivariate) log-normal distribution.

Parameters

t	time vector (not used, included for consistency and possible extensions)
x	vector of the means and covariances (not used, included for consistency and possible extensions)
$dxdxi$	derivatives of the means and covariances
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/stimulus (not used, included for consistency and possible extensions)
n_dim	dimension of measurements

Return values

$dmudxi$	derivative of μ of the (multivariate) log-normal distribution
----------	---

Definition at line 17 of file func_dmudxi_logn_mean.m.

7.4 distributions/logn/func_dmudxi_logn_median.m File Reference

This function calculates the derivative of μ of the (multivariate) log-normal distribution for the case of linking the mean of the observables to the median of the (multivariate) log-normal distribution. This should only be used for the case of using only a mechanistic description of the means of the observables.

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 μ of the (multivariate) log-normal distribution for the case of linking the mean of the observables to the median of the (multivariate) log-normal distribution. This should only be used for the case of using only a mechanistic description of the means of the observables.

7.4.1 Detailed Description

This function calculates the derivative of μ of the (multivariate) log-normal distribution for the case of linking the mean of the observables to the median of the (multivariate) log-normal distribution. This should only be used for the case of using only a mechanistic description of the means of the observables.

7.4.2 Function Documentation

- 7.4.2.1 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 μ of the (multivariate) log-normal distribution for the case of linking the mean of the observables to the median of the (multivariate) log-normal distribution. This should only be used for the case of using only a mechanistic description of the means of the observables.

Parameters

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

Return values

<i>dmudxi</i>	derivative of μ of the (multivariate) log-normal distribution
---------------	---

Definition at line 17 of file func_dmudxi_logn_median.m.

7.5 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.

7.5.1 Detailed Description

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

7.5.2 Function Documentation

7.5.2.1 `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,'additive')`
`dsigma2dxi = func_dsigma2dxi_logn(t,x,dxdxi,xi,noise,dnoisedxi,'multiplicative')`

Parameters

t	time vector (not used, included for consistency and possible extensions)
x	vector of the means and variances (not used, included for consistency and possible extensions)
$dxdxi$	derivatives of the means and variances
xi	parameter vector(not used, included for consistency and possible extensions)
$varargin$	<ul style="list-style-type: none"> • noise: parameters for measurement noise • dnoisedxi: derivative of measurement noise • noisemodel: 'multiplicative' or 'additive'

Return values

$d\sigma^2_{dxi}$	derivative of σ^2 of a log-normal distribution
-------------------	---

Definition at line 17 of file func_dsigma2dxi_logn.m.

7.6 distributions/logn/func_dSigmadxi_logn.m File Reference

This function maps the means and covariances of the observables for a subpopulation to Σ of the multivariate 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 the means and covariances of the observables for a subpopulation to Σ of the multivariate log-normal distribution.

7.6.1 Detailed Description

This function maps the means and covariances of the observables for a subpopulation to Σ of the multivariate log-normal distribution.

7.6.2 Function Documentation

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

This function maps the means and covariances of the observables for a subpopulation to Σ of the multivariate 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')
dSigmadxi = func_dSigmadxi_logn(t,x,dxdxi,xi,n_n_dim,noise,dnoisedxi,'additive')
```

Definition at line 17 of file func_dSigmadxi_logn.m.

7.7 distributions/logn/func_Sigma_logn.m File Reference

This function maps the means and covariances to Σ of a (multivariate) log-normal distribution.

Functions

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

This function maps the means and covariances to Σ of a (multivariate) log-normal distribution.

7.7.1 Detailed Description

This function maps the means and covariances to Σ of a (multivariate) log-normal distribution.

7.7.2 Function Documentation

- ##### 7.7.2.1
- `mlhsInnerSubst< matlabtypesubstitute, Sigma > func_Sigma_logn` (matlabtypesubstitute `t`, matlabtypesubstitute `x`, matlabtypesubstitute `xi`, matlabtypesubstitute `n_dim`, matlabtypesubstitute `varargin`)

This function maps the means and covariances to Σ of a (multivariate) log-normal distribution.

USAGE

```
Sigma = func_Sigma_logn(t,x,xi,n_dim)
Sigma = func_Sigma_logn(t,x,xi,n_dim,noise,'additive')
Sigma = func_Sigma_logn(t,x,xi,n_dim,noise,'multiplicative')
```

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

7.8 distributions/logn/logoflognpdf.m File Reference

Modified version of MATLAB function LOGNPDF such that the log-density is returned.

Functions

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

Modified version of MATLAB function LOGNPDF such that the log-density is returned.

7.8.1 Detailed Description

Modified version of MATLAB function LOGNPDF such that the log-density is returned.

7.8.2 Function Documentation

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

Modified version of MATLAB function LOGNPDF such that the log-density is returned.

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().

7.9 distributions/logn/logofmvnpdf.m File Reference

Modified version of MATLAB function MVNPDF such that the log-density is returned.

Functions

- `mlhsInnerSubst< matlabtypesubstitute, varargout > logofmvnpdf (matlabtypesubstitute varargin)`
Modified version of MATLAB function MVNPDF such that the log-density is returned.

7.9.1 Detailed Description

Modified version of MATLAB function MVNPDF such that the log-density is returned.

7.10 distributions/norm/func_dmudxi_norm.m File Reference

This function calculates the derivative of μ of the (multivariate) 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 μ of the (multivariate) normal distribution.

7.10.1 Detailed Description

This function calculates the derivative of μ of the (multivariate) normal distribution.

7.10.2 Function Documentation

7.10.2.1 `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 μ of the (multivariate) 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 μ of the (multivariate) normal distribution.
----------	--

Definition at line 17 of file func_dmudxi_norm.m.

7.11 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 ξ , matlabtypesubstitute $varargin$)`

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

7.11.1 Detailed Description

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

7.11.2 Function Documentation

7.11.2.1 `mlhsInnerSubst < matlabtypesubstitute, dsigma2dxi > func_dsigma2dxi_norm (matlabtypesubstitute t , matlabtypesubstitute x , matlabtypesubstitute $dxdxi$, matlabtypesubstitute ξ , 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,'additive')`

Parameters

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

Parameters

x	vector of the means and variances of the observables (not used, included for consistency and possible extensions)
$dxdxi$	derivatives of the means and variances of the obseables
xi	parameter vector(not used, included for consistency and possible extensions)
$varargin$	<ul style="list-style-type: none"> • noise: parameter for measurement noise • dnoisedxi: derivative of measurement noise • noisemodel: (so far only 'additive' supported)

Return values

$d\sigma^2/dxi$	derivative of σ^2 of the normal distribution.
-----------------	--

Definition at line 17 of file func_dsigma2dxi_norm.m.

7.12 distributions/norm/func_dSigmadxi_norm.m File Reference

This function maps the means and variances to Σ of the multivariate 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 Σ of the multivariate normal distribution.

7.12.1 Detailed Description

This function maps the means and variances to Σ of the multivariate normal distribution.

7.12.2 Function Documentation

- 7.12.2.1** 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 Σ of the multivariate normal distribution.

USAGE

