The dendrite number change in ODE are show

|  |  |
| --- | --- |
|  | (1) |

𝑛 is the number of dendrites, r is the length of dendrite, is branching frequency, and is the terminating frequency.

The and of dendrites are estimated by Kaplan Meier Estimate (survival analysis, ):

|  |  |
| --- | --- |
|  | (2) |

Prior study indicates that both frequency are Poisson distribution in segment less than 4 the slope of based on log2 scale is negative.

We simulate the stochastic differential equation by Gillespie algorithm.

**Programming:**

I would program the data process by MATLAB 2019b and the stimulation by Python 3.7.3 in the platform, Spyder 4 under setting of the Anaconda and CentOS Linux 7.

We include 3 MATLAB programs and 3 python programs with the present work.

**MATLAB:**

1. neuronTypesFilesNamesAndPaths\_Pyramidal.m
2. dendriticalLengthDistribution\_fromRoot\_pyramidal.m
3. plotKbAndKtDistribution.m

**Python:**

1. plotMeanOfACF.py
2. nueronclass.py
3. stochastic\_Tm\_nueron.py

Preparation:

1. Download the neuron morphology data in SWC format from

NeuroMoroph.Org [(http://neuromorpho.org/)]((http:/neuromorpho.org/))

1. Download two toolboxes:

Python: btmorph\_v2 (<https://github.com/btorb/btmorph_v2>)

MATLAB: TREES toolbox (<https://www.treestoolbox.org/>)

Programs:

**neuronTypesFilesNamesAndPaths\_Tm.m**

Input:

Mat files location

SWC files loading location

Date

Loading filenames:

Lab\_names

Define folder name:

for classified by Yu-Chuan (classified = 1) Files names are not clear for neuron name or condition

for classified by original file name (classified =2)

Output:

workingDateAndLab.mat

neuronTypesFilesNamesAndPaths\_Tm\_**lab\_name\_date**.mat

**dendriticalLengthDistribution\_fromRoot\_Tm.m**

Check the Structure of neuron (number and type)

Count the branching and terminal segment from root

Calculate kb and kt

Input:

workingDateAndLab.mat

neuronTypesFilesNamesAndPaths\_Pyramidal\_**lab\_name\_date**.mat

Output:

dendriticLengthDistributionData\_PyramidalRoot\_**lab\_name\_date**.mat

**plotKbAndKtDistribution\_Tm.m**

Plot program of all output

Input:

workingDateAndLab.mat

dendriticLengthDistributionData\_PyramidalRoot\_**lab\_name\_date**.mat

Output:

Figures(.png):

boxplot\_SegmentNumberRoot\_**lab\_name**\_**neuron\_names**\_**dendriteType**

kb/kt\_region\_**neuronType**\_**lab\_name**\_**neuron\_names**\_**dendriteType**

segmentAtRiskRegion\_**neuronType**\_**lab\_name**\_**neuron\_names**\_**dendriteType**

NumberChange\_**neuronType**\_**lab\_name**\_**neuron\_names**\_**dendriteType**

kbkt\_RegionM\_**neuronType**\_**lab\_name**\_**neuron\_names**\_**dendriteType**

TimeSeriesOf\_kb/kt\_**neuronType**\_**lab\_name**\_**neuron\_names**\_**dendriteType**

boxplot\_SegmentNumberRoot\_**lab\_name**\_**neuron\_names**\_**dendriteType**

boxplot\_averageSegmentLengthRoot\_**lab\_name**\_**neuron\_names**\_**dendriteType**

delta\_kbkt\_**neuronType**\_**lab\_name**\_**neuron\_names**\_**dendriteType**

ACFof\_kbkt\_**neuronType**\_**lab\_name**\_**neuron\_names**\_**dendriteType(nx)**

ACFofkb\_mean\_std\_**label**\_**neuronType**\_**lab\_name**\_**neuron\_names**\_**dendriteType**

ACFofkb\_mean\_std\_s\_**label**\_**neuronType**\_**lab\_name**\_**neuron\_names**\_**dendriteType**

ACFof\_mean\_kbkt\_**neuronType**\_**lab\_name**\_**neuron\_names**\_**dendriteType**

ACFof\_mean\_kb/kt\_**neuronType**\_**lab\_name**\_**neuron\_names**\_**dendriteType**

Br/TeNumChange\_**neuronType**\_**lab\_name**\_**neuron\_names**\_**dendriteTyp**

NumChangeMerger\_**neuronType**\_**lab\_name**\_**neuron\_names**\_**dendriteType**

CSV files:

KB/KT\_series\_**neuronType**\_**labname**\_**name**\_**Dendrite**

MeanOfACF\_**neuronType**\_**labname**\_**name**\_**dendrite**\_**Br/Te**

**Python:**

**plotMeanOfACF.py**

Analyze the autocorrelation of branching and terminating frequency

Input:

MeanOfACF\_**neuronType**\_**labname**\_**name**\_**dendrite**\_**Br/Te**.csv

Output:

meanOfACF\_**labname**\_**neuronType**\_**Segment\_name**.png

**nueronclass.py**

Define neuron base properties, such as the Gillespie algorithm and the parameters from MATLAB

Input:

KB/KT\_series\_**neuronType**\_**labname**\_**name**\_**Dendrite**.csv

**stochastic\_Tm\_nueron.py**

input:

Nueronclass.py (in the same folder)

Output:

**neuronType**\_**labname**\_**neuron\_type**.png