The neurobiology of drug withdrawal: insights and hints from computational modeling

Computer exercises using the NEURON Simulator

5th Baltic-Nordic Summer School on Neuroinformatics
October 6-7, 2017 Kaunas, Lithuania

Author: Dr. Paolo Enrico, Dept. of Biomedical Sciences,
University of Sassari, Italy
Email: *enrico@uniss.it*

Overview

This tutorial will guide you exploring the characteristics of our model of a dopamine (DA) cell of the Ventral Tegmental Area (VTA) of the rat. The model has been developed and used in Enrico et al. (2016) Neuroscience 322: 195-207. doi: 10.1016/j.neuroscience.2016.02.006, to study how morphine withdrawal-induced neurochemical and morphological changes could affect the electrophysiological behaviour of VTA DA cells. The model findings suggested how and to what extent a change in the balance of GABA/GLU inputs can take into account the experimentally observed hypofunction of VTA DA neurons during acute and prolonged withdrawal, whereas morphological changes may play a role in the increased excitability of VTA DA cell to opiate administration observed during opiate withdrawal.

Our model make use of the NEURON simulator. While it is not necessary to be familiar with NEURON before attempting to run it, previous experience will allow you to progress faster and to better understand the model.

Prewritten code is available for all the exercises; the model code is commented for your convenience and you're requested to directly work on it. For this you will need some familiarity with the *hoc* scripting language (a little *C* will also do), and a text editor (Emacs is suggested, but other text editors such as Notepad, Gedit or Nano are ok too). If you would like to develop more experience with NEURON and *hoc* before using the model, then you'll find useful material in the “Documentation” and “Programmer's Reference” sections on the NEURON simulator web site at: www.neuron.yale.edu, as well as in the NEURON Forum: www.neuron.yale.edu/phpBB.

The model code

You'll find the code of the model stored in an appropriately named folder containing several “.hoc” files containing the code that defines the simulation. These are accompanied by a “.ses” files that contains graphic user interface (GUI) definitions, and several “.mod” files that define model components, such as ion channels.

Running the code

Linux

1. In a terminal window, change to the folder containing the desired HOC file (command “cd”).
2. Rebuild NEURON to incorporate model components (“.mod” files) by typing: “nrnivmodl”.
3. Run NEURON by typing “nrngui filename.hoc” eg “nrngui DA_release.hoc”.

OSX

1. GUI-based: The GUI way to compile the ".mod" files is to drag the model folder onto the mknrndll icon, and then drag the seed ".hoc" file onto the nrngui icon.
2. non-GUI (suggested): open a terminal window and follow the Linux instructions (see above).

Windows

1. In Windows Explorer, change to the folder containing the desired ".hoc" file.
2. If the folder contains ".mod" files, first rebuild NEURON to incorporate these model components by running "mknrndll" on this folder (you will find this command in the NEURON Start group).
3. Run NEURON by double-clicking the relevant ".hoc" file eg "DA_release.hoc".

Working with the NEURON simulator

The NEURON simulator provides you with two different methods to build models (both individual neurons and networks of neurons) and managing simulations.

- GUI tools - Neuron models and small networks can be constructed in NEURON using built-in GUI tools, with no programming skills required. Of course, your control on the simulation will be limited.
- Hoc programming – NEURON command line interface (CLI) gives you access to the *hoc* scripting language by typing commands at the *hoc* prompt (oc>). To gain basic experience in constructing NEURON simulations by programming them using *hoc*, try the Neuron tutorial by Gillies and Sterratt (web.mit.edu/neuron_v7.4/nrntuthtml/index.html).

In this tutorial you'll use both GUI and CLI to run and manage the simulations, all *hoc* commands in this documents are typewritten in italic and underlined (eg *psection()*).

At any moment you may safely stop the running code using the - *crtl+c* - key combination. To exit the simulator using the GUI, select from the NEURON toolbar the File menu and then Quit; otherwise use the command *quit()* at the *hoc* prompt.

NEURON and Python

While you can still develop and run NEURON models with *hoc*, this scripting language is now quite outdated. In order to overcome *hoc* limitations, a Python interface to NEURON has been developed

and is now extensively used. Python is an interpreted language with a precise and efficient syntax and is one of the most diffuse languages in the scientific community. Python makes available within NEURON a very extensive suite of analysis tools and in particular SciPy (https://www.scipy.org), a Python-based ecosystem of open-source software for mathematics, science, and engineering. The SciPy package includes Numpy and Scipy for core scientific functionality, Matplotlib for 2-D plotting, Pandas for data analysis, and Jupyter for a great interactive environment. (**Go for it!**)
As usual, the NEURON web site is a great starting point for learning how to use NEURON within Python: neuron.yale.edu/neuron/static/docs/neuronpython/firststeps.html. Other valuable informations can be found in this paper by Hines et al.: (2009:) www.ncbi.nlm.nih.gov/pmc/articles/PMC2636686.