```
dSigmadxi = func_dSigmadxi_norm(t,x,dxdxi,xi,n_n_dim)
dSigmadxi = func_dSigmadxi_norm(t,x,dxdxi,xi,n_n_dim,noise,dnoisedxi,'additive')
```

Definition at line 17 of file func_dSigmadxi_norm.m.

7.13 distributions/norm/func_Sigma_norm.m File Reference

This function maps the means and variances to Σ of the multivariate 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 Σ of the multivariate normal distribution.

7.13.1 Detailed Description

This function maps the means and variances to Σ of the multivariate normal distribution.

7.13.2 Function Documentation

- 7.13.2.1 `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 Σ of the multivariate normal distribution.

USAGE

`Sigma = func_Sigma_norm(t,x,xi,n_dim,noise,'additive')`

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>	<ul style="list-style-type: none"> • noise: parameter for measurement noise • noisemodel: (so far only 'additive' supported)

Return values

<i>Sigma</i>	(<i>n_t</i> x <i>n_dim</i> x <i>n_dim</i>) Σ of the multivariate normal distribution.
--------------	--

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

7.14 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.

7.14.1 Detailed Description

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

7.14.2 Function Documentation

- 7.14.2.1 `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()`.

7.15 examples/conversion_reaction/data/generate_data_cr.m File Reference

This script generates the artificial data of a conversion process.

Functions

- `noret::substitute generate_data_cr` ()

This script generates the artificial data of a conversion process.

7.15.1 Detailed Description

This script generates the artificial data of a conversion process.

7.16 examples/conversion_reaction/main_optimization_cr.m File Reference

This is the main script for the first example, the conversion reaction. All Reaction Rate Equation (RRE) models are generated and their parameters are estimated by calling [models_RRE.m](#). All SP models are estimated, for which e.g. [model_SP_k1.m](#) indicates that k1 has an additional cell-to-cell variability modeled by the Sigma point approximation.

Functions

- `noret::substitute main_optimization_cr ()`

This is the main script for the first example, the conversion reaction. All Reaction Rate Equation (RRE) models are generated and their parameters are estimated by calling [models_RRE.m](#). All SP models are estimated, for which e.g. [model_SP_k1.m](#) indicates that k1 has an additional cell-to-cell variability modeled by the Sigma point approximation.

7.16.1 Detailed Description

This is the main script for the first example, the conversion reaction. All Reaction Rate Equation (RRE) models are generated and their parameters are estimated by calling [models_RRE.m](#). All SP models are estimated, for which e.g. [model_SP_k1.m](#) indicates that k1 has an additional cell-to-cell variability modeled by the Sigma point approximation.

7.17 examples/conversion_reaction/main_sampling_cr.m File Reference

This is the main script for the Bayesian model selection for the conversion reaction.

Functions

- `noret::substitute main_sampling_cr ()`

This is the main script for the Bayesian model selection for the conversion reaction.

7.17.1 Detailed Description

This is the main script for the Bayesian model selection for the conversion reaction.

7.18 examples/conversion_reaction/main_singlecell_prediction.m File Reference

This script shows the analysis of the single-cell trajectory prediction. First the single-cell parameter based on one single-cell trajectory are sampled and the predicted results are visualized. Then the parameters are optimized for 100 single-cell trajectories. This analysis corresponds to Loos et al., Cell Systems (2018), Figure 3E-G.

Functions

- `noret::substitute main_singlecell_prediction ()`

This script shows the analysis of the single-cell trajectory prediction. First the single-cell parameter based on one single-cell trajectory are sampled and the predicted results are visualized. Then the parameters are optimized for 100 single-cell trajectories. This analysis corresponds to Loos et al., Cell Systems (2018), Figure 3E-G.

7.18.1 Detailed Description

This script shows the analysis of the single-cell trajectory prediction. First the single-cell parameter based on one single-cell trajectory are sampled and the predicted results are visualized. Then the parameters are optimized for 100 single-cell trajectories. This analysis corresponds to Loos et al., Cell Systems (2018), Figure 3E-G.

7.18.2 Function Documentation

7.18.2.1 noret::substitute main_singlecell_prediction ()

This script shows the analysis of the single-cell trajectory prediction. First the single-cell parameter based on one single-cell trajectory are sampled and the predicted results are visualized. Then the parameters are optimized for 100 single-cell trajectories. This analysis corresponds to Loos et al., Cell Systems (2018), Figure 3E-G.

Script for the single cell prediction for neurons cultured on poly-D-lysine.

Definition at line 17 of file main_singlecell_prediction.m.

References load_plot_settings(), and sclogLikelihood_cr().

7.19 examples/subpopulation_differences/main_singlecell_prediction.m File Reference

Script for the single cell prediction for neurons cultured on poly-D-lysine.

Functions

- noret::substitute **main_singlecell_prediction** ()

7.19.1 Detailed Description

Script for the single cell prediction for neurons cultured on poly-D-lysine.

7.20 examples/conversion_reaction/model_SP_all.m File Reference

This function generates model and estimates parameters for the model accounting for means and variances (using Sigma Points (SP)).

All kinetic parameters are assumed to vary between individual cells:

k1: inter- and intra-subpopulation variable

k2: cell-to-cell variable

k3: cell-to-cell variable

.

Functions

- noret::substitute **model_SP_all** ()

This function generates model and estimates parameters for the model accounting for means and variances (using Sigma Points (SP)).

All kinetic parameters are assumed to vary between individual cells:

k1: inter- and intra-subpopulation variable

k2: cell-to-cell variable

k3: cell-to-cell variable

.

7.20.1 Detailed Description

This function generates model and estimates parameters for the model accounting for means and variances (using Sigma Points (SP)).

All kinetic parameters are assumed to vary between individual cells:

k1: inter- and intra-subpopulation variable

k2: cell-to-cell variable

k3: cell-to-cell variable

.

7.21 examples/conversion_reaction/model_SP_k1.m File Reference

This function generates the model and estimates the parameters in case of accounting for means and variances (using Sigma Points (SP)) The parameter are assumed to be:

k1: inter- and intra-subpopulation variable

k2: homogeneous

k3: homogeneous

.

Functions

- noret::substitute [model_SP_k1](#) ()

This function generates the model and estimates the parameters in case of accounting for means and variances (using Sigma Points (SP)) The parameter are assumed to be:

k1: inter- and intra-subpopulation variable

k2: homogeneous

k3: homogeneous

.

7.21.1 Detailed Description

This function generates the model and estimates the parameters in case of accounting for means and variances (using Sigma Points (SP)) The parameter are assumed to be:

k1: inter- and intra-subpopulation variable

k2: homogeneous

k3: homogeneous

.

7.22 examples/conversion_reaction/model_SP_k1k2.m File Reference

This function generates the model and estimates the parameters in case of accounting for means and variances (using Sigma Points (SP)) The parameter are assumed to be:

k1: inter- and intra-subpopulation variable

k2: cell-to-cell variable

k3: homogeneous.

Functions

- noret::substitute [model_SP_k1k2](#) ()

This function generates the model and estimates the parameters in case of accounting for means and variances (using Sigma Points (SP)) The parameter are assumed to be:

k1: inter- and intra-subpopulation variable

k2: cell-to-cell variable

k3: homogeneous.

7.22.1 Detailed Description

This function generates the model and estimates the parameters in case of accounting for means and variances (using Sigma Points (SP)) The parameter are assumed to be:

k1: inter- and intra-subpopulation variable

k2: cell-to-cell variable

k3: homogeneous.

7.23 examples/conversion_reaction/model_SP_k1k3.m File Reference

This function generates the model and estimates the parameters in case of accounting for means and variances (using Sigma Points (SP)) The parameter are assumed to be:

k1: inter- and intra-subpopulation variable

k2: homogeneous

k3: cell-to-cell variable

Functions

- noret::substitute [model_SP_k1k3](#) ()

This function generates the model and estimates the parameters in case of accounting for means and variances (using Sigma Points (SP)) The parameter are assumed to be:

k1: inter- and intra-subpopulation variable

k2: homogeneous

k3: cell-to-cell variable

7.23.1 Detailed Description

This function generates the model and estimates the parameters in case of accounting for means and variances (using Sigma Points (SP)) The parameter are assumed to be:

k1: inter- and intra-subpopulation variable

k2: homogeneous

k3: cell-to-cell variable

7.24 examples/conversion_reaction/model_SP_k2.m File Reference

This function generates the model and estimates the parameters in case of accounting for means and variances (using Sigma Points (SP)). The parameter are assumed to be:

k1: subpopulation variable

k2: cell-to-cell variable

k3: homogeneous.

Functions

- noret::substitute [model_SP_k2](#) ()

This function generates the model and estimates the parameters in case of accounting for means and variances (using Sigma Points (SP)). The parameter are assumed to be:

k1: subpopulation variable

k2: cell-to-cell variable

k3: homogeneous.

7.24.1 Detailed Description

This function generates the model and estimates the parameters in case of accounting for means and variances (using Sigma Points (SP)). The parameter are assumed to be:

k1: subpopulation variable

k2: cell-to-cell variable

k3: homogeneous.

7.25 examples/conversion_reaction/model_SP_k2k3.m File Reference

This function generates the model and estimates the parameters in case of accounting for means and variances (using Sigma Points (SP)). The parameter are assumed to be:

k1: subpopulation variable

k2: cell-to-cell variable

k3: cell-to-cell variable.

Functions

- noret::substitute [model_SP_k2k3](#) ()

This function generates the model and estimates the parameters in case of accounting for means and variances (using Sigma Points (SP)). The parameter are assumed to be:

k1: subpopulation variable

k2: cell-to-cell variable

k3: cell-to-cell variable.

7.25.1 Detailed Description

This function generates the model and estimates the parameters in case of accounting for means and variances (using Sigma Points (SP)). The parameter are assumed to be:

k1: subpopulation variable

k2: cell-to-cell variable

k3: cell-to-cell variable.

7.26 examples/conversion_reaction/model_SP_k3.m File Reference

This function generates the model and estimates the parameters in case of accounting for means and variances (using Sigma Points (SP)). The parameter are assumed to be:

k1: subpopulation variable

k2: homogenous

k3: cell-to-cell variable

This model is the ground truth.

Functions

- noret::substitute [model_SP_k3](#) ()

This function generates the model and estimates the parameters in case of accounting for means and variances (using Sigma Points (SP)). The parameter are assumed to be:

k1: subpopulation variable

k2: homogenous

k3: cell-to-cell variable

This model is the ground truth.

7.26.1 Detailed Description

This function generates the model and estimates the parameters in case of accounting for means and variances (using Sigma Points (SP)). The parameter are assumed to be:

k1: subpopulation variable

k2: homogenous

k3: cell-to-cell variable

This model is the ground truth.

7.27 examples/conversion_reaction/models_RRE.m File Reference

This function generates Reaction Rate Euqation model and estimates the parameters for the models with varying number of parameters for the variances.

Functions

- noret::substitute [models_RRE](#) ()

This function generates Reaction Rate Euqation model and estimates the parameters for the models with varying number of parameters for the variances.

7.27.1 Detailed Description

This function generates Reaction Rate Euqation model and estimates the parameters for the models with varying number of parameters for the variances.

7.28 examples/conversion_reaction/plot_cr.m File Reference

This script visualizes the results for the conversion reaction.

Functions

- noret::substitute [plot_cr](#) ()

This script visualizes the results for the conversion reaction.

7.28.1 Detailed Description

This script visualizes the results for the conversion reaction.

7.29 examples/conversion_reaction/plot_cr_SIAppendix.m File Reference

This script generates the figures for the supplement for the conversion reaction example.

Functions

- noret::substitute [plot_cr_SIAppendix](#) ()

This script generates the figures for the supplement for the conversion reaction example.

7.29.1 Detailed Description

This script generates the figures for the supplement for the conversion reaction example.

7.30 examples/conversion_reaction/plot_cr_variabilityReduction.m File Reference

