

Package ‘AIUQ’

May 25, 2026

Type Package

Title Ab Initio Uncertainty Quantification

Version 0.5.4

Author Yue He [aut],
Xubo Liu [aut],
Tong Lin [aut],
Mengyang Gu [aut, cre]

Maintainer Mengyang Gu <mengyang@pstat.ucsb.edu>

Description Uncertainty quantification and inverse estimation by probabilistic generative models from the beginning of the data analysis. An example is a Fourier basis method for inverse estimation in scattering analysis of microscopy videos. It does not require specifying a certain range of Fourier bases and it substantially reduces computational cost via the generalized Schur algorithm. See the reference: Mengyang Gu, Yue He, Xubo Liu and Yimin Luo (2023), <[doi:10.48550/arXiv.2309.02468](https://doi.org/10.48550/arXiv.2309.02468)>.

License GPL (>= 3)

Encoding UTF-8

RoxygenNote 7.3.1

Imports fftwtools (>= 0.9.11), SuperGauss (>= 2.0.3), methods, plot3D (>= 1.4), RobustGaSP, geometry

NeedsCompilation no

Suggests knitr, rmarkdown, bioimager

VignetteBuilder knitr

Repository CRAN

Date/Publication 2026-05-25 19:20:02 UTC

Contents

aniso_SAM	2
aniso_SAM-class	5
aniso_simulation	6
aniso_simulation-class	8

get_dqt	9
Get_Gp_Gpp_GSER	10
get_isf	11
modeled_dqt	12
modeled_isf	13
plot_intensity	14
plot_MSD	15
plot_traj	16
SAM	17
SAM-class	21
show.aniso_sam	23
show.aniso_simulation	24
show.sam	25
show.simulation	26
simulation	27
simulation-class	29

Index	30
--------------	-----------

aniso_SAM	<i>Scattering analysis of microscopy for anisotropic processes</i>
-----------	--

Description

Fast parameter estimation in scattering analysis of microscopy for anisotropic processes, using AIUQ method.

Usage

```
aniso_SAM(
  intensity = NA,
  intensity_str = "T_SS_mat",
  pxsz = 1,
  sz = c(NA, NA),
  mindt = 1,
  AIUQ_thr = c(1, 1),
  model_name = "BM",
  sigma_0_2_ini = NaN,
  param_initial = NA,
  num_optim = 1,
  msd_fn = NA,
  msd_grad_fn = NA,
  num_param = NA,
  uncertainty = FALSE,
  M = 50,
  sim_object = NA,
  msd_truth = NA,
  method = "AIUQ",
```

```

    index_q_AIUQ = NA,
    message_out = TRUE,
    square = FALSE
)

```

Arguments

intensity	intensity profile. See 'Details'.
intensity_str	structure of the intensity profile, options from ('SST_array', 'S_ST_mat', 'T_SS_mat'). See 'Details'.
pxsz	size of one pixel in unit of micron, 1 for simulated data
sz	frame size of the intensity profile in x and y directions, number of pixels contained in each frame equals sz_x by sz_y.
mindt	minimum lag time, 1 for simulated data
AIUQ_thr	threshold for wave number selection, numeric vector of two elements with values between 0 and 1. See 'Details'.
model_name	fitted model, options from ('BM', 'OU', 'FBM', 'OU+FBM', 'user_defined'), with Brownian motion as the default model. See 'Details'.
sigma_0_2_ini	initial value for background noise. If NA, use minimum value of absolute square of intensity profile in reciprocal space.
param_initial	initial values for param estimation.
num_optim	number of optimization.
msd_fn	user defined mean squared displacement(MSD) structure, a function of parameters and lag times. NA if model_name is not 'user_defined'.
msd_grad_fn	gradient for user defined mean squared displacement structure. If NA, then numerical gradient will be used for parameter estimation in 'user_defined' model.
num_param	number of parameters need to be estimated in the intermediate scattering function, need to be non-NA value for user_defined' model.
uncertainty	a logical evaluating to TRUE or FALSE indicating whether parameter uncertainty should be computed.
M	number of particles. See 'Details'.
sim_object	NA or an S4 object of class simulation.
msd_truth	true MSD or reference MSD value.
method	methods for parameter estimation, options from ('AIUQ', 'DDM').
index_q_AIUQ	index range for wave number when using AIUQ method. See 'Details'.
message_out	a logical evaluating to TRUE or FALSE indicating whether or not to output the message.
square	a logical evaluating to TRUE or FALSE indicating whether or not to crop the original intensity profile into square image.

Details

For simulated data using `aniso_simulation` in AIUQ package, intensity will be automatically extracted from `aniso_simulation` class.

By default `intensity_str` is set to "T_SS_mat", a time by space x space matrix, which is the structure of intensity profile obtained from `aniso_simulation` class. For `intensity_str="SST_array"`, input intensity profile should be a space by space by time array, which is the structure from loading a tif file. For `intensity_str="S_ST_mat"`, input intensity profile should be a space by space x time matrix.

By default `AIUQ_thr` is set to `c(1, 1)`, uses information from all complete q rings. The first element affects maximum wave number selected, and second element controls minimum proportion of wave number selected. By setting 1 for the second element, if maximum wave number selected is less than the wave number length, then maximum wave number selected is coerced to use all wave number unless user defined another index range through `index_q_AIUQ`.

If `model_name` equals "user_defined", or NA (will coerced to "user_defined"), then `msd_fn` and `num_param` need to be provided for parameter estimation.

Number of particles `M` is set to 50 or automatically extracted from simulation class for simulated data using simulation in AIUQ package.

By default, using all wave vectors from complete q ring for both AIUQ, unless user defined index range through `index_q_AIUQ`.

Value

Returns an S4 object of class `aniso_SAM`.

Author(s)

Yue He [aut], Xubo Liu [aut], Tong Lin [aut], Mengyang Gu [aut, cre]

References

Gu, M., He, Y., Liu, X., & Luo, Y. (2023). Ab initio uncertainty quantification in scattering analysis of microscopy. arXiv preprint arXiv:2309.02468.

Gu, M., Luo, Y., He, Y., Helgeson, M. E., & Valentine, M. T. (2021). Uncertainty quantification and estimation in differential dynamic microscopy. *Physical Review E*, 104(3), 034610.