Modeling a Ventral Tegmental Area dopamine cell

The cell model is based on the morphology of an actual DA VTA cell from a rat. Golgi-Cox-stained tyrosine hydroxylase-positive rat VTA neurons were identified using a Leica confocal microscope and their morphology was reconstructed using the Filament Tracer tool of Bitplane Imaris software (figure 1).

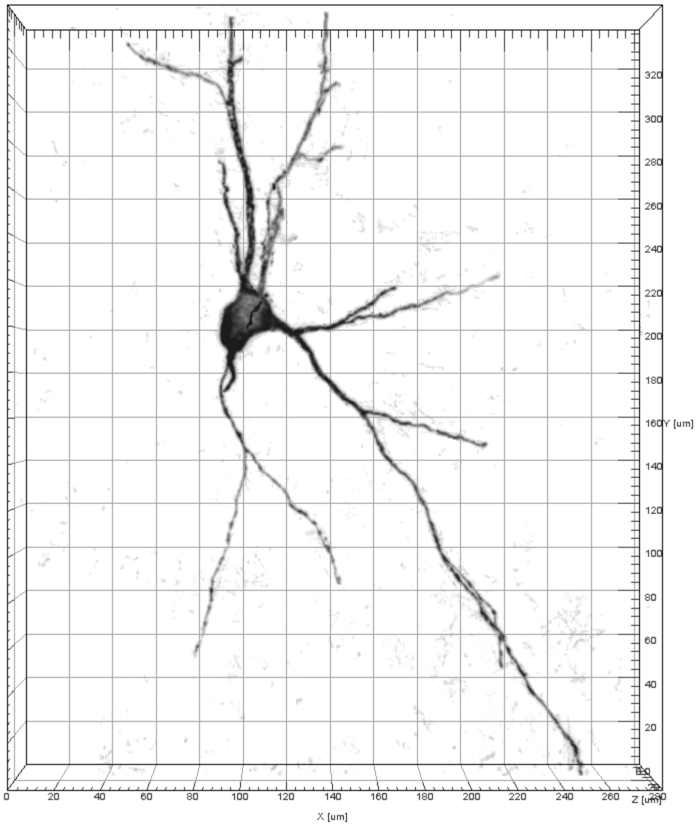
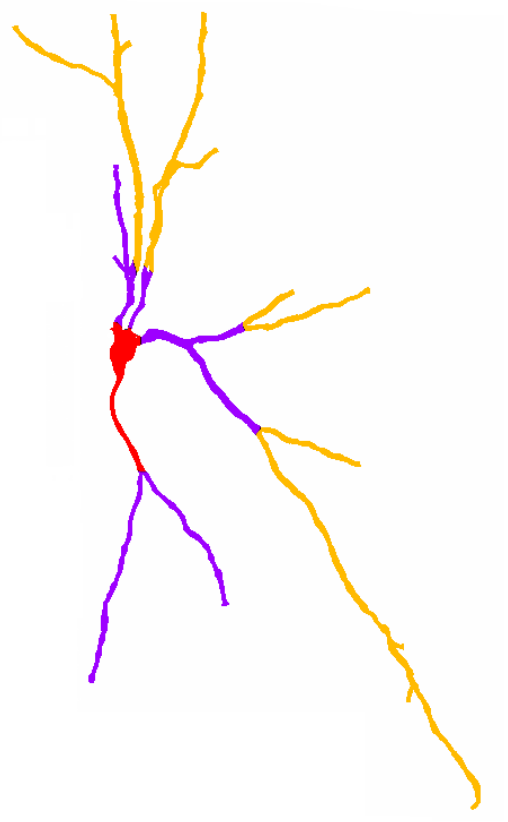


Fig. 1

The morphology is divided into 38 electrical compartments (sections): 1 soma and 27 dendrites (11 proximal and 16 distal); sections are further split into 89 membrane segments using d_Lambda rule, to ensure that each section is smaller than 0.1 λ, the space constant of this morphology.
Spatially distributed ion channels consist of the following families: fast Na^+ current, delayed rectifying K^+ channel, transient outward K^+ current, and Na/K pump. Ca^{2+} dynamics voltage- activated T-, N-, and L-type Ca^{2+} currents, and a SK-type K^+ current are added to the soma section (figure 2).



