Integrate and Fire

Balazs B Ujfalussy

25/09/2017

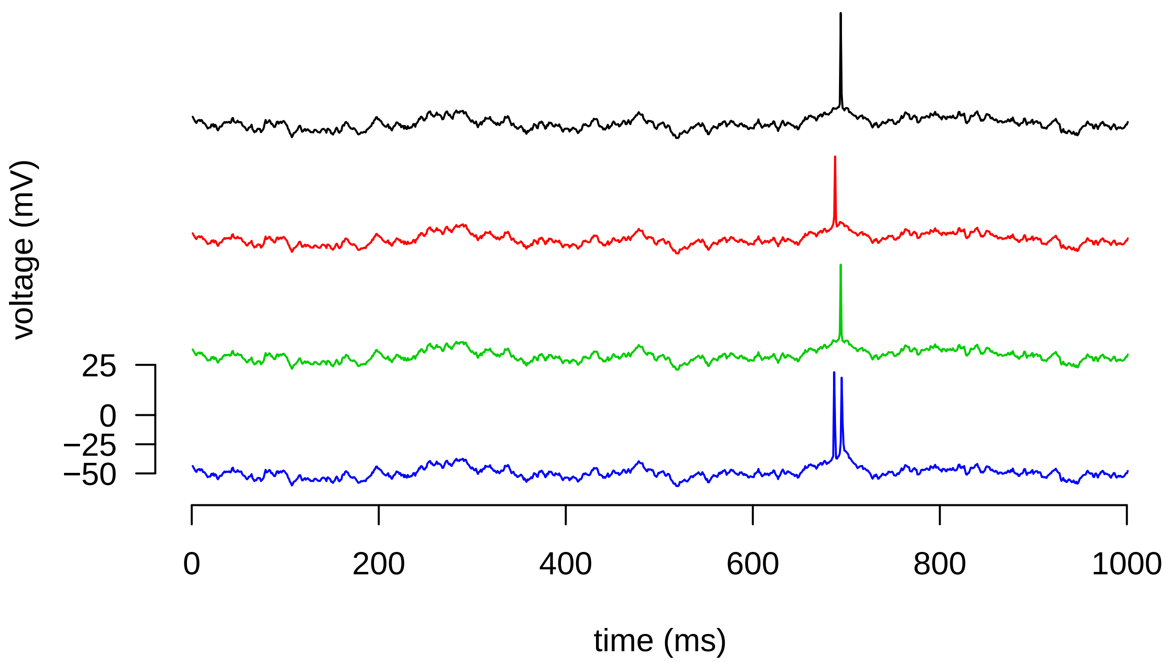
This is a demo for illustrating the response of different neuronal models to somatic current injections.

```
source('./IF_sim.R', chdir=T)
source('./Izhikevich_sim.R', chdir=T)
source('./HH_sim.R', chdir=T)
```

The models are compared with experimental data from *The quantitative single-neuron modeling competition*. In this experiments a cortical pyramidal neuron was stimulated by injection of randomly fluctuating currents of various amplitudes. Current was injected and voltage responses were recorded at the soma. For more details, see Fig. 1 of Jolivet et al., **The quantitative single-neuron modeling competition** Biol Cybern (2008) 99:417–426.

The experimental data is loaded from the `Data` folder - Jolivet_resp.RData and `Jolivet_stim.RData`. First we load the data and hen plot a short example of it.

```
load("./Jolivet/Jolivet_stim.RData")
load("./Jolivet/Jolivet_resp.RData")
matplot(t(resp[1:4+4*2,3000:4000] - c(0, 1, 2, 3)*100), t="l", lty=1, axes=F, xlab="time (ms)", ylab="vo
```

