Make your Own Spikes

Neural Data Analyses

Neural Data Analysis: Incremental process

The ‘master plan’

Experiment(s) → Data

Analysis 1
Analysis 2
Analysis 3
…
Analysis N

Typical, classical

- ‘Conclusion’ driven
- Results

Analysis

Qualitative (phenomenological)

Quantitative

statistical
analytical

Neural Data Analyses

- Analyses results can suggest new analyses: Branching process

Analysis 1
Analysis 2 → Analysis 2-1
Analysis 3 Analysis 2-2
… Analysis 2-3
Analysis N …
Analysis 2-P

<u>Compromise between depth/breadth first</u>
<u>Combinatorial explosions (analyses for ever!)</u>

- Analyses results can suggest new experiments: Long time scales

Analysis 1
Analysis 2 → Experiment1 → data1 → Analysis 1.1, Analysis 2.1, Analysis 3.1, …, Analysis Q.1
Analysis 3
…
Analysis N

<u>Careful planning/design of the initial experiment(s)</u>
<u>(controls, alternative hypotheses…)</u>

Data Representation

Raw data = data + noise *(e.g. multi-unit spikes)*

Denoising
(off-line)

Re-coding *(e.g. time series)*
(theoretically, no/neg. information loss)

Representation 1
(data feature 1)
(e.g. Histogram of ISIs)

Representation 2
(data feature 2)
(e.g. PSTH)

Representation 3
(data feature 3)
(e.g. matrix of pairwise correlations)

Representation 2.1
(e.g. peaks of the PTSH)

Representation 2.2

Representation 2.3
(e.g. width of largest PSTH peak)

Information loss

Data Representation

Representation *i*
(data feature *i*)

→ Analysis level/type *j*

→ Archiving (database)

Backups Strategy

- Raw data (permanent, multiple copies)
- Various representations (depends on amount of processing from raw data)
- Code (permanent, multiple copies)

Showing Data Analyses: The typical progression

1- Analysis method. Use surrogate dataset, simulation data set, cartoon.

2- Show typical single cell examples (raw data): voltage traces, rasterplots.

3- Show a single cell analysis: Extract interesting feature(s) from step 2.

4- Show population results: statistical analyses, population features, controls.

5- Propose an interpretation (explanation), prediction(s):
Use a (conceptual or computational) model.

Good examples: Reinagel and Reid, J Neuroscience, 2002
Usrey, Sceniak Chapman, J Neurophys, 2003

Example

Place cells: Basic facts

- Excitatory cells in the hippocampus (memory structure of the temporal lobes).

- Fire action potentials when the animal is at a particular spatial location (place field).

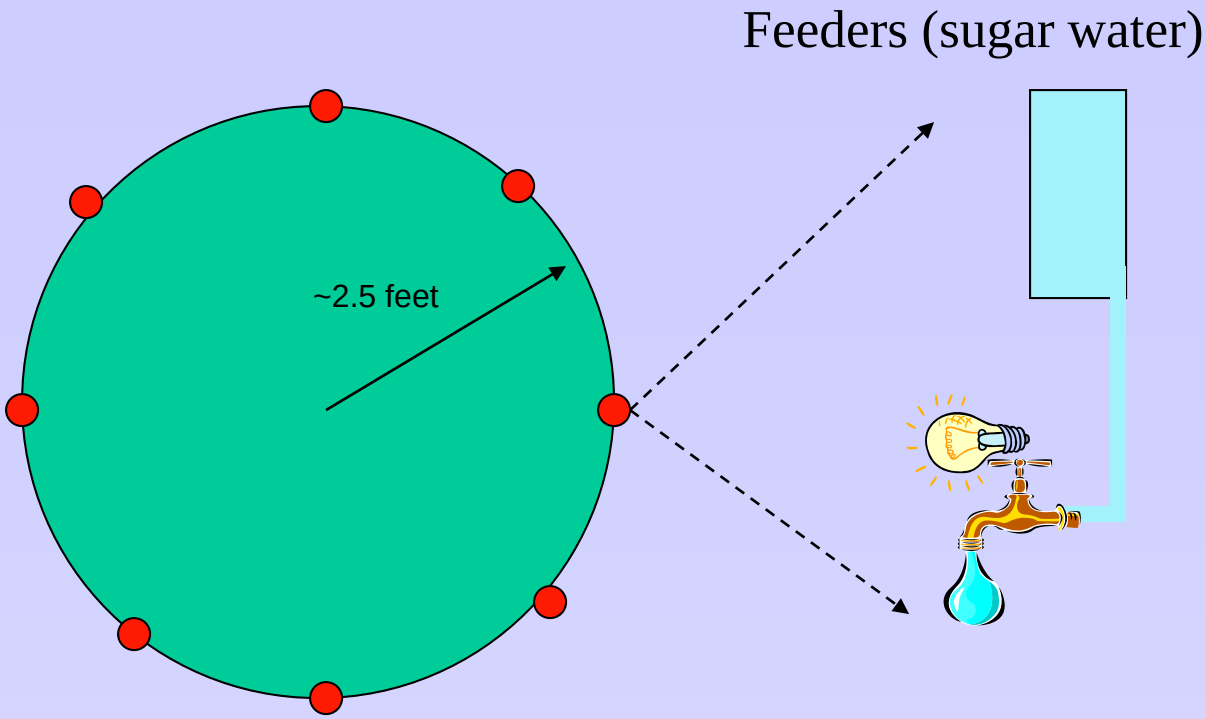
- Form in 20 minutes, persist in the dark.

- Depend on visual, and idiothetic (self-motion) cues.

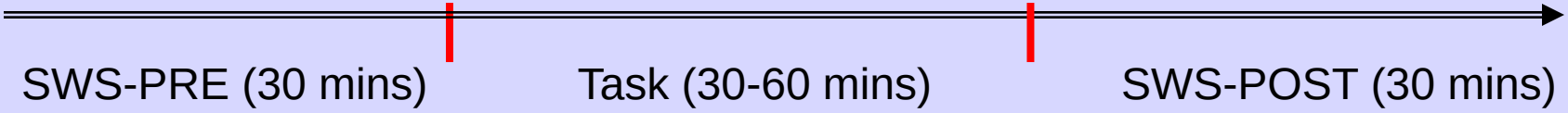
- Place fields form sometimes at ‘significant’ locations.

Experimental paradigm

Experimental setup:
(sequence learning)

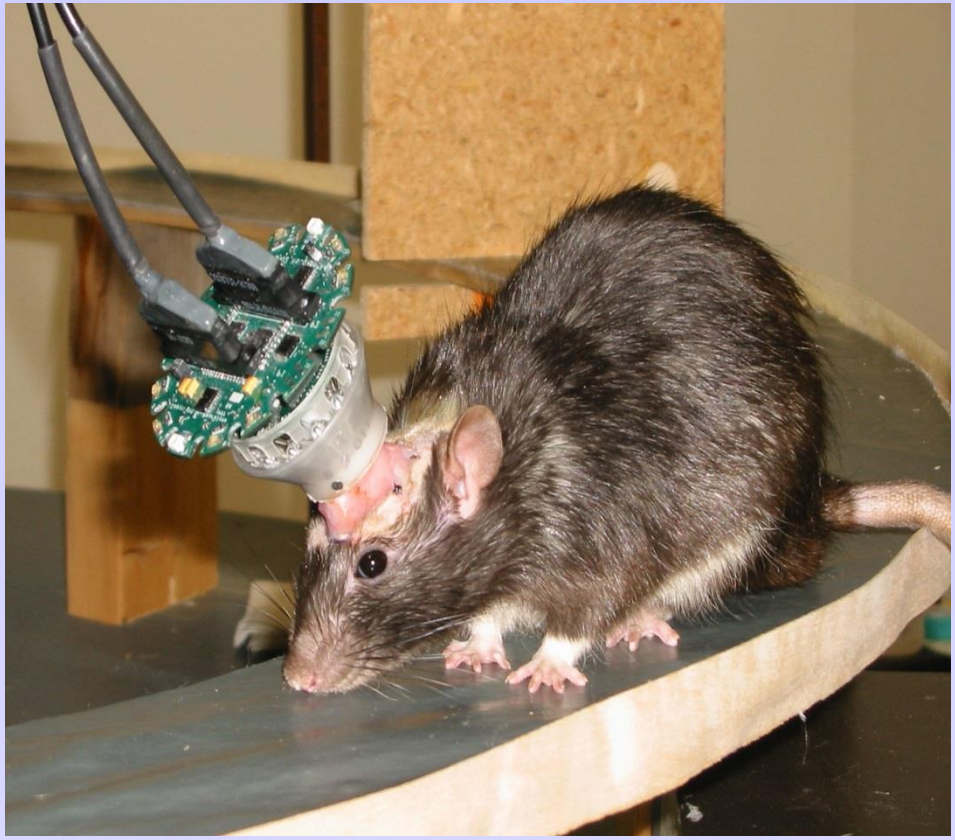


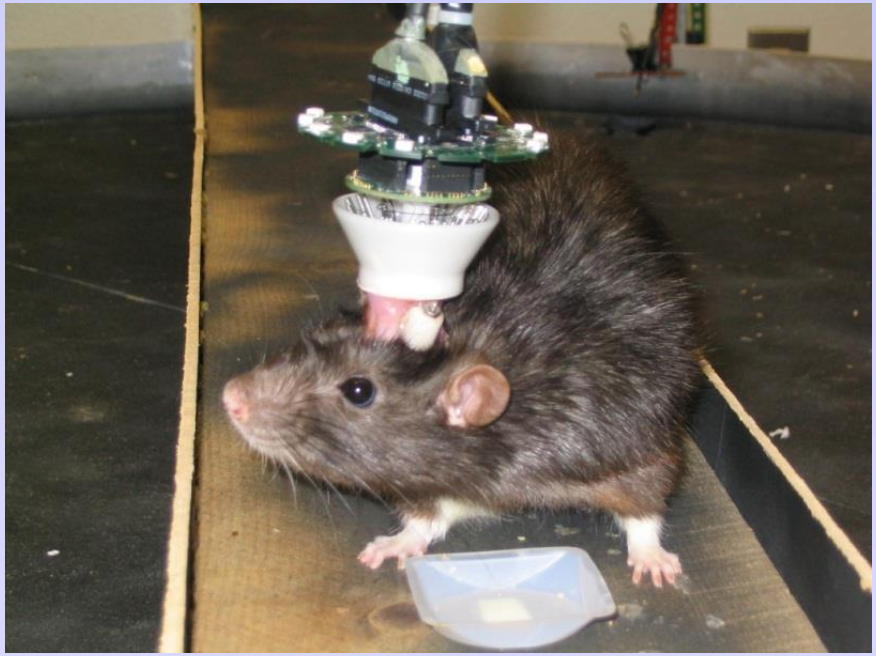
Experimental session:



