

# Package ‘SpikeInference’

April 14, 2021

**Maintainer** Yiqun Chen <yiqunc@uw.edu>

**Author** Yiqun Chen, Sean Jewell

**Version** 1.0

**License** GPL-3

**Title** Quantifying uncertainty for spikes estimated from calcium imaging data

**Description** Implements selective inference procedures for spikes estimated via the L0 penalty, as described in Chen, Jewell, and Witten (2021+) <<https://arxiv.org/abs/2103.07818>>

**Depends** R (>= 3.5)

**Imports** Rcpp, latex2exp, ggplot2, dplyr, magrittr

**LinkingTo** Rcpp

**RoxygenNote** 7.1.1

**Suggests** testthat,  
knitr,  
rmarkdown

**VignetteBuilder** knitr, rmarkdown

**SystemRequirements** GNU make

## R topics documented:

construct_v . . . . .	2
estimate_spike_by_spike_number . . . . .	2
plot.simdata . . . . .	3
plot.spike_estimates . . . . .	4
plot.spike_inference . . . . .	4
print.simdata . . . . .	5
print.spike_estimates . . . . .	6
print.spike_inference . . . . .	6
simulate_ar1 . . . . .	7
spike_estimates . . . . .	8
spike_inference . . . . .	9
summary.spike_estimates . . . . .	11
summary.spike_inference . . . . .	12

<b>Index</b>	<b>13</b>
--------------	-----------

---

construct_v	<i>Generate the contrast vector for testing the null hypothesis of interest. See Section 2 of Chen et al. (2021+) for details.</i>
-------------	--

---

### Description

Generate the contrast vector for testing the null hypothesis of interest. See Section 2 of Chen et al. (2021+) for details.

### Usage

```
construct_v(n, thj, window_size, gam)
```

### Arguments

n	length of the vector
thj	location of the estimated spike of interest
window_size	parameter h
gam	AR-1 decaying parameter

### Value

Returns a vector of length n

### References

Chen YT, Jewell SW, Witten DM. (2021) Quantifying uncertainty in spikes estimated from calcium imaging data. arXiv:2103.0781 [statME].

### Examples

```
nu_vector <- construct_v(n = 100, thj = 2, window_size = 2, gam = 0.95)
```

---

estimate_spike_by_spike_number	<i>Search for the right tuning parameter lambda to meet the target firing rate</i>
--------------------------------	--

---

### Description

Search for the right tuning parameter lambda to meet the target firing rate

**Usage**

```
estimate_spike_by_spike_number(
  dat,
  decay_rate,
  target_firing_rate,
  lam_min = 1e-06,
  lam_max = 1,
  max_iters = 50,
  tolerance = 5
)
```

**Arguments**

dat	A simdata object generated by simulate_ar1.
decay_rate	Numeric; specified AR-1 decay rate $\gamma$ , a number between 0 and 1 (non-inclusive).
target_firing_rate	Numeric; a number between 0 to 1 indicating the average probability of firing
lam_min	Numeric; minimal lambda to consider
lam_max	Numeric; maximal lambda to consider
max_iters	Numeric; maximal iterations to search
tolerance	Numeric; tolerance level for differences in firing rate

**Examples**

```
curr_sim <- simulate_ar1(n = 1000, gam = 0.95, poisMean = 0.01, sd = 0.1, seed = 1)
fit_spike <- estimate_spike_by_spike_number(curr_sim, decay_rate = 0.95,
target_firing_rate = 0.01, max_iters=10, tolerance=max(5))
```

---

plot.simdata	<i>Plot simulated data</i>
--------------	----------------------------

---

**Description**

Plot simulated data

**Usage**

```
## S3 method for class 'simdata'
plot(x, xlims = NULL, ...)
```

**Arguments**

x	output data from simulate_ar1
xlims	optional parameter to specify the x-axis limits
...	to be passed to methods

**Value**

Plot with simulated fluorescence (dark grey circles), calcium concentration (dark green line) and spikes (dark green tick marks on x-axis)