Soma (1 section)

- Cm 1 μF/cm^2, Ra 400 MΩ/cm
- fast Na$^+$ current, delayed rectifying K$^+$ channel, transient outward K$^+$ current, Na/K pump
- Ca^{2+} dynamics voltage-activated T-, N-, and L-type Ca^{2+} currents, and SK-type K$^+$ current

Proximal dendrites (11 sections)

- Cm 1 μF/cm^2, Ra 400 MΩ/cm
- fast Na$^+$ current, delayed rectifying K$^+$ channel, transient outward K$^+$ current, Na/K pump

Distal dendrites (16 section)

- Cm 1 μF/cm^2, Ra 400 MΩ/cm
- fast Na$^+$ current, delayed rectifying K$^+$ channel, transient outward K$^+$ current, Na/K pump

Fig. 2

Clearly, this is just a very simplified view of the neurobiological properties of the VTA DA cell membrane, nevertheless it is sufficient to capture key features of the electrical behaviour of the cell.

Synaptic input

GABA-A input is modeled as a leak current with a reversal potential (Erev) of -70 mV. GABA-A conductance is located on both soma and dendrites, where GABA-A conductance is 1/10 that on the soma (2750 μS/cm^2, soma and 275 μS/cm^2 dendrites).

AMPA and NMDA synapses (Erev = 0 mV) are colocalized on distal dendrites. AMPA and NMDA synapses are activated in such a way to generate a random bursting activity at an average interburst frequency of 1 Hz, average duration of 200 ms, and average intraburst frequency in the range 20–50

Hz, allowing to reproduce a good approximation of the firing patterns experimentally observed in VTA DA neurons (figure 3).

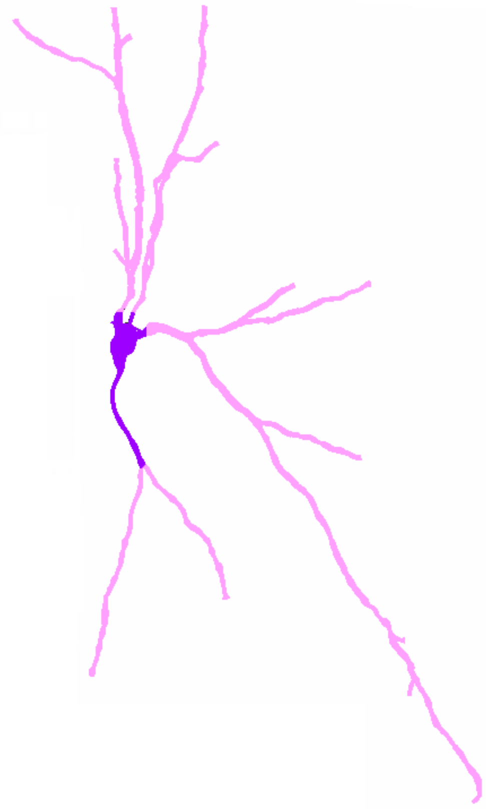
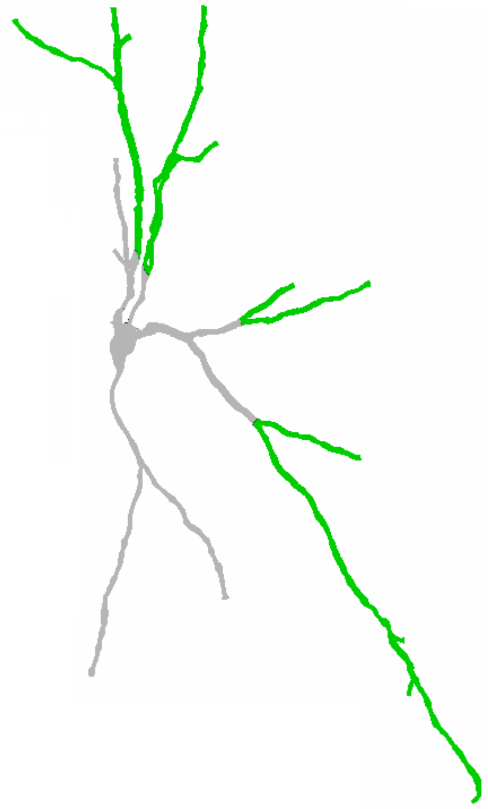
	
GABA-A input Soma: 2750 µS/cm^2 , Rin 27.4 MΩ Proximal: 275 µS/cm^2 , Rin 35.6-121.5 MΩ Distal: 275 µS/cm^2 , Rin 65.8-173.5 MΩ Erev = -70 mV	**GLU** input Distal dendrites: AMPA and NMDA Random (Poisson) activation Erev = 0 mV

Fig. 3

DA release

The amount of DA to be released is calculated by assuming an instantaneous and constant release at each action potential, and a Michaelis–Menten uptake with an affinity constant km = 0.2 µM and a maximal velocity for uptake Vmax of 5 µM/s.