SWS=Slow Wave Sleep

Methods: Hyperdrive





Kawahara et al, 2003

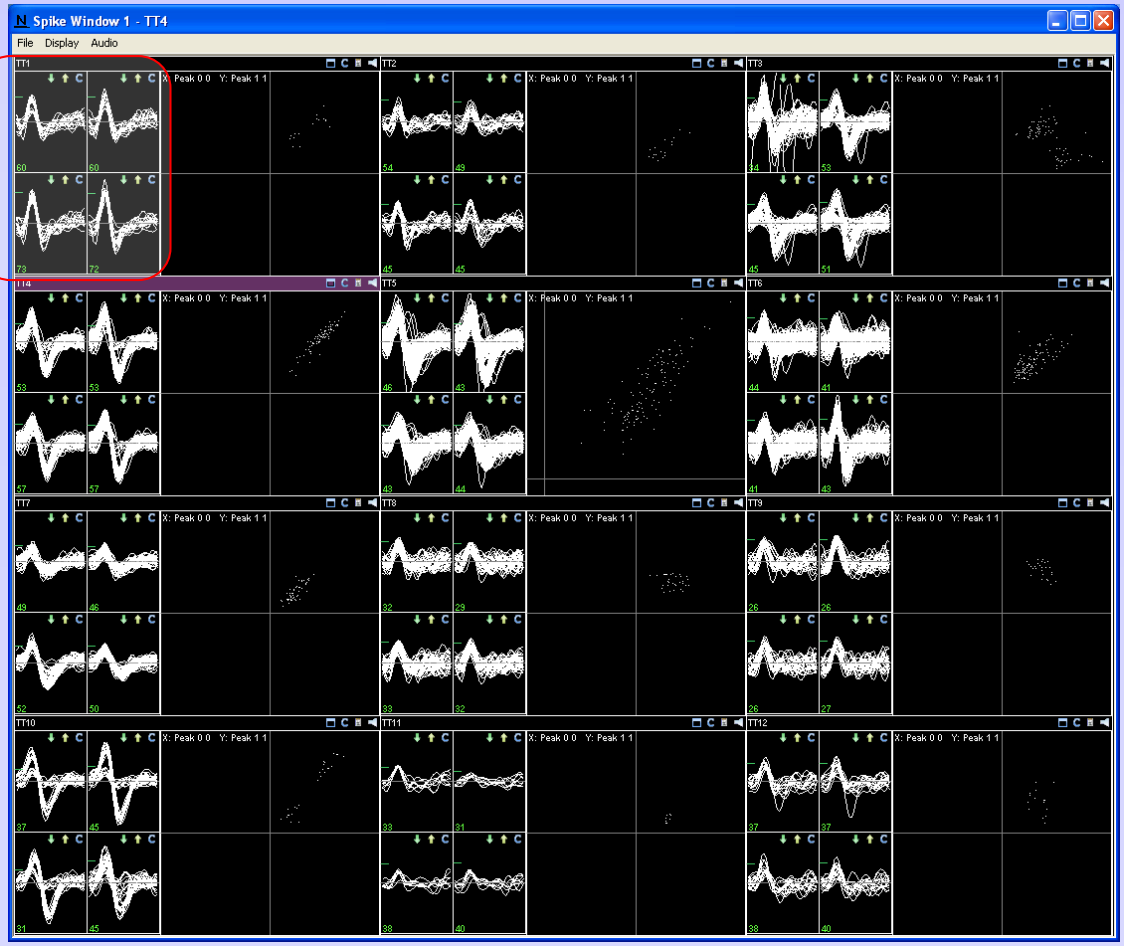
The Data

Question: How do we define/characterize a place field ?

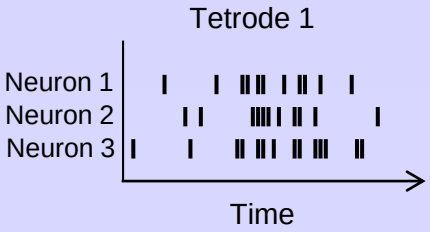
Raw Data: Spike data

tetrode

Action potential on
4 very close wires

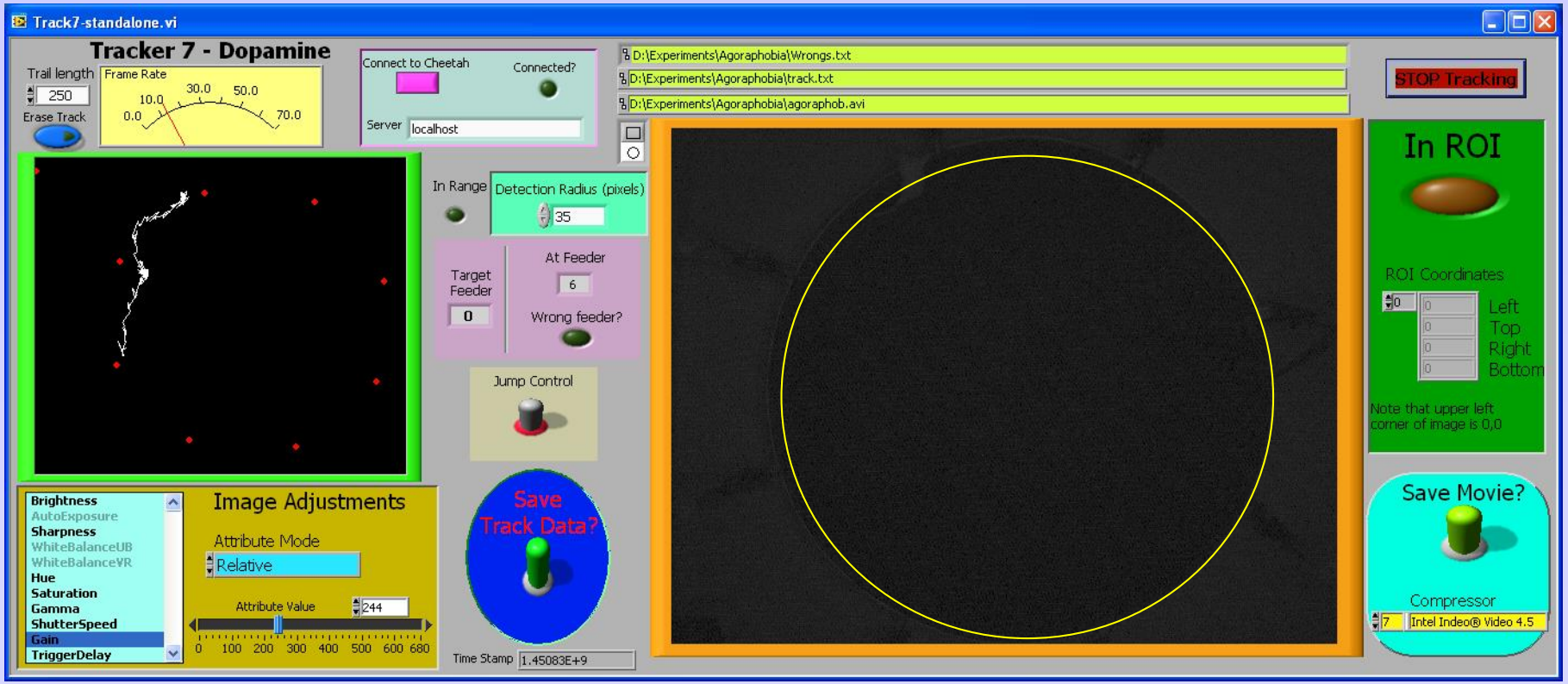


Spike Sorting



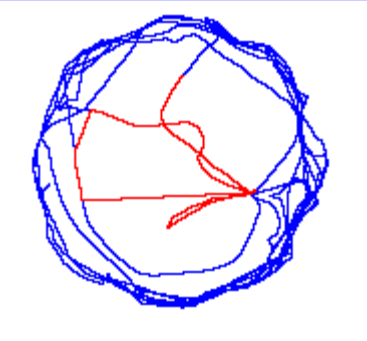
Data

Raw data: Position



Data

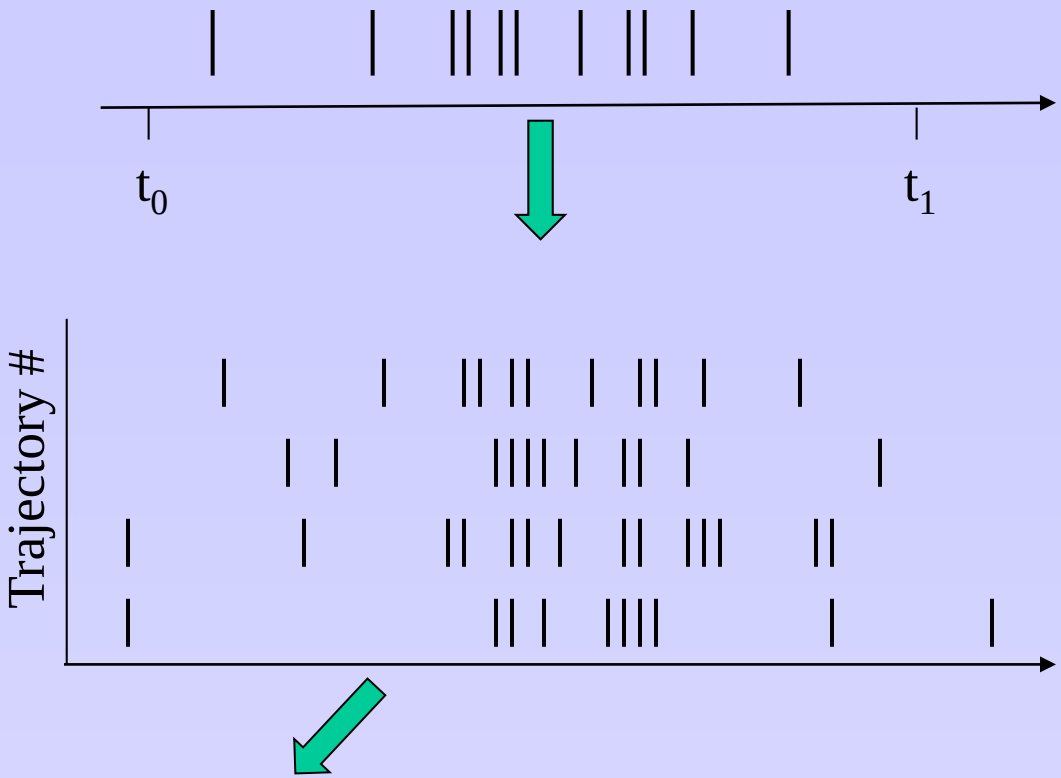
Re-coding: Spike cutting, position smoothing

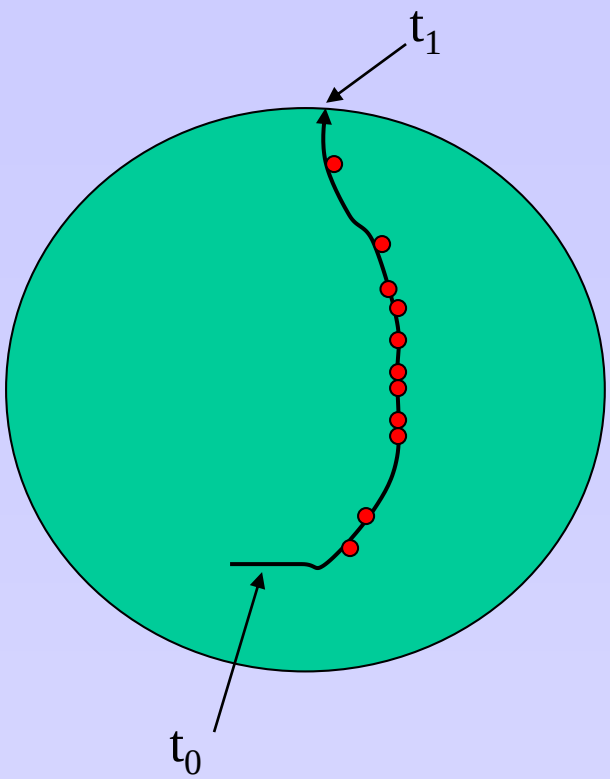


Data: time stamped spike trains + time stamped rat location

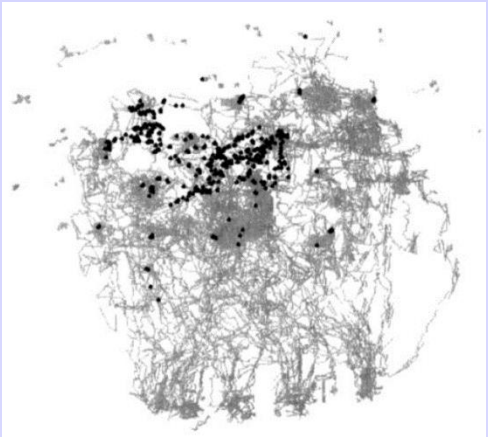
12.05 (secs)	11.1 s: (100,45) (pixels)
13.45	11.2 s:(120,58)
13.95	12.1 s:(156,71)
14.20	12.3 s:(130,79)
14.35	13.4 s:(137,121)
14.65	13.8 s:(145,150)
14.80	14.2 s:(129,170)
15.90	14.7 s:(133,180)
16.21	15.1 s:(120,201)
17.50	15.6 s:(116,230)
	16.4 s:(100,290)