```
clear all close all clc load_plot_settings
```

Functions

- noret::substitute [plot_cr_variabilityReduction](#) ()

clear all close all clc load_plot_settings

7.30.1 Detailed Description

```
clear all close all clc load_plot_settings
```

7.30.2 Function Documentation

7.30.2.1 noret::substitute plot_cr_variabilityReduction ()

```
clear all close all clc load_plot_settings
```

```
% SP all
```

Definition at line 17 of file plot_cr_variabilityReduction.m.

Referenced by plot_cr_SIAppendix().

7.31 examples/conversion_reaction/run_lppd_cr.m File Reference

This function runs the sampling for the log pointwise predictive density for the conversion reaction reaction example.

Functions

- mlhsInnerSubst< matlabtypesubstitute, lppd > [run_lppd_cr](#) (matlabtypesubstitute model_name, matlabtype-substitute burnin, matlabtypesubstitute nlter, matlabtypesubstitute saveFlag)

This function runs the sampling for the log pointwise predictive density for the conversion reaction reaction example.

7.31.1 Detailed Description

This function runs the sampling for the log pointwise predictive density for the conversion reaction reaction example.

7.31.2 Function Documentation

7.31.2.1 `mlhsInnerSubst< matlabtypesubstitute, lppd > run_lppd_cr (matlabtypesubstitute model_name,
matlabtypesubstitute burnin, matlabtypesubstitute nIter, matlabtypesubstitute saveFlag)`

This function runs the sampling for the log pointwise predictive density for the conversion reaction reaction example.

USAGE

```
parameters_lppd = run_lppd_cr(model_name,burnin,nIter,saveFlag)
```

Parameters

<i>model_name</i>	string indicating model (RRE_timedep,RRE_subpop, RRE_onlyone,SP_k1,SP_k2,SP_k3,SP_k1k2,SP_k1k3, SP_k2k3,or SP_all)
<i>burnin</i>	number of burn-in steps for sampling
<i>nIter</i>	number of total iterations for sampling
<i>saveFlag</i>	if results should be saved

Return values

<i>lppd</i>	log pointwise predictive density
-------------	----------------------------------

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

References `computeMixtureProbability()`, and `logLikelihood()`.

Referenced by `main_sampling_cr()`.

7.32 examples/conversion_reaction/run_thermIntegration_cr.m File Reference

This function runs thermodynamic integration for the conversion reaction example.

Functions

- `mlhsInnerSubst< matlabtypesubstitute, Q > run_thermIntegration_cr (matlabtypesubstitute model_name,
matlabtypesubstitute burnin, matlabtypesubstitute nIter, matlabtypesubstitute saveFlag)`

This function runs thermodynamic integration for the conversion reaction example.

7.32.1 Detailed Description

This function runs thermodynamic integration for the conversion reaction example.

7.32.2 Function Documentation

7.32.2.1 `mlhsInnerSubst< matlabtypesubstitute, Q > run_thermIntegration_cr (matlabtypesubstitute model_name,
matlabtypesubstitute burnin, matlabtypesubstitute nIter, matlabtypesubstitute saveFlag)`

This function runs thermodynamic integration for the conversion reaction example.

USAGE