This model has been developed and used in Enrico et al. (2016) Neuroscience 322: 195-207. doi: 10.1016/j.neuroscience.2016.02.006, to study how morphine withdrawal-induced neurochemical and morphological changes could affect the electrophysiological behaviour of VTA DA cells. The model findings to understand how and to what extent a change in the balance of GABA/GLU inputs can take into account the experimentally observed hypofunction of VTA DA neurons during acute and prolonged withdrawal. Further, the model predicted that withdrawal-induced morphological changes (reduction of soma diameter) may play a role in the increased excitability of VTA DA cell to opiate administration observed during opiate withdrawal.

Working with the model.

1- pacemaker firing (*file: VTA_01.hoc*)

DA neurons in the VTA, fire spontaneously in a pacemaker-like manner. This is due to a complex interplay between Na^+, Ca^{2+} and K^+ channels. Tetrodotoxin (Na_V blocker) administration stops spontaneous pacemaking activity and usually results in stable resting potentials near −60 mV to −55 mV.

Run the simulation by opening the file VTA_01.hoc in the BNNI17_VTA folder in NEURON (first remember to compile .mod files in this folder with *nrnivmodl*). This will open several windows: the NEURON main menu, the RunControl, the Model Conductances, a Shape windows, and two graphs: Graph[0] for plotting soma membrane potential, and Graph[1] for plotting DA release. Before launching the simulation, right-click on the shape plot window and select "shape plot" from the menu in order to visualize membrane potential on the cell shape (color-coded, figure 4). Now launch the simulation by clicking on the "Init & Run" button, you'll see the VTA cell in absence of external stimuli firing in a pacemaker-like pattern (figure 5).

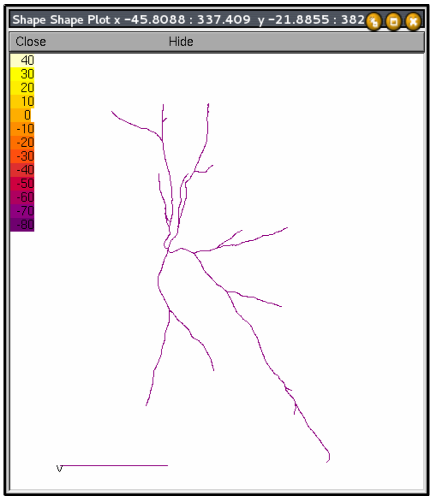

Shape Shape Plot x -45.8088 : 337.409 y -21.8855 : 382

Fig. 4

Exercise: by modifying ionic conductance values in the Model conductances panel, you can stop cell pacemaker-like firing activity. How?

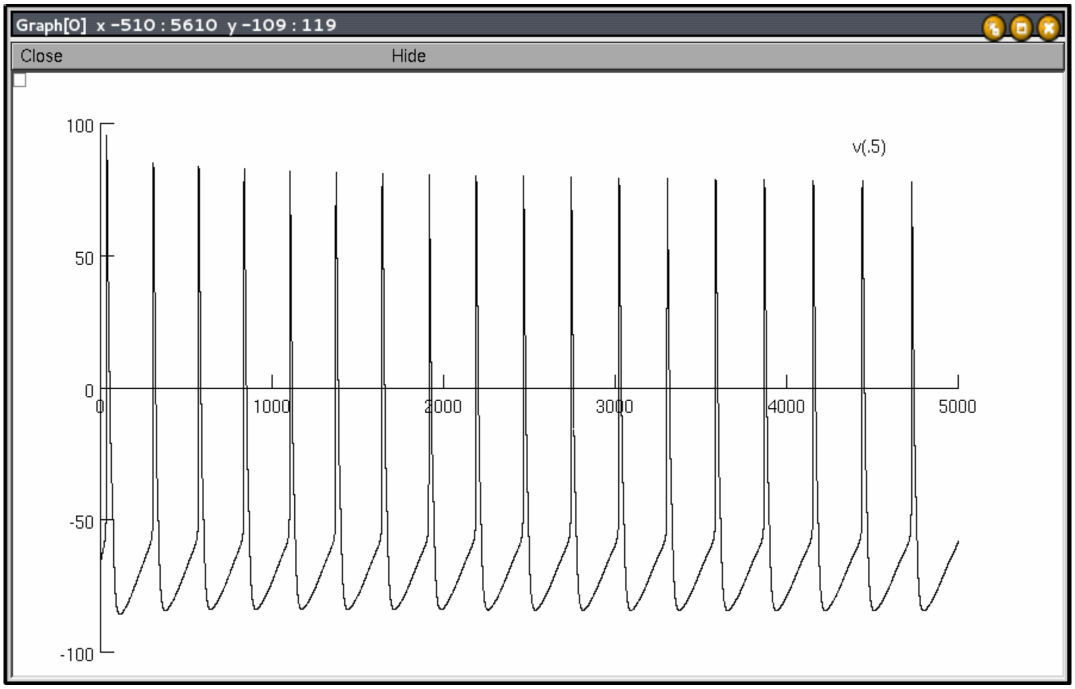
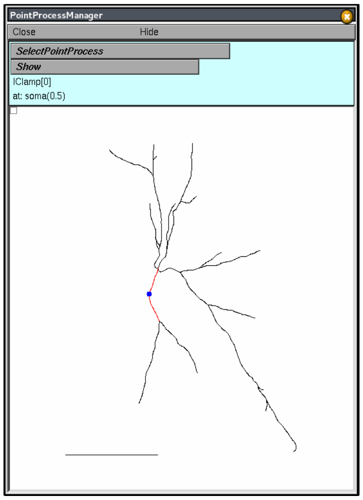