Data Visualization

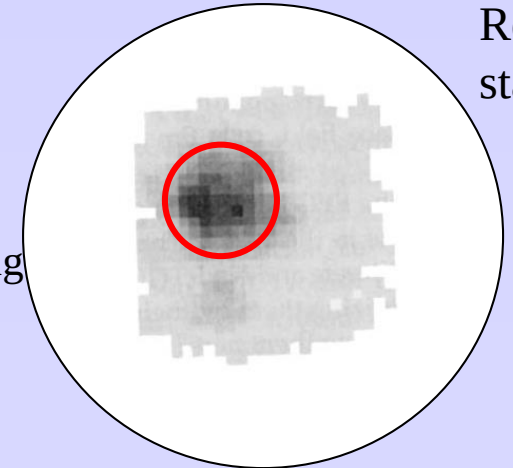




Representation of spikes (time stamps) and position (x,y coordinate

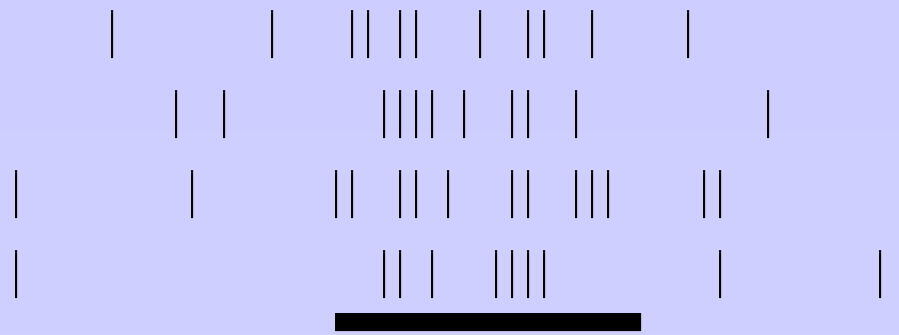


smoothing

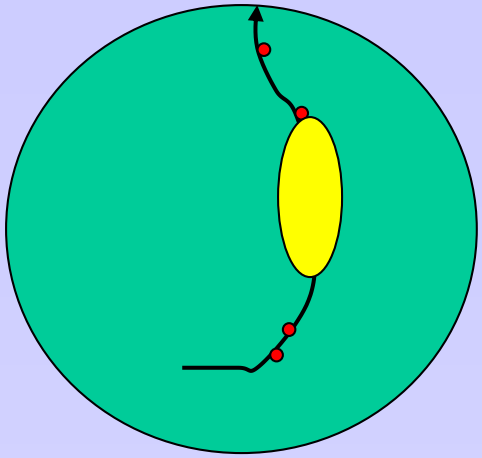


Size of place field
Location of place field
…

Data Analyses



Place field



Analysis1 → Spike count

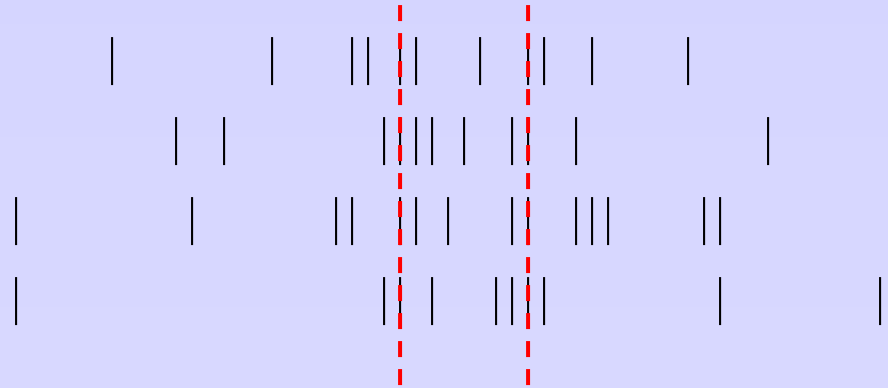
11,11,14, … -> average, std

Analysis2 → Spike rate

Analysis3 → Spike timing

(e.g. phase precession)

…



Surrogate Data Set(s)

How do we know that:

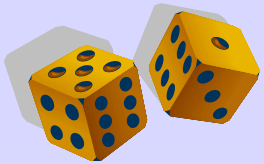
- the analysis algorithm ‘really’ works
- the result is not due to chance

<u>Construct a **surrogate data set**</u>

→ ◘ Use a biophysical model (detailed neuron)

◘ *Use a phenomenological model (abstract neuron(s): integrate and fire)*

→ ◘ Use a random process



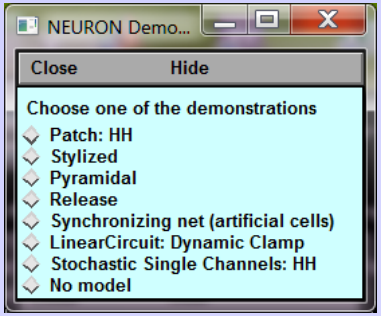
Surrogate datasets I: A Simple Biophysical Model

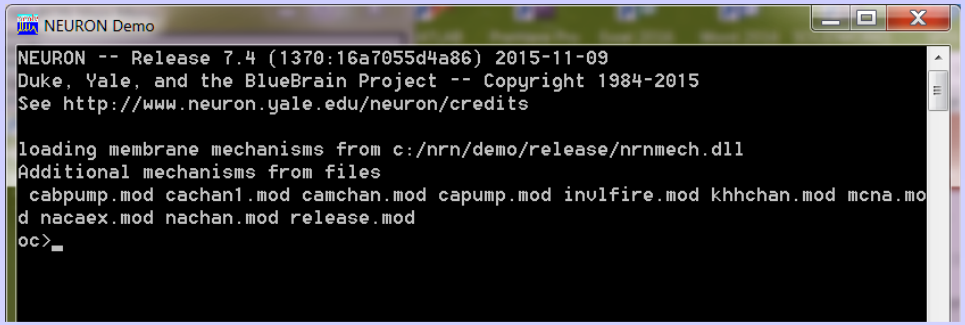
Goal: get (fake but realistic) data!

Install in default directories
Run 'NEURON Demo'

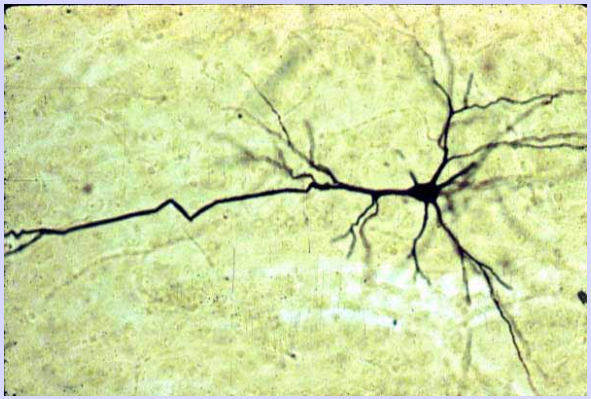
Use NEURON: www.neuron.yale.edu

https://neuron.yale.edu/neuron/docs





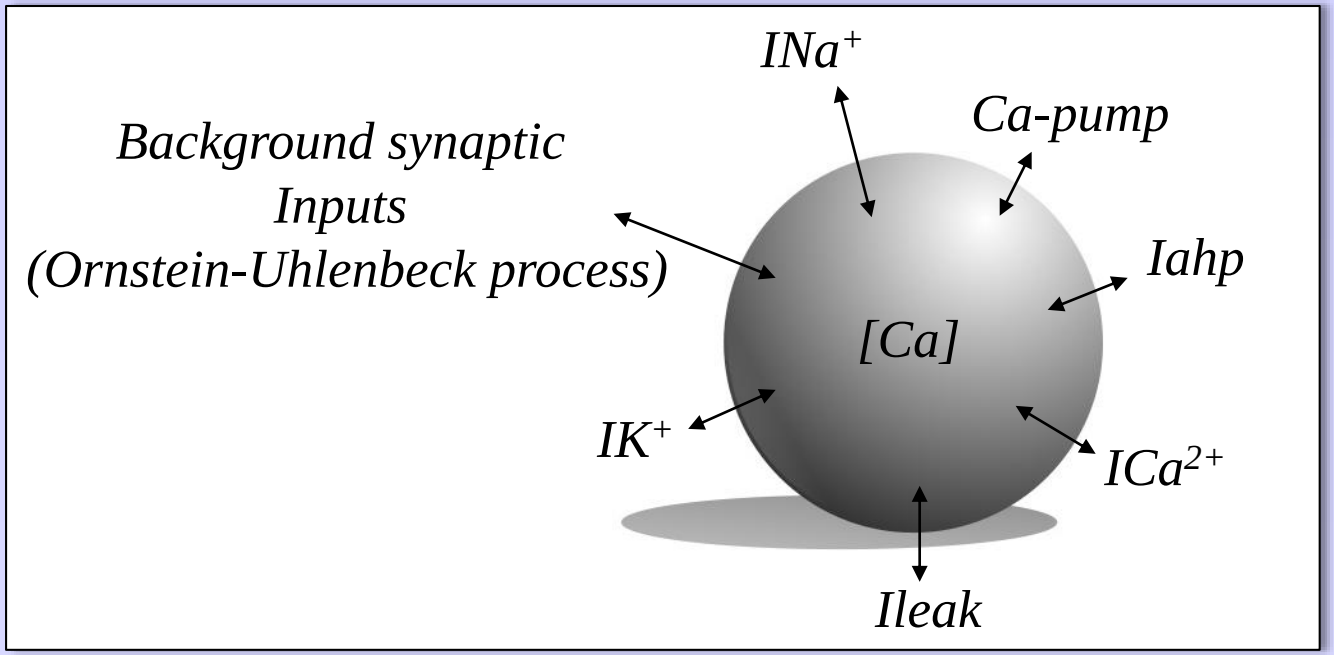
Note: we will not use Python in this class



- Simulate the spontaneous activity of a neuron (sub and super threshold).
- Accept arbitrary patterns of stimulation.
- Output (fake but realistic) spike times.