```
Q = run_thermIntegration_cr(model_name,burnin,nIter,saveFlag)
```

Parameters

<i>model_name</i>	string indicating model (RRE_timedep,RRE_subpop,RRE_onlyone,SP_k1,SP_k2,SP_k3,SP_k1k2,SP_k1k3,SP_k2k3,or SP_all)
<i>burnin</i>	number of burn-in steps for sampling
<i>nIter</i>	number of total iterations for sampling
<i>saveFlag</i>	if results should be saved

Return values

<i>Q</i>	log marginal likelihood
----------	-------------------------

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

References `logLikelihood()`.

Referenced by `main_sampling_cr()`.

7.33 examples/conversion_reaction/sclogLikelihood_cr.m File Reference

This function provides the likelihood for the optimization of the single cell trajectory in the conversion reaction example.

Functions

- `mlhsSubst< mlhsInnerSubst< matlabtypesubstitute, J >,mlhsInnerSubst< matlabtypesubstitute, dJ >> sclogLikelihood_cr (matlabtypesubstitute t, matlabtypesubstitute y, matlabtypesubstitute xi, matlabtypesubstitute k2, matlabtypesubstitute m_k3, matlabtypesubstitute sigma_k3, matlabtypesubstitute sigma_noise, matlabtypesubstitute w, matlabtypesubstitute k1_1, matlabtypesubstitute k1_2, matlabtypesubstitute time-point)`

This function provides the likelihood for the optimization of the single cell trajectory in the conversion reaction example.

7.33.1 Detailed Description

This function provides the likelihood for the optimization of the single cell trajectory in the conversion reaction example.

7.33.2 Function Documentation

7.33.2.1 `mlhsSubst< mlhsInnerSubst< matlabtypesubstitute, J >,mlhsInnerSubst< matlabtypesubstitute, dJ > >`
`sclogLikelihood_cr (matlabtypesubstitute t, matlabtypesubstitute y, matlabtypesubstitute xi, matlabtypesubstitute k2,
matlabtypesubstitute m_k3, matlabtypesubstitute sigma_k3, matlabtypesubstitute sigma_noise, matlabtypesubstitute
w, matlabtypesubstitute k1_1, matlabtypesubstitute k1_2, matlabtypesubstitute timepoint)`

This function provides the likelihood for the optimization of the single cell trajectory in the conversion reaction example.

USAGE

`[J,dJ] = sclogLikelihood_cr(t,y,xi,k2,m_k3,sigma_k3,sigma_noise,w,k1_1,k1_2,timepoint)`

Parameters

<i>t</i>	time for simulation
<i>y</i>	data
<i>xi</i>	single-cell parameter vector which is optimized
<i>k2</i>	pre-optimized parameter value for k2
<i>m_k3</i>	pre-optimized parameter value for m_k3
<i>sigma_k3</i>	pre-optimized parameter value for sigma_k3
<i>sigma_noise</i>	pre-optimized parameter value for sigma_noise
<i>w</i>	pre-optimized parameter value for 2
<i>k1_1</i>	pre-optimized parameter value for k1_1
<i>k1_2</i>	pre-optimized parameter value for k1_2
<i>timepoint</i>	on which time point the function should be evaluated, <i>last</i> , or <i>first</i>

Return values

<i>J</i>	log-likelihood value
<i>dJ</i>	gradient of log-likelihood functoin

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

Referenced by `main_singlecell_prediction()`.

7.34 examples/conversion_reaction/simulation/CR_log_syms.m File Reference

This function defines the model of the conversion process with the logarithm of the output, which are used for the sigma point approximation.

Functions

- `mlhsInnerSubst< matlabtypesubstitute, model > CR_log_syms ()`

This function defines the model of the conversion process with the logarithm of the output, which are used for the sigma point approximation.

7.34.1 Detailed Description

This function defines the model of the conversion process with the logarithm of the output, which are used for the sigma point approximation.

7.34.2 Function Documentation

7.34.2.1 mlhsInnerSubst< matlabtypesubstitute, model > CR_log_syms ()

This function defines the model of the conversion process with the logarithm of the output, which are used for the sigma point approximation.

Return values

<i>model</i>	model struct used with amiwrap
--------------	--------------------------------

Generated fields of model:

Definition at line 17 of file CR_log_syms.m.

7.35 examples/conversion_reaction/simulation/CR_syms.m File Reference

This function defines the model of the conversion process with the linear output, which is used for the Reaction Rate Equations.

Functions

- mlhsInnerSubst< matlabtypesubstitute, model > [CR_syms](#) ()

This function defines the model of the conversion process with the linear output, which is used for the Reaction Rate Equations.

7.35.1 Detailed Description

This function defines the model of the conversion process with the linear output, which is used for the Reaction Rate Equations.

7.35.2 Function Documentation

7.35.2.1 mlhsInnerSubst< matlabtypesubstitute, model > CR_syms ()

This function defines the model of the conversion process with the linear output, which is used for the Reaction Rate Equations.

Return values

<i>model</i>	model struct used with amiwrap
--------------	--------------------------------

Generated fields of model:

Definition at line 17 of file CR_syms.m.

7.36 examples/conversion_reaction/simulation/generate_simFiles_cr.m File Reference

This script generates the functions for the conversion reaction using the toolbox AMICI.

Functions

- noret::substitute [generate_simFiles_cr](#) ()

This script generates the functions for the conversion reaction using the toolbox AMICI.

7.36.1 Detailed Description

This script generates the functions for the conversion reaction using the toolbox AMICI.

7.37 examples/differential_protein_expression/data/generate_data_oneStage.m File Reference

This script generates the artificial data of the differential gene protein expression.

Functions

- noret::substitute [generate_data_oneStage](#) ()

This script generates the artificial data of the differential gene protein expression.

7.37.1 Detailed Description

This script generates the artificial data of the differential gene protein expression.

7.38 examples/differential_protein_expression/main_oneStage_SP_1D.m File Reference

This script generates the model for the 1-dimensional measurements of the differential gene expression example using the sigma point approximation. The parameters are estimated and also the profile likelihoods are calculated using the toolbox PESTO.

Functions

- noret::substitute [main_oneStage_SP_1D](#) ()

This script generates the model for the 1-dimensional measurements of the differential gene expression example using the sigma point approximation. The parameters are estimated and also the profile likelihoods are calculated using the toolbox PESTO.

7.38.1 Detailed Description

This script generates the model for the 1-dimensional measurements of the differential gene expression example using the sigma point approximation. The parameters are estimated and also the profile likelihoods are calculated using the toolbox PESTO.

7.39 examples/differential_protein_expression/main_oneStage_SP_2D.m File Reference

This script generates the model for the 2-dimensional measurements of the differential gene expression example using the sigma point approximation. The parameters are estimated and also the profile likelihoods are calculated using the toolbox PESTO.

Functions

- noret::substitute [main_oneStage_SP_2D](#) ()

This script generates the model for the 2-dimensional measurements of the differential gene expression example using the sigma point approximation. The parameters are estimated and also the profile likelihoods are calculated using the toolbox PESTO.

7.39.1 Detailed Description

This script generates the model for the 2-dimensional measurements of the differential gene expression example using the sigma point approximation. The parameters are estimated and also the profile likelihoods are calculated using the toolbox PESTO.

7.40 examples/differential_protein_expression/plot_fit_oneStage.m File Reference

This script visualizes the model fit for the two-dimensional data.

Functions

- noret::substitute [plot_fit_oneStage](#) ()

This script visualizes the model fit for the two-dimensional data.

7.40.1 Detailed Description

This script visualizes the model fit for the two-dimensional data.

7.41 examples/differential_protein_expression/plot_uncertainty_oneStage.m File Reference

This script visualizes the confidence intervals for the one- and the two-dimensional models.

Functions

- noret::substitute [plot_uncertainty_oneStage](#) ()
This script visualizes the confidence intervals for the one- and the two-dimensional models.

7.41.1 Detailed Description

This script visualizes the confidence intervals for the one- and the two-dimensional models.

7.42 examples/differential_protein_expression/simulation/generate_simFile_oneStage.m File Reference

Compilation of one stage simulation file.

Functions

- noret::substitute [generate_simFile_oneStage](#) ()
Compilation of one stage simulation file.

7.42.1 Detailed Description

Compilation of one stage simulation file.

7.43 examples/differential_protein_expression/simulation/oneStage_syms.m File Reference

This function defines the model of the conversion process with the logarithm of the output, which are used for the sigma point approximation.

Functions

- mlhsInnerSubst< matlabtypesubstitute, model > [oneStage_syms](#) ()
This function defines the model of the conversion process with the logarithm of the output, which are used for the sigma point approximation.

7.43.1 Detailed Description

This function defines the model of the conversion process with the logarithm of the output, which are used for the sigma point approximation.

7.43.2 Function Documentation

7.43.2.1 mlhsInnerSubst< matlabtypesubstitute, model > oneStage_syms ()

This function defines the model of the conversion process with the logarithm of the output, which are used for the sigma point approximation.

Return values

<i>model</i>	model struct used with amiwrap
--------------	--------------------------------

Generated fields of model:

Definition at line 17 of file oneStage_syms.m.

7.44 examples/ECM_differences/data/load_datasets_PDL_Coll.m File Reference

Script for loading the datasets of extracellular scaffolds poly-D-lysine (PDL) and Collagen I (Coll).

Functions

- noret::substitute [load_datasets_PDL_Coll](#) ()
Script for loading the datasets of extracellular scaffolds poly-D-lysine (PDL) and Collagen I (Coll).

7.44.1 Detailed Description

Script for loading the datasets of extracellular scaffolds poly-D-lysine (PDL) and Collagen I (Coll).

7.45 examples/ECM_differences/generate_ECM_models.m File Reference

This function generates the 128 hierarchical models accounting for all possible differences between the extracellular scaffolds. The underlying model for each scaffolds assumes inter- and intra-subpopulation variability of TrkA activity and cell-to-cell variability of relative Erk1/2 levels.

Functions

