



Dopaminergic and Cholinergic Modulation of Large Scale Networks *in silico* Using *Snudda*

Johanna Frost Nylen^{1*}, Jarl Jacob Johannes Hjorth², Sten Grillner¹ and Jeanette Hellgren Kotaleski^{2*}

¹ Department of Neuroscience, Karolinska Institute, Stockholm, Sweden, ² Science for Life Laboratory, School of Electrical Engineering and Computer Science, Royal Institute of Technology, Stockholm, Sweden

Neuromodulation is present throughout the nervous system and serves a critical role for circuit function and dynamics. The computational investigations of neuromodulation in large scale networks require supportive software platforms. *Snudda* is a software for the creation and simulation of large scale networks of detailed microcircuits consisting of multicompartmental neuron models. We have developed an extension to *Snudda* to incorporate neuromodulation in large scale simulations. The extended *Snudda* framework implements neuromodulation at the level of single cells incorporated into large-scale microcircuits. We also developed *Neuromodcell*, a software for optimizing neuromodulation in detailed multicompartmental neuron models. The software adds parameters within the models modulating the conductances of ion channels and ionotropic receptors. Both application of neuromodulators is simulated and models which reproduce the experimentally measured effects are selected. In *Snudda*, we developed an extension to accommodate large scale simulations of neuromodulation. The simulator has two modes of simulation – denoted *replay* and *adaptive*. In the *replay* mode, transient levels of neuromodulators can be defined as a time-varying function which modulates the receptors and ion channels within the network in a cell-type specific manner. In the *adaptive* mode, spiking neuromodulatory neurons are connected via integrative modulating mechanisms to ion channels and receptors. Both modes of simulating neuromodulation allow for simultaneous modulation by several neuromodulators that can interact dynamically with each other. Here, we used the *Neuromodcell* software to simulate dopaminergic and muscarinic modulation of neurons from the striatum. We also demonstrate how to simulate different neuromodulatory states with dopamine and acetylcholine using *Snudda*. All software is freely available on Github, including tutorials on *Neuromodcell* and *Snudda-neuromodulation*.

OPEN ACCESS

Edited by:

Srikanth Ramaswamy,
Newcastle University, United Kingdom

Reviewed by:

Christiane Linster,
Cornell University, United States
Wen-Jie Song,
Kumamoto University Hospital, Japan

*Correspondence:

Johanna Frost Nylen
johanna.frost.nylen@ki.se
Jeanette Hellgren Kotaleski
jeanette@csc.kth.se

Received: 28 July 2021

Accepted: 16 September 2021

Published: 21 October 2021

Citation:

Frost Nylen J, Hjorth JJJ,
Grillner S and Hellgren Kotaleski J
(2021) Dopaminergic and Cholinergic
Modulation of Large Scale Networks
in silico Using *Snudda*.
Front. Neural Circuits 15:748989.
doi: 10.3389/fncir.2021.748989

Keywords: neuromodulation, simulation – computers, microcircuit, dopamine, acetylcholine, striatum

INTRODUCTION

The nervous system depends on fast interaction via ionotropic receptors, which are activated by a variety of neurotransmitters including glutamate, GABA, and glycine. Besides these ionotropic receptors, neuromodulation via metabotropic receptors have a profound effect on network dynamics via slower processes. On a single cell level, these neuromodulators act via a variety

of receptor subtypes, which modulate the excitability of neurons and influence their synaptic properties (Nadim and Bucher, 2014). The interactions between neuromodulators and their targets are complex. For example, a single ion channel type can be modulated by several different neuromodulators (Swensen and Marder, 2000; Park and Spruston, 2012). While a single neuromodulator can affect multiple ion channel types and signaling pathways within neurons (Greengard, 2001). On a circuit level, neuromodulation of neurons and synapses can massively alter the network activity. A challenge is to bridge the levels between single neurons and the circuit to understand the role of neuromodulators in shaping network behavior (Nadim and Bucher, 2014).

There are models of both invertebrate and different types of vertebrate which attempt to bridge the gap from cellular and synaptic properties to circuit function, and where the effects of neuromodulation is considered (Lansner et al., 1998; Kozlov et al., 2001, 2009; Hamood and Marder, 2014; Nadim and Bucher, 2014; Colangelo et al., 2019). For neuromodulation, the challenge is that the effect of a single neuromodulator in a single cell setting could be very different once that cell is embedded within a circuit. In the circuit, other components, such as other neurons and synapses, would also be under neuromodulatory control and hence contribute to more complex circuit interactions (Oh et al., 2012). Therefore, computational models have to incorporate the ability to bridge these levels. Furthermore, it has been shown that different parts of a neuron (dendrites/soma) can be modulated differently depending on receptor sub-type and target (Bender et al., 2010). Such compartmental differences can be implemented in multicompartmental models where reconstructed neuronal morphologies are used and ion channel models are distributed throughout the morphology (Frost Nylén et al., 2020; Hjorth et al., 2020; Lindroos and Hellgren Kotaleski, 2021).

We use the *in silico* striatal microcircuit (Hjorth et al., 2020) as a demonstrative example, which consists of 95% striatal projection neurons (SPNs) and 5% interneurons (fast-spiking (FS), low-threshold spiking and cholinergic interneurons). Here, we simulate neuromodulation of SPNs and FS. The striatum is the input stage of the basal ganglia, a group of subcortical nuclei involved in action-selection, motor control, and habitual learning. Neuromodulation is important in the striatum and especially dopamine is essential for basal ganglia function (Gerfen and Surmeier, 2011; Surmeier et al., 2014; Da Silva et al., 2018). The cholinergic interneuron (ChIN) is a spontaneously active interneuron within the striatum, which releases acetylcholine (ACh). Hence, in addition to the dopaminergic modulation, cholinergic modulation can modulate several components of the striatal network, which has extensive expression of muscarinic receptors (Abudukeyoumu et al., 2019). Furthermore, several studies have demonstrated the complex interactions between DA and ACh within the striatum, ranging from modulating dopamine release via presynaptic nicotinic receptors to direct dopaminergic modulation of ChINs via D2 receptors (Threlfell et al., 2012; Howe et al., 2019).

Previously, *Snudda*, a Python package for creating data-driven networks of neurons, placing synapses using touch detection between axons and dendrites and setting up large scale

simulations was developed by Hjorth et al. (2021). The software included fast synaptic transmission but neuromodulation was limited to dopamine. Hence, the *Snudda* package required further development to accommodate the generalized implementation of neuromodulation, together with all the other functionalities contained within *Snudda*. Following the development of *Snudda.neuromodulation*, it was necessary to create a software for generating and selecting parameter sets which reproduce neuromodulation. Therefore, we decided to develop *Neuromodcell*, which extracts the modulatory parameters necessary to reproduce neuromodulation on a single cell level for multicompartmental neuron models (Figure 1).

In this study, we applied *Neuromodcell* and *Snudda.neuromodulation* to the dopaminergic and cholinergic modulation of the *in silico* striatal microcircuit to demonstrate the functionalities included in these software packages. In general, *Neuromodcell* and *Snudda* can be used to generate and simulate networks of multicompartmental models with a wide range of neuromodulators. Within these simulations, a variety of questions could be addressed. Hence, not merely to simulate neuromodulation, but to predict which components are important for certain features and also redundancies within the network. Furthermore, simulations of multiple neuromodulators could predict how these interact on a network level. Both *Neuromodcell* and *Snudda* are open-source and freely available on Github, for further enhancement and expansion.