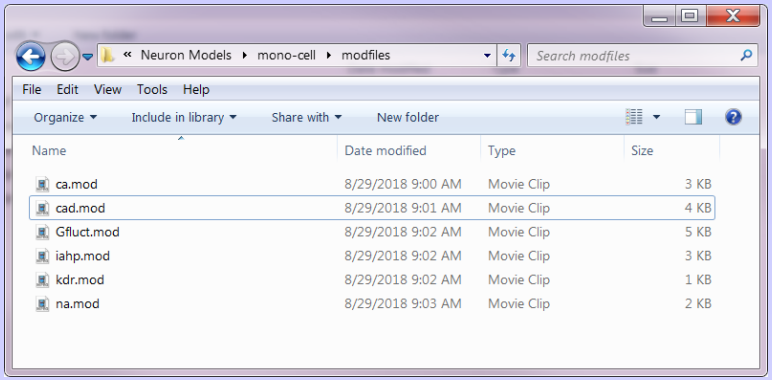
The model (SimpleNeuron.zip)

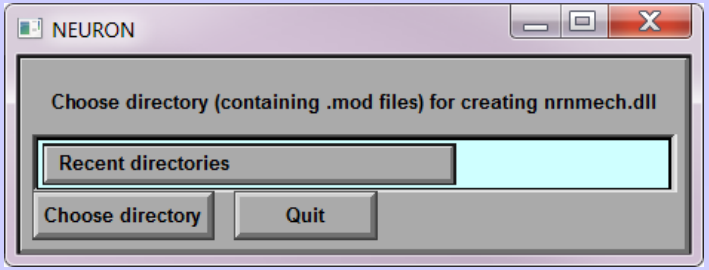
- Single compartment ('ball neuron').
- Generic cortical neuron tuning/parameters
- Multiple currents
    - Passive properties (Ileak, capacitance)
    - Action potential currents (INa^+, IK^+)
    - Spike frequency adaptation: Ca-dependent K^+ current (I_{ahp})
    - Calcium dynamics (ICa^{2+}, Ca-pump)
    - In vivo-like background/noise synaptic currents (inhibitory and excitatory)

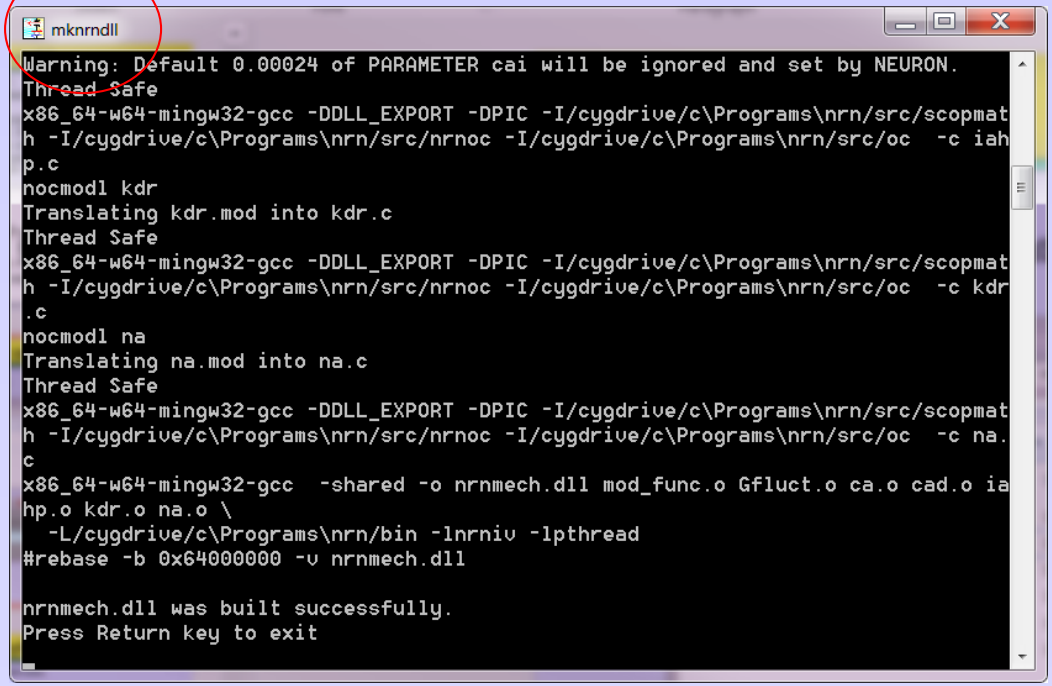


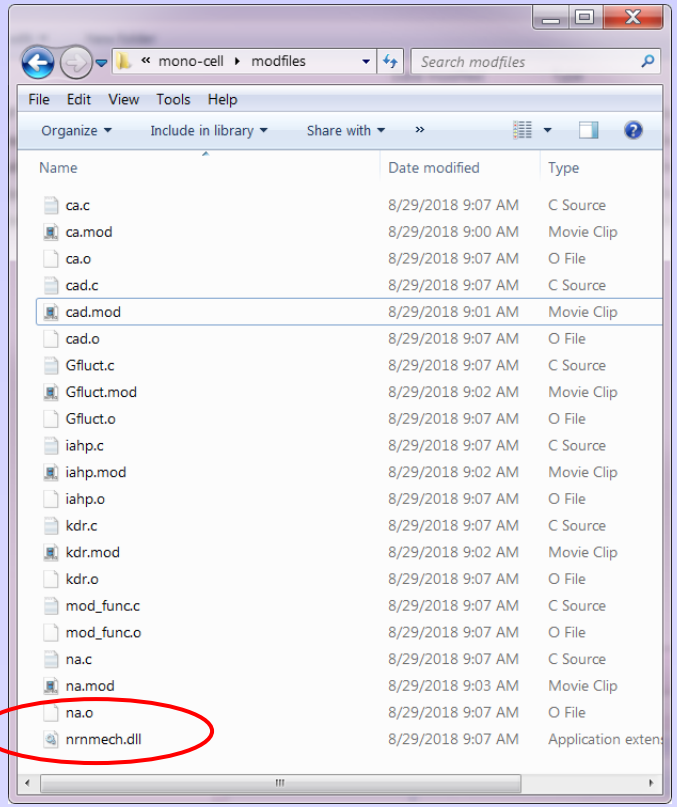
The model

Step1: Compile the currents (.mod files) – mknrndll.exe









Move ‘nrnmech.dll’ one folder up …

A Simple Biophysical Model: **simpleneuron.hoc**

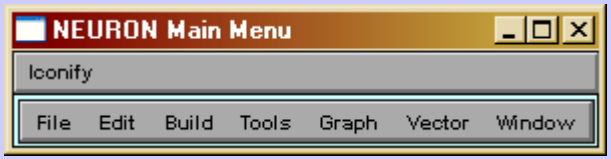
```
load_file("nrngui.hoc")                    // load default NEURON interface
nrnmainmenu()
nrncontrolmenu()


// Global simulation settings

dt=0.1
tstop = 1500
runStopAt = tstop
steps_per_ms = 10
celsius = 36
v_init = -70

//number of neurons
Nneurons=1

load_file("neuron.tem")                    // load the template for a single neuron
objectvar neurs[Nneurons]                  // declare how many neurons will be created

for i=0, Nneurons-1{
        neurs[i]=new neuron()              // create the neurons
}


xopen("graphprocs.hoc")                    // load basic graphic procedures
xopen("inout.hoc")          // load basic input/output procedures: InsertStim(), InsertVStim(), ReadStimVec(), RecordAP(), SaveAP()

addgraph("neurs[0].soma.v(0)",-90,30)      // plot the membrane voltage of neurs[0] measured at the soma.
```

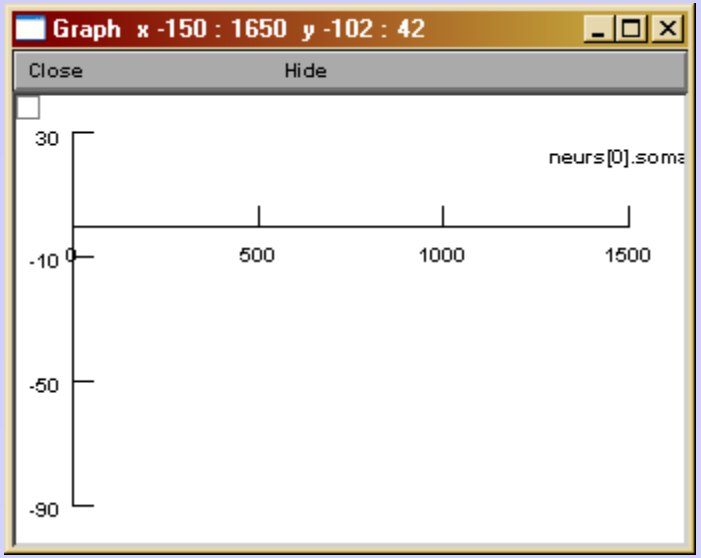
A Simple Biophysical Model: The ball neuron

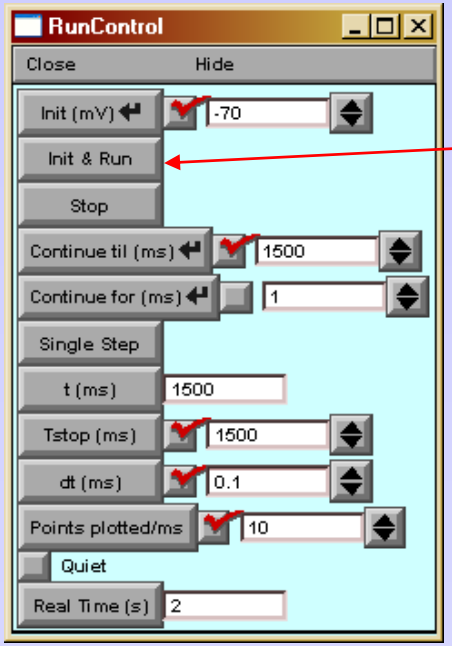
```
begintemplate neuron
public soma, apc, noise

objref apc, noise

proc init(){
    EK=-80
    ENa=55
    create soma

    soma{                   // only one compartment, the soma.
            nseg=1
            diam=70             // actually a cylinder…
            L=70
            Ra=250
    }

    access soma

    insert pas                  // leak current
    e_pas = -70
    g_pas = 4e-5
    cm=1

    insert Na    // insert the sodium Channels
    ena=    55
    g_Na= 0.03

    insert Kdr   // insert the potassium Channels
    ek=    -90
    g_Kdr=  0.005

    insert iahp                 // insert a calcium-dependent K current
    cac_iahp = 0.025
    beta_iahp = 0.03
    taumin_iahp = 0.1
    ek=EK
    gkbar_iahp = 0.5

     insert ca                  // insert a calcium current
    gbar_ca = 0.4

    insert cad                  // insert a calcium pump
    cainf_cad=2.4e-4
    caini=2.4e-4
    kd_cad = 1e-4
    kt_cad = 1e-4/5
    depth_cad = 1
    taur_cad = 1e10
     decay_cad=0.7

                                // insert background synaptic noise
    noise = new Gfluct2(0.5)

    noise.std_e = 0.008         // standard deviation excitatory
    noise.std_i = 0.02          // standard deviation inhibitory
    noise.g_e0=0.002            // mean excitatory
    noise.g_i0=0.05             // mean excitatory

    apc = new APCount(.5)       // insert an action potential detector

}
endtemplate neuron
```