Cerbino, R., & Trappe, V. (2008). Differential dynamic microscopy: probing wave vector dependent dynamics with a microscope. *Physical review letters*, 100(18), 188102.

Examples

```
library(AIUQ)
# Example 1: Estimation for simulated data
set.seed(1)
aniso_sim = aniso_simulation(sz=100, len_t=100, model_name="BM", M=100, sigma_bm=c(0.5, 0.3))
show(aniso_sim)
plot_traj(object=aniso_sim)
aniso_sam = aniso_SAM(sim_object=aniso_sim, model_name="BM", AIUQ_thr = c(0.999, 0))
show(aniso_sam)
plot_MSD(aniso_sam, msd_truth = aniso_sam@msd_truth)
```

aniso_SAM-class *Anisotropic SAM class*

Description

S4 class for fast parameter estimation in scattering analysis of microscopy for anisotropic processes, using either AIUQ or DDM method.

Slots

pxsz numeric. Size of one pixel in unit of micron with default value 1.

mindt numeric. Minimum lag time with default value 1.

sz vector. Frame size of the intensity profile in x and y directions, number of pixels contained in each frame equals sz_x by sz_y.

len_t integer. Number of time steps.

len_q integer. Number of wave vector.

q vector. Wave vector in unit of μm^{-1} .

d_input vector. Sequence of lag times.

B_est_ini numeric. Estimation of B. This parameter is determined by the noise in the system. See "References".

A_est_ini vector. Estimation of A(q). Note this parameter is determined by the properties of the imaged material and imaging optics. See "References".

I_o_q_2_ori vector. Absolute square of Fourier transformed intensity profile, ensemble over time.

q_ori_ring_loc_unique_index list. List of location index of non-duplicate values for each q ring.

model_name character. Fitted model, options from "BM", "OU", "FBM", "OU+FBM", and "user_defined".

param_est matrix. Estimated parameters contained in MSD.

sigma_2_0_est vector. Estimated variance of background noise.

msd_est matrix. Estimated MSD.

uncertainty logical. A logical evaluating to TRUE or FALSE indicating whether parameter uncertainty should be computed.

msd_truth matrix. True MSD or reference MSD value.

sigma_2_0_truth vector. True variance of background noise, non NA for simulated data using simulation.

param_truth matrix. True parameters used to construct MSD, non NA for simulated data using aniso_simulation.

index_q vector. Selected index of wave vector.

I_q matrix. Fourier transformed intensity profile with structure "SS_T_mat".

AIC numeric. Akaike information criterion score.

mle numeric. Maximum log likelihood value.

msd_x_lower vector. Lower bound of 95% confidence interval of MSD in x directions.
 msd_x_upper vector. Upper bound of 95% confidence interval of MSD in x directions.
 msd_y_lower vector. Lower bound of 95% confidence interval of MSD in y directions.
 msd_y_upper vector. Upper bound of 95% confidence interval of MSD in y directions.
 param_uq_range matrix. 95% confidence interval for estimated parameters.

Author(s)

Yue He [aut], Xubo Liu [aut], Tong Lin [aut], Mengyang Gu [aut, cre]

References

Gu, M., He, Y., Liu, X., & Luo, Y. (2023). Ab initio uncertainty quantification in scattering analysis of microscopy. arXiv preprint arXiv:2309.02468.
 Gu, M., Luo, Y., He, Y., Helgeson, M. E., & Valentine, M. T. (2021). Uncertainty quantification and estimation in differential dynamic microscopy. *Physical Review E*, 104(3), 034610.
 Cerbino, R., & Trappe, V. (2008). Differential dynamic microscopy: probing wave vector dependent dynamics with a microscope. *Physical review letters*, 100(18), 188102.

aniso_simulation

Simulate anisotropic 2D particle movement

Description

Simulate anisotropic 2D particle movement from a user selected stochastic process, and output intensity profiles.

Usage

```
aniso_simulation(
  sz = c(200, 200),
  len_t = 200,
  M = 50,
  model_name = "BM",
  noise = "gaussian",
  I0 = 20,
  Imax = 255,
  pos0 = matrix(NaN, nrow = M, ncol = 2),
  rho = c(0.95, 0.9),
  H = c(0.4, 0.3),
  sigma_p = 2,
  sigma_bm = c(1, 0.5),
  sigma_ou = c(2, 1.5),
  sigma_fbm = c(2, 1.5)
)
```

Arguments

sz	frame size of simulated image with default $c(200, 200)$.
len_t	number of time steps with default 200.
M	number of particles with default 50.
model_name	stochastic process simulated, options from ('BM', 'OU', 'FBM', 'OU+FBM'), with default 'BM'.
noise	background noise, options from ('uniform', 'gaussian'), with default 'gaussian'.
I0	background intensity, value between 0 and 255, with default 20.
Imax	maximum intensity at the center of the particle, value between 0 and 255, with default 255.
pos0	initial position for M particles, matrix with dimension M by 2.
rho	correlation between successive step and previous step in O-U process, in x, y-directions. A vector of length 2 with values between 0 and 1, default $c(0.95, 0.9)$.
H	Hurst parameter of fractional Brownian Motion, in x, y-directions. A vector of length 2, value between 0 and 1, default $c(0.4, 0.3)$.
sigma_p	radius of the spherical particle (3sigma_p), with default 2.
sigma_bm	distance moved per time step of Brownian Motion, in x,y-directions. A vector of length 2 with default $c(1, 0.5)$.
sigma_ou	distance moved per time step of Ornstein–Uhlenbeck process, in x, y-directions. A vector of length 2 with default $c(2, 1.5)$.
sigma_fbm	distance moved per time step of fractional Brownian Motion, in x, y-directions. A vector of length 2 with default $c(2, 1.5)$.

Value

Returns an S4 object of class `anisotropic_simulation`.

Author(s)

Yue He [aut], Xubo Liu [aut], Tong Lin [aut], Mengyang Gu [aut, cre]

References

- Gu, M., He, Y., Liu, X., & Luo, Y. (2023). Ab initio uncertainty quantification in scattering analysis of microscopy. arXiv preprint arXiv:2309.02468.
- Gu, M., Luo, Y., He, Y., Helgeson, M. E., & Valentine, M. T. (2021). Uncertainty quantification and estimation in differential dynamic microscopy. *Physical Review E*, 104(3), 034610.
- Cerbino, R., & Trappe, V. (2008). Differential dynamic microscopy: probing wave vector dependent dynamics with a microscope. *Physical review letters*, 100(18), 188102.