MATERIALS AND METHODS

Software Setup

Neuromodcell is freely available to be downloaded from its Github site¹, written in Python with requirements specified in *requirements.txt* including NEURON² and Jupyter Notebook. To install *Neuromodcell* use `pip install neuromodcell`. The software is compatible with Linux operating system and super-computer clusters (Cray XC40 system). The simulations of large networks use *Snudda* (Hjorth et al., 2020, 2021), a Python package for creating data-driven networks of neurons available from its Github site³. To simulate neuromodulation, within the *Snudda* framework, additional simulation classes and associated neuromodulation subpackages were created, *snudda.neuromodulation*. To use *Snudda*, install via `pip install snudda` and follow the instructions on its wiki page (see text footnote 3).

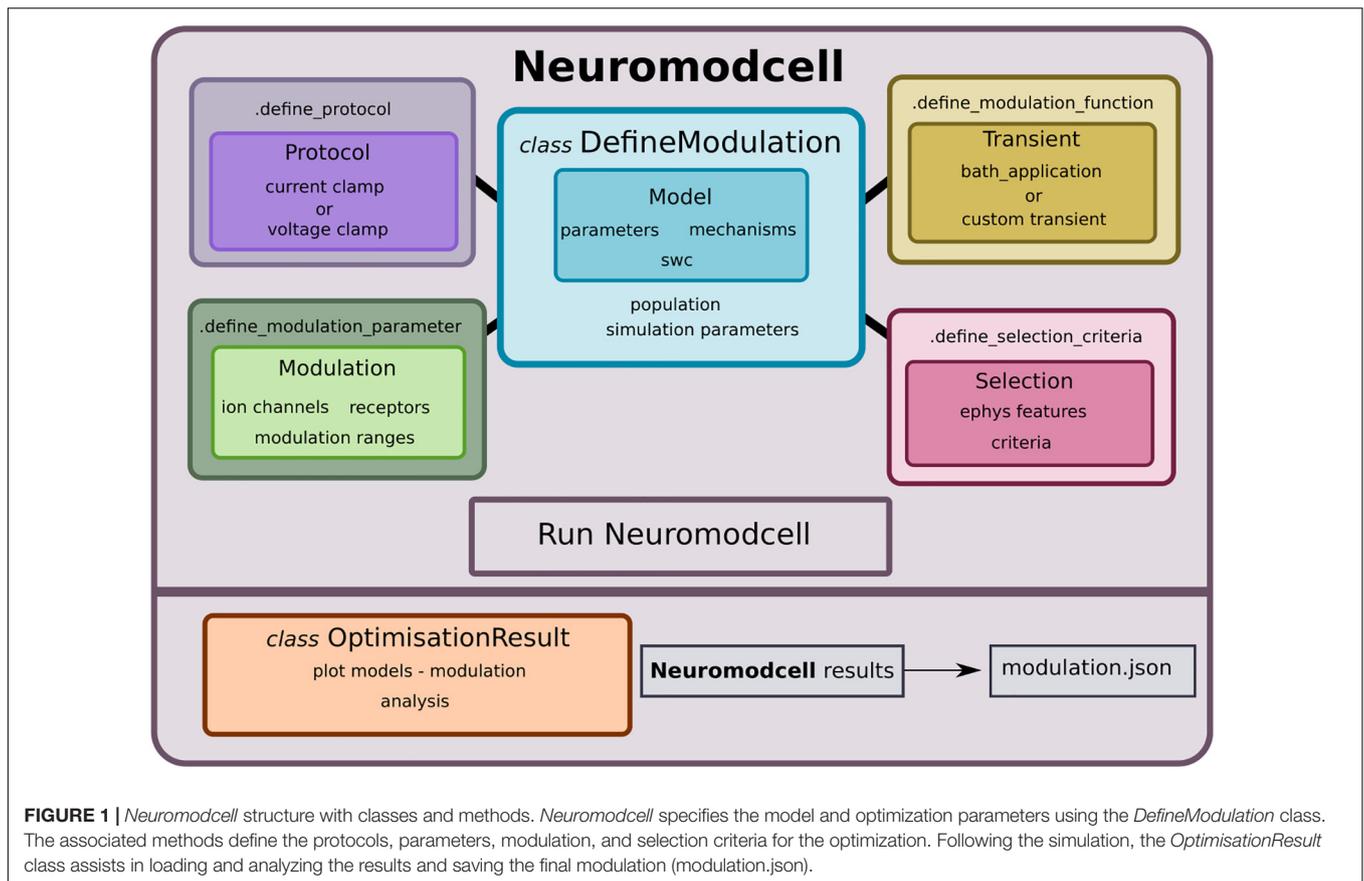
Multicompartmental Models

Neuromodcell and *Snudda* require multicompartmental models specified for NEURON simulator described by three files: a morphology file (SWC), a parameter file (JSON) and a mechanism file (JSON) including a directory with the ion channel model files (.mod files, NEURON model description language). The models used in this publication were optimized

¹github.com/jofronny/Neuromodcell

²github.com/neuronsimulator/nrn

³github.com/Hjorthmedh/Snudda



using BluePyOpt (Van Geit et al., 2016) in previous publications (Hjorth et al., 2020; Lindroos and Hellgren Kotaleski, 2021) with modification for muscarinic modulation; and include direct and indirect striatal projection neuron (dSPN and iSPN) and fast-spiking interneuron (FS) models from the striatum.

Parameterization

To modulate the models during the simulation, we introduce additional parameters for the specific neuromodulator within the .mod files. The modulation is implemented in a phenomenological manner, where a modulation parameter is multiplied with the target (for example the conductance of an ion channel). Hence, the conductance (in this case) can be regulated through the simulation. For example, for dopamine, the three parameters are maxModDA, modDA and levelDA (using the key, DA to indicate dopamine), based on previous formalism developed in Hjorth et al. (2020). In general, the convention is 'maxMod*', 'level*' and 'mod*' and the * should be replaced with a specific name for the neuromodulator. The maxMod* defines the modulation degree (which can vary, with 0.6 being 40% reduction). The level* defines the transient level of modulation throughout the simulation. The mod* defines the activation of modulation (0 inactive, 1 fully active). Within the examples, dopaminergic and muscarinic modulation was included (with muscarinic modulation using key *ACh*). The list of modulated ion channels for each cell type is presented in **Supplementary**

Table 1. For examples of .mod-files with these modulation parameters (see text footnote 1).

Neuromodcell

Neuromodcell is a software for optimizing and simulating neuromodulation in detailed multicompartmental neuron models. The software loads and simulates a population of models with user-defined parameter variations of maxMod*, which is how the program induces the changes associated with neuromodulation. The software also incorporates current clamp and voltage clamp parameters to simulate specific biological experiments. Hence, the user can define the simulated experiment in terms of both neuromodulator and the specific protocol to be used. Following the simulation, the parameter sets that reproduce the experimental data are selected based on a user-specified selection criteria. We provide several electrophysiological features by default, but custom features can be included.

Here, *Neuromodcell* was used to optimize dopaminergic modulation for dSPN, iSPN and FS, as well as muscarinic modulation for dSPN and iSPN. The optimizations for dSPN and iSPN were validated against data from Planert et al. (2013) and Lahiri and Bevan (2020) (see **Supplementary Figure 1**) for dopaminergic modulation and Lv et al. (2017) for muscarinic modulation. The dopaminergic modulation for FS was validated against data from Bracci et al. (2002).