- noret::substitute [generate_ECM_models](#) ()
This function generates the 128 hierarchical models accounting for all possible differences between the extracellular scaffolds. The underlying model for each scaffolds assumes inter- and intra-subpopulation variability of TrkA activity and cell-to-cell variability of relative Erk1/2 levels.

7.45.1 Detailed Description

This function generates the 128 hierarchical models accounting for all possible differences between the extracellular scaffolds. The underlying model for each scaffolds assumes inter- and intra-subpopulation variability of TrkA activity and cell-to-cell variability of relative Erk1/2 levels.

7.46 examples/ECM_differences/main_ECM_differences.m File Reference

Main script for the analysis of the influence of extracellular scaffolds on pain signaling in sensory neurons.

Functions

- noret::substitute [main_ECM_differences](#) ()

Main script for the analysis of the influence of extracellular scaffolds on pain signaling in sensory neurons.

7.46.1 Detailed Description

Main script for the analysis of the influence of extracellular scaffolds on pain signaling in sensory neurons.

7.47 examples/ECM_differences/plot_ECM_fit.m File Reference

Visualization script for the results of the influence of extracellular scaffolds on pain signaling in sensory neurons.

Functions

- noret::substitute [plot_ECM_fit](#) ()

Visualization script for the results of the influence of extracellular scaffolds on pain signaling in sensory neurons.

7.47.1 Detailed Description

Visualization script for the results of the influence of extracellular scaffolds on pain signaling in sensory neurons.

7.48 examples/ECM_differences/plot_ECM_ranks.m File Reference

Visualization script for ranking of the models and differences.

Functions

- noret::substitute [plot_ECM_ranks](#) ()

Visualization script for ranking of the models and differences.

7.48.1 Detailed Description

Visualization script for ranking of the models and differences.

7.49 examples/ECM_differences/plot_ECM_valid.m File Reference

Visualization script for the validation of the TrkA and Erk differences.

Functions

- noret::substitute [plot_ECM_valid](#) ()

Visualization script for the validation of the TrkA and Erk differences.

7.49.1 Detailed Description

Visualization script for the validation of the TrkA and Erk differences.

7.50 examples/ECM_differences/run_fittings_ECM.m File Reference

This function performs optimization for the all model accounting for differences between PDL and Coll.

Functions

- mlhsInnerSubst< matlabtypesubstitute, parameters > [run_fittings_ECM](#) ()

This function performs optimization for the all model accounting for differences between PDL and Coll.

7.50.1 Detailed Description

This function performs optimization for the all model accounting for differences between PDL and Coll.

7.50.2 Function Documentation

7.50.2.1 mlhsInnerSubst< matlabtypesubstitute, parameters > run_fittings_ECM ()

This function performs optimization for the all model accounting for differences between PDL and Coll.

Return values

<i>parameters</i>	struct of parameters for each of the 128 models obtained by getMultiStarts.m
-------------------	--

Generated fields of parameters:

Definition at line 17 of file run_fittings_ECM.m.

References collectConditions(), and logLikelihood().

Referenced by main_ECM_differences().

7.51 examples/ECM_differences/run_profile_ECM.m File Reference

This function calculates the profile likelihoods for the final model accounting for differences in TrkA activity, Erk levels and Erk inactivation.

Functions

- mlhsInnerSubst< matlabtypesubstitute, parameters > [run_profile_ECM](#) ()

This function calculates the profile likelihoods for the final model accounting for differences in TrkA activity, Erk levels and Erk inactivation.

7.51.1 Detailed Description

This function calculates the profile likelihoods for the final model accounting for differences in TrkA activity, Erk levels and Erk inactivation.

7.52 examples/subpopulation_differences/dephosphorylation/exp_decay.m File Reference

Likelihood function for the fit of the dephosphorylation rate.

Functions

- `mlhsInnerSubst< matlabtypesubstitute, J > exp_decay (matlabtypesubstitute k5, matlabtypesubstitute t, matlabtypesubstitute pErk)`

Likelihood function for the fit of the dephosphorylation rate.

7.52.1 Detailed Description

Likelihood function for the fit of the dephosphorylation rate.

7.53 examples/subpopulation_differences/dephosphorylation/import_dephospho_data.m File Reference

Script for the import of the dephosphorylation data.

Functions

- `noret::substitute import_dephospho_data ()`

Script for the import of the dephosphorylation data.

7.53.1 Detailed Description

Script for the import of the dephosphorylation data.

7.54 examples/subpopulation_differences/dephosphorylation/main_analysis_dephospho.m File Reference

Main script for the analysis of the dephosphorylation data.

Functions

- `noret::substitute main_analysis_dephospho ()`

Main script for the analysis of the dephosphorylation data.

7.54.1 Detailed Description

Main script for the analysis of the dephosphorylation data.

7.55 examples/subpopulation_differences/dephosphorylation/plot_subpop_dephoshp_supplement.m File Reference

Visualization script for the dephosphorylation analysis.

Functions

- noret::substitute [plot_subpop_dephoshp_supplement](#) ()
Visualization script for the dephosphorylation analysis.

7.55.1 Detailed Description

Visualization script for the dephosphorylation analysis.

7.56 examples/subpopulation_differences/generate_nosubpop_file.m File Reference

Generate file for model without any subpopulations.

Functions

- noret::substitute [generate_nosubpop_file](#) ()
Generate file for model without any subpopulations.

7.56.1 Detailed Description

Generate file for model without any subpopulations.

7.57 examples/subpopulation_differences/generate_subpop_files.m File Reference

Generation of all models accounting for subpopulation differences.

Functions

- noret::substitute [generate_subpop_files](#) ()
Generation of all models accounting for subpopulation differences.

7.57.1 Detailed Description

Generation of all models accounting for subpopulation differences.

7.58 examples/subpopulation_differences/generate_subpop_TrkA.m File Reference

Generation of final model including differences in TrkA levels.

Functions

- noret::substitute [generate_subpop_TrkA](#) ()
Generation of final model including differences in TrkA levels.

7.58.1 Detailed Description

Generation of final model including differences in TrkA levels.

7.59 examples/subpopulation_differences/main_subpopulation_analysis.m File Reference

Script for the analysis of subpopulation differences in sensory neurons cultured on poly-D-lysine.

Functions

- noret::substitute [main_subpopulation_analysis](#) ()
Script for the analysis of subpopulation differences in sensory neurons cultured on poly-D-lysine.

7.59.1 Detailed Description

Script for the analysis of subpopulation differences in sensory neurons cultured on poly-D-lysine.

7.60 examples/subpopulation_differences/plot_subpop_fit.m File Reference

Visualization script for the fits of cells on poly-D-lysine.

Functions

- noret::substitute [plot_subpop_fit](#) ()
Visualization script for the fits of cells on poly-D-lysine.

7.60.1 Detailed Description

Visualization script for the fits of cells on poly-D-lysine.

7.61 examples/subpopulation_differences/plot_subpop_profiles.m File Reference

Visualization script for the profile likelihoods.

Functions

- noret::substitute [plot_subpop_profiles](#) ()
Visualization script for the profile likelihoods.

7.61.1 Detailed Description

Visualization script for the profile likelihoods.

7.62 examples/subpopulation_differences/plot_subpop_valid.m File Reference

Visualization script for the validation Figure 5D-F of Loos et al., Cell Systems (2018).

Functions

- noret::substitute [plot_subpop_valid](#) ()
Visualization script for the validation Figure 5D-F of Loos et al., Cell Systems (2018).

7.62.1 Detailed Description

Visualization script for the validation Figure 5D-F of Loos et al., Cell Systems (2018).

7.63 examples/subpopulation_differences/run_fitting_subpop.m File Reference

This function optimizes the model of NGF-induced Erk1/2 signaling accounting for the differences in subpopulations denoted by icomb.

Functions

- mlhsInnerSubst< matlabtypesubstitute, parameters > [run_fitting_subpop](#) (matlabtypesubstitute icomb)
This function optimizes the model of NGF-induced Erk1/2 signaling accounting for the differences in subpopulations denoted by icomb.

7.63.1 Detailed Description

This function optimizes the model of NGF-induced Erk1/2 signaling accounting for the differences in subpopulations denoted by icomb.

7.63.2 Function Documentation

7.63.2.1 mlhsInnerSubst< matlabtypesubstitute, parameters > run_fitting_subpop (matlabtypesubstitute icomb)

This function optimizes the model of NGF-induced Erk1/2 signaling accounting for the differences in subpopulations denoted by icomb.

Generated fields of parameters:

Definition at line 17 of file run_fitting_subpop.m.

References collectConditions(), and logLikelihood().

7.64 examples/subpopulation_differences/run_logmarg_subpop.m File Reference

This function calculates the log marginals for the model of NGF-induced Erk1/2 signaling accounting for the differences in subpopulations denoted by icomb.

Functions

- mlhsInnerSubst< matlabtypesubstitute, logmarg > [run_logmarg_subpop](#) (matlabtypesubstitute icomb)
This function calculates the log marginals for the model of NGF-induced Erk1/2 signaling accounting for the differences in subpopulations denoted by icomb.

7.64.1 Detailed Description

This function calculates the log marginals for the model of NGF-induced Erk1/2 signaling accounting for the differences in subpopulations denoted by icomb.

7.65 examples/subpopulation_differences/run_lppd_subpop.m File Reference

This function calculates the log pointwise predictive density for the model of NGF-induced Erk1/2 signaling accounting for the differences in subpopulations denoted by icomb.

Functions

- mlhsInnerSubst< matlabtypesubstitute, lppd > [run_lppd_subpop](#) (matlabtypesubstitute icomb)
This function calculates the log pointwise predictive density for the model of NGF-induced Erk1/2 signaling accounting for the differences in subpopulations denoted by icomb.

7.65.1 Detailed Description

This function calculates the log pointwise predictive density for the model of NGF-induced Erk1/2 signaling accounting for the differences in subpopulations denoted by icomb.

7.66 examples/subpopulation_differences/simulation/generate_simFile_ErkSignaling.m File Reference