**Examples**

```
sim <- simulate_ar1(n = 500, gam = 0.998, poisMean = 0.009, sd = 0.05, seed = 1)
plot(sim)
```

---

plot.spike\_estimates    *Plot the solution to an L0 segmentation problem*

---

**Description**

Plot the solution to an L0 segmentation problem

**Usage**

```
## S3 method for class 'spike_estimates'
plot(x, xlims = NULL, ...)
```

**Arguments**

x	output from running spike_estimates
xlims	optional parameter to specify the x-axis limits
...	to be passed to methods

**Examples**

```
### Generate sample data
sim <- simulate_ar1(n = 500, gam = 0.998, poisMean = 0.01, sd = 0.05, seed = 1)
### Fit the spike
fit_spike <- spike_estimates(sim$f1, decay_rate = 0.998, tuning_parameter = 0.01)
### Plot estimated spikes
plot(fit_spike)
### summarize estimated spike times
summary_fit_spike <- summary(fit_spike)
```

---

plot.spike\_inference    *Plot the result of quantifying the uncertainty of spikes estimated via an L0 penalty*

---

**Description**

Plot the result of quantifying the uncertainty of spikes estimated via an L0 penalty

**Usage**

```
## S3 method for class 'spike_inference'
plot(x, xlims = NULL, ...)
```

**Arguments**

x	output from running spike_inference
xlims	optional parameter to specify the x-axis limits
...	to be passed to methods

**References**

Chen YT, Jewell SW, Witten DM. (2021) Quantifying uncertainty in spikes estimated from calcium imaging data. arXiv:2103.0781 [statME].

**Examples**

```
gam <- 0.98
LAMBDA <- 0.7
sigma <- 0.3
n_length <- 1000
curr_sim <- simulate_ar1(n = n_length, gam = gam, poisMean = 0.01, sd = sigma, seed = 2)
curr_inference_spike <- spike_inference(dat = curr_sim$f1, decay_rate = gam,
  tuning_parameter = LAMBDA, window_size = 2, sig2 = sigma*sigma, return_ci = TRUE)
### The observed fluorescence is plotted in gray, along with two sets of
### vertical ticks: each orange tick represents an estimated spike associated with
### a positive increase in fluorescence, and blue ticks are the subset of spikes
### with a selective p-value < 0.05.
plot(curr_inference_spike)
```

---

print.simdata	<i>Print simulated data</i>
---------------	-----------------------------

---

**Description**

Print simulated data

**Usage**

```
## S3 method for class 'simdata'
print(x, ...)
```

**Arguments**

x	simulated data
...	arguments to be passed to methods

**Examples**

```
sim <- simulate_ar1(n = 500, gam = 0.998, poisMean = 0.009, sd = 0.05, seed = 1)
print(sim)
```

---

`print.spike_estimates` *Print estimated spikes*

---

### Description

Print estimated spikes

### Usage

```
## S3 method for class 'spike_estimates'  
print(x, ...)
```

### Arguments

<code>x</code>	estimated spikes
<code>...</code>	arguments to be passed to methods

### Examples

```
### Generate sample data  
sim <- simulate_ar1(n = 500, gam = 0.998, poisMean = 0.01, sd = 0.05, seed = 1)  
### Fit the spike  
fit_spike <- spike_estimates(sim$f1, decay_rate = 0.998, tuning_parameter = 0.01)  
print(fit_spike)
```

---

`print.spike_inference` *Print estimated spikes*

---

### Description

Print estimated spikes

### Usage

```
## S3 method for class 'spike_inference'  
print(x, ...)
```

### Arguments

<code>x</code>	estimated spikes
<code>...</code>	arguments to be passed to methods

### Examples

```
### Generate sample data  
sim <- simulate_ar1(n = 500, gam = 0.998, poisMean = 0.01, sd = 0.05, seed = 1)  
### Fit the spike  
fit_spike <- spike_estimates(sim$f1, decay_rate = 0.998, tuning_parameter = 0.01)  
print(fit_spike)
```