Neuromodulation in *Snudda*

Neuromodulation in *Snudda* is implemented as a separate simulation module called *snudda.neuromodulation*. Within the module, there are two simulation classes for neuromodulation, *SnuddaSimulateNeuromodulation* and *SnuddaSimulateNeuromodulationSynapse*. These classes represent the two strategies used to simulate neuromodulation, *replay* and *adaptive*. The *replay* mode plays a predetermined transient through the *level**-parameters and hence modulates each component of the circuit in a predefined time dependent manner. On the other hand, the *adaptive* mode originates from the fact that a neuromodulatory neuron integrated within the microcircuit could not modulate via a predetermined transient. Instead, the spiking activity of such a neuron would have to be continuously translated into an instantaneous level of neuromodulation. Therefore, an alternative approach was developed where an intermediate mechanism would integrate the spiking activity. *Adaptive* simulations couple one or several presynaptic neurons to postsynaptic neurons via an integrating mechanism called *conc**, which modifies the neuromodulatory parameters (*level**) in the circuit (for example *concDA* to *levelDA* for dopamine). Hence, neuromodulation within the simulation can either follow a fixed recipe ('replay' mode), or dynamically change based on the activity in the network ('adaptive' mode). The mode used within a particular simulation will therefore depend on the network structure. Using either approach, the level of neuromodulation would be updated every time step via the *level**-parameter. The *replay* mode loads an array which has set values for the *level** for each time step of the simulation. This would suffice if the neuromodulation is synchronous and the neuromodulatory neurons are not affected by activity of the circuit. On the other hand, if the activity of the microcircuit could for example inhibit the neuromodulatory neurons, like striosomal dSPNs in the striatum, *replay* mode does not suffice and *adaptive* mode would be used instead, as the presynaptic neuromodulatory neurons can interact with the microcircuit throughout the simulation.

For *replay*, the *maxMod**, *mod** and *level** are introduced as *RANGE* variables. In the *adaptive* mode, which is based on pointers, the parameters are introduced with the keyword *POINTERS*⁴ and 'ptr' is added to the filename (i.e., *na.mod* to *na_ptr.mod*).

Lastly, both types of simulations require a configuration file (JSON) which is created using either *Neuromodulation* class (*snudda.neuromodulation.modulation_network*, for *replay* simulations) or *NeuromodulationSynapse* (*snudda.neuromodulation.modulation_synapse*, for *adaptive* simulations). For examples of how to use these classes (see text footnote 4).

To demonstrate the features of *Snudda.neuromodulation*, we simulated several networks using both *replay* and *adaptive* mode. Firstly, using the *Snudda* framework, a network of 10 000 neurons was created and simulated with dopamine and acetylcholine transients; and with a cortical activation at 1 s, with cortical and thalamic background activity. The network is based on previous

publications (Hjorth et al., 2020) and the connection probabilities follows the diagram in **Supplementary Figure 2**. The transients consisted of a tonic level of ACh accompanied with a burst or pause of ACh and a DA burst. The simulations were performed on a super-computer cluster (Cray XC40 system at PDC Center for High Performance Computing, KTH). A smaller network with dSPNs was simulated to show the effect of DA and ACh without network interactions from iSPNs and FS. Furthermore, several dopamine transients with different start times were created using the *Neuromodulation* class and applied to a network of 20 dSPNs with a cortical activation lasting for 500 ms. Lastly, a network of dSPNs was simulated with presynaptic spiking neurons using the *adaptive* mode. The presynaptic neurons represented dopaminergic neurons and were simulated with and without activation (i.e., bursting). The activity of the neurons within the simulations were measured by the percentage of spiking neurons and the mean firing frequency during the cortical activations using custom Python code and Electrophysiology Analysis Toolkit⁵.

Tutorials

A tutorial on *Neuromodcell* and an example of dopaminergic modulation of dSPN are available (see text footnote 1).

Examples of simulations using *Snudda.neuromodulation* module are available (see text footnote 4).

RESULTS

Overview

Neuromodcell performs parameter variation of ion channels and receptor models within multicompartmental models to simulate neuromodulation. The components of the software are presented in **Figure 1** with the main class, *DefineModulation* (**Figure 2**) and its associated class methods, which are used to define the parameters, protocols, selection criteria and modulation for the specific cell and neuromodulator. The user specifies the range of parameter variation for each ion channel and/or receptor as well as the experimental protocol to be replicated in the simulation (**Figures 3, 4**). The models with parameter variations are simulated using the *optimise* module, which uses *mpi4py* for parallelization.

Following the simulation, a selection process is performed, where the models are compared to the experimental data provided in the setup (using *DefineModulation*). The parameter variations of the models which pass validation are saved in a separate file. The code for defining the selection criteria is freely available (module *selection_criteria* and *selection_function*) and can be modified to measure a specific electrophysiological feature of the simulation, for example firing frequency or the number of action potentials.

Below, we describe the process of setting up, optimizing and selecting the models that pass validation. Following the completion of the optimization, the results are transferred to

⁴github.com/jofronny/Neuromodulation-software

⁵github.com/NeuralEnsemble/elephant

```

from neuromodcell.modulation_set import DefineModulation

model_folder = 'path_to_model'
output_dir = 'dspn'
parameter_id = 0 # several model variations for each morphology

modset = DefineModulation(parameterID = parameter_id,
                           cell_name = 'dSPN',
                           cell_dir = model_folder,
                           output_dir = output_dir,
                           population = 1000,
                           tstop = 1000,
                           time_step = 0.025)

modset.define_modulation(modulationDA = 'dopamine')

```

FIGURE 2 | Code for setting up the parameters for neuromodulation using *Neuromodcell*. The parameter, “path_to_model,” defines the path to the model and which specific parameterID, *parameter_id*. *mod_set.define_neuromodulation*, the naming defines the neuromodulation key, DA for dopamine. The example has a simulation time of 1000 ms and population of 1000 different models.

```

modset.define_modulation_parameter('naf','dopamine','somatic', [0.6,0.8])
modset.define_modulation_parameter('naf','dopamine','basal', [0.6,0.8])

modset.define_modulation_parameter('kas','dopamine','somatic', [0.65,0.85])
modset.define_modulation_parameter('kas','dopamine','basal', [0.65,0.85])

modset.define_modulation_parameter('kaf','dopamine','somatic', [0.75,0.85])
modset.define_modulation_parameter('kaf','dopamine','basal', [0.75,0.85])

modset.define_modulation_parameter('kir','dopamine','somatic', [0.85,1.25])
modset.define_modulation_parameter('kir','dopamine','basal', [0.85,1.25])

modset.define_modulation_parameter('cal12','dopamine','somatic', [1,2])
modset.define_modulation_parameter('cal12','dopamine','basal', [1,2])

modset.define_modulation_parameter('cal13','dopamine','somatic', [1,2])
modset.define_modulation_parameter('cal13','dopamine','basal', [1,2])

modset.define_modulation_parameter('can','dopamine','somatic', [0.2,1])

```

FIGURE 3 | Code for setting up the parameter variation for neuromodulation. *modset.define_modulation_parameter* accepts the ion channel name, neuromodulator, part of the morphology and the intervals of modulation. Here we have used data from previous publications (Lindroos et al., 2018; Lindroos and Hellgren Kotaleski, 2021).

Snudda and simulated using *snudda.neuromodulation* module. A step-by-step tutorial is available (see text footnote 4).