Fig. 5

2- depolarization block (*file: VTA_02.hoc*)

A particular electrophysiological state of DA cell activity is depolarization block (DB) of spike generation. During DB DA cells (both in the substantia nigra and in the VTA) are in a state of overexcitation, resulting in the inactivation of spike firing. Interestingly enough, the efficacy of some antipsychotic drugs acting as DA receptor blockers (such as haloperidol), appears to be correlated with induction of chronic DP block in DA neurons.

Run the simulation by opening the file VTA_02.hoc in the BNNI17_VTA folder in NEURON (first remember to compile .mod files in this folder with *nrnivmodl*). This will open several windows: the NEURON main menu, the RunControl, the PointProcessManager, and two graphs: Graph[0] for plotting soma membrane potential, and Graph[1] for plotting DA release. If you launch the simulation by clicking on the "Init & Run" button, and the VTA cell will start firing in a pacemaker-like manner as in the previous example. Now we insert a current source (*IClamp*) into the soma (now enhanced in red on the cell shape, figure 6): from the PointProcessManager SelectPointProcess menu select IClamp and use the following parameters: delay (*del*) = 2000 ms; duration (*dur*) = 3000



IClamp[0]
at: soma(0.5)

Fig. 6

ms; current (*amp*): 4 nA. Run the simulation again.

Exercise: delete the IClamp located on the soma (just select None on the SelectPointProcess menu) and insert a new one (with the same parameters) on the farther dendrite section (just click on it to select). Run the simulation. How could you explain the electrical behaviour you see?

3 – Synaptic input and burst firing (*file: VTA_03.hoc*)

VTA DA cells activity is under control of a large network. The primary afferents include excitatory inputs from prefrontal cortex and tegmental nuclei; while the inhibitory input includes local interneurons, feedback fibres from striatal nuclei, and the rostromedial tegmental nucleus (figure 7).

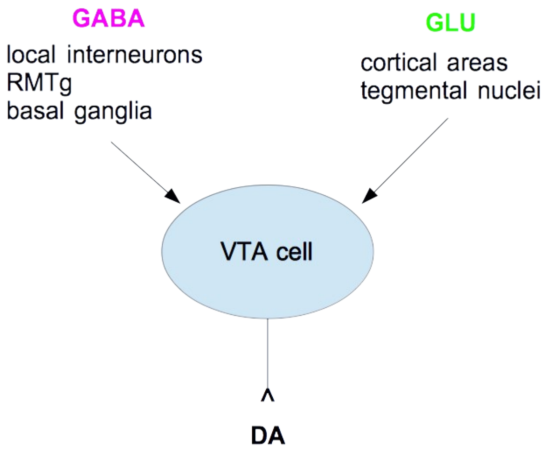
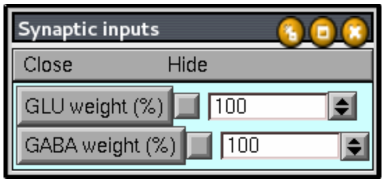


Fig. 7

In response to cogent sensory stimuli, VTA DA cells physiologically fire bursts of activity. This type of cell firing give rise to a supra-additive release of DA in forebrain areas, providing a critical modulatory influence on cognitive and rewarding brain functions. It is known that burst firing in VTA DA neurons is engendered by the activity of GLU cortical afferents via AMPA and NMDA receptors, while GABA input acts .