---

simulate_ar1	<i>Simulate fluorescence trace based on a simple AR-1 generative model</i>
--------------	--

---

## Description

Simulate fluorescence trace based on a simple AR-1 generative model

## Usage

```
simulate_ar1(n, gam, poisMean, sd, seed, c0 = 0)
```

## Arguments

n	Numeric; length of the time series
gam	Numeric; AR-1 decay rate
poisMean	Numeric; mean for Poisson distributed spikes
sd	Numeric; standard deviation
seed	Numeric; random seed
c0	Numeric; initial calcium concentration, default to 0

## Details

Simulate fluorescence trace based on a simple AR-1 generative model:

$$y_t = c_t + \epsilon_t, \epsilon_t \sim N(0, \sigma^2),$$

$$c_t = \text{gam} * c_{t-1} + s_t,$$

$$s_t \sim \text{Poisson}(\text{poisMean}).$$

## Value

- spikes – A list of timesteps at which a spike occurs
- fl – The noisy fluorescence  $y_t$
- conc – The true calcium concentration  $c_t$

## Examples

```
sim <- simulate_ar1(n = 500, gam = 0.998, poisMean = 0.009, sd = 0.05, seed = 1)
plot(sim)
```

---

spike_estimates	<i>Estimate spikes with an L0 penalty</i>
-----------------	---

---

## Description

Estimate spikes with an L0 penalty

## Usage

```
spike_estimates(
  dat,
  decay_rate,
  tuning_parameter,
  functional_pruning_out = FALSE
)
```

## Arguments

dat	Numeric vector; observed data.
decay_rate	Numeric; specified AR(1) decay rate $\gamma$ , a number between 0 and 1 (non-inclusive).
tuning_parameter	Numeric; tuning parameter $\lambda$ for L0 spike estimation, a non-negative number.
functional_pruning_out	Logical; if TRUE, return cost functions for L0 spike estimation. Defaults to FALSE.

## Details

Estimation: This function estimates spikes via an L0 penalty based on the following optimization problem:

$$\underset{c_1, \dots, c_T \geq 0}{\text{minimize}} \frac{1}{2} \sum_{t=1}^T (y_t - c_t)^2 + \lambda \sum_{t=2}^T 1(c_t \neq \gamma c_{t-1}),$$

where  $y_t$  is the observed fluorescence at the t-th timestep.

## Value

For L0 spike estimation, returns a list with elements:

- `estimated_calcium` Estimated calcium levels
- `spikes` The set of estimated spikes
- `cost` The cost at each time point
- `n_intervals` The number of piecewise quadratics used at each point
- `piecewise_square_losses` A data frame of optimal cost functions  $\text{Cost}_s^*(\mu)$  for  $s = 1, \dots, T$ .



## References

- Jewell, S. W., Hocking, T. D., Fearnhead, P., & Witten, D. M. (2019). Fast nonconvex deconvolution of calcium imaging data. *Biostatistics*.
- Maidstone, R., Hocking, T., Rigaiill, G., & Fearnhead, P. (2017). On optimal multiple changepoint algorithms for large data. *Statistics and Computing*, 27(2), 519-533.
- Rigaiill, G. (2015). A pruned dynamic programming algorithm to recover the best segmentations with 1 to  $K_{\max}$  change-points. *Journal de la Societe Francaise de Statistique*, 156(4), 180-205.