Defining Neuromodulation *in silico*

We performed optimizations of dopaminergic and muscarinic modulation for dSPN, iSPN, and FS. In the following examples, the dopaminergic optimization of dSPNs is used to demonstrate the features of *Neuromodcell*. We instantiated the *DefineModulation* class as shown in **Figure 2**. The class requires several arguments to define the optimization, including the

```

modset.define_modulation_function(modulation_function = 'bath_application',
                                 gmax = 1)

modset.define_protocol(typeEx = 'current_clamp',
                       start = 0,
                       duration = 1000,
                       amp = 0.08)

modset.define_protocol(typeEx = 'current_clamp',
                       start = 200,
                       duration = 500,
                       amp = 0.416)

modset.define_selection_criteria(function = "number_AP_increase",
                                mean= 6.44, std = 0.89, threshold = 1.5,
                                tstart = 200, tstop = 700)

modset.save_modulation_setup()

```

FIGURE 4 | Code for setting up the selection criteria. Dopamine level is simulated as a bath application. Two simulated current clamps are added to the set up to replicate the experiment from Planert et al. (2013). The effect of dopamine is measured by the number of action potentials and compared with experimental data.

model directory and the number of model variations to simulate (population). The next step is parameter variation. In **Figure 3**, the ion channels which are modulated by DA within dSPN (based on previous publications Hjorth et al., 2020 and Lindroos et al., 2018) were set using the *define_modulation_parameter* method within the class. The modulation can also be defined to be somatic, dendritic (basal or apical, due to the SWC naming convention) or axonal.

Experimental Setup and Selection Criteria

The experimental data used in the example optimization was taken from Figure 6C in Planert et al. (2013) for dSPNs. The experiment used patch clamp recordings and bath application of dopamine to measure the effect of DA on dSPNs. Following the application of DA, there was an increase in the intrinsic excitability of dSPNs measured by the number of action potentials. Hence, an experimental setup was introduced into *modset*, with a current clamp protocol and bath application of dopamine (**Figure 4**). Furthermore, the method *define_selection_criteria* was used to introduce the measurement used in the validation step (i.e., the change in the number of action potentials). Both the *define_modulation_function* and *define_selection_criteria* can select several pre-defined functions but they can also be customized to accommodate specific transients or criteria, respectively.

Optimization

The optimization uses the *OptimiseModulation*, from *optimise* module in *Neuromodcell*. The optimization for dSPN used a custom Python script called *optimise_dspn.py*, which loaded the *path_to_model* and the specific *seed* for the optimization (**Figure 5**). The seed is used to randomize the modulation parameters defined using *define_modulation_parameter*. The modulation parameters sets (i.e., population) are simulated in

parallel with the specific model (defined in *DefineModulation*), as illustrated in **Figure 1**.

Analysis

The result of the optimization was visualized using *dSPNanalysis* (inheritance from *OptimiseAnalysis* class in *Neuromodcell*) (**Figure 6**). The class loads the output files following the optimization and enables the user to plot and/or analyze the results. In **Figure 7**, the result of dopaminergic modulation of dSPN is shown with the model variations (modulated) and control simulation (in black). The dopaminergic modulation increases the intrinsic excitability of dSPNs via D1 receptors. The result of the *Neuromodcell* optimization produced model variations which reproduce the experimental data from Planert et al. (2013). The control simulation (in black, **Figure 7A**) has four spikes following current injection. The simulated dopamine modulation increases the number of spikes and the change in the number of action potentials are within the range extracted from the experimental data (**Figure 7B**).

Transfer Files to Snudda Simulation

Following the optimization, the parameter sets which reproduce the neuromodulation are saved as *modulation.json* and moved to the model directory (i.e., the model is now defined by *parameters.json*, *morphology.swc*, *mechanisms.json* and *modulation.json*). The modified *.mod* files should also be transferred to a common *Snudda* mechanism directory.

Dopamine

Using the *replay* mode in *Snudda.neuromodulation*, we simulated a network of 10 000 (**Figure 8A**) multicompartmental neuron

```
$ nrnivmodl mechanisms-dspn/
$ parameter_id_model = 'path_to_setup_file'
$ seed = 12345
$ mpiexec -n 4 python optimise_dspn.py $parameter_id_model $seed
```

FIGURE 5 | Code for running the optimization.

```
from dspn_analysis import dSPNanalysis
opt_result = dSPNanalysis('dspn')
opt_result.load()
parameter_id = optR.modulation_setup['parameterID']
opt_result.plot_control(title='D1 modulation dSPN',
                        save=True,filename='control.pdf')
opt_result.plot_all_traces(title='D1 modulation dSPN',
                           save=True,filename='all_traces.pdf')
opt_result.plot_validated_traces_with_control(title='D1 modulation dSPN',
                                             save=True,filename='validated.pdf')
```