Examples

```

library(AIUQ)
# -----
# Example 1: Simple diffusion for 200 images with
#           200 by 200 pixels and 50 particles
# -----
aniso_sim_bm = aniso_simulation()
show(aniso_sim_bm)

# -----
# Example 2: Simple diffusion for 100 images with
#           100 by 100 pixels and slower speed
# -----
aniso_sim_bm = aniso_simulation(sz=100,len_t=100,sigma_bm=c(0.5,0.1))
show(aniso_sim_bm)

# -----
# Example 3: Ornstein-Uhlenbeck process
# -----
aniso_sim_ou = aniso_simulation(model_name="OU")
show(aniso_sim_ou)

```

aniso_simulation-class

Anisotropic simulation class

Description

S4 class for anisotropic 2D particle movement simulation.

Details

intensity should has structure "T_SS_mat", matrix with dimension len_t by sz x sz.

pos should be the position matrix with dimension M x len_t. See [bm_particle_intensity](#), [ou_particle_intensity](#), [fbm_particle_intensity](#), [fbm_ou_particle_intensity](#).

Slots

sz vector. Frame size of the intensity profile, number of pixels contained in each frame equals sz[1] by sz[2].

len_t integer. Number of time steps.

noise character. Background noise, options from ('uniform','gaussian').

model_name character. Simulated stochastic process, options from ('BM','OU','FBM','OU+FBM').

M integer. Number of particles.

pxsz numeric. Size of one pixel in unit of micron, 1 for simulated data.

mindt numeric. Minimum lag time, 1 for simulated data.

pos matrix. Position matrix for particle trajectory, see 'Details'.

intensity matrix. Filled intensity profile, see 'Details'.

num_msd matrix. Numerical mean squared displacement (MSD).

param matrix. Parameters used to construct MSD.

theor_msd matrix. Theoretical MSD.

sigma_2_0 vector. Variance of background noise.

Author(s)

Yue He [aut], Xubo Liu [aut], Tong Lin [aut], Mengyang Gu [aut, cre]

References

Gu, M., He, Y., Liu, X., & Luo, Y. (2023). Ab initio uncertainty quantification in scattering analysis of microscopy. arXiv preprint arXiv:2309.02468.

Gu, M., Luo, Y., He, Y., Helgeson, M. E., & Valentine, M. T. (2021). Uncertainty quantification and estimation in differential dynamic microscopy. *Physical Review E*, 104(3), 034610.

Cerbino, R., & Trappe, V. (2008). Differential dynamic microscopy: probing wave vector dependent dynamics with a microscope. *Physical review letters*, 100(18), 188102.

get_dqt

Compute observed dynamic image structure function

Description

Compute observed dynamic image structure function (Dqt) using object of SAM class.

Usage

```
get_dqt(object, index_q = NA)
```

Arguments

object	an S4 object of class SAM
index_q	wavevector range used for computing Dqt

Value

A matrix of observed dynamic image structure function with dimension len_q by len_t-1.

Author(s)

Yue He [aut], Xubo Liu [aut], Tong Lin [aut], Mengyang Gu [aut, cre]

References

- Gu, M., He, Y., Liu, X., & Luo, Y. (2023). Ab initio uncertainty quantification in scattering analysis of microscopy. arXiv preprint arXiv:2309.02468.
- Gu, M., Luo, Y., He, Y., Helgeson, M. E., & Valentine, M. T. (2021). Uncertainty quantification and estimation in differential dynamic microscopy. *Physical Review E*, 104(3), 034610.
- Cerbino, R., & Trappe, V. (2008). Differential dynamic microscopy: probing wave vector dependent dynamics with a microscope. *Physical review letters*, 100(18), 188102.

Examples

```
library(AIUQ)
sim_bm = simulation(len_t=100,sz=100,sigma_bm=0.5)
show(sim_bm)
sam = SAM(sim_object = sim_bm)
show(sam)
Dqt = get_dqt(object=sam)
```

Get_Gp_Gpp_GSER

Estimate Viscoelastic Moduli by GSER

Description

Estimate the storage and loss moduli from mean squared displacement (MSD) using the generalized Stokes-Einstein relation. Because the transformation from MSD to the storage and loss moduli is nonlinear, log MSD curves are sampled and used to calculate storage and loss moduli. The pointwise medians of the sampled moduli are used as the estimates.

Usage

```
Get_Gp_Gpp_GSER(
  MSD_est,
  d_input,
  MSD_lower,
  MSD_upper,
  sim_object = NA,
  MSD_unit = 1,
  kB = 1.380649 * 10^{
    -23
  },
  Temp = 1,
  a = 1
)
```

Arguments

MSD_est	A numeric vector of estimated MSD values. This argument is used for compatibility and for computing theoretical moduli when a simulation object is provided. The estimated moduli are obtained from MSD_lower and MSD_upper.
d_input	A numeric vector of lag times.
MSD_lower	A numeric vector of lower bound of 95% confidence interval of MSD.
MSD_upper	A numeric vector of upper bound of 95% confidence interval of MSD.
sim_object	Optional simulation object. If provided, the function also returns the theoretical storage and loss moduli for the simulated data.
MSD_unit	Scaling factor applied to MSD values used to convert the estimated MSD to the desired physical unit before modulus estimation.
kB	Boltzmann constant.
Temp	Absolute temperature.
a	Particle radius.

Value

A list containing estimated frequency, storage modulus, and loss modulus. If `sim_object` is provided, the theoretical moduli are also returned.

get_isf *Compute empirical intermediate scattering function*

Description

Compute empirical intermediate scattering function (ISF) using object of SAM class.

Usage

```
get_isf(object, index_q = NA, msd_truth = NA)
```

Arguments

object	an S4 object of class SAM
index_q	wavevector range used for computing ISF
msd_truth	true or reference MSD

Value

A matrix of empirical intermediate scattering function with dimension `len_q` by `len_t-1`.