Run the simulation by opening the file VTA_03.hoc in the BNNI17_VTA folder in NEURON (first remember to compile .mod files in this folder with *nrnivmodl*). This will open several windows: the NEURON main menu, the RunControl, Synaptic input, and two graphs: Graph[0] for plotting soma membrane potential, and Graph[1] for plotting DA release.



GLU weight (%) 100

GABA weight (%) 100

Launch the simulation by clicking on the “Init & Run” button; notice the peculiar firing pattern of VTA DA cells, characterized by short bursts distributed amongst single spikes (figure 8) . At the end of the run, at the hoc prompt (oc>) call the procedure *average()* in order to see a simple analysis of the soma spikes pattern (number of: bursts, single spikes, soma action potentials, and the mean number of spikes/burst).

Exercise: Try to explore the effects of diverse GABA/GLU input values on model firing pattern by running appropriate simulations and comparing the results.

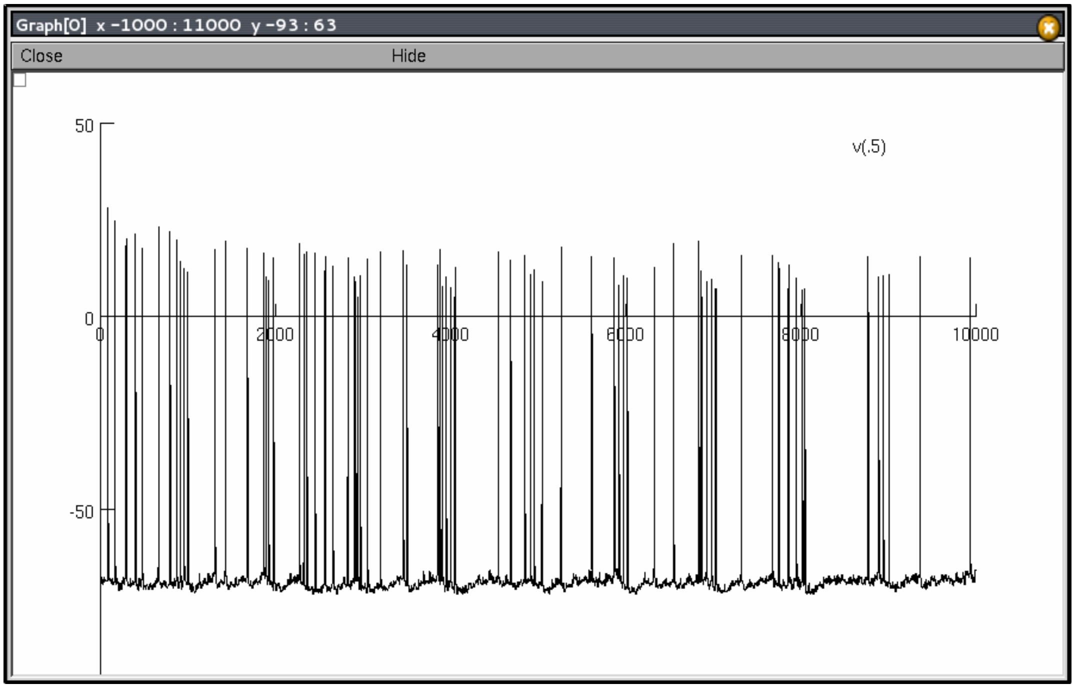


Fig. 8

5 – Opioids effect (*file: VTA_03.hoc*)

Opioids administration inhibit GABA neurons within the VTA and in the rostromedial tegmental nucleus. Since GLU excitatory input appears not affected by opioids administration and opioid receptors are not represented on VTA DA cell, morphine indirectly increases VTA DA cell firing. Recent *in vitro* data show that morphine administration reduces GABA-A-mediated inhibitory postsynaptic currents on VTA DA cells to 68.4 ± 4.4% of control in naive subjects, and to 53.9 ± 3.0% of control in morphine tolerant subjects.

Run the simulation by opening the file VTA_03.hoc in the BNNI17_VTA folder in NEURON (first remember to compile .mod files in this folder with *nrnivmodl*). This will open several windows: the NEURON main menu, the RunControl, Synaptic input, and two graphs: Graph[0] for plotting soma membrane potential, and Graph[1] for plotting DA release. At the end of the run, at the hoc prompt (oc>) call the procedure *average()* in order to see a simple analysis of the soma spikes pattern (number of: bursts, single spikes, soma action potentials, and the mean number of spikes/burst).

Exercise: Simulate the effect of morphine administration in naïve and tolerant subjects and evaluate the differences in the response.