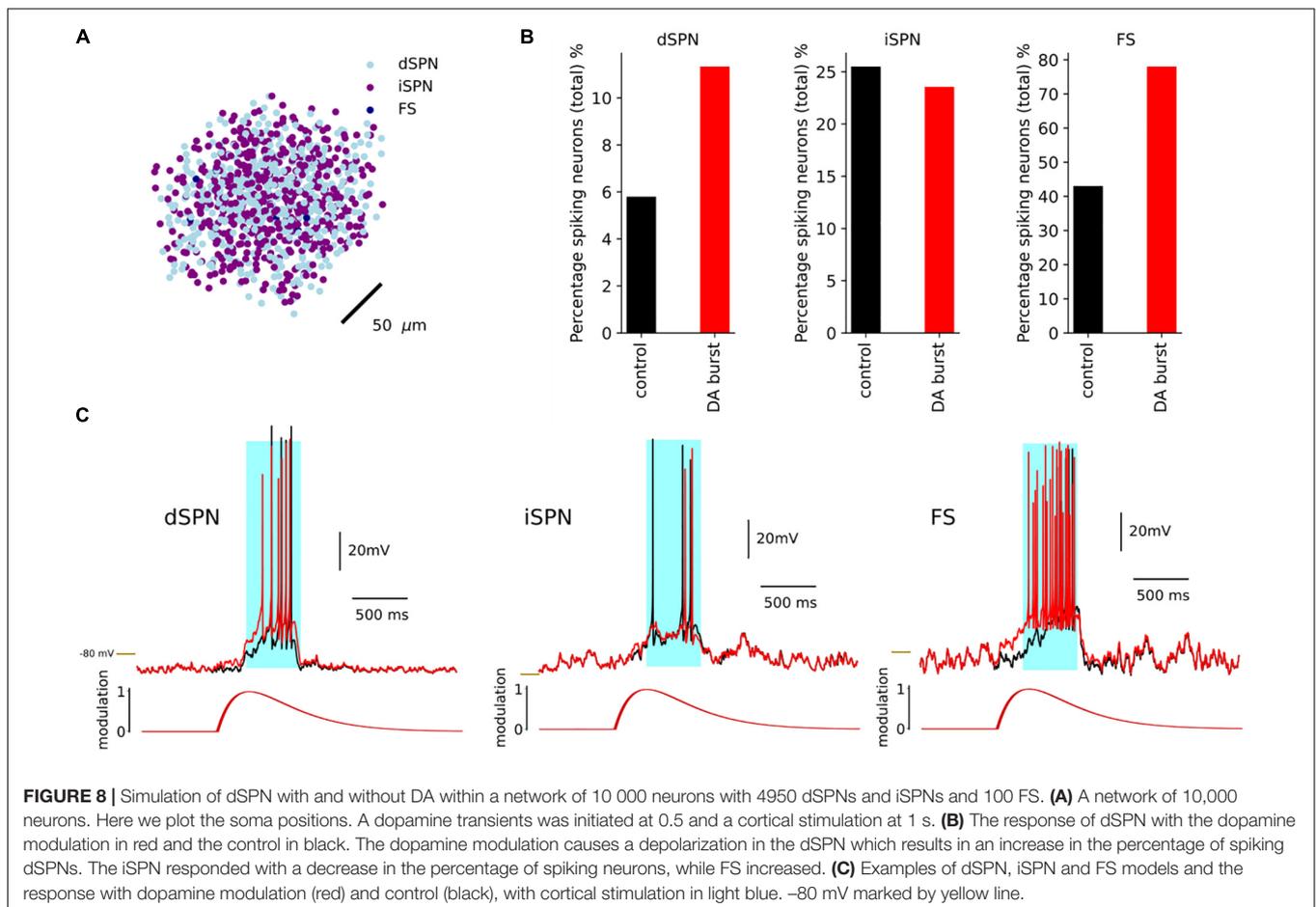
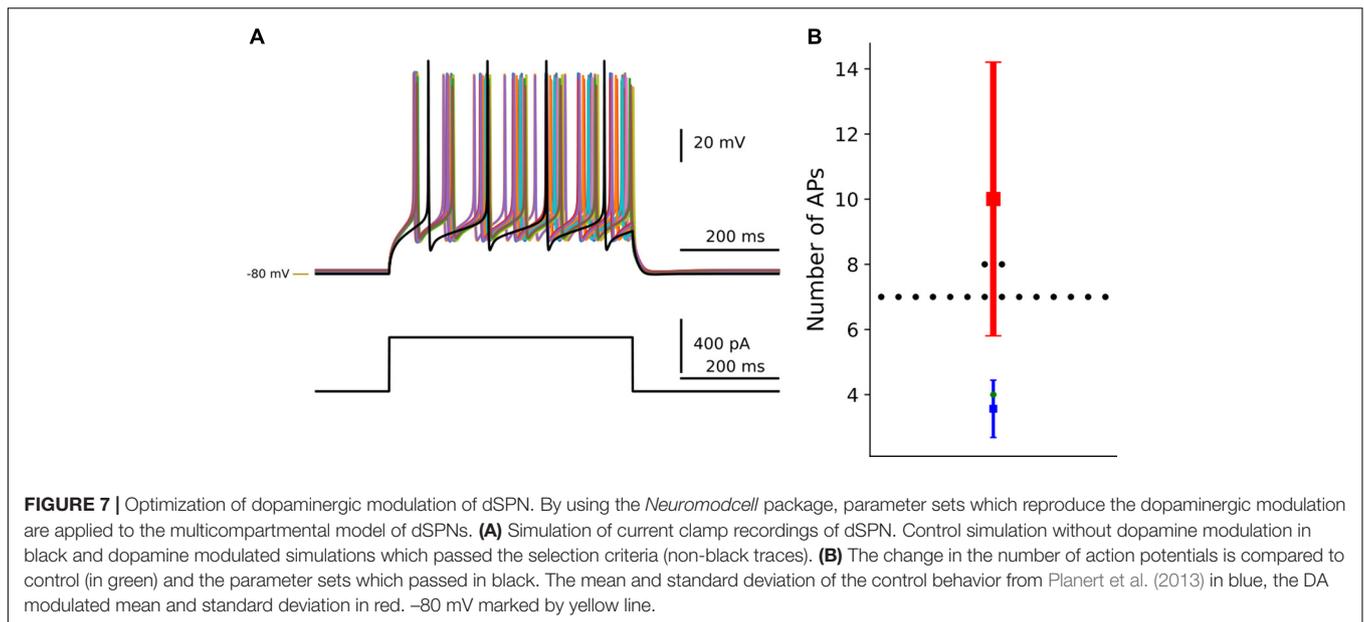
FIGURE 6 | Code for analyzing the results from the optimization using *dSPNanalysis*. *dSPNanalysis* (inheritance from *OptimiseAnalysis* class) loads the results from the optimization. The results can be plotted by custom methods and the final modulation is saved.

models of dSPNs, iSPNs, and FS to exemplify the effect of the dopamine modulation. The simulation included a cortical stimulation of dSPNs, iSPNs and FS at 1 s, lasting for 0.5 s. In **Figure 8**, the effect of dopamine on network activity is shown. Following cortical activation, 6% of dSPNs within the network were spiking. By modulating the ion channels defined in the optimization result of *Neuromodcell*, dopamine modulation increases the percentage of spiking dSPNs to the double (**Figures 8B,C**). iSPNs are modulated by D2 receptors, which reduces the intrinsic excitability. Hence, in **Figure 8B**, the percentage of spiking iSPNs within the network decreases by approximately 3%. The middle panel in **Figure 8C** shows an example trace of an iSPN, where the dopamine modulation reduces the number of spikes. FS are modulated by D1-like receptors, which depolarizes the membrane potential, as seen in **Figure 8C**. Following, an optimization using *Neuromodcell*, the parameter sets for FS dopamine modulation increase the number of spiking neurons within the network (**Figure 8B**). The cortical activation occurs via activating glutamate ionotropic receptor models. These receptors can also be modulated by the same formalism as previously described which leads to changes in the amplitude and release probability (**Supplementary Figure 3**).

Dopamine and Acetylcholine

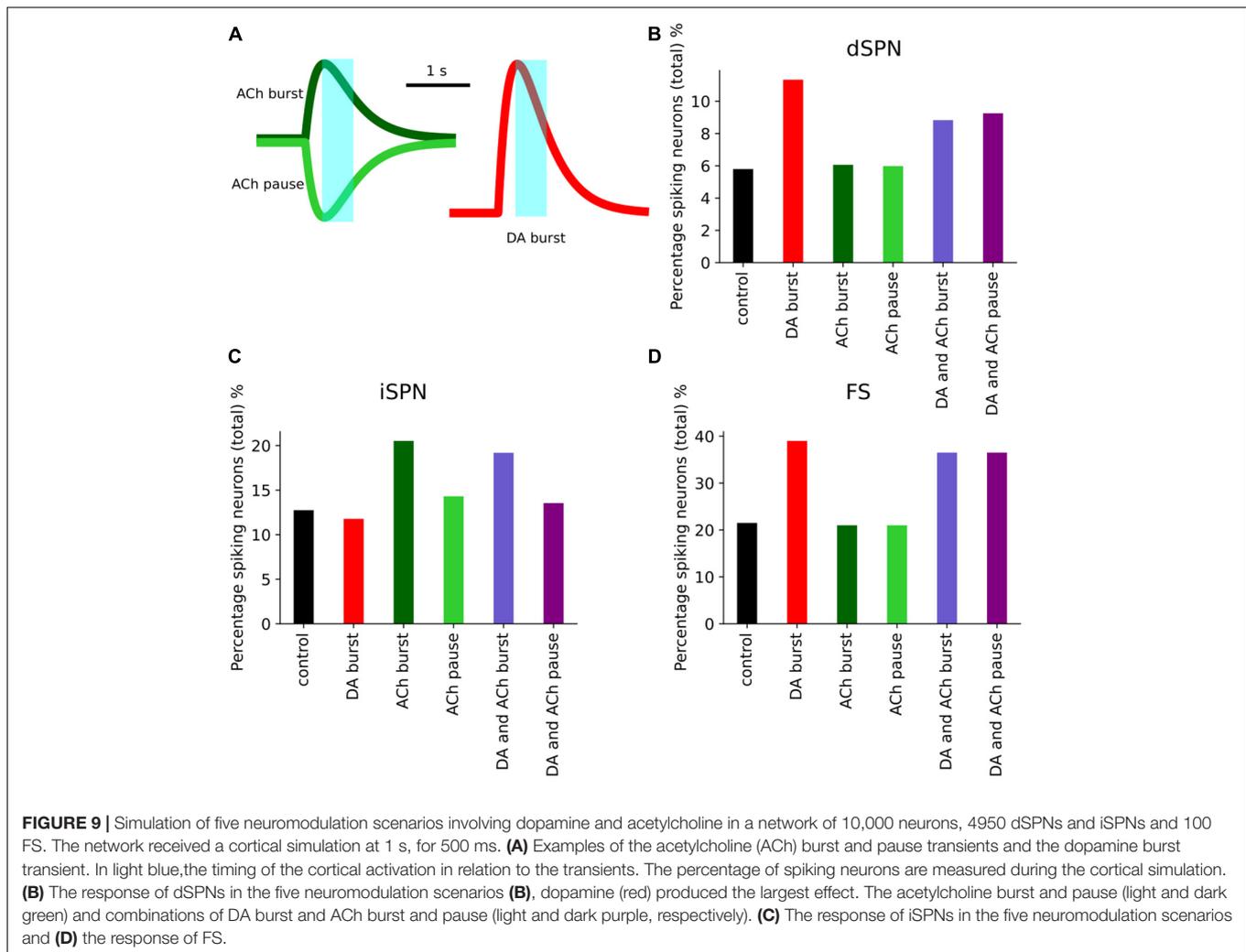
The neuromodulation extension to *Snudda* can also simulate multiple neuromodulators. To demonstrate this, we simulated five scenarios of dopamine and acetylcholine modulation with the same network presented in **Figure 8A**. As previously described, dSPNs and FS are modulated by D1 receptors while iSPNs are modulated by D2 receptors. In addition, we used *Neuromodcell* to optimize for muscarinic modulation for dSPNs and iSPNs.