Author(s)

Yue He [aut], Xubo Liu [aut], Tong Lin [aut], Mengyang Gu [aut, cre]

References

- Gu, M., He, Y., Liu, X., & Luo, Y. (2023). Ab initio uncertainty quantification in scattering analysis of microscopy. arXiv preprint arXiv:2309.02468.
- Gu, M., Luo, Y., He, Y., Helgeson, M. E., & Valentine, M. T. (2021). Uncertainty quantification and estimation in differential dynamic microscopy. *Physical Review E*, 104(3), 034610.
- Cerbino, R., & Trappe, V. (2008). Differential dynamic microscopy: probing wave vector dependent dynamics with a microscope. *Physical review letters*, 100(18), 188102.

Examples

```
library(AIUQ)
sim_bm = simulation(len_t=100, sz=100, sigma_bm=0.5)
show(sim_bm)
sam = SAM(sim_object = sim_bm)
show(sam)
ISF = get_isf(object=sam)
```

modeled_dqt

Compute modeled dynamic image structure function

Description

Compute modeled dynamic image structure function (Dqt) using object of SAM class.

Usage

```
modeled_dqt(object, index_q = NA, uncertainty = FALSE)
```

Arguments

object	an S4 object of class SAM
index_q	wavevector range used for computing Dqt
uncertainty	logic evaluation

Value

A matrix of modeled dynamic image structure function with dimension len_q by len_t-1.

Author(s)

Yue He [aut], Xubo Liu [aut], Tong Lin [aut], Mengyang Gu [aut, cre]

References

- Gu, M., He, Y., Liu, X., & Luo, Y. (2023). Ab initio uncertainty quantification in scattering analysis of microscopy. arXiv preprint arXiv:2309.02468.
- Gu, M., Luo, Y., He, Y., Helgeson, M. E., & Valentine, M. T. (2021). Uncertainty quantification and estimation in differential dynamic microscopy. *Physical Review E*, 104(3), 034610.
- Cerbino, R., & Trappe, V. (2008). Differential dynamic microscopy: probing wave vector dependent dynamics with a microscope. *Physical review letters*, 100(18), 188102.

Examples

```
library(AIUQ)
sim_bm = simulation(len_t=100,sz=100,sigma_bm=0.5)
show(sim_bm)
sam = SAM(sim_object = sim_bm)
show(sam)
modeled_Dqt = modeled_dqt(object=sam)
```

modeled_isf

Compute modeled intermediate scattering function

Description

Compute modeled intermediate scattering function (ISF) using object of SAM class.

Usage

```
modeled_isf(object, index_q = NA)
```

Arguments

object	an S4 object of class SAM
index_q	wavevector range used for computing ISF

Value

A matrix of modeled intermediate scattering function with dimension len_q by len_t-1.

Author(s)

Yue He [aut], Xubo Liu [aut], Tong Lin [aut], Mengyang Gu [aut, cre]

References

Gu, M., He, Y., Liu, X., & Luo, Y. (2023). Ab initio uncertainty quantification in scattering analysis of microscopy. arXiv preprint arXiv:2309.02468.

Gu, M., Luo, Y., He, Y., Helgeson, M. E., & Valentine, M. T. (2021). Uncertainty quantification and estimation in differential dynamic microscopy. *Physical Review E*, 104(3), 034610.

Cerbino, R., & Trappe, V. (2008). Differential dynamic microscopy: probing wave vector dependent dynamics with a microscope. *Physical review letters*, 100(18), 188102.

Examples

```
library(AIUQ)
sim_bm = simulation(len_t=100,sz=100,sigma_bm=0.5)
show(sim_bm)
sam = SAM(sim_object = sim_bm)
show(sam)
modeled_ISF = modeled_isf(object=sam)
```

plot_intensity	<i>Plot 2D intensity</i>
----------------	--------------------------

Description

Function to plot 2D intensity profile for a certain frame, default is to plot the first frame. Input can be a matrix (2D) or an array (3D).

Usage

```
plot_intensity(
  intensity,
  intensity_str = "T_SS_mat",
  frame = 1,
  sz = NA,
  title = NA,
  color = FALSE
)
```

Arguments

intensity	intensity profile
intensity_str	structure of the intensity profile, options from ('SST_array', 'S_ST_mat', 'T_SS_mat', 'SS_T_mat'). See 'Details'.
frame	frame index
sz	frame size of simulated image with default c(200, 200).
title	main title of the plot. If NA, title is "intensity profile for frame n" with n being the frame index in frame.
color	a logical evaluating to TRUE or FALSE indicating whether a colorful plot is generated

Details

By default `intensity_str` is set to "T_SS_mat", a time by space x space matrix, which is the structure of intensity profile obtained from `simulation` class. For `intensity_str="SST_array"`, input intensity profile should be a space by space by time array, which is the structure from loading a tif file. For `intensity_str="S_ST_mat"`, input intensity profile should be a space by space x time matrix. For `intensity_str="SS_T_mat"`, input intensity profile should be a space x space by time matrix.

Value

2D plot in gray scale (or with color) of selected frame.

Author(s)

Yue He [aut], Xubo Liu [aut], Tong Lin [aut], Mengyang Gu [aut, cre]

Examples

```
library(AIUQ)
sim_bm = simulation(sz=100,len_t=100,sigma_bm=0.5)
show(sim_bm)
plot_intensity(sim_bm@intensity, sz=sim_bm@sz)
```

plot_MSD

Plot estimated MSD with uncertainty from SAM class

Description

Function to plot estimated MSD with uncertainty from SAM class, versus true mean squared displacement(MSD) or given reference values.

Usage

```
plot_MSD(object, msd_truth = NA, title = NA, log10 = TRUE)
```

Arguments

<code>object</code>	an S4 object of class SAM
<code>msd_truth</code>	a vector/matrix of true MSD or reference MSD value, default is NA
<code>title</code>	main title of the plot. If NA, title is "model_name" with <code>model_name</code> being a field in SAM class representing fitted model.
<code>log10</code>	a logical evaluating to TRUE or FALSE indicating whether a plot in log10 scale is generated

Value

A plot of estimated MSD with uncertainty versus truth/reference values.

Author(s)

Yue He [aut], Xubo Liu [aut], Tong Lin [aut], Mengyang Gu [aut, cre]

References

Gu, M., He, Y., Liu, X., & Luo, Y. (2023). Ab initio uncertainty quantification in scattering analysis of microscopy. arXiv preprint arXiv:2309.02468.

Gu, M., Luo, Y., He, Y., Helgeson, M. E., & Valentine, M. T. (2021). Uncertainty quantification and estimation in differential dynamic microscopy. *Physical Review E*, 104(3), 034610.