## Examples

```
### Generate sample data
sim <- simulate_ar1(n = 500, gam = 0.998, poisMean = 0.01, sd = 0.05, seed = 1)
### Fit the spike
fit_spike <- spike_estimates(sim$f1, decay_rate = 0.998, tuning_parameter = 0.01)
### Plot estimated spikes
plot(fit_spike)
### summarize estimated spike times
summary_fit_spike <- summary(fit_spike)
```

---

spike_inference	<i>Estimation and inference for AR-1 spike problem using an L0 penalty</i>
-----------------	--

---

## Description

Estimation and inference for AR-1 spike problem using an L0 penalty

## Usage

```
spike_inference(
  dat,
  decay_rate,
  tuning_parameter,
  window_size,
  sig2 = NULL,
  return_conditioning_sets = FALSE,
  return_ci = FALSE,
  two_sided = FALSE,
  alpha = 0.05
)
```

## Arguments

dat	Numeric vector; observed data.
decay_rate	Numeric; specified AR-1 decay rate $\gamma$ , a number between 0 and 1 (non-inclusive).
tuning_parameter	Numeric; tuning parameter $\lambda$ for L0 spike estimation, a non-negative number.
window_size	Numeric; window size for fixed window hypothesis testing, a non-negative integer.

sig2	Numeric; noise variance for the observed data, a non-negative number. If unknown (NULL), sample variance of residuals is used instead.
return_conditioning_sets	Logical; Should the conditioning set S be returned?
return_ci	Logical; if TRUE, the confidence interval for the change in calcium is computed and returned.
two_sided	Logical; if TRUE, a 2-sided p-value is computed and returned.
alpha	Numeric; significance level for the hypothesis test, a number between 0 and 1 (non-inclusive).

## Details

Consider the AR-1 generative model

$$Y_t = c_t + \epsilon_t, \epsilon_t \sim N(0, \sigma^2),$$

where  $c_t = \gamma c_{t-1} + z_t$  and  $z_t \sim \text{Poisson}(\text{poisMean})$ . In words, this says between spikes (when  $z_t = 0$ ), calcium decays exponentially at a known rate  $\gamma \in (0, 1)$ , which is taken to be known. Further denote the locations of true spikes,  $\{t : z_t \geq 0\}$  as  $\{0 = \tau_0 < \tau_1 < \dots < \tau_K < \tau_{K+1} = T\}$ .

This function first estimates spikes via L0 penalty based on noisy observations  $y_t, t = 1, \dots, T$  by solving the following optimization problem

$$\text{minimize}_{c_1, \dots, c_T \geq 0} \frac{1}{2} \sum_{t=1}^T (y_t - c_t)^2 + \lambda \sum_{t=2}^T 1(c_t \neq \gamma c_{t-1}).$$

Estimated spikes correspond to the time  $t$  such that estimated calcium does not decay exponentially, i.e.,  $\{\dots, \hat{\tau}_j, \dots\} = \{t : \hat{c}_{t+1} - \gamma \hat{c}_t \neq 0\}$ .

Now suppose that we want to test whether the calcium is exponentially decaying near an estimated spike  $\hat{\tau}_j$ ; or equivalently, the null hypothesis of the form  $H_0 : \nu^T c = 0$  versus  $H_1 : \nu^T c > 0$  for suitably chosen  $\nu$  (see Section 2 in Chen et al. (2021+) for details). This function computes the following p-value

$$P(\nu^T Y \geq \nu^T y | \hat{\tau}_j \in M(Y), \nu^T Y > 0, \Pi_\nu^\perp Y = \Pi_\nu^\perp y)$$

where  $M(Y)$  is the set of spikes estimated from  $Y$  via the L0 method,  $\Pi_\nu^\perp$  is the orthogonal projection to the orthogonal complement of  $\nu$ . In particular, this p-value controls the selective Type I error (see Section 3 in Chen et al. (2021+) for details).

In addition, we implement a  $1 - \alpha$  confidence interval for the parameter  $\nu^T c$ , the increase in calcium associated with an estimated spike  $\hat{\tau}_j$  (see Section 4 in Chen et al. (2021+) for details).

## Value

Returns a list with elements:

- spikes the set of spikes,
- pvals p-values associated with each spike,
- LCB lower confidence band for each spike,
- UCB upper confidence band for each spike.