Surrogate datasets I: A Simple Biophysical Model

Main menu

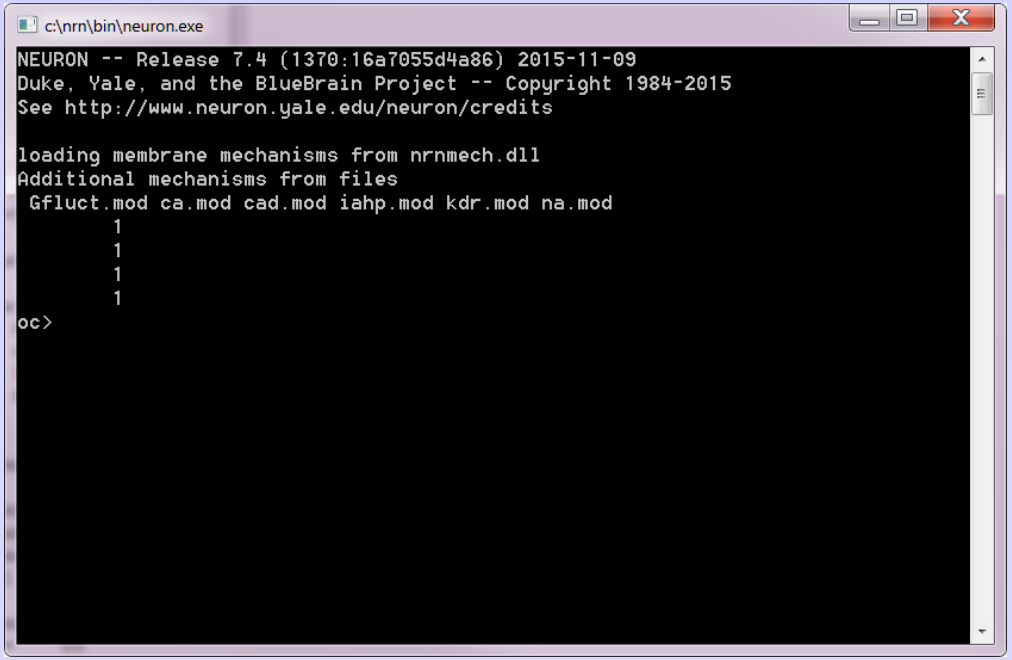






Initializes and starts the simulation

Shell window (interactive)



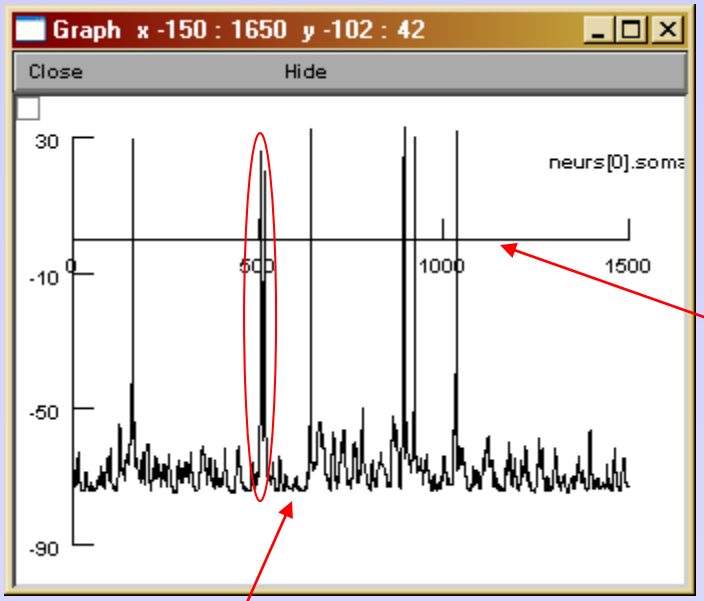
Surrogate datasets I: A Simple Biophysical Model

```
// Example session: Double click on simpleneuron.hoc (this file)
// in the shell window, create and initialize a stimulation electrode with variable stimulation
//                    InsertVStim(0)          // when prompted enter (for ex.) Stimpulse.txt (just a 1nA 20ms-long pulse at t=500ms)
//  then type
//                    RecordAP(0)             // to indicate that you want to record the actions potentials of neuron 0
// Click 'Init&Run' in the RunControl panel. You should see the membrane voltage fluctuate, and some spikes
// in the shell window type
//                    SaveAP()                // and give a file name when prompted. This will save the spike times currently recorded.
```

Stimulus input (Stimpulse.txt)

15000
0
0
…
1
1
…
0
…
0

nA



Time (ms)

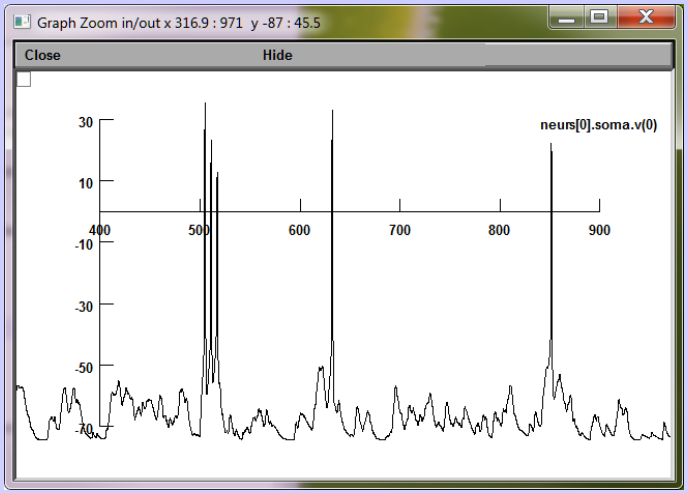
Membrane Voltage (mV)

Spike train(s)

160.4
509.2
516.6
642.1
892.5
921.5
1033.1

Analyses

Let's practice a bit…



- Zoom the voltage around 500ms

  (right click, View, Zoom in/out)

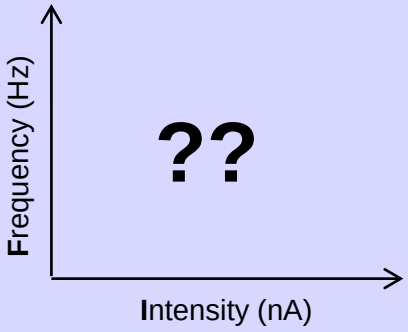
- Run the simulation ~10 times. What do you see ?

- Now increase the stimulus pulse a bit to, say, 1.2nA

  (Shell window: **oc> vec.mul(1.2)**)

- Run the simulation ~10 times. What do you see?

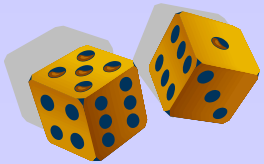
- Challenge…optional, home practice: Build the **IF curve** of this neuron (1 second pulse)



- See README.txt regarding the format of the stimulus file *Stimpulse.txt*
- You can certainly do 'everything' in Matlab, but hand counting and Excel are also just fine…

Surrogates II: Use a Random Process (Matlab)

Generate sequences of random numbers between 0 and 1500 ms with certain properties. Neurons are 'point processes' (see Johnson 1996)



- Absolute refractory period (2 ms)

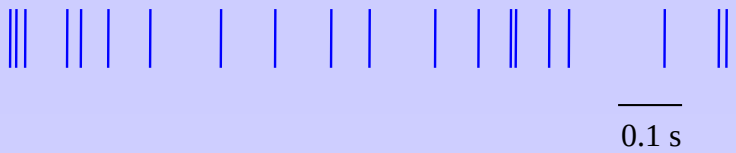
- Relative refractory period

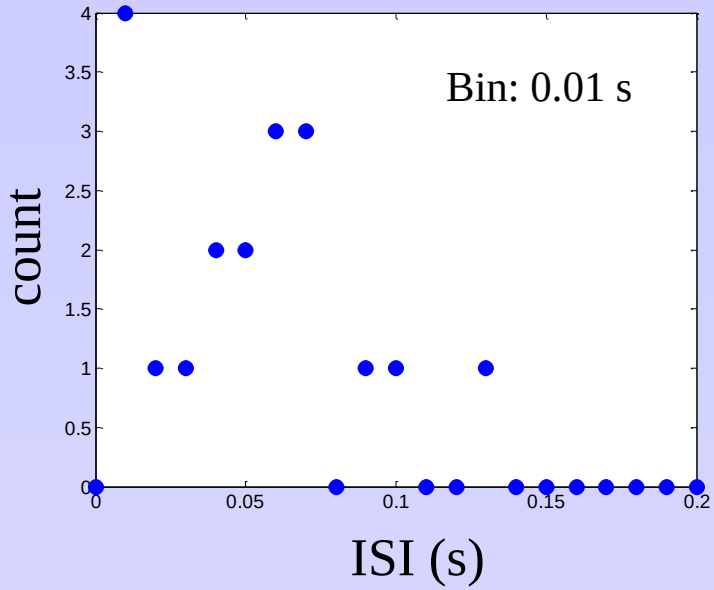
  - Probabilistic: $p(t_{n+1})=1-\exp((t_n-t)/\tau)$, with $\tau=300$ ms.
  - Activity/history dependent: $p(t_{n+1}) = f(ISI_n, ISI_{n-1}, \ldots)$.
    (**Possible Project**: Determine f() for the NEURON model)

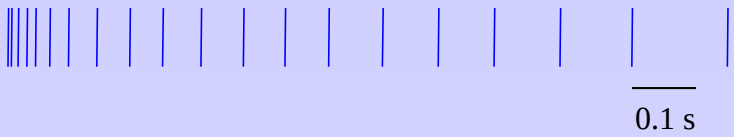
- Statistical Distribution of **Inter-Spike Intervals** (ISIs). Typical ones:

  - Uniform
  - Gaussian (mean, standard deviation)
  - Poisson (given rate)
  - Gamma

Inter Spike Interval - ISI

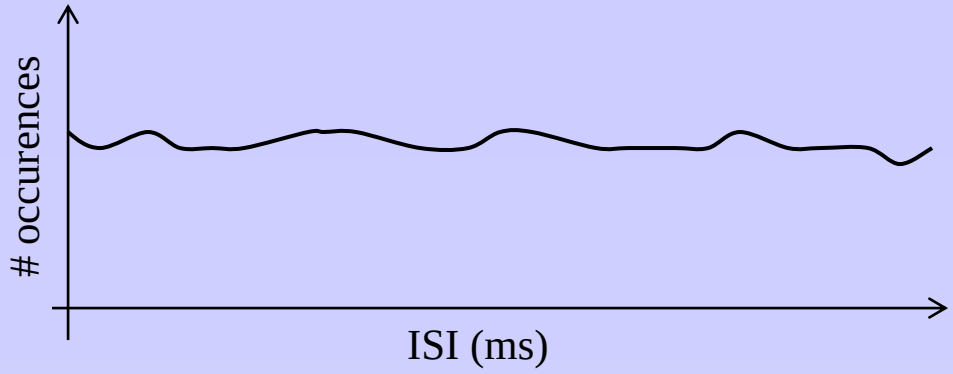




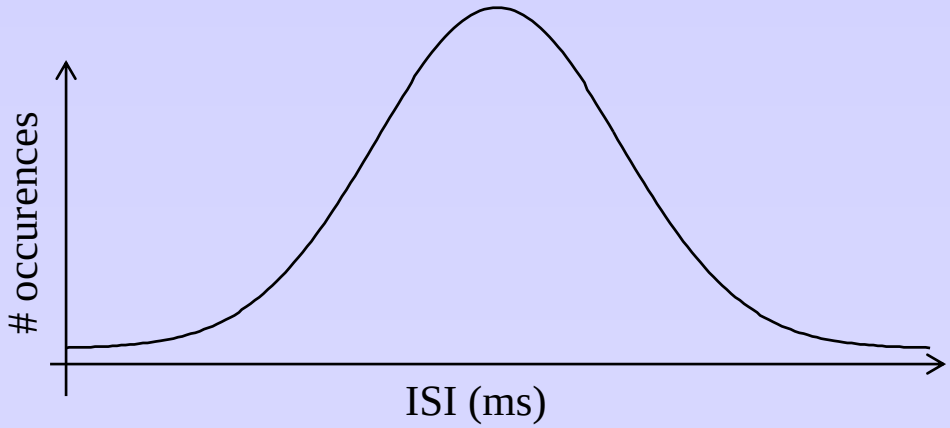


- Characterization of a spike train.
- Note: A spike train has a unique ISI distribution, but many spike trains may have the *same* ISI distributions.