Examples

```
library(AIUQ)

## Simulate BM and get estimated parameters with uncertainty using BM model
# Simulation
set.seed(1)
sim_bm = simulation(sz=100, len_t=100, sigma_bm=0.5)
show(sim_bm)

# AIUQ method: fitting using BM model
sam = SAM(sim_object=sim_bm, uncertainty=TRUE, AIUQ_thr=c(0.999, 0))
show(sam)

plot_MSD(object=sam, msd_truth=sam@msd_truth) #in log10 scale
plot_MSD(object=sam, msd_truth=sam@msd_truth, log10=FALSE) #in real scale
```

plot_traj

Plot 2D particle trajectory

Description

Function to plot the particle trajectory after the simulation class has been constructed.

Usage

```
plot_traj(object, title = NA)
```

Arguments

object	an S4 object of class simulation
title	main title of the plot. If NA, title is "model_name with M particles" with model_name and M being field in simulation class.

Value

2D plot of particle trajectory for a given simulation from simulation class.

Author(s)

Yue He [aut], Xubo Liu [aut], Tong Lin [aut], Mengyang Gu [aut, cre]

Examples

```
library(AIUQ)
sim_bm = simulation(sz=100, len_t=100, sigma_bm=0.5)
show(sim_bm)
plot_traj(sim_bm)
```

SAM

Scattering analysis of microscopy

Description

Fast parameter estimation in scattering analysis of microscopy, using either AIUQ or DDM method.

Usage

```
SAM(
  intensity = NA,
  intensity_str = "T_SS_mat",
  pxsz = 1,
  sz = c(NA, NA),
  mindt = 1,
  AIUQ_thr = c(1, 1),
  model_name = "BM",
  sigma_0_2_ini = NaN,
  param_initial = NA,
  num_optim = 1,
  msd_fn = NA,
  msd_grad_fn = NA,
  num_param = NA,
  uncertainty = FALSE,
  M = 50,
  sim_object = NA,
  msd_truth = NA,
  method = "AIUQ",
  index_q_AIUQ = NA,
  index_q_DDM = NA,
  message_out = TRUE,
  A_neg = "abs",
  square = FALSE,
  output_dqt = FALSE,
  output_isf = FALSE,
  output_modeled_isf = FALSE,
  output_modeled_dqt = FALSE,
```

```

model_free_estimation = FALSE,
substack = FALSE,
estimate_moduli = FALSE,
MSD_unit = 1,
kB = 1.380649 * 10^{
  -23
},
Temp = NA,
a = NA
)

```

Arguments

intensity	intensity profile. See 'Details'.
intensity_str	structure of the intensity profile, options from ('SST_array', 'S_ST_mat', 'T_SS_mat'). See 'Details'.
pxsz	size of one pixel in unit of micron, 1 for simulated data
sz	frame size of the intensity profile in x and y directions, number of pixels contained in each frame equals sz_x by sz_y.
mindt	minimum lag time, 1 for simulated data
AIUQ_thr	threshold for wave number selection, numeric vector of two elements with values between 0 and 1. See 'Details'.
model_name	fitted model, options from ('BM', 'OU', 'FBM', 'OU+FBM', 'user_defined'), with Brownian motion as the default model. See 'Details'.
sigma_0_2_ini	initial value for background noise. If NA, use minimum value of absolute square of intensity profile in reciprocal space.
param_initial	initial values for param estimation.
num_optim	number of optimization.
msd_fn	user defined mean squared displacement(MSD) structure, a function of parameters and lag times. NA if model_name is not 'user_defined'.
msd_grad_fn	gradient for user defined mean squared displacement structure. If NA, then numerical gradient will be used for parameter estimation in 'user_defined' model.
num_param	number of parameters need to be estimated in the intermediate scattering function, need to be non-NA value for user_defined' model.
uncertainty	a logical evaluating to TRUE or FALSE indicating whether parameter uncertainty should be computed.
M	number of particles. See 'Details'.
sim_object	NA or an S4 object of class simulation.
msd_truth	true MSD or reference MSD value.
method	methods for parameter estimation, options from ('AIUQ', 'DDM_fixedAB', 'DDM_estAB').
index_q_AIUQ	index range for wave number when using AIUQ method. See 'Details'.
index_q_DDM	index range for wave number when using DDM method. See 'Details'.

message_out	a logical evaluating to TRUE or FALSE indicating whether or not to output the message.
A_neg	controls modification for negative $A(q)$, options from ('abs','zero'), with setting negative $A(q)$ to its absolute value as the default.
square	a logical evaluating to TRUE or FALSE indicating whether or not to crop the original intensity profile into square image.
output_dqt	a logical evaluating to TRUE or FALSE indicating whether or not to compute observed dynamic image structure function(Dqt).
output_isf	a logical evaluating to TRUE or FALSE indicating whether or not to compute empirical intermediate scattering function(ISF).
output_modeled_isf	a logical evaluating to TRUE or FALSE indicating whether or not to compute modeled intermediate scattering function(ISF).
output_modeled_dqt	a logical evaluating to TRUE or FALSE indicating whether or not to compute modeled dynamic image structure function(Dqt).
model_free_estimation	a logical evaluating to TRUE or FALSE indicating whether or not to use the model-free version for MSD estimation.
substack	a logical evaluating to TRUE or FALSE indicating whether or not to substack the large videos (for example, videos with 2000 frames) for MSD estimation. Only used when model_free_estimation = TRUE.
estimate_moduli	a logical evaluating to TRUE or FALSE indicating whether or not to estimate the storage and loss moduli from the fitted mean squared displacement. Only used when model_free_estimation = TRUE and uncertainty = TRUE.
MSD_unit	scaling factor used to convert the estimated MSD to the desired physical unit before modulus estimation. Only used when model_free_estimation = TRUE.
kB	Boltzmann constant used in the generalized Stokes-Einstein relation. Only used when model_free_estimation = TRUE.
Temp	absolute temperature used for modulus estimation. Only used when model_free_estimation = TRUE.
a	particle radius used for modulus estimation. Only used when model_free_estimation = TRUE.

Details

For simulated data using simulation in the AIUQ package, intensity will be automatically extracted from the simulation class.