This script generates the functions for the NGF-induced Erk1/2 signaling using the toolbox AMICI.

Functions

- `noret::substitute generate_simFile_ErkSignaling ()`
This script generates the functions for the NGF-induced Erk1/2 signaling using the toolbox AMICI.

7.66.1 Detailed Description

This script generates the functions for the NGF-induced Erk1/2 signaling using the toolbox AMICI.

7.67 examples/subpopulation_differences/simulation/ODEmodel_syms_sPsET_loglog.m File Reference

This function provides the model of NGF-induced Erk1/2 signaling. It is compiled using AMICI.

Functions

- `mlhsInnerSubst< matlabtypesubstitute, model > ODEmodel_syms_sPsET_loglog ()`
This function provides the model of NGF-induced Erk1/2 signaling. It is compiled using AMICI.

7.67.1 Detailed Description

This function provides the model of NGF-induced Erk1/2 signaling. It is compiled using AMICI.

7.67.2 Function Documentation

7.67.2.1 `mlhsInnerSubst< matlabtypesubstitute, model > ODEmodel_syms_sPsET_loglog ()`

This function provides the model of NGF-induced Erk1/2 signaling. It is compiled using AMICI.

Generated fields of model:

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

7.68 examples/two_stage_exandintrinsic/data/generate_data_twostage.m File Reference

This script generates the data for the two stage gene expression example using the Stochastic Simulation Algorithm (SSA) implemented in the toolbox CERENA.

Functions

- noret::substitute [generate_data_twostage](#) ()

This script generates the data for the two stage gene expression example using the Stochastic Simulation Algorithm (SSA) implemented in the toolbox CERENA.

7.68.1 Detailed Description

This script generates the data for the two stage gene expression example using the Stochastic Simulation Algorithm (SSA) implemented in the toolbox CERENA.

7.69 examples/two_stage_exandintrinsic/main_MA.m File Reference

This script generates the model incorporating the moment approximation, estimates the parameters and calculates the profile likelihoods.

Functions

- noret::substitute [main_MA](#) ()

This script generates the model incorporating the moment approximation, estimates the parameters and calculates the profile likelihoods.

7.69.1 Detailed Description

This script generates the model incorporating the moment approximation, estimates the parameters and calculates the profile likelihoods.

7.70 examples/two_stage_exandintrinsic/main_MA_1subpop.m File Reference

This script generates the model incorporating the moment approximation, estimates the parameters and calculates the profile likelihoods.

Functions

- noret::substitute [main_MA_1subpop](#) ()

This script generates the model incorporating the moment approximation, estimates the parameters and calculates the profile likelihoods.

7.70.1 Detailed Description

This script generates the model incorporating the moment approximation, estimates the parameters and calculates the profile likelihoods.

7.71 examples/two_stage_exandintrinsic/main_MA_1subpop_extrinsic.m File Reference

This script generates the model incorporating the moment approximation, estimates the parameters and calculates the profile likelihoods.

Functions

- noret::substitute [main_MA_1subpop_extrinsic \(\)](#)

This script generates the model incorporating the moment approximation, estimates the parameters and calculates the profile likelihoods.

7.71.1 Detailed Description

This script generates the model incorporating the moment approximation, estimates the parameters and calculates the profile likelihoods.

7.72 examples/two_stage_exandintrinsic/main_RRE.m File Reference

This script generates the model incorporating only the means by reaction rate equations, estimates the parameters and calculates the profile likelihoods.

Functions

- noret::substitute [main_RRE \(\)](#)

This script generates the model incorporating only the means by reaction rate equations, estimates the parameters and calculates the profile likelihoods.

7.72.1 Detailed Description

This script generates the model incorporating only the means by reaction rate equations, estimates the parameters and calculates the profile likelihoods.

7.73 examples/two_stage_exandintrinsic/simulation/compile_simFiles_geneExp.m File Reference

This script generate the simulation files for the example of two stage gene expression including intrinsic and extrinsic noise. This requires the MATLAB Toolbox CERENA. The `_syms.m` files are generate using CERENA and the model definition files [modelDef_geneExp.m](#) for the reaction rate equations and the moment approximation, and [modelDef_geneExp_extrinsic.m](#) for the moment approximation with additional extrinsic noise.

Functions

- noret::substitute [compile_simFiles_geneExp \(\)](#)

This script generate the simulation files for the example of two stage gene expression including intrinsic and extrinsic noise. This requires the MATLAB Toolbox CERENA. The `_syms.m` files are generate using CERENA and the model definition files [modelDef_geneExp.m](#) for the reaction rate equations and the moment approximation, and [modelDef_geneExp_extrinsic.m](#) for the moment approximation with additional extrinsic noise.

7.73.1 Detailed Description

This script generate the simulation files for the example of two stage gene expression including intrinsic and extrinsic noise. This requires the MATLAB Toolbox CERENA. The `_syms.m` files are generate using CERENA and the model definition files `modelDef_geneExp.m` for the reaction rate equations and the moment approximation, and `modelDef_geneExp_extrinsic.m` for the moment approximation with additional extrinsic noise.

7.74 examples/two_stage_exandintrinsic/simulation/MEC_2_LD_2_c_geneExp_extrinsic_syms.m File Reference

This function provides the model definition for the moment approximation with additional extrinsic noise of the two stage gene expression. It is generated by the toolbox CERENA.

Functions

- `mlhsInnerSubst< matlabtypesubstitute, model > MEC_2_LD_2_c_geneExp_extrinsic_syms` (matlabtype-substitute varargin)

This function provides the model definition for the moment approximation with additional extrinsic noise of the two stage gene expression. It is generated by the toolbox CERENA.

7.74.1 Detailed Description

This function provides the model definition for the moment approximation with additional extrinsic noise of the two stage gene expression. It is generated by the toolbox CERENA.

7.74.2 Function Documentation

7.74.2.1 `mlhsInnerSubst< matlabtypesubstitute, model > MEC_2_LD_2_c_geneExp_extrinsic_syms (matlabtypesubstitute varargin)`

This function provides the model definition for the moment approximation with additional extrinsic noise of the two stage gene expression. It is generated by the toolbox CERENA.

Generated fields of model:

Definition at line 17 of file `MEC_2_LD_2_c_geneExp_extrinsic_syms.m`.

7.75 examples/two_stage_exandintrinsic/simulation/MEC_2_LD_2_c_geneExp_MA_syms.m File Reference

This function provides the model definition for the moment approximation of the two stage gene expression. It is generated by the toolbox CERENA.

Functions

- `mlhsInnerSubst< matlabtypesubstitute, model > MEC_2_LD_2_c_geneExp_MA_syms` (`matlabtypesubstitute varargin`)

This function provides the model definition for the moment approximation of the two stage gene expression. It is generated by the toolbox CERENA.

7.75.1 Detailed Description

This function provides the model definition for the moment approximation of the two stage gene expression. It is generated by the toolbox CERENA.

7.75.2 Function Documentation

7.75.2.1 `mlhsInnerSubst< matlabtypesubstitute, model > MEC_2_LD_2_c_geneExp_MA_syms (matlabtypesubstitute varargin)`

This function provides the model definition for the moment approximation of the two stage gene expression. It is generated by the toolbox CERENA.

Generated fields of model:

Definition at line 17 of file MEC_2_LD_2_c_geneExp_MA_syms.m.

7.76 examples/two_stage_exandintrinsic/simulation/modelDef_geneExp.m File Reference

Model definition of the reaction network for the two stage gene expression.

Functions

- `noret::substitute modelDef_geneExp ()`

Model definition of the reaction network for the two stage gene expression.

7.76.1 Detailed Description

Model definition of the reaction network for the two stage gene expression.

7.77 examples/two_stage_exandintrinsic/simulation/modelDef_geneExp_extrinsic.m File Reference

Model definition of the reaction network for the two stage gene expression with additional extrinsic noise.

Functions

- `noret::substitute modelDef_geneExp_extrinsic ()`

Model definition of the reaction network for the two stage gene expression with additional extrinsic noise.

7.77.1 Detailed Description

Model definition of the reaction network for the two stage gene expression with additional extrinsic noise.

7.78 examples/two_stage_exandintrinsic/simulation/RRE_geneExp_RRE_syms.m File Reference

This function provides the model definition for the reaction rate equations of the two stage gene expression. It is generated by the toolbox CERENA.

Functions

- `mlhsInnerSubst< matlabtypesubstitute, model > RRE_geneExp_RRE_syms (matlabtypesubstitute varargin)`

This function provides the model definition for the reaction rate equations of the two stage gene expression. It is generated by the toolbox CERENA.

7.78.1 Detailed Description

This function provides the model definition for the reaction rate equations of the two stage gene expression. It is generated by the toolbox CERENA.

7.78.2 Function Documentation

7.78.2.1 `mlhsInnerSubst< matlabtypesubstitute, model > RRE_geneExp_RRE_syms (matlabtypesubstitute varargin)`

This function provides the model definition for the reaction rate equations of the two stage gene expression. It is generated by the toolbox CERENA.

Generated fields of model:

Definition at line 17 of file RRE_geneExp_RRE_syms.m.

7.79 generateODEMM.m File Reference

This function generates a file that defines the ODE-constrained mixture model.

Functions

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

This function generates a file that defines the ODE-constrained mixture model.

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

7.79.1 Detailed Description

This function generates a file that defines the ODE-constrained mixture model.

7.79.2 Function Documentation

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

This function generates a file that defines the 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 (options for the generation)

Required fields of M:

- `name` -- name of the model
- `model` -- simulation file with input (T,theta,u) (e.g., generated by amiwrap of the toolbox AMICI), the first output needs to be the status of the simulation (whether it failed or not) the 4th the simulation output, and the 6th the sensitivities