Uniform and Gaussian Distributions



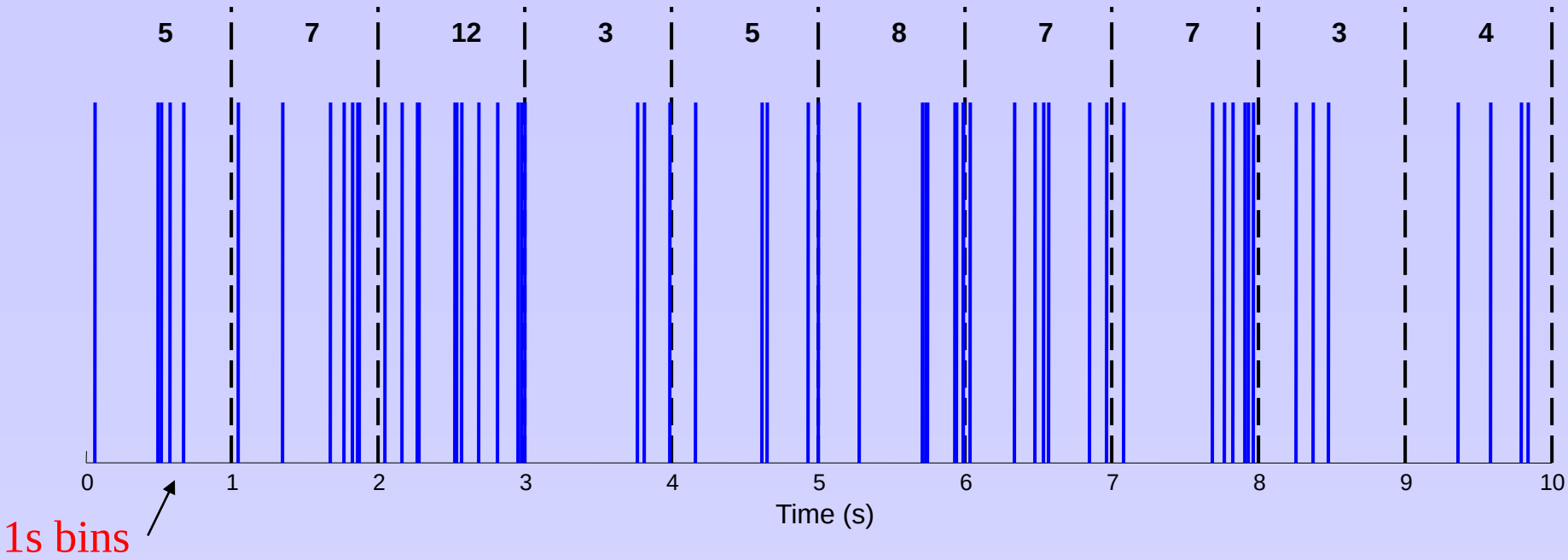
- The time between spikes (ISI) is truly random. No ISI appears more often than any other.

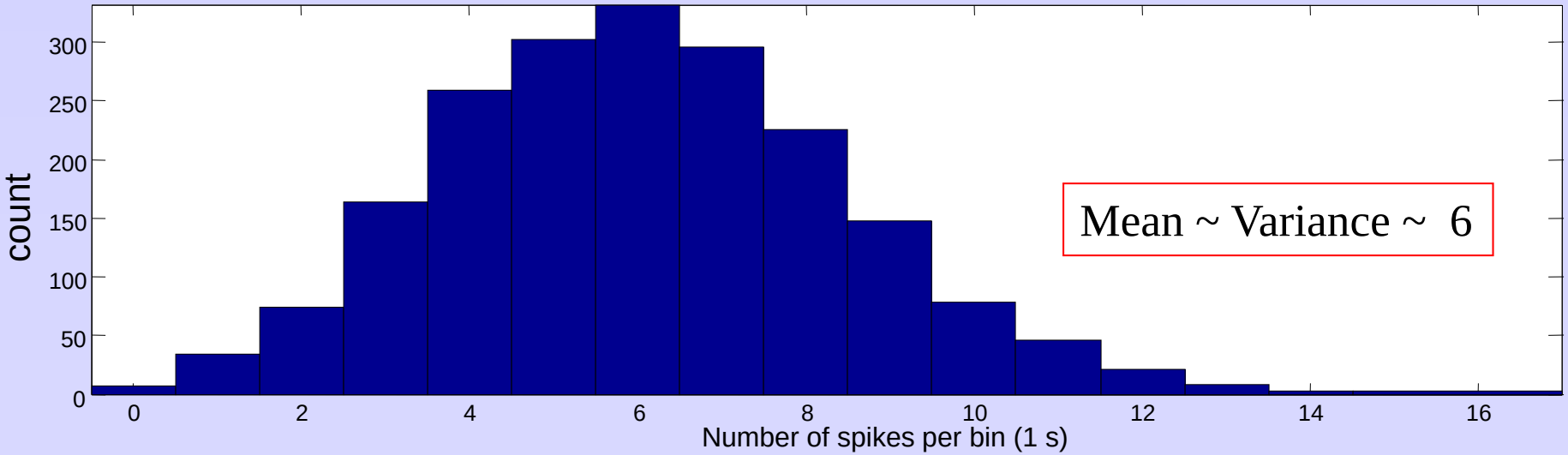


Johann Carl Friedrich Gauss
(1777-1855)

- Some ISIs appear more often than any other. Frequency preference. Also called ‘normal’ distribution. The most frequent ISI (‘mode’) is also the mean ISI.

Homogeneous Poisson Distribution (6 Hz)





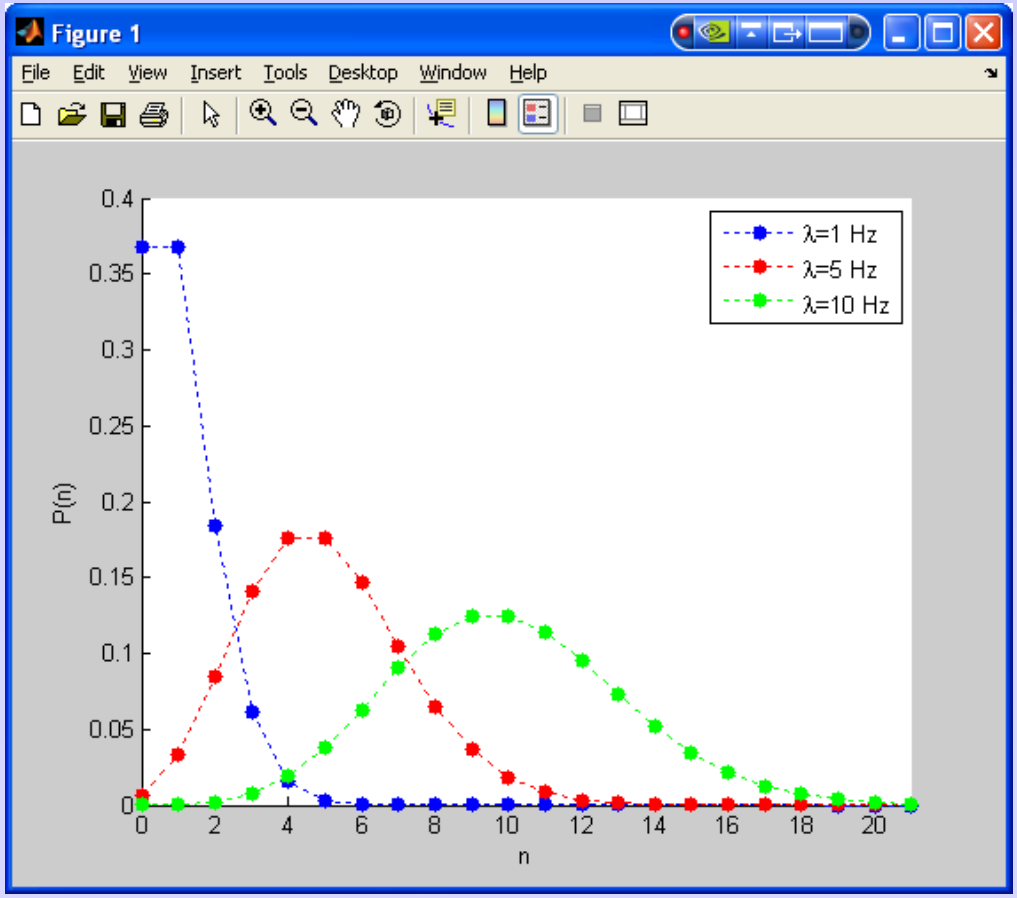
Homogeneous Poisson Distribution

- Spikes are independent from each other. Probability of getting n spikes (in T seconds), when λ are desired on average

$$P(n) = \frac{\lambda^n e^{-\lambda}}{n!}$$

(see derivation in Johnson 1996)

Siméon-Denis Poisson. 1781-1840



(T= 1 sec)

Homogeneous Poisson Distribution

- Mean = λ = variance = σ^2 of the spike count.

- The probability (P) of having a spike between τ and $\tau + \Delta t$ (i.e. an inter-spike interval τ) is given by:

$$P(\tau) = \Delta t . r e^{-r\tau}$$

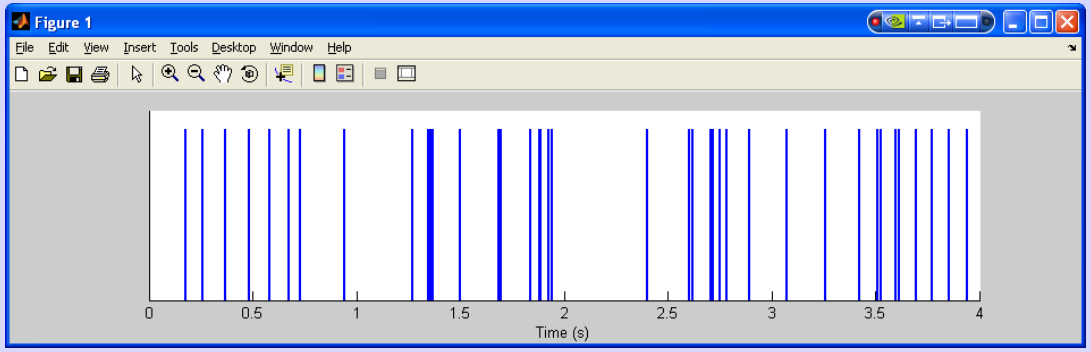
(see Johnson 1996)

where r is the mean firing rate (Hz). In T seconds, λ=rT.

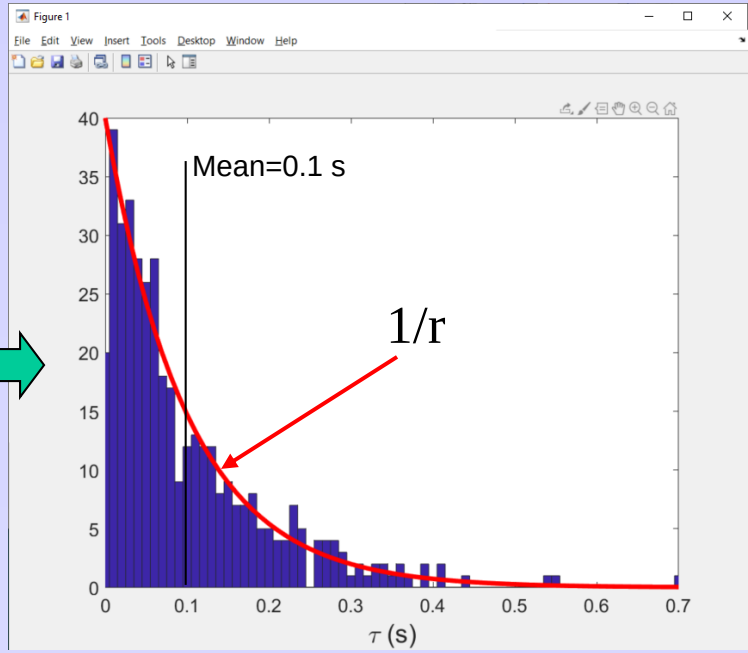
- The distribution of τ (ISIs) is exponential.