By default, intensity_str is set to "T_SS_mat", a time-by-space-by-space matrix, which is the structure of the intensity profile obtained from the simulation class. For intensity_str = "SST_array", the input intensity profile should be a space-by-space-by-time array, which is the structure obtained from loading a tif file. For intensity_str = "S_ST_mat", the input intensity profile should be a space-by-space-by-time matrix.

By default, `AIUQ_thr` is set to `c(1, 1)` and uses information from all complete `q` rings. The first element affects the maximum selected wave number, and the second element controls the minimum proportion of wave numbers selected. By setting 1 for the second element, if the maximum wave number selected is less than the wave number length, then the maximum wave number selected is coerced to use all wave numbers unless the user defines another index range through `index_q_AIUQ`.

If `model_name` equals "user_defined", or NA (which is coerced to "user_defined"), then `msd_fn` and `num_param` need to be provided for parameter estimation.

The number of particles `M` is set to 50 by default, or automatically extracted from the simulation class for simulated data generated using `simulation` in the `AIUQ` package.

By default, all wave vectors from complete `q` rings are used unless the user defines an index range through `index_q_AIUQ` or `index_q_DDM`.

Value

Returns an S4 object of class `SAM`.

Author(s)

Yue He [aut], Xubo Liu [aut], Tong Lin [aut], Mengyang Gu [aut, cre]

References

Gu, M., He, Y., Liu, X., & Luo, Y. (2023). Ab initio uncertainty quantification in scattering analysis of microscopy. arXiv preprint arXiv:2309.02468.

Gu, M., Luo, Y., He, Y., Helgeson, M. E., & Valentine, M. T. (2021). Uncertainty quantification and estimation in differential dynamic microscopy. *Physical Review E*, 104(3), 034610.

Cerbino, R., & Trappe, V. (2008). Differential dynamic microscopy: probing wave vector dependent dynamics with a microscope. *Physical review letters*, 100(18), 188102.

Examples

```
library(AIUQ)

# Example 1: Model-dependent estimation for simulated data
sim_bm = simulation(len_t=100,sz=100,sigma_bm=0.5)
show(sim_bm)
sam = SAM(sim_object = sim_bm,model_name='BM')
show(sam)
plot(log10(sam@d_input), log10(sam@msd_est), type = "l", col = "cyan",
      xlab = expression(log[10](Delta * t)), ylab = expression(log[10](MSD)))
lines(log10(sam@d_input), log10(sam@msd_truth),
      col = "black", type="l")

## Not run:
##with uncertainty
sim_bm = simulation(len_t=100,sz=100,sigma_bm=0.5)
sam_uq = SAM(sim_object = sim_bm,model_name='BM',uncertainty=TRUE)
lines(log10(sam_uq@d_input), log10(sam_uq@msd_truth),
      col = "black", type="l")
```

```

lines(log10(sam@d_input), log10(sam@msd_truth),
      col = "black", type="l")
lines(log10(sam_uq@d_input), log10(sam_uq@msd_lower), type = "l", col = "cyan",lty=2)
lines(log10(sam_uq@d_input), log10(sam_uq@msd_upper), type = "l", col = "cyan",lty=2)

## End(Not run)

# Example 2: Model-free MSD estimation for simulated data
## Not run:
sim_bm = simulation(len_t=200,sz=200,sigma_bm=1)
sam_mf = SAM(sim_object = sim_bm, model_free_estimation = TRUE,uncertainty=TRUE)
show(sam_mf)
plot(log10(sam_mf@d_input), log10(sam_mf@msd_est), type = "l", col = "cyan",
      xlab = expression(log[10](Delta * t)), ylab = expression(log[10](MSD)),
      ylim=c(min(log10(sam_mf@msd_lower[-1])), max(log10(sam_mf@msd_upper[-1]))))
lines(log10(sam_mf@d_input), log10(sam_mf@msd_truth),
      col = "black", type="l")
lines(log10(sam_mf@d_input), log10(sam_mf@msd_lower), type = "l", col = "cyan",lty=2)
lines(log10(sam_mf@d_input), log10(sam_mf@msd_upper), type = "l", col = "cyan",lty=2)

## End(Not run)

# Example 3: Model-free MSD estimation for real data
## Not run:
sam_mf = SAM(intensity = intensity, intensity_str = "SST_array",
             mindt = mindt, pxsz = pxsz, model_free_estimation = TRUE)

## End(Not run)

# Example 4: Model-free MSD estimation with storage and loss moduli
## Not run:
sam_mf_moduli = SAM(intensity = intensity, intensity_str = "SST_array",
                   mindt = mindt, pxsz = pxsz, model_free_estimation = TRUE,
                   uncertainty = TRUE, M = 200, estimate_moduli = TRUE,
                   MSD_unit = 10^{-12}, Temp = 293, a = 250 * 10^{-9})
plot(log10(sam_mf_moduli@omega), log10(sam_mf_moduli@Gp), type = 'l')
lines(log10(sam_mf_moduli@omega), log10(sam_mf_moduli@Gpp), type = 'l', col = "red")

## End(Not run)

```

SAM-class

SAM class

Description

S4 class for fast parameter estimation in scattering analysis of microscopy, using either AIUQ or DDM method.

Slots

pxsz numeric. Size of one pixel in unit of micron with default value 1.
mindt numeric. Minimum lag time with default value 1.
sz vector. Frame size of the intensity profile in x and y directions, number of pixels contained in each frame equals sz_x by sz_y .
len_t integer. Number of time steps.
len_q integer. Number of wave vector.
q vector. Wave vector in unit of μm^{-1} .
d_input vector. Sequence of lag times.
B_est_ini numeric. Estimation of B. This parameter is determined by the noise in the system. See "References".
A_est_ini vector. Estimation of $A(q)$. Note this parameter is determined by the properties of the imaged material and imaging optics. See "References".
I_o_q_2_ori vector. Absolute square of Fourier transformed intensity profile, ensemble over time.
q_ori_ring_loc_unique_index list. List of location index of non-duplicate values for each q ring.
model_name character. Fitted model, options from "BM", "OU", "FBM", "OU+FBM", and "user_defined".
param_est vector. Estimated parameters contained in MSD.
sigma_2_0_est numeric. Estimated variance of background noise.
msd_est vector. Estimated MSD.
uncertainty logical. A logical evaluating to TRUE or FALSE indicating whether parameter uncertainty should be computed.
msd_lower vector. Lower bound of 95% confidence interval of MSD.
msd_upper vector. Upper bound of 95% confidence interval of MSD.
msd_truth vector. True MSD or reference MSD value.
sigma_2_0_truth vector. True variance of background noise, non NA for simulated data using simulation.
param_truth vector. True parameters used to construct MSD, non NA for simulated data using simulation.
index_q vector. Selected index of wave vector.
Dqt matrix. Dynamic image structure function $D(q, \Delta t)$.
ISF matrix. Empirical intermediate scattering function $f(q, \Delta t)$.
I_q matrix. Fourier transformed intensity profile with structure "SS_T_mat".
AIC numeric. Akaike information criterion score.
mle numeric. Maximum log likelihood value.
param_uq_range matrix. 95% confidence interval for estimated parameters.
modeled_Dqt matrix. Modeled dynamic image structure function $D(q, \Delta t)$.
modeled_ISF matrix. Modeled intermediate scattering function $f(q, \Delta t)$.
omega vector. Frequencies corresponding to the estimated moduli.