6 – Morphine withdrawal (*file: VTA_04.hoc*)

During opioids withdrawal the mesolimbic DA system is hypofunctional, with reduced activity of VTA DA cells and low DA levels in projection fields. This hypodopaminergic state is regarded as one of the main causes that triggers relapse to drug use, even after prolonged abstinence. The neurobiological mechanisms underlying this phenomenon are still unclear, although it has been commonly related to a dysfunctional potentiation of the GABA input. In our previous studies with this model, we found that GABA increase is not enough and that the *in vivo* effects of morphine withdrawal may be reproduced by increasing GABA-A input by 30% as well as decreasing GLU input by 30%, in presence of withdrawal-induced morphofunctional alterations (figure 9).

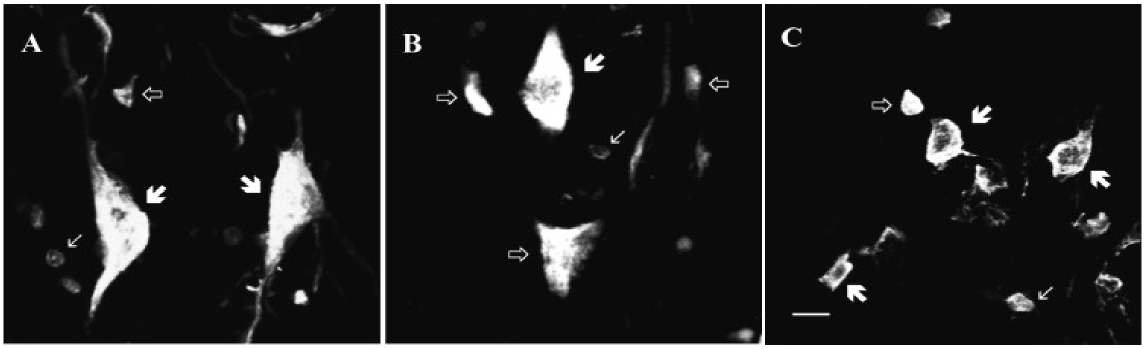


Fig. 9 *Control* *Chronic morphine* *Withdrawal*

Run the simulation by opening the file VTA_04.hoc in the BNNI17_VTA folder in NEURON (first remember to compile .mod files in this folder with *nrnivmodl*). This will open several windows: the NEURON main menu, the RunControl, Synaptic input, and two graphs: Graph[0] for plotting soma membrane potential, and Graph[1] for plotting DA release. Launch the simulation by clicking on the “Init & Run” button; notice the profound reduction in the electrophysiological activity of VTA DA cells and in DA release (figure 10). At the end of the run, at the hoc prompt (*oc*>) call the procedure *average()* in order to see a simple analysis of the soma spikes pattern (number of: bursts, single spikes, soma action potentials, and the mean number of spikes/burst).

Exercise: Simulate the effect of morphine administration (using the appropriate values for both naïve and tolerant subjects) while the cell is in withdrawal conditions, evaluate the responses and compare the results with the effects of morphine obtained in base conditions (previous exercise).

Exercise: By using the previous model file (*file: VTA_03.hoc*) try to understand why the increase of GABA by itself cannot fully account for the firing pattern observed during withdrawal conditions.