In **Figure 9A**, we show the transients used in the simulations, which consisted of an acetylcholine burst or pause and a dopamine burst. In the striatum, acetylcholine is released continuously by cholinergic interneurons due to their spontaneous activity. Hence, the acetylcholine transients had a tonic modulation level of 0.5 with burst and pause response causing an increase or decrease (by 0.5), respectively. These transients were simulated individually but also simultaneously (DA and ACh burst; DA and ACh pause). As shown in **Figure 8**, dopamine increased the percentage of spiking dSPNs within the network. The acetylcholine burst also slightly increases the percentage of spiking dSPNs, although in the simultaneous simulation (**Figure 9B**, light purple), the effect is not additive. Instead, the dopamine burst and acetylcholine pause produces a larger response. This is contrary to previous results in Lindroos and Hellgren Kotaleski (2021), where DA and ACh had additive effects when single neurons were simulated. Hence, we performed a simulation with only dSPNs to investigate the effect of DA and ACh. Without network interactions, the effect of DA and ACh was indeed additive (**Supplementary Figures 4A,C**). Hence, the decrease in the percentage of spiking dSPNs could be attributed to the network interactions and the changes seen in iSPNs and FS. As described in **Figure 8**, the intrinsic excitability of iSPNs is decreased following dopamine modulation, while ACh modulates iSPNs via M1 receptor, which lead to the opposite effect. Compared to dSPNs, iSPNs have a larger increase in the



number of spiking neurons following an ACh burst (**Figure 9C**). During the ACh pause, the percentage of spiking iSPNs is expected to decrease, due to the reduction in ACh levels. In

contrast, the percentage of spiking iSPNs was larger than in the control. In Zucca et al. (2018), SPNs were recorded *in vivo* and cholinergic interneurons were inhibited optogenetically.



They observed a reduction in the spontaneous activity of SPNs. The control simulation in **Figure 9** was, however, run without any neuromodulation, which could explain the inconsistency. Hence, we simulated a small network of SPNs with a cholinergic pause, but we modified the control simulation to contain a continuous tonic level of ACh. By comparing the mean firing frequency of SPNs with and without the pause, we observed the expected reduction in SPN activity (**Supplementary Figure 4**). Therefore, the tonic level of ACh has an influence on the activity of SPNs. For FS, the main effect is seen within the dopamine burst modulation (**Figure 9D**), while for ACh pause and burst there is no visible effect (due to the lack of direct muscarinic modulation of FS).

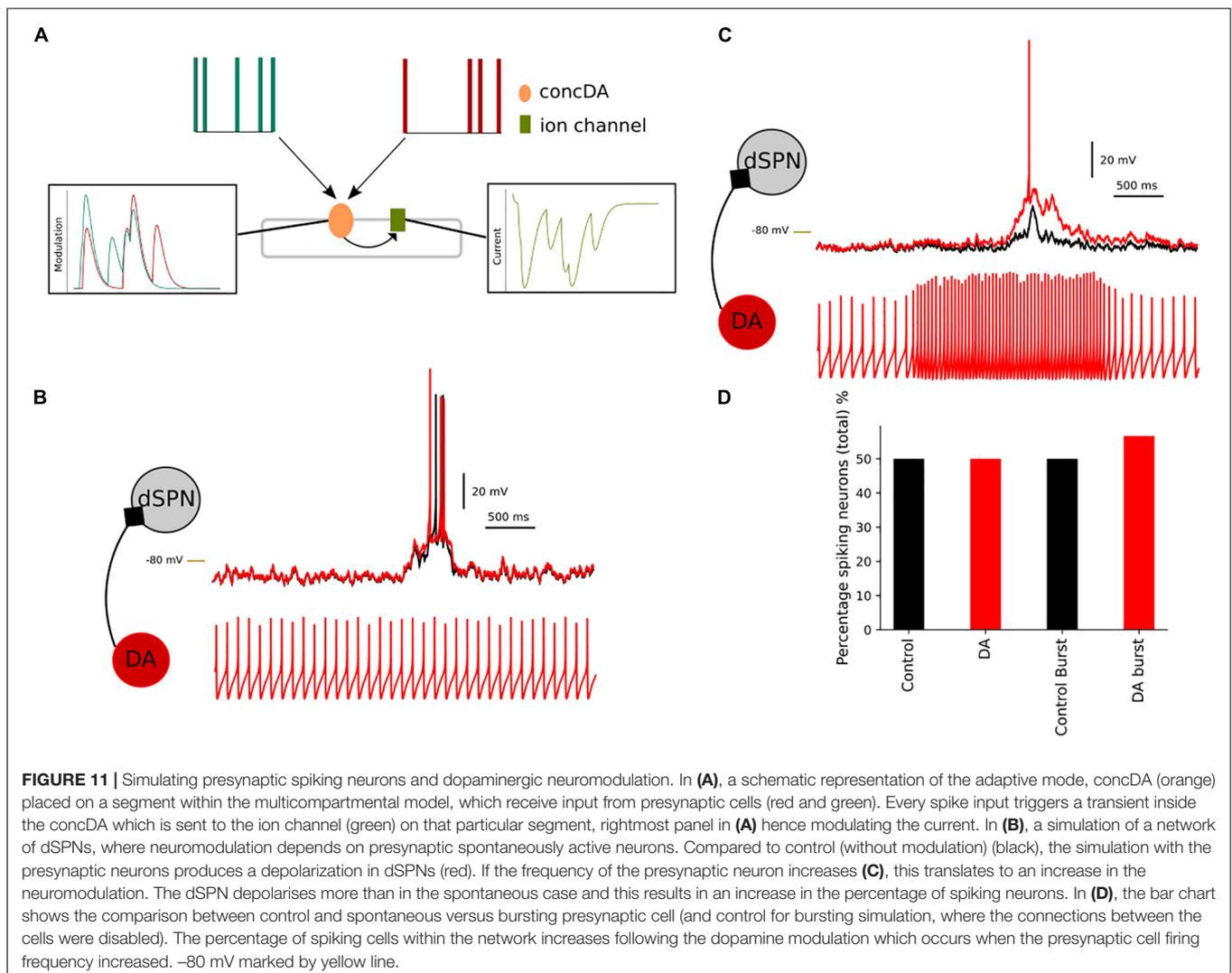
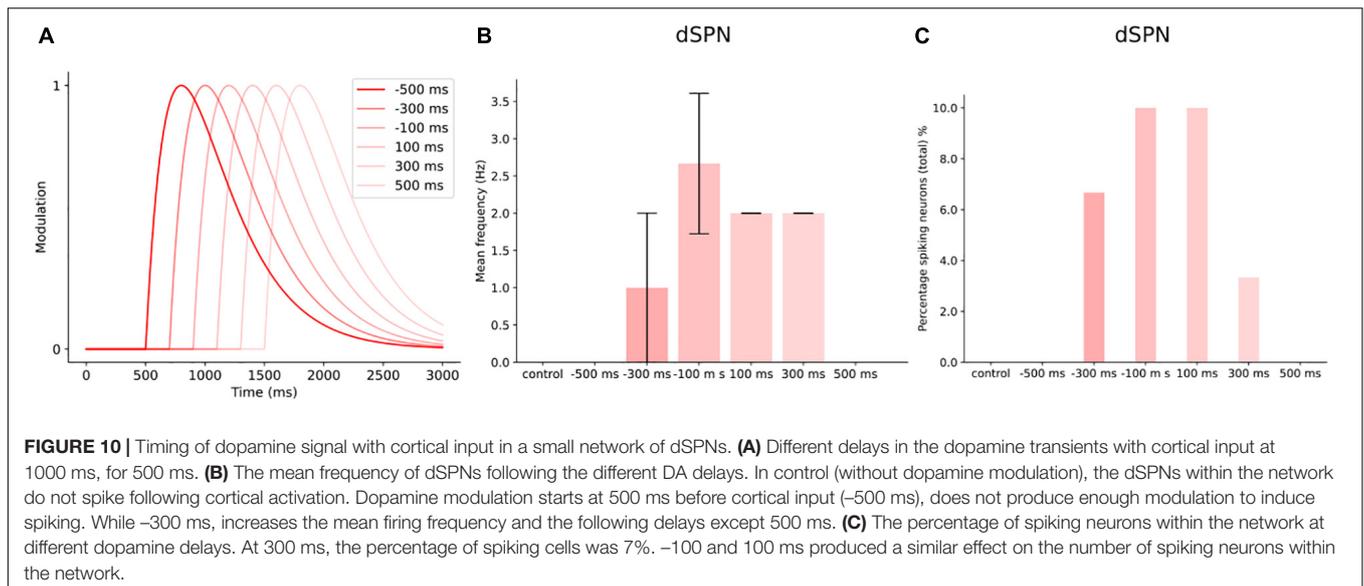
The Timing of Dopamine