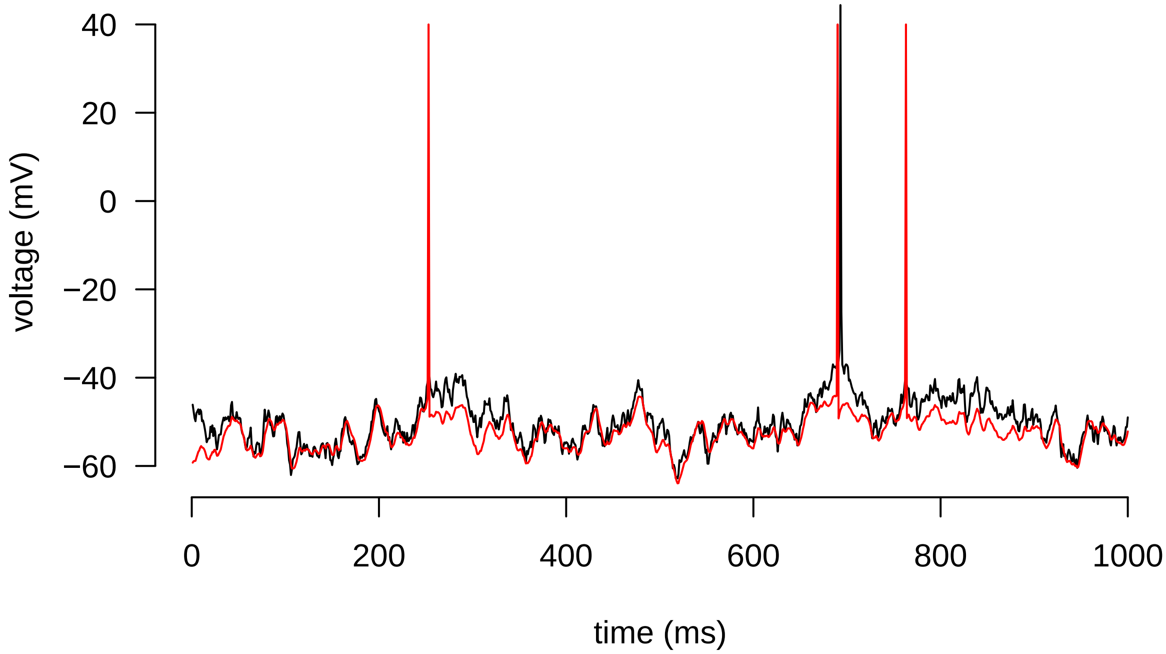


There are 8 different stimulus, and 4 recordings with each - so we have altogether 32 stimulus response pair. The stimuli differ in their mean and variance, otherwise they are filtered Gaussian noise.

Next, we select a 1s long portion of the data - both the stimulus and the response. We select the 9th stimulus response pair, which has a large input variance but low mean. We will inject the same current to an integrate and fire neuron and record its response. We adjusted the parameters of the IF neuron to roughly match the real neuron's response.

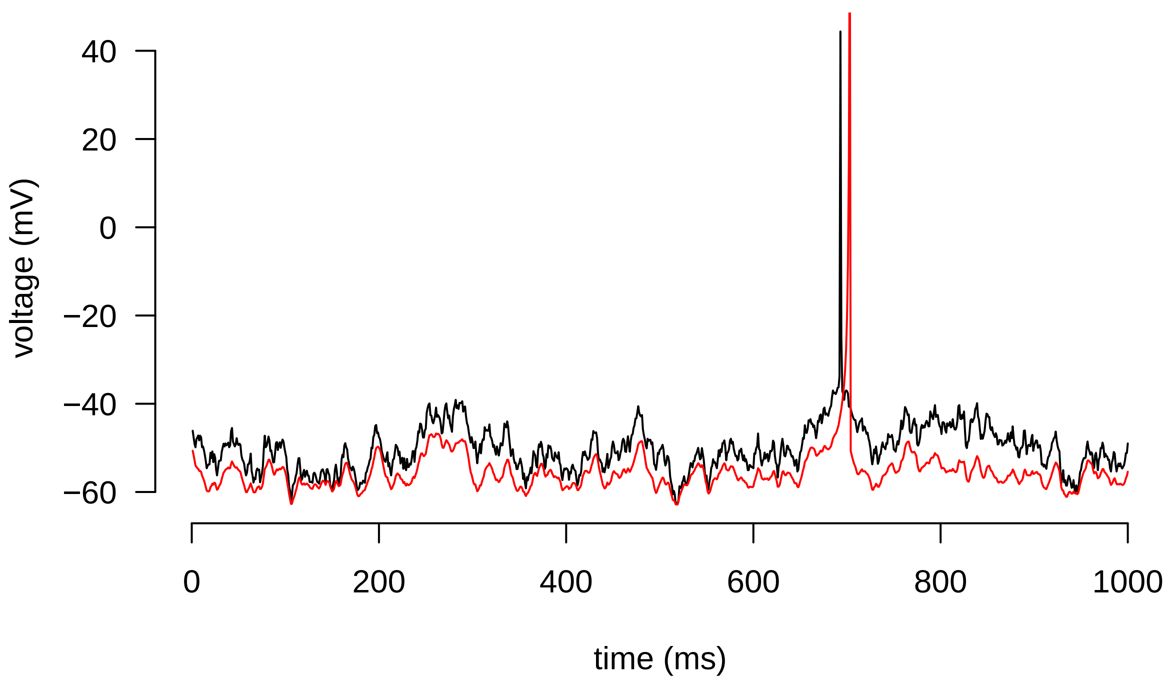
```
r.cell <- resp[9,3001:4000 ]
input <- stim[9, 3001:4000]
```

```
v.IF <- sim.IF(I=input, v.rest=-60, Rm=0.045, tau=10, v.threshold=-44, v.reset=-50)
plot(r.cell, t="l", lty=1, axes=F, xlab="time (ms)", ylab="voltage (mV)"); axis(1); axis(2, las=2)
lines(v.IF, col=2)
```