- `sim_type` -- simulation type ('RRE' for a mechanistic description of the mean, e.g., by reaction rate equations, 'HO' for a mechanistic description of the mean and covariance, e.g., by moment-closure approximation or sigma-point approximation)
- `n_subpop` -- number of subpopulations
- `distribution{s,e}` — distribution assumption
 - = 'norm' for normal distribution assumption
 - = 'logn_median' for log-normal distribution assumption when mean of simulation linked to median of distribution
 - = 'logn_mean' for log-normal distribution assumption when mean of simulation is % linked to mean of distribution
- `mean_ind{s,e}` — indices of simulation output describing the mean of the measurand(s) of experiment e
- `var_ind{s,e}` — indices of simulation output describing the variance (empty if RREs used)
- `u{s,e}` — input vector describing differences between subpopulations and experiments
- `sym` -- symbolic description of properties of the model with fields
 - `w{s,e}`: weights of subpopulation s in experiment e
 - `theta`: parameter needed for simulation of individual subpopulations
 - `scaling{r,e}`: scaling factor for replicate r in experiment e, if the replicates are not considered separately, use `r=1`
 - `offset{r,e}`: offset parameter for replicate r in experiment e

Required fields of D:

- `conditions` -- obtained by [collectConditions.m](#)

Required fields of parameters:

Optional fields of options:

- `write_parameter` -- write parameter definition in file (true by default)
- `measurement_noise` -- if measurement noise is included
 - = true
 - = false (default)
- `replicates` -- if individual replicates are modeled = true
 - = false (default)
- `sigmas` -- parametrization of the variance in case of using only a mechanistic description of the mean
 - = 'condition-dependent': (default) assign sigma for every time point
 - = 'time-independent': sigma stays the same for subpopulation and some dosage
 - = 'only-one': only one sigma for everything

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

Referenced by `generate_ECM_models()`, `generate_nosubpop_file()`, `generate_subpop_files()`, `generate_subpop_TrkA()`, `main_MA()`, `main_MA_1subpop()`, `main_MA_1subpop_extrinsic()`, `main_oneStage_SP_1D()`, `main_oneStage_SP_2D()`, `main_RRE()`, `model_SP_all()`, `model_SP_k1()`, `model_SP_k1k2()`, `model_SP_k1k3()`, `model_SP_k2()`, `model_SP_k2k3()`, `model_SP_k3()`, and `models_RRE()`.

7.80 getLognMeanVar.m File Reference

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

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 μ and Σ .

7.80.1 Detailed Description

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

7.80.2 Function Documentation

- 7.80.2.1 `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 μ and Σ .

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 μ and Σ
<i>n_dim</i>	dimension of the multivariate log-normal distribution
<i>varargin</i>	<p>1 getLognMeanVar (..., dZdtheta)</p> <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 derivative)

Return values

<i>Zout</i>	(n_dim + n_dim(n_dim+1)/2) x n_t vector with mean and coariance
<i>dZdthetaout</i>	derivative of Zout

Definition at line 17 of file getLognMeanVar.m.

Referenced by logLikelihood(), and plotODEMM().

7.81 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, vararginout > > 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.

7.81.1 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.

7.81.2 Function Documentation

- 7.81.2.1 `mlhsSubst< mlhsInnerSubst< matlabtypesubstitute, parameters >,mlhsInnerSubst< matlabtypesubstitute, conditions >,mlhsInnerSubst< matlabtypesubstitute, vararginout > > 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 collectConditions.m)
<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) • M model struct (see generateODEMM.m)

Return values

<i>parameters</i>	updated parameters struct
<i>conditions</i>	updated conditions struct
<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` -- boundaries for optimization for the sigma parameters with fields
 - `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`.

Referenced by `main_RRE()`, and `models_RRE()`.

7.82 `getScalingFactors.m` File Reference

Calculates scaling factors for replicates such that the distance between the means in log-space are minimal.

Functions

- `mlhsInnerSubst` < `matlabtypesubstitute`, `s` > `getScalingFactors` (`matlabtypesubstitute` `varargin`)
Calculates scaling factors for replicates such that the distance between the means in log-space are minimal.

7.82.1 Detailed Description

Calculates scaling factors for replicates such that the distance between the means in log-space are minimal.

7.82.2 Function Documentation**7.82.2.1 `mlhsInnerSubst` < `matlabtypesubstitute`, `s` > `getScalingFactors` (`matlabtypesubstitute` `varargin`)**

Calculates scaling factors for replicates such that the distance between the means in log-space are minimal.

USAGE

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

Parameters

<i>varargin</i>	<pre>1 getScalingFactors (ExpC)</pre> <p><i>Required Parameters for varargin:</i></p> <ul style="list-style-type: none"> • ExpC struct of experiments
-----------------	--

Return values

<i>s</i>	(1 x n_r) vector including scaling factor for every replicate
----------	---

Required fields of ExpC:

- `name` -- string specifying the conditions
- `time` -- time point of measurement
- `stimulus` -- stimulus for measurement
- `replicate` -- struct of replicates
 - `name`: string specifying the replicate
 - `measurands`: names of measurands
 - `ndata`: matrices under different conditions (one row represents one observed cell with the data in the order of the measurands. The different rows provide measurement data for different cells)

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

Referenced by `load_datasets_PDL_Coll()`.

7.83 getSigmaPointApp_status_mod.m File Reference

Modified version of the `getSigmaPointApp.m` function of the SPToolbox.

Functions

- `mlhsSubst< mlhsInnerSubst< matlabtypesubstitute, status >,mlhsInnerSubst< matlabtypesubstitute, SP >`
`> getSigmaPointApp_status_mod (matlabtypesubstitute varargin)`
Modified version of the `getSigmaPointApp.m` function of the SPToolbox.

7.83.1 Detailed Description

Modified version of the `getSigmaPointApp.m` function of the SPToolbox.

7.83.2 Function Documentation

7.83.2.1 mlhsSubst< mlhsInnerSubst< matlabtypesubstitute, status >,mlhsInnerSubst< matlabtypesubstitute, SP > >
getSigmaPointApp_status_mod (matlabtypesubstitute *varargin*)

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

Generated fields of SP:

Definition at line 17 of file getSigmaPointApp_status_mod.m.

Referenced by testSigmaPointApp_mod().

7.84 install_ODEMM.m File Reference

Script that adds the required paths to the MATLAB search path.

Functions

- noret::substitute [install_ODEMM](#) ()
Script that adds the required paths to the MATLAB search path.

7.84.1 Detailed Description

Script that adds the required paths to the MATLAB search path.

7.85 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.

7.85.1 Detailed Description

Functions to set font sizes and colors for the visualization.

7.86 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.

7.86.1 Detailed Description

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

7.86.2 Function Documentation

- 7.86.2.1 `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,I)
[...] = 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>	<pre>1 logLikelihood (..., options, conditions, I)</pre> <p><i>Required Parameters for varargin:</i></p> <ul style="list-style-type: none"> • options struct • conditions generated by function collectConditions.m • I 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 ($n_t \times n_{obs} \times n_{theta}$)
- `mean_ind` -- indices of output for mean
- `var_ind` -- indices of output for variances (empty if using RREs)
- `theta` -- parameters needed for simulation dependend 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': log-normal distribution assumption, mean of simulation linked to median of distribution
= 'logn_mean': log-normal 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 μ for subpopulation s and experiment e
- `dmudxi{s,e}` — gradient of μ
- `sigma{s,e}` — specification of mixture parameter σ (M.Sigma in multivariate case (covariance matrix))
- `dsigmadx{i}{s,e}` — gradient of sigma (M.dSigmadxi in multivariate case)
- `w{s,e}` — specification of weights w_s
- `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
- `t` -- $1 \times n_t$ vector of timepoints
- `u` -- $n_{maxu} \times n_u$ vector of inputs with `n_maxu`: maximal number of inputs simulatenously used
- `y` -- $n_u \times n_t \times n_{cells} \times n_{dim}$ data matrix (only needed if replicates are merged and already scaled), `dim` is the dimension of the measurement
- `c` -- $n_{subpop} \times (n_u + n_{differences})$ corresponding condition (automatically added by calling `collectCondition.m`)
- `replicate(r).y` — $n_u \times n_t \times n_{cells} \times n_{dim}$ 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()`.

Referenced by `main_MA()`, `main_MA_1subpop()`, `main_MA_1subpop_extrinsic()`, `main_oneStage_SP_1D()`, `main_oneStage_SP_2D()`, `main_RRE()`, `main_subpopulation_analysis()`, `model_SP_all()`, `model_SP_k1()`, `model_SP_k1k2()`, `model_SP_k1k3()`, `model_SP_k2()`, `model_SP_k2k3()`, `model_SP_k3()`, `models_RRE()`, `run_↵fitting_subpop()`, `run_fittings_ECM()`, `run_logmarg_subpop()`, `run_lppd_cr()`, `run_lppd_subpop()`, `run_profile_EC↵M()`, and `run_thermIntegration_cr()`.

7.87 plotODEMM.m File Reference

Routine to plot the ODE-constrained mixture model.

Functions

- `mlhsInnerSubst< matlabtypesubstitute, vararginout > plotODEMM` (matlabtypesubstitute varargin)
Routine to plot the ODE-constrained mixture model.
- `mlhsInnerSubst< matlabtypesubstitute, str_dose > mtoc_subst_plotODEMM_m_tsbus_cotm_getStr↵
Dose` (matlabtypesubstitute D, matlabtypesubstitute e, matlabtypesubstitute d)
- `noret::substitute mtoc_subst_plotODEMM_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_plotODEMM_m_tsbus_cotm_set↵
YminmaxHists` (matlabtypesubstitute D, matlabtypesubstitute e, matlabtypesubstitute d, matlabtypesubstitute options, matlabtypesubstitute ind)
- `noret::substitute mtoc_subst_plotODEMM_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)

7.87.1 Detailed Description

Routine to plot the ODE-constrained mixture model.

7.87.2 Function Documentation

7.87.2.1 `mlhsInnerSubst< matlabtypesubstitute, vararginout > plotODEMM (matlabtypesubstitute varargin)`

Routine to plot the ODE-constrained mixture model.

USAGE