The user can construct different transients to investigate the effect of timing on network activity. We created a smaller network of 30 dSPNs to illustrate the effect such an investigation could have on network activity. We defined six transients which were identical, in terms of time constants and modulation level (**Figure 10A**). We then adjusted the start time in relation to

cortical stimulation at 1s (of a 3 s simulation). The different delays caused the peak modulation to occur at different times during the cortical stimulation. **Figure 10B** shows that the transient -100 ms results in the largest increase in mean firing frequency. While $-/+500$ ms transients are not different from control. This is due to the dopamine level being too low and the peak dopamine response occurring much earlier and later compared to the cortical stimulation. The $+100$ ms delay produced a lower increase in the mean firing frequency but the percentage of spiking neurons were similar to the -100 ms case (**Figure 10C**). The user could conduct similar investigations into other features of the transients such as the time constants, tonic levels or more complex transients (Lahiri and Bevan, 2020).

Adaptive Modulation

The previous simulations have used a transient(s) to modulate the network. In **Figure 11**, we simulate neuromodulation using the *adaptive* mode in *Snudda*. In *adaptive* mode, the spiking activity of the presynaptic neurons are translated to modulation levels as described in **Figure 11A**. The top panel in **Figure 11A** shows the spike times of two presynaptic neurons (green and red).



These neurons are connected to an integrating mechanisms called concDA (for dopamine, in orange in **Figure 11A**). The concDA integrates the spiking activity of the two cells into modulation levels (as seen in **Figure 11A** in left most box). The concDA sends the modulation level from both neurons into the ion channel model (right most box in **Figure 11A**). The level of current follows the modulation as the conductance of the ion channel is modulated and hence affecting the neuronal activity.

We created a small network of dSPNs to illustrate the effect of changing the firing frequency of the presynaptic neurons. We placed spontaneously active neurons to modulate the dSPNs via D1 receptors. In control, 50% of the dSPNs within the network were spiking and the spontaneous activity of the presynaptic neurons did not change the percentage (**Figure 11D**). Although in **Figure 11B**, the spontaneous activity has an effect on the membrane potential of the dSPNs. By stimulation the presynaptic spontaneous neurons, we increased the firing frequency (**Figure 11C**). As expected the level of neuromodulation increased as the concDA mechanism integrates the number of spikes into neuromodulation. This leads to a depolarization of the dSPN membrane potential. On a network scale, the increased firing frequency leads to simulated D1 receptor modulation and an increase in the percentage of spiking dSPNs.

DISCUSSION

In this article, we present a framework for simulating neuromodulation in detailed multicompartmental neuronal models being part of large-scale microcircuits, using *Neuromodcell* and *Snudda*. Our priority was to create an extension to *Snudda*, a software for large scale simulation of networks of multicompartmental models, which also simulates transients of neuromodulator(s). Firstly, we developed *Neuromodcell* which enables the user to optimize and select parameters for a specific type of neuromodulator. The neuromodulation is defined by introducing parameters within the ion channel and receptors models, hence enabling dynamic control of for example conductance. Secondly, we developed an extension to *Snudda*, which incorporates the neuromodulation parameters found using *Neuromodcell*. Thirdly, we simulated a range of different neuromodulatory states with the striatal microcircuit to show the versatility of the tool.

We simulated an example network of dSPNs, iSPNs, and FS. These neurons are modulated by dopamine D1 and D2 receptors. We used *Neuromodcell* to optimize for parameters sets which reproduced the available experimental data on dopaminergic modulation of these cell types. The simulations of dopaminergic modulation showed that DA can modulate the excitability of the network. Hence, the simulations of dopamine are in line with known effects of DA in the striatum (Gerfen and Surmeier, 2011; Maltese et al., 2021). Currently, the *Neuromodcell* optimization produces a population of modulated models, which are selected according to predefined parameters. In the future, the *Neuromodcell* optimization could be improved and utilize more elaborate optimization algorithms, for example the genetic algorithm used within *BluePyOpt* software (Van Geit et al., 2016).

Within the striatum, cholinergic interneurons release acetylcholine which modulates the network via nicotinic and muscarinic receptors. Both dSPNs and iSPNs are modulated by the M1 receptor, while only the M4 receptor modulates dSPNs (Abudukeyoumu et al., 2019). Using *Snudda.neuromodulation*, we simulated muscarinic modulation throughout the whole network. We defined a burst and pause transient with a tonic background level of muscarinic modulation to replicate the acetylcholine levels reported in the striatum. We could show that this changes the excitability of dSPNs and iSPNs which is consistent with reports (Abudukeyoumu et al., 2019). Furthermore, we could replicate the effect of ACh pause on SPNs reported by Zucca et al. (2018).