We repeat the same experiment now using an Izhikevich neuron.

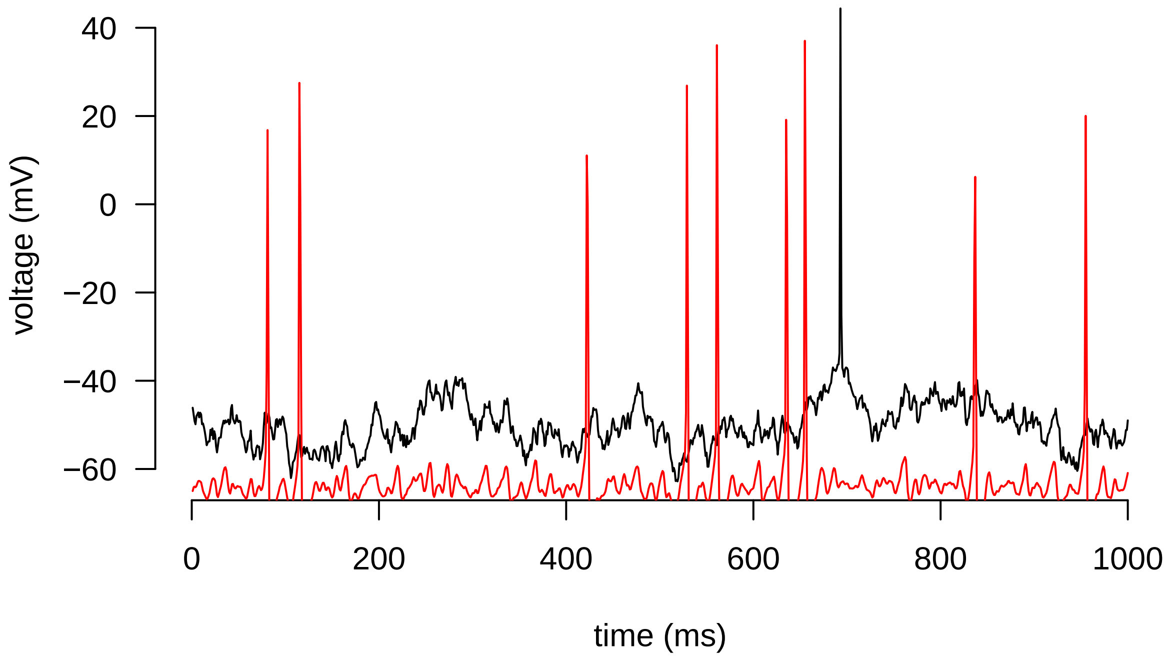
```
v.Iz <- sim.Izhikevich(I=input, Rm=0.045, tau=10, v.init=-50)
plot(r.cell, t="l", lty=1, axes=F, xlab="time (ms)", ylab="voltage (mV)"); axis(1); axis(2, las=2)
lines(v.Iz, col=2)
```



We repeat the same experiment now using a Hodgkin-Huxley cell.

```r
I.ext <- approxfun(1:1000, input/1000/1000/10, method = "linear", rule = 2) # this is necessary to prov
params <- c(gK=gK, gNa=gNa, gL=gL, cm=cm, E.Na=E.Na, E.K=E.K, E.L=E.L) # parameters of the system.
state <- c(v=-65, m=.053, h=.596, n=.317) # initial state of the system
times <- 1:1000
v.HH <- ode(y = state, times = times, func = sim.HH, parms = params)
```

```r
plot(r.cell, t="l", lty=1, axes=F, xlab="time (ms)", ylab="voltage (mV)"); axis(1); axis(2, las=2)
lines(v.HH[,1], v.HH[,2], col=2)
```



Homework

Select a stimulus response pair, and change the parameters of the three models to match the actual data! Summarise your experience: Which of the three models was the easiest to fit? Why? Are you conteent with the quality of fitting? If not, what are the typical errors you observe, and how could you correct them?