Note: the mean ISI is **not** the preferred ISI!

(the mean is not the mode).



10 Hz



Homogeneous Poisson Distribution

A practical algorithm…

For a constant firing rate (i.e. ‘homogeneous’, i.e. ‘stationary’) spike times can be computed with:

$$t_{i+1} = t_i - \frac{\ln(x)}{r}$$

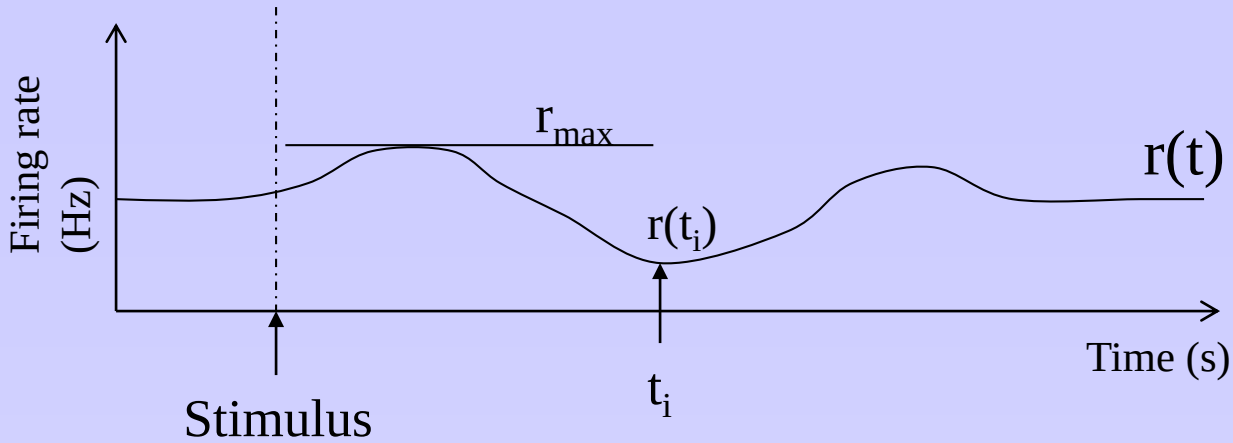
With x uniformly distributed in]0 1[

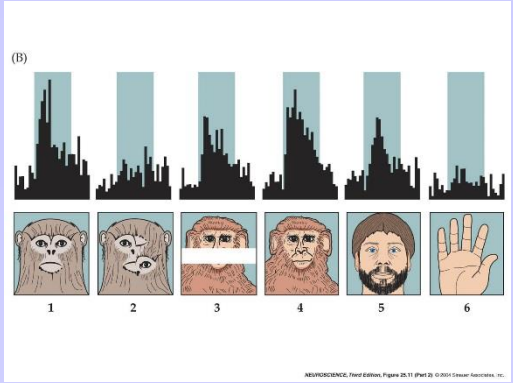
⇨ ‘Renewal Process’
(dependence on previous spike time only)

Problem: if x close to 1 ⇨ $t_{i+1} \approx t_i$ (not possible in general)

(i.e. no refractory period)

Inhomogeneous Poisson Distribution





- Generate a homogeneous Poisson process

$$t_{i+1} = t_i - \frac{\ln(x)}{r_{\max}} \Longrightarrow \text{‘Renewal Process’}$$

- Rejection sampling (‘thinning’) method

For each i, estimate r: $r(t_i)$, and get a random number x’ in]0 1[
If $r(t_i)/r_{max}$<x’, delete spike i

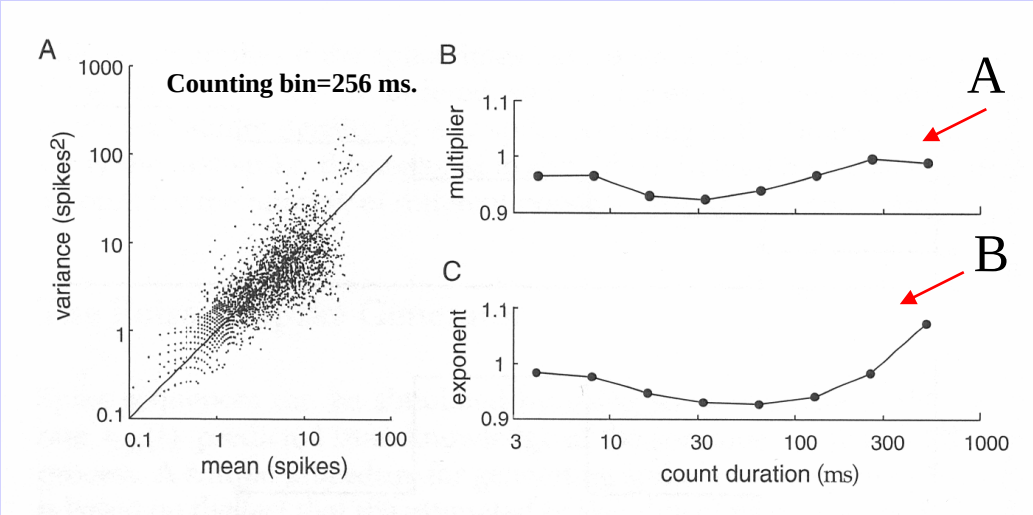
(see also Berry and Meister 1998)

- **Project:** How does one decide if a Poisson spike train is stationary or not?

(see Johnson 1996)

The real data is (often) not strictly Poisson

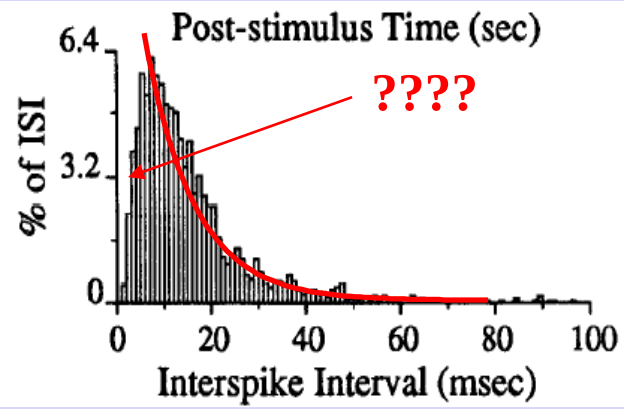
Area MT, awake monkey, moving visual images.
(Dayan and Abbott book, p32-)



$$\sigma_n^2 = A\langle n\rangle^B$$

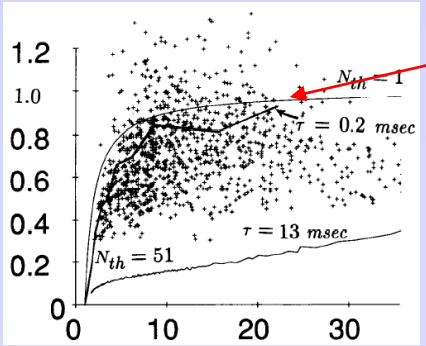
In general, A and B are in [0.5 1.5]

Area MT, awake monkey, moving random dots.
(Dayan and Abbott book, p33-)



V1 and MT, awake monkey
(Softky and Koch 1993)

$$CV = \frac{\sigma_{isi}}{<isi>}$$

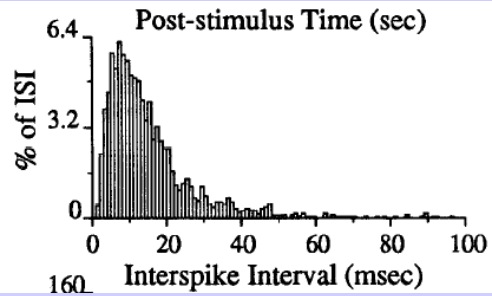


Poisson
(with refractory period)

<*isi*> (mean ISI, ms)

Gamma Distribution

- Distribution of ISI (τ ms) follows a **gamma** distribution

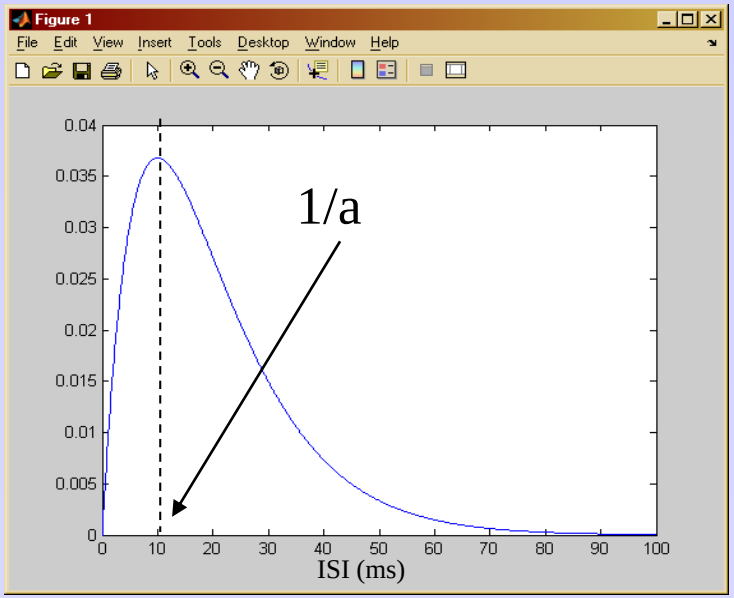


$$p(\tau) = \frac{a(a\tau)^k e^{-a\tau}}{k!}$$

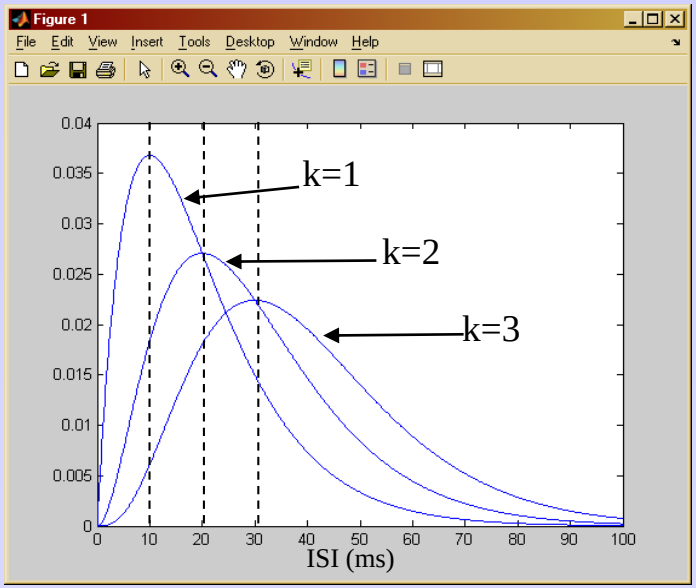
k integer >0

k='shape'

a=0.1, k=1



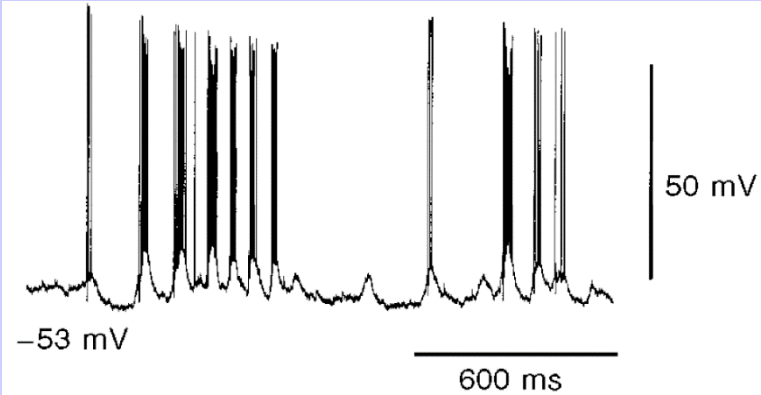
a=0.1



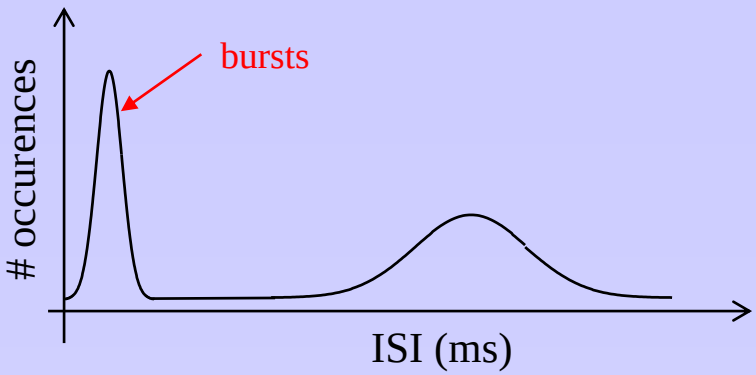
- Note: Poisson distribution: k=0.