Within the striatum, there are several *in silico* investigations which can be performed using *Snudda.neuromodulation*. In Parkinson's disease (PD), the degeneration of dopaminergic neurons results in a change in the balance between dopamine and acetylcholine (McKinley et al., 2019) and anti-cholinergic drugs were used to treat PD before DOPA therapy emerged (Carlsson, 2001). Although, the interaction between dopamine and acetylcholine is not fully understood. Therefore, future simulations could investigate how the changes of DA and ACh on the single cell level affect network activity; and potentially dissect the important components using *Neuromodcell* and *Snudda*. Furthermore, in Howe et al. (2019), they showed that cholinergic activity and DA levels were coordinated during spontaneous movement. Using *Snudda.neuromodulation* such transients could be simulated to understand how the coordination between ACh and DA affects the activity of dSPNs and iSPNs.

Recent advances in biosensor technology are enabling research to visualize neuromodulator levels within neuronal networks (Leopold et al., 2019). These biosensors can provide high spatial and temporal resolution on the action of neuromodulators like dopamine, serotonin, and opioids amongst others. Hence, a simulation platform like *Snudda*, can incorporate such data to understand how the underlying neural circuit responds to these transients.

Neuromodulation affects many aspects of neurons and neural circuits. Presently, *Neuromodcell* and *Snudda* focused on the effect on ion channels and receptors. Hence, we have included a limited part of the effects that neuromodulators have on network activity (Nadim and Bucher, 2014). Many of the targets of neuromodulators are not incorporated into multicompartmental models, including transcription factors and other subcellular processes. Simulations of such effects would require other types of models and simulators. Although, a possible future development would be to include synaptic plasticity within the large scale simulations. Currently, *Snudda* includes receptor models of glutamate (NMDA/AMPA) and GABA receptors with short-term plasticity. In several brain areas, neuromodulators can regulate the long-term potentiation (LTP) and long-term depression (LTD) (Huang et al., 2012; Xu et al., 2018; Perrin and Venance, 2019). Hence, synaptic plasticity could be coupled to neuromodulation levels during the simulation and simulate the effect of LTP vs. LTD, although currently this is not included in *Snudda*.

Snudda is a general tool for simulating large scale networks of neurons in any part of the nervous system with NEURON simulator. Our aim was to develop an extension to *Snudda*, which simulates neuromodulation on a large scale and can include any neuromodulator. We then developed *Neuromodcell*, which provides a tool for investigating neuromodulation at single cell level, in addition to the integration into the *Snudda* framework. These tools can provide a link between the single cell experiments and circuit level experiments of neuromodulation, which is currently not possible with the available simulation tools.

DATA AVAILABILITY STATEMENT

The datasets presented in this study can be found in online repositories. The names of the repository/repositories and accession number(s) can be found below: The code for single neuron optimizations available at github.com/jofrony/Neuromodcell.git and *Snudda.neuro modulation* is available at github.com/Hjorthmedh/Snudda.git.

AUTHOR CONTRIBUTIONS

JF designed *Neuromodcell* and generalization of neuromodulation in *Snudda* and wrote the manuscript with some input from the co-authors. JHj, SG, and JH contributed throughout the study by discussing the results. All authors contributed to the article and approved the submitted version.

REFERENCES

- Abudukeyoumu, N., Hernandez-Flores, T., Garcia-Munoz, M., and Arbuthnott, G. W. (2019). Cholinergic modulation of striatal microcircuits. *Eur. J. Neurosci.* 49, 604–622. doi: 10.1111/ejn.13949
- Bender, K. J., Ford, C. P., and Trussell, L. O. (2010). Dopaminergic modulation of axon initial segment calcium channels regulates action potential initiation. *Neuron* 68, 500–511. doi: 10.1016/j.neuron.2010.09.026
- Bracci, E., Centonze, D., Bernardi, G., and Calabresi, P. (2002). Dopamine excites fast-spiking interneurons in the striatum. *J. Neurophysiol.* 87, 2190–2194. doi: 10.1152/jn.00754.2001
- Carlsson, A. (2001). A paradigm shift in brain research. *Science (New York, N.Y.)* 294, 1021–1024. doi: 10.1126/science.1066969
- Colangelo, C., Shichkova, P., Keller, D., Markram, H., and Ramaswamy, S. (2019). Cellular, synaptic and network effects of acetylcholine in the neocortex. *Front. Neural Circuits* 13:24. doi: 10.3389/fncir.2019.00024
- Da Silva, J. A., Tecuapetla, F., Paixão, V., and Costa, R. M. (2018). Dopamine neuron activity before action initiation gates and invigorates future movements. *Nature* 554, 244–248. doi: 10.1038/nature25457
- Frost Nylén, J., Carannante, I., Grillner, S., and Hellgren Kotaleski, J. (2020). Reciprocal interaction between striatal cholinergic and low-threshold spiking interneurons — a computational study. *Eur. J. Neurosci.* 53, 2135–2148. doi: 10.1111/ejn.14854
- Gerfen, C. R., and Surmeier, D. J. (2011). Modulation of striatal projection systems by dopamine. *Annu. Rev. Neurosci.* 34, 441–466. doi: 10.1146/annurev-neuro-061010-113641
- Greengard, P. (2001). The neurobiology of slow synaptic transmission. *Science* 294, 1024–1030. doi: 10.1126/science.294.5544.1024
- Hamood, A. W., and Marder, E. (2014). Animal-to-Animal variability in neuromodulation and circuit function. *Cold Spring Harb.*

FUNDING

This study received funding from Swedish Research Council (VR-M-K2013-62X-03026, VR-M-2015-02816, and VR-M-2018-02453) to SG and (VR-M-2017-02806 and VR-M-2020-01652) to JH. Swedish e-Science (SeRC) KTH Digital Futures to JH. European Union (FP7/2007-2013) No. 604102 (HBP), EU/Horizon 2020 No. 720270 (HBP SGA1), No. 785907 (HBP SGA2), and No. 945539 (HBP SGA3) to SG and JH. Karolinska Institutet to SG.

ACKNOWLEDGMENTS

We thank Ilaria Carannante for the help on the early stages and testing *Neuromodcell*. The optimizations and simulations were performed on resources provided by the Swedish National Infrastructure for Computing (SNIC) at the PDC Center for High Performance Computing, KTH Royal Institute of Technology. We also thank the funding agencies for their support.

SUPPLEMENTARY MATERIAL

The Supplementary Material for this article can be found online at: <https://www.frontiersin.org/articles/10.3389/fncir.2021.748989/full#supplementary-material>

Symposia Quantitat. Biol. 79, 21–28. doi: 10.1101/sqb.2014.79.024828

- Hjorth, J. J. J., Hellgren Kotaleski, J., and Kozlov, A. (2021). Predicting synaptic connectivity for large-scale microcircuit simulations using *Snudda*. *Neuroinformatics [Online ahead of print]* doi: 10.1007/s12021-021-09531-w
- Hjorth, J. J. J., Kozlov, A., Carannante, I., Frost Nylén, J., Lindroos, R., Johansson, Y., et al. (2020). The microcircuits of striatum in silico. *Proc. Natl. Acad. Sci. U.S.A.* 117, 9554–9565. doi: 10.1073/pnas.2000671117
- Howe, M., Ridouh, I., Allegra Mascaro, A. L., Larios, A., Azcorra, M., Dombeck, D. A., et al. (2019). Coordination of rapid cholinergic and dopaminergic signaling in striatum during spontaneous movement. *Elife* 28:e44903. doi: 10.7554/eLife.44903
- Huang, S., Treviño, M., He, K., Ardiles, A., Pasquale, R. D., Guo, Y., et al. (2012). Pull-push neuromodulation of LTP and LTD enables bidirectional experience-induced synaptic scaling in visual cortex. *Neuron* 73, 497–510. doi: 10.1016/j.neuron.2011.11.023
- Kozlov, A., Huss, M., Lansner, A., Hellgren Kotaