

# Package ‘pro’

October 14, 2022

**Type** Package

**Title** Point-Process Response Model for Optogenetics

**Version** 0.1.1

**Date** 2015-09-17

**Author** Xi (Rossi) LUO with contributions from Dylan Small and Vikaas Sohal

**Maintainer** Xi (Rossi) LUO <xi.rossi.luo@gmail.com>

**Suggests** cin

**Description** Optogenetics is a new tool to study neuronal circuits that have been genetically modified to allow stimulation by flashes of light. This package implements the methodological framework, Point-process Response model for Optogenetics (PRO), for analyzing data from these experiments. This method provides explicit nonlinear transformations to link the flash point-process with the spiking point-process. Such response functions can be used to provide important and interpretable scientific insights into the properties of the biophysical process that governs neural spiking in response to optogenetic stimulation.

**License** GPL-2

**NeedsCompilation** no

**Repository** CRAN

**Date/Publication** 2015-09-17 22:50:12

## R topics documented:

model.pro . . . . .	2
pro . . . . .	3
sim.lif . . . . .	3

<b>Index</b>	<b>5</b>
--------------	----------

---

`model.pro`*Model matrix for point-process responses*

---

### Description

Constructs a `data.frame` to be fitted using `pro`. Reference: X Luo, S Gee, V Sohal, D Small (In Press). A Point-process Response Model for Optogenetics Experiments on Neural Circuits. *Statistics in Medicine*.

### Usage

```
model.pro(spike, flash, fixed = NULL, kv = F)
```

### Arguments

<code>spike</code>	A binary vector represents spiking (1) or no spiking (0).
<code>flash</code>	A binary vector of the same length of <code>spike</code> , 1 for flashing and 0 for non-flashing.
<code>fixed</code>	Whether a fixed time window of <code>spike/flash</code> history should be used. If it is <code>NULL</code> , a varying time window of history will be used as described in the reference. If it is a integer <code>j</code> , a fixed window from index <code>t-j</code> to <code>t</code> will be used.
<code>kv</code>	Whether the history dependence model in Kass and Ventura (2001) (A Spike-Train Probability Model, <i>Neural Computation</i> 13, 1713-1720) should be employed. This differs from the history dependence model in the reference.

### Value

a `data.frame` of the three response functions (PF, CF, SF) and other intermediate functions (for future modeling use).

### Examples

```
n <- 500
set.seed(100)
re <- sim.lif(n, rbinom(n, 1, 0.14), 7, 3)
d <- model.pro(re$sbin, re$I)
d[1:10, ]
```

---

pro *Fit the PRO model*

---

### Description

Fit the PRO model to data. Reference: X Luo, S Gee, V Sohal, D Small (In Press). A Point-process Response Model for Optogenetics Experiments on Neural Circuits. *\_Statistics in Medicine\_*.

### Usage

```
pro(spike, flash, ...)
```

### Arguments

spike	A binary vector represents spiking (1) or no spiking (0).
flash	A binary vector of the same length of spike, 1 for flashing and 0 for non-flashing.
...	Additional parameters, see <a href="#">model.pro</a> .

### Value

a [glm](#) object of the fitted PRO coefficients.

### Examples

```
n <- 500
set.seed(100)
re <- sim.lif(n, rbinom(n, 1, 0.14), 7, 3)
fit.pro <- pro(re$sbin, re$I)
summary(fit.pro)
```

---

sim.lif *Simulate optogenetic stimulation on a leaky-integrate-fire neuron*

---

### Description

Simulate various kinds of neural measures (e.g. membrane potentials and spikes) from a LIF neuron.

### Usage

```
sim.lif(n, I, C, R, Vth = 1, V0 = 0, bin = 5, dt = 0.05)
```

**Arguments**

n	Number of time bins. The total time is n times bin.
I	Input stimulus vector of length n.
C	Membrane capacitance of the simulated neuron.
R	Membrane resistance of the simulated neuron.
Vth	Membrane potential threshold for spiking.
V0	Membrane potential reset value after spiking.
bin	Time length for each time bin. Default 5 millisecond.
dt	Time length for each simulation step. Default 0.05 millisecond.

**Value**

a list of simulated neural spikes, optogenetic light flashes, and simulation parameters.

**Examples**

```
n<- 500
set.seed(100)
re <- sim.lif(n, rbinom(n, 1, 0.14), 7, 3)
```

# Index

`glm`, 3

`model.pro`, 2, 3

`pro`, 2, 3

`sim.lif`, 3