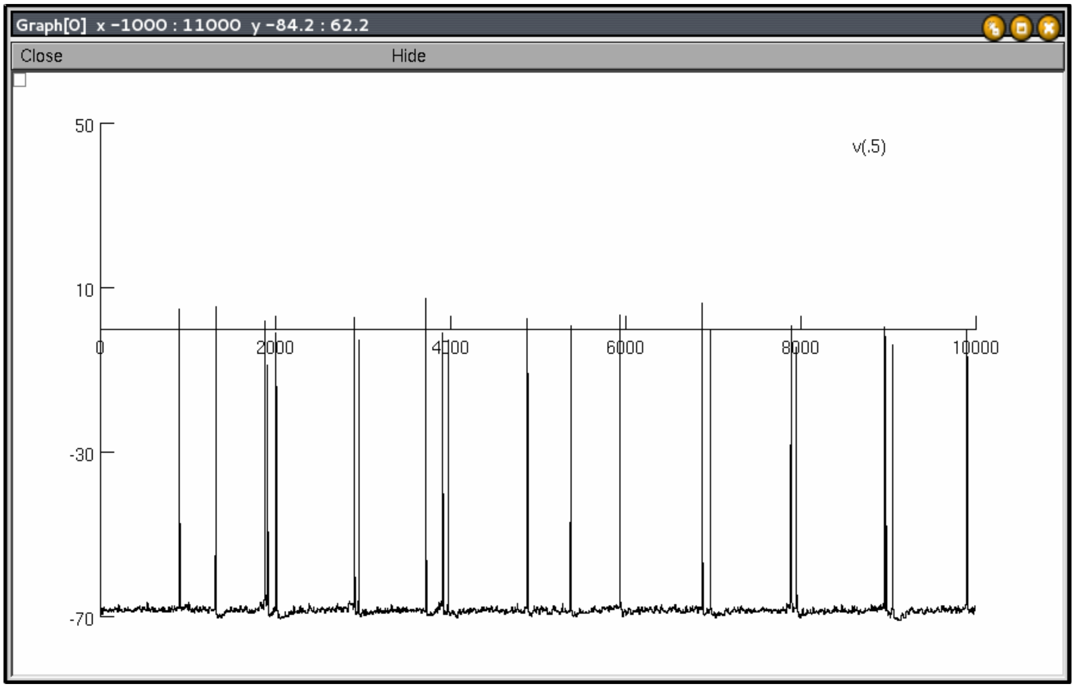


Fig. 10

One last word

Hope you enjoyed experimenting with our VTA DA cell model. Of course there are many other facets of the VTA DA cell behaviour deserving attentions, and that our model may help study.
If you want use the model for your research and update it or improve it you're absolutely welcome. In my opinion, the development of a neural models is not a linear process; actually I think it is more a spiral-shaped path (figure 11). You start from a basic structure and while you progress detailing its complexity and comparing your model results with the experimental data, quite often you (or other researchers) have to go back and reconsider all the previous work in order to obtain an useful model.
If you improve the model, please let me know!

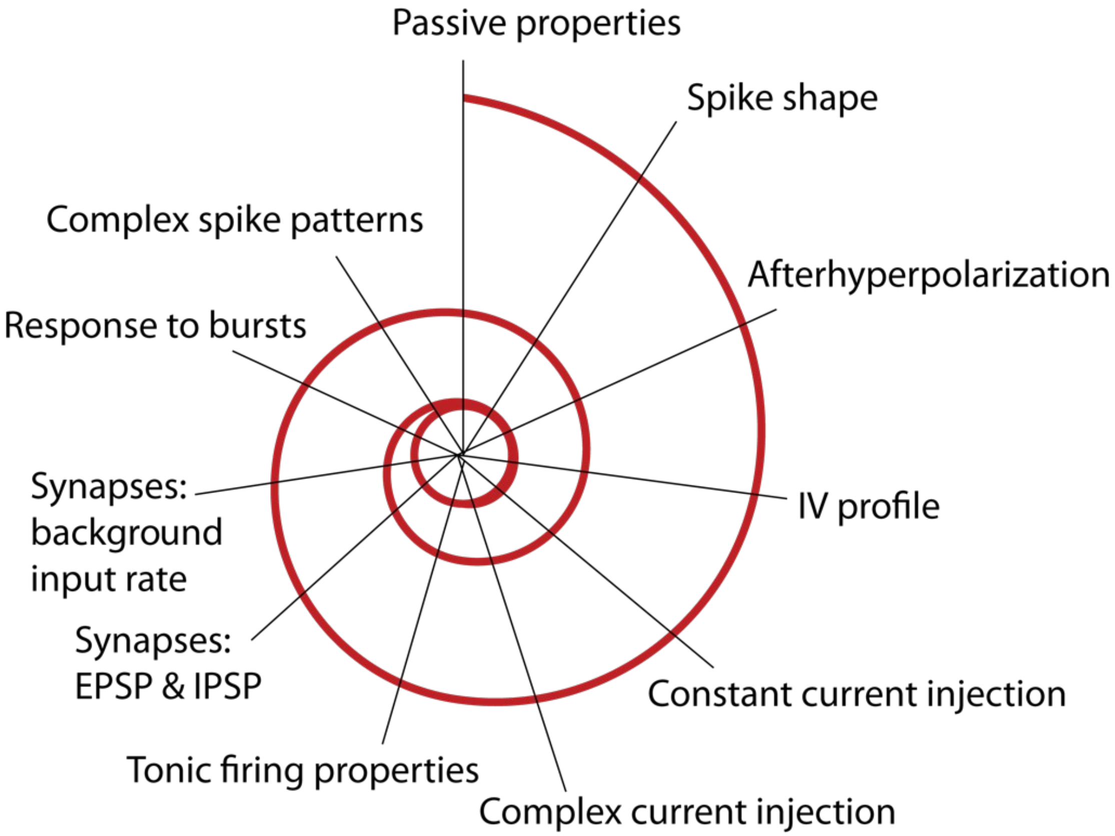


Figure 11 (courtesy of Dr. Sergio Solinas)