Gp vector. Estimated storage modulus.
Gpp vector. Estimated loss modulus.
theor_Gp vector. Theoretical storage modulus.
theor_Gpp vector. Theoretical loss modulus.

Author(s)

Yue He [aut], Xubo Liu [aut], Tong Lin [aut], Mengyang Gu [aut, cre]

References

Gu, M., He, Y., Liu, X., & Luo, Y. (2023). Ab initio uncertainty quantification in scattering analysis of microscopy. arXiv preprint arXiv:2309.02468.
Gu, M., Luo, Y., He, Y., Helgeson, M. E., & Valentine, M. T. (2021). Uncertainty quantification and estimation in differential dynamic microscopy. Physical Review E, 104(3), 034610.
Cerbino, R., & Trappe, V. (2008). Differential dynamic microscopy: probing wave vector dependent dynamics with a microscope. Physical review letters, 100(18), 188102.

show.aniso_sam	<i>Show scattering analysis of microscopy for anisotropic processes (aniso_SAM) object</i>
----------------	--

Description

Function to print the aniso_SAM class object after the aniso_SAM model has been constructed.

Usage

```
show.aniso_sam(object)
```

Arguments

object an S4 object of class aniso_SAM

Value

Show a list of important parameters in class aniso_SAM.

Author(s)

Yue He [aut], Xubo Liu [aut], Tong Lin [aut], Mengyang Gu [aut, cre]

References

Gu, M., He, Y., Liu, X., & Luo, Y. (2023). Ab initio uncertainty quantification in scattering analysis of microscopy. arXiv preprint arXiv:2309.02468.
Gu, M., Luo, Y., He, Y., Helgeson, M. E., & Valentine, M. T. (2021). Uncertainty quantification and estimation in differential dynamic microscopy. Physical Review E, 104(3), 034610.

Examples

```
library(AIUQ)

## Simulate BM and get estimated parameters using BM model
# Simulation
aniso_sim_bm = aniso_simulation(sz=100, len_t=100, sigma_bm=c(0.5, 0.3))
show(aniso_sim_bm)

# AIUQ method: fitting using BM model
aniso_sam = aniso_SAM(sim_object=aniso_sim_bm, AIUQ_thr=c(0.99, 0))
show(aniso_sam)
```

show.aniso_simulation *Show anisotropic simulation object*

Description

Function to print the `aniso_simulation` class object after the `aniso_simulation` model has been constructed.

Usage

```
show.aniso_simulation(object)
```

Arguments

`object` an S4 object of class `aniso_simulation`

Value

Show a list of important parameters in class `aniso_simulation`.

Author(s)

Yue He [aut], Xubo Liu [aut], Tong Lin [aut], Mengyang Gu [aut, cre]

References

Gu, M., He, Y., Liu, X., & Luo, Y. (2023). Ab initio uncertainty quantification in scattering analysis of microscopy. arXiv preprint arXiv:2309.02468.

Gu, M., Luo, Y., He, Y., Helgeson, M. E., & Valentine, M. T. (2021). Uncertainty quantification and estimation in differential dynamic microscopy. *Physical Review E*, 104(3), 034610.

Examples

```
library(AIUQ)

# Simulate simple diffusion for 100 images with 100 by 100 pixels
aniso_sim_bm = aniso_simulation(sz=100, len_t=100, sigma_bm=c(0.5, 0.1))
show(aniso_sim_bm)
```

show.sam

Show scattering analysis of microscopy (SAM) object

Description

Function to print the SAM class object after the SAM model has been constructed.

Usage

```
show.sam(object)
```

Arguments

object an S4 object of class SAM

Value

Show a list of important parameters in class SAM.

Author(s)

Yue He [aut], Xubo Liu [aut], Tong Lin [aut], Mengyang Gu [aut, cre]

References

Gu, M., He, Y., Liu, X., & Luo, Y. (2023). Ab initio uncertainty quantification in scattering analysis of microscopy. arXiv preprint arXiv:2309.02468.

Gu, M., Luo, Y., He, Y., Helgeson, M. E., & Valentine, M. T. (2021). Uncertainty quantification and estimation in differential dynamic microscopy. Physical Review E, 104(3), 034610.

Examples

```
library(AIUQ)

## Simulate BM and get estimated parameters using BM model
# Simulation
sim_bm = simulation(sz=100, len_t=100, sigma_bm=0.5)
show(sim_bm)

# AIUQ method: fitting using BM model
sam = SAM(sim_object=sim_bm)
show(sam)
```

show.simulation	<i>Show simulation object</i>
-----------------	-------------------------------

Description

Function to print the simulation class object after the simulation model has been constructed.

Usage

```
show.simulation(object)
```

Arguments

object an S4 object of class simulation

Value

Show a list of important parameters in class simulation.

Author(s)

Yue He [aut], Xubo Liu [aut], Tong Lin [aut], Mengyang Gu [aut, cre]

References

Gu, M., He, Y., Liu, X., & Luo, Y. (2023). Ab initio uncertainty quantification in scattering analysis of microscopy. arXiv preprint arXiv:2309.02468.

Gu, M., Luo, Y., He, Y., Helgeson, M. E., & Valentine, M. T. (2021). Uncertainty quantification and estimation in differential dynamic microscopy. Physical Review E, 104(3), 034610.