Most frequent ISI= k/a,
<ISI>=(k+1)/a

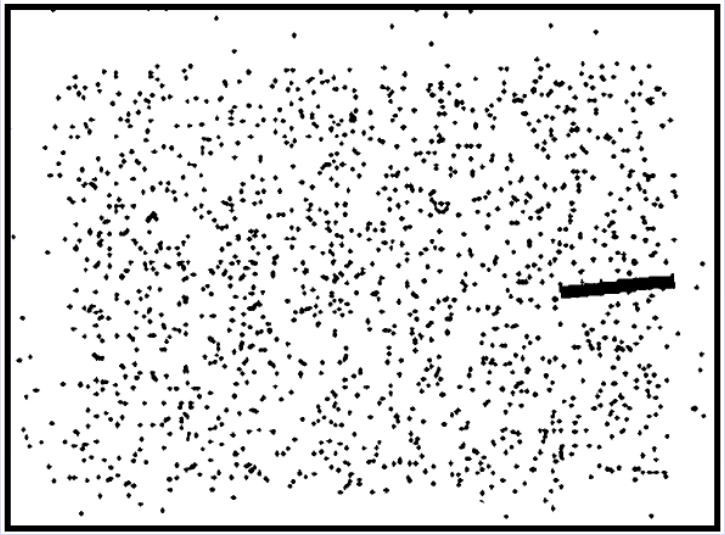
Special ISI Distributions….Bursting



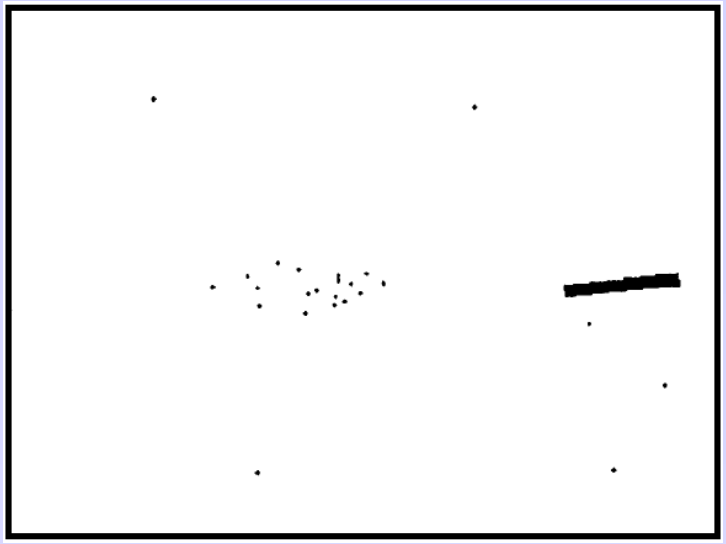
(rat thalamic neuron during sleep)



- Are bursts special?

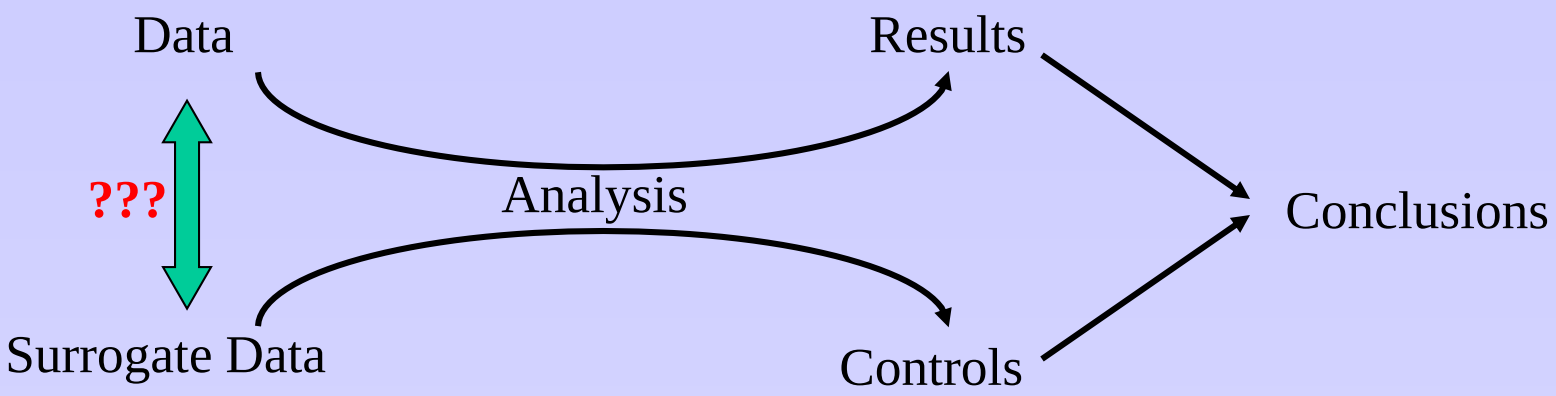


Bursts only (*4 spikes 10 ms–1)*



(Awake monkey watching a flashing bar, cell in V1)

Assessing the quality/adequacy of the surrogate set

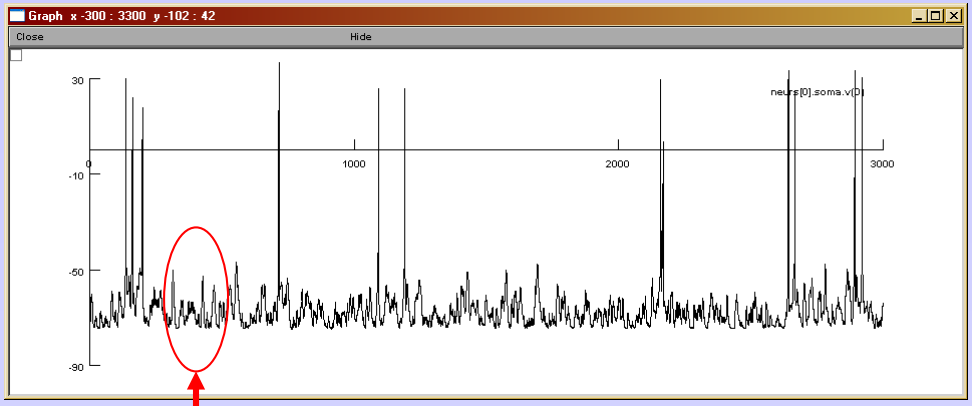


- Data contains features above and beyond those that would be obtained by ‘chance’.
- Results do/do not depend on some parameter(s) used to generate the surrogate set.

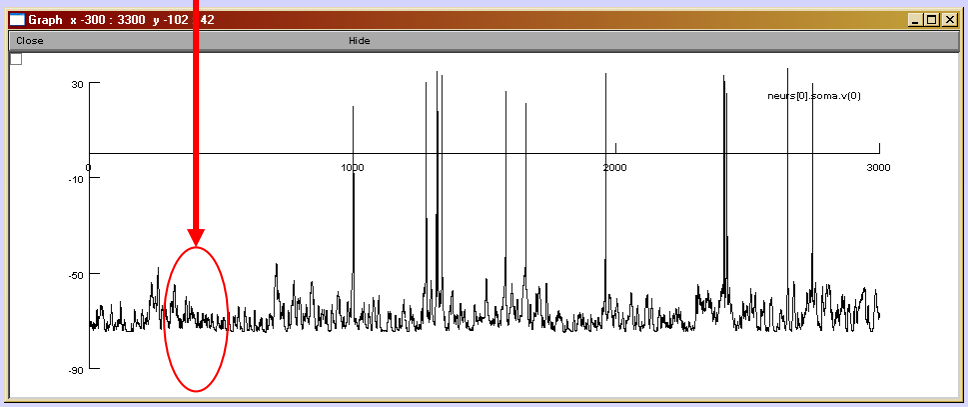
Need a ‘convincing’ surrogate set
Need to **compare** real/surrogate datasets

Comparing Neural Responses: An Open Question

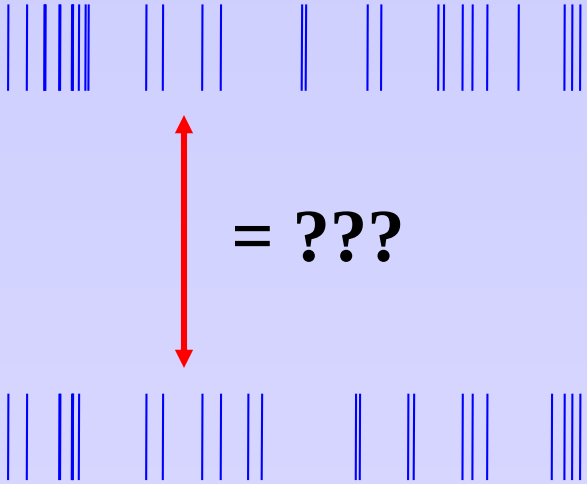
Membrane potential Metrics…



= ???



Spike train Metrics…..



= ???

More on this soon…..

Interim Summary

- General introduction:
    - Neurons and synapses
    - Basic neuroanatomy
    - Basic neurophysiology (action potential, E/IPSPs, integration)
    - Methods in brain Research

- General Issues in Neural Data Analyses
    - Quantitative Vs Qualitative Analyses
    - Breadth-first Vs Depth-first Analyses
    - Data Representations

- Surrogate Datasets
    - Simulation data (NEURON models)
    - Point processes
        - Refractory period and stationarity
        - Distribution of ISIs (Gaussian, Poisson, Gamma)
    - Comparing Neural responses

Homework1: Due next week

- Write a function that takes N spike trains (response of a neuron to N trials), and display all spikes, all trials in a graphical form (i.e. rasterplot).
Note: Each spike train could be a hard-coded MATLAB ‘cell’ {}.

- Write a function that will return N spike trains, T seconds long, distributed in a homogeneous Poisson manner (rate *r*). Make sure the spikes have an absolute refractory period of 2 ms (hard-coded). Display 5 such realizations using the function above (N=20, T=2, *r*=20Hz).

Note about format (1 printout and email a zip file containing):
- A word file with screen shots of the results and text/figure caption containing all the parameters you used to obtain the figure.
- Whatever *commented* matlab code necessary to reproduce the figure. The naming convention for file names is:

<your initials (3 letters)><version number (2 digits)><function name>.m

Ex: JMF01DisplaySpikes.m