```
[...] = plotODEMM(D,M,xi)
[...] = plotODEMM(D,M,xi,options)
[...] = plotODEMM(D,M,xi,options,fh)
[fh] = plotODEMM(...)
[fh,fhm] = plotODEMM(...)
```

Parameters

<i>varargin</i>	<ul style="list-style-type: none"> • D: data struct • M: model struct • xi: parameter vector • options: plotting options • fh: figure handle where the plots are added
-----------------	---

Return values

<i>fh</i>	struct of function handles for each data set
<i>fhm</i>	struct of function handles for the plots of the marginals

Definition at line 17 of file plotODEMM.m.

References collectConditions(), and getLognMeanVar().

Referenced by main_MA(), main_MA_1subpop(), main_MA_1subpop_extrinsic(), main_RRE(), plot_cr(), plot_cr↔_SIAppendix(), plot_ECM_fit(), plot_fit_oneStage(), and plot_subpop_fit().

7.88 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.

7.88.1 Detailed Description

Help function to print parameters names and values.

7.88.2 Function Documentation

7.88.2.1 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 together with parameter names

Required fields of parameters:

- name -- struct with names of parameters

Definition at line 17 of file printParams.m.

7.89 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.

7.89.1 Detailed Description

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

7.89.2 Function Documentation**7.89.2.1 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.

References getSigmaPointApp_status_mod().

Index

CR_log_syms
 CR_log_syms.m, 33
CR_log_syms.m
 CR_log_syms, 33
CR_syms
 CR_syms.m, 33
CR_syms.m
 CR_syms, 33
collectConditions
 collectConditions.m, 11
collectConditions.m, 10
 collectConditions, 11
computeMixtureProbability
 computeMixtureProbability.m, 12
computeMixtureProbability.m, 12
 computeMixtureProbability, 12
conversion_reaction/main_singlecell_prediction.m
 main_singlecell_prediction, 24

distributions/logn/func_Sigma_logn.m, 17
distributions/logn/func_dSigmadxi_logn.m, 16
distributions/logn/func_dmudxi_logn_mean.m, 13
distributions/logn/func_dmudxi_logn_median.m, 14
distributions/logn/func_dsigma2dxi_logn.m, 15
distributions/logn/logoflognpdf.m, 17
distributions/logn/logofmvnpdf.m, 18
distributions/norm/func_Sigma_norm.m, 21
distributions/norm/func_dSigmadxi_norm.m, 20
distributions/norm/func_dmudxi_norm.m, 18
distributions/norm/func_dsigma2dxi_norm.m, 19
distributions/norm/logofnormpdf.m, 21

examples/ECM_differences/data/load_datasets_PDL_←
 _Coll.m, 37
examples/ECM_differences/generate_ECM_models.m, 37
examples/ECM_differences/main_ECM_differences.m, 37
examples/ECM_differences/plot_ECM_fit.m, 38
examples/ECM_differences/plot_ECM_ranks.m, 38
examples/ECM_differences/plot_ECM_valid.m, 38
examples/ECM_differences/run_fittings_ECM.m, 39
examples/ECM_differences/run_profile_ECM.m, 39
examples/conversion_reaction/data/generate_data_←
 cr.m, 22
examples/conversion_reaction/main_optimization_cr.m, 23
examples/conversion_reaction/main_sampling_cr.m, 23
examples/conversion_reaction/main_singlecell_←
 prediction.m, 23
examples/conversion_reaction/model_SP_all.m, 24
examples/conversion_reaction/model_SP_k1.m, 25
examples/conversion_reaction/model_SP_k1k2.m, 25
examples/conversion_reaction/model_SP_k1k3.m, 26
examples/conversion_reaction/model_SP_k2.m, 26
examples/conversion_reaction/model_SP_k2k3.m, 27
examples/conversion_reaction/model_SP_k3.m, 27
examples/conversion_reaction/models_RRE.m, 28
examples/conversion_reaction/plot_cr.m, 28
examples/conversion_reaction/plot_cr_SIAppendix.m, 28
examples/conversion_reaction/plot_cr_variability_←
 Reduction.m, 29
examples/conversion_reaction/run_lppd_cr.m, 29
examples/conversion_reaction/run_thermIntegration_←
 cr.m, 30
examples/conversion_reaction/sclogLikelihood_cr.m, 31
examples/conversion_reaction/simulation/CR_log_←
 syms.m, 32
examples/conversion_reaction/simulation/CR_syms.m, 33
examples/conversion_reaction/simulation/generate_←
 simFiles_cr.m, 34
examples/differential_protein_expression/data/generate_←
 _data_oneStage.m, 34
examples/differential_protein_expression/main_one_←
 Stage_SP_1D.m, 34
examples/differential_protein_expression/main_one_←
 Stage_SP_2D.m, 35
examples/differential_protein_expression/plot_fit_one_←
 Stage.m, 35
examples/differential_protein_expression/plot_←
 uncertainty_oneStage.m, 36
examples/differential_protein_expression/simulation/generate_←
 _simFile_oneStage.m, 36
examples/differential_protein_expression/simulation/one_←
 Stage_syms.m, 36
examples/subpopulation_differences/dephosphorylation/exp_←
 decay.m, 40
examples/subpopulation_differences/dephosphorylation/import_←
 _dephospho_data.m, 40
examples/subpopulation_differences/dephosphorylation/main_←
 _analysis_dephospho.m, 40
examples/subpopulation_differences/dephosphorylation/plot_←
 _subpop_dephosphp_supplement.m, 41
examples/subpopulation_differences/generate_←
 nosubpop_file.m, 41
examples/subpopulation_differences/generate_←
 subpop_TrkA.m, 42
examples/subpopulation_differences/generate_←
 subpop_files.m, 41
examples/subpopulation_differences/main_singlecell_←
 prediction.m, 24
examples/subpopulation_differences/main_subpopulation_←
 analysis.m, 42
examples/subpopulation_differences/plot_subpop_fit.m, 42
examples/subpopulation_differences/plot_subpop_←
 profiles.m, 42
examples/subpopulation_differences/plot_subpop_←
 valid.m, 43

- examples/subpopulation_differences/run_fitting_↔
subpop.m, 43
- examples/subpopulation_differences/run_logmarg_↔
subpop.m, 44
- examples/subpopulation_differences/run_lppd_↔
subpop.m, 44
- examples/subpopulation_differences/simulation/OD_↔
Emodel_syms_sPsET_loglog.m, 45
- examples/subpopulation_differences/simulation/generate_↔
_simFile_ErkSignaling.m, 45
- examples/two_stage_exandintrinsic/data/generate_↔
data_twostage.m, 45
- examples/two_stage_exandintrinsic/main_MA.m, 46
- examples/two_stage_exandintrinsic/main_MA_↔
1subpop.m, 46
- examples/two_stage_exandintrinsic/main_MA_↔
1subpop_extrinsic.m, 47
- examples/two_stage_exandintrinsic/main_RRE.m, 47
- examples/two_stage_exandintrinsic/simulation/MEC_↔
2_LD_2_c_geneExp_MA_syms.m, 48
- examples/two_stage_exandintrinsic/simulation/MEC_↔
2_LD_2_c_geneExp_extrinsic_syms.m, 48
- examples/two_stage_exandintrinsic/simulation/RRE_↔
geneExp_RRE_syms.m, 50
- examples/two_stage_exandintrinsic/simulation/compile_↔
_simFiles_geneExp.m, 47
- examples/two_stage_exandintrinsic/simulation/model_↔
Def_geneExp.m, 49
- examples/two_stage_exandintrinsic/simulation/model_↔
Def_geneExp_extrinsic.m, 49
- func_Sigma_logn
func_Sigma_logn.m, 17
- func_Sigma_logn.m
func_Sigma_logn, 17
- func_Sigma_norm
func_Sigma_norm.m, 21
- func_Sigma_norm.m
func_Sigma_norm, 21
- func_dSigmadxi_logn
func_dSigmadxi_logn.m, 16
- func_dSigmadxi_logn.m
func_dSigmadxi_logn, 16
- func_dSigmadxi_norm
func_dSigmadxi_norm.m, 20
- func_dSigmadxi_norm.m
func_dSigmadxi_norm, 20
- func_dmudxi_logn_mean
func_dmudxi_logn_mean.m, 13
- func_dmudxi_logn_mean.m
func_dmudxi_logn_mean, 13
- func_dmudxi_logn_median
func_dmudxi_logn_median.m, 14
- func_dmudxi_logn_median.m
func_dmudxi_logn_median, 14
- func_dmudxi_norm
func_dmudxi_norm.m, 18
- func_dmudxi_norm.m
func_dmudxi_norm, 18
- func_dsigma2dxi_logn
func_dsigma2dxi_logn.m, 15
- func_dsigma2dxi_logn.m
func_dsigma2dxi_logn, 15
- func_dsigma2dxi_norm
func_dsigma2dxi_norm.m, 19
- func_dsigma2dxi_norm.m
func_dsigma2dxi_norm, 19
- generateODEMM.m, 50
- generateODEMM, 51
- generateODEMM
generateODEMM.m, 51
- getLognMeanVar
getLognMeanVar.m, 53
- getLognMeanVar.m, 52
- getLognMeanVar, 53
- getRREsigmas
getRREsigmas.m, 54
- getRREsigmas.m, 53
- getRREsigmas, 54
- getScalingFactors
getScalingFactors.m, 55
- getScalingFactors.m, 55
- getScalingFactors, 55
- getSigmaPointApp_status_mod
getSigmaPointApp_status_mod.m, 57
- getSigmaPointApp_status_mod.m, 56
- getSigmaPointApp_status_mod, 57
- install_ODEMM.m, 57
- load_plot_settings.m, 57
- logLikelihood
logLikelihood.m, 58
- logLikelihood.m, 57
- logLikelihood, 58
- logoflognpdf
logoflognpdf.m, 18
- logoflognpdf.m
logoflognpdf, 18
- logofnormpdf
logofnormpdf.m, 22
- logofnormpdf.m
logofnormpdf, 22
- MEC_2_LD_2_c_geneExp_MA_syms
MEC_2_LD_2_c_geneExp_MA_syms.m, 49
- MEC_2_LD_2_c_geneExp_MA_syms.m
MEC_2_LD_2_c_geneExp_MA_syms, 49
- MEC_2_LD_2_c_geneExp_extrinsic_syms
MEC_2_LD_2_c_geneExp_extrinsic_syms.m, 48
- MEC_2_LD_2_c_geneExp_extrinsic_syms.m
MEC_2_LD_2_c_geneExp_extrinsic_syms, 48
- main_singlecell_prediction
conversion_reaction/main_singlecell_prediction.m,
24
- ODEmodel_syms_sPsET_loglog

- ODEmodel_syms_sPsET_loglog.m, [45](#)
- ODEmodel_syms_sPsET_loglog.m
 - ODEmodel_syms_sPsET_loglog, [45](#)
- oneStage_syms
 - oneStage_syms.m, [36](#)
- oneStage_syms.m
 - oneStage_syms, [36](#)
- plot_cr_variabilityReduction
 - plot_cr_variabilityReduction.m, [29](#)
- plot_cr_variabilityReduction.m
 - plot_cr_variabilityReduction, [29](#)
- plotODEMM.m, [60](#)
 - plotODEMM, [60](#)
- plotODEMM
 - plotODEMM.m, [60](#)
- printParams
 - printParams.m, [61](#)
- printParams.m, [61](#)
 - printParams, [61](#)
- RRE_geneExp_RRE_syms
 - RRE_geneExp_RRE_syms.m, [50](#)
- RRE_geneExp_RRE_syms.m
 - RRE_geneExp_RRE_syms, [50](#)
- run_fitting_subpop
 - run_fitting_subpop.m, [44](#)
- run_fitting_subpop.m
 - run_fitting_subpop, [44](#)
- run_fittings_ECM.m
 - run_fittings_ECM, [39](#)
- run_fittings_ECM
 - run_fittings_ECM.m, [39](#)
- run_lppd_cr
 - run_lppd_cr.m, [30](#)
- run_lppd_cr.m
 - run_lppd_cr, [30](#)
- run_thermIntegration_cr
 - run_thermIntegration_cr.m, [31](#)
- run_thermIntegration_cr.m
 - run_thermIntegration_cr, [31](#)
- sclogLikelihood_cr
 - sclogLikelihood_cr.m, [32](#)
- sclogLikelihood_cr.m
 - sclogLikelihood_cr, [32](#)
- testSigmaPointApp_mod
 - testSigmaPointApp_mod.m, [62](#)
- testSigmaPointApp_mod.m, [62](#)
 - testSigmaPointApp_mod, [62](#)