Examples

```
library(AIUQ)

# Simulate simple diffusion for 100 images with 100 by 100 pixels
sim_bm = simulation(sz=100, len_t=100, sigma_bm=0.5)
show(sim_bm)
```

simulation	<i>Simulate 2D particle movement</i>
------------	--------------------------------------

Description

Simulate 2D particle movement from a user selected stochastic process, and output intensity profiles.

Usage

```
simulation(
  sz = c(200, 200),
  len_t = 200,
  M = 50,
  model_name = "BM",
  noise = "gaussian",
  I0 = 20,
  Imax = 255,
  pos0 = matrix(NaN, nrow = M, ncol = 2),
  rho = 0.95,
  H = 0.3,
  sigma_p = 2,
  sigma_bm = 1,
  sigma_ou = 2,
  sigma_fbm = 2
)
```

Arguments

<code>sz</code>	frame size of simulated image with default <code>c(200, 200)</code> .
<code>len_t</code>	number of time steps with default 200.
<code>M</code>	number of particles with default 50.
<code>model_name</code>	stochastic process simulated, options from ('BM', 'OU', 'FBM', 'OU+FBM'), with default 'BM'.
<code>noise</code>	background noise, options from ('uniform', 'gaussian'), with default 'gaussian'.
<code>I0</code>	background intensity, value between 0 and 255, with default 20.
<code>Imax</code>	maximum intensity at the center of the particle, value between 0 and 255, with default 255.
<code>pos0</code>	initial position for M particles, matrix with dimension M by 2.
<code>rho</code>	correlation between successive step and previous step in O-U process, value between 0 and 1, with default 0.95.
<code>H</code>	Hurst parameter of fractional Brownian Motion, value between 0 and 1, with default 0.3.
<code>sigma_p</code>	radius of the spherical particle (<code>3sigma_p</code>), with default 2.

sigma_bm	distance moved per time step in Brownian Motion, with default 1.
sigma_ou	distance moved per time step in Ornstein–Uhlenbeck process, with default 2.
sigma_fbm	distance moved per time step in fractional Brownian Motion, with default 2.

Value

Returns an S4 object of class `simulation`.

Author(s)

Yue He [aut], Xubo Liu [aut], Tong Lin [aut], Mengyang Gu [aut, cre]

References

Gu, M., He, Y., Liu, X., & Luo, Y. (2023). Ab initio uncertainty quantification in scattering analysis of microscopy. arXiv preprint arXiv:2309.02468.

Gu, M., Luo, Y., He, Y., Helgeson, M. E., & Valentine, M. T. (2021). Uncertainty quantification and estimation in differential dynamic microscopy. *Physical Review E*, 104(3), 034610.

Cerbino, R., & Trappe, V. (2008). Differential dynamic microscopy: probing wave vector dependent dynamics with a microscope. *Physical review letters*, 100(18), 188102.

Examples

```
library(AIUQ)
# -----
# Example 1: Simple diffusion for 200 images with
#           200 by 200 pixels and 50 particles
# -----
sim_bm = simulation()
show(sim_bm)

# -----
# Example 2: Simple diffusion for 100 images with
#           100 by 100 pixels and slower speed
# -----
sim_bm = simulation(sz=100, len_t=100, sigma_bm=0.5)
show(sim_bm)

# -----
# Example 3: Ornstein-Uhlenbeck process
# -----
sim_ou = simulation(model_name="OU")
show(sim_ou)
```

simulation-class	<i>Simulation class</i>
------------------	-------------------------

Description

S4 class for 2D particle movement simulation.

Details

`intensity` should have structure "T_SS_mat", matrix with dimension `len_t` by `sz` x `sz`.

`pos` should be the position matrix with dimension `M` x `len_t`. See [bm_particle_intensity](#), [ou_particle_intensity](#), [fbm_particle_intensity](#), [fbm_ou_particle_intensity](#).

Slots

`sz` vector. Frame size of the intensity profile, number of pixels contained in each frame equals `sz[1]` by `sz[2]`.

`len_t` integer. Number of time steps.

`noise` character. Background noise, options from ('uniform','gaussian').

`model_name` character. Simulated stochastic process, options from ('BM','OU','FBM','OU+FBM').

`M` integer. Number of particles.

`pxsz` numeric. Size of one pixel in unit of micron, 1 for simulated data.

`mindt` numeric. Minimum lag time, 1 for simulated data.

`pos` matrix. Position matrix for particle trajectory, see 'Details'.

`intensity` matrix. Filled intensity profile, see 'Details'.

`num_msd` vector. Numerical mean squared displacement (MSD).

`param` vector. Parameters for simulated stochastic process.

`theor_msd` vector. Theoretical MSD.

`sigma_2_0` vector. Variance of background noise.

Author(s)

Yue He [aut], Xubo Liu [aut], Tong Lin [aut], Mengyang Gu [aut, cre]

References

Gu, M., He, Y., Liu, X., & Luo, Y. (2023). Ab initio uncertainty quantification in scattering analysis of microscopy. arXiv preprint arXiv:2309.02468.

Gu, M., Luo, Y., He, Y., Helgeson, M. E., & Valentine, M. T. (2021). Uncertainty quantification and estimation in differential dynamic microscopy. *Physical Review E*, 104(3), 034610.

Cerbino, R., & Trappe, V. (2008). Differential dynamic microscopy: probing wave vector dependent dynamics with a microscope. *Physical review letters*, 100(18), 188102.

Index

* classes

- aniso_SAM-class, 5
- aniso_simulation-class, 8
- SAM-class, 21
- simulation-class, 29

- aniso_SAM, 2
- aniso_SAM-class, 5
- aniso_simulation, 6
- aniso_simulation-class, 8

- bm_particle_intensity, 8, 29

- fbm_ou_particle_intensity, 8, 29
- fbm_particle_intensity, 8, 29

- get_dqt, 9
- Get_Gp_Gpp_GSER, 10
- get_isf, 11

- modeled_dqt, 12
- modeled_isf, 13

- ou_particle_intensity, 8, 29

- plot_intensity, 14
- plot_MSD, 15
- plot_traj, 16

- SAM, 17
- SAM-class, 21
- show.aniso_sam, 23
- show.aniso_simulation, 24
- show.sam, 25
- show.simulation, 26
- simulation, 27
- simulation-class, 29