## References

- Chen YT, Jewell SW, Witten DM. (2021+) Quantifying uncertainty in spikes estimated from calcium imaging data. arXiv:2103.0781 [statME].
- Jewell, S. W., Hocking, T. D., Fearnhead, P., & Witten, D. M. (2019). Fast nonconvex deconvolution of calcium imaging data. *Biostatistics*.
- Jewell, S., Fearnhead, P., and Witten, D. (2019+). Testing for a change in mean after changepoint detection. Technical report.
- Maidstone, R., Hocking, T., Rigaiil, G., & Fearnhead, P. (2017). On optimal multiple changepoint algorithms for large data. *Statistics and Computing*, 27(2), 519-533.
- Rigaiil, G. (2015). A pruned dynamic programming algorithm to recover the best segmentations with 1 to K\_max change-points. *Journal de la Societe Francaise de Statistique*, 156(4), 180-205.

## Examples

```
gam <- 0.98
LAMBDA <- 0.7
sigma <- 0.3
n_length <- 1000
curr_sim <- simulate_ar1(n = n_length, gam = gam, poisMean = 0.01, sd = sigma, seed = 2)
curr_fit_spike <- spike_estimates(dat = curr_sim$f1, decay_rate = gam, tuning_parameter = LAMBDA)
curr_inference_spike <- spike_inference(dat = curr_sim$f1, decay_rate = gam,
  tuning_parameter = LAMBDA, window_size = 2, sig2 = sigma*sigma, return_ci = TRUE)
### a summary of the inferential results
summary(curr_inference_spike)
```

---

```
summary.spike_estimates
```

*Summarize the solution to an L0 segmentation problem*

---

## Description

Summarize the solution to an L0 segmentation problem

## Usage

```
## S3 method for class 'spike_estimates'
summary(object, ...)
```

## Arguments

object	output from running spike_estimates
...	to be passed to methods

## Value

A data frame with a column of estimated spikes

**Examples**

```
gam <- 0.98
LAMBDA <- 0.7
sigma <- 0.3
n_length <- 1000
curr_sim <- simulate_ar1(n = n_length, gam = gam, poisMean = 0.01, sd = sigma, seed = 2)
curr_fit_spike <- spike_estimates(dat = curr_sim$f1, decay_rate = gam, tuning_parameter = LAMBDA)
summary(curr_fit_spike)
```

---

```
summary.spike_inference
```

*Summarize the inference result for spikes estimated via an L0 inference problem*

---

**Description**

Summarize the inference result for spikes estimated via an L0 inference problem

**Usage**

```
## S3 method for class 'spike_inference'
summary(object, ...)
```

**Arguments**

object	output from running spike_estimates
...	to be passed to methods

**Value**

A data frame with a column of estimated spikes with associated p-values and confidence intervals.

**Examples**

```
gam <- 0.98
LAMBDA <- 0.7
sigma <- 0.3
n_length <- 1000
curr_sim <- simulate_ar1(n = n_length, gam = gam, poisMean = 0.01, sd = sigma, seed = 2)
curr_fit_spike <- spike_estimates(dat = curr_sim$f1, decay_rate = gam, tuning_parameter = LAMBDA)
curr_inference_spike <- spike_inference(dat = curr_sim$f1, decay_rate = gam,
tuning_parameter = LAMBDA, window_size = 2, sig2 = sigma*sigma, return_ci = TRUE)
summary(curr_inference_spike)
```

# Index

`construct_v`, [2](#)  
`estimate_spike_by_spike_number`, [2](#)  
  
`plot.simdata`, [3](#)  
`plot.spike_estimates`, [4](#)  
`plot.spike_inference`, [4](#)  
`print.simdata`, [5](#)  
`print.spike_estimates`, [6](#)  
`print.spike_inference`, [6](#)  
  
`simulate_ar1`, [7](#)  
`spike_estimates`, [8](#)  
`spike_inference`, [9](#)  
`summary.spike_estimates`, [11](#)  
`summary.spike_inference`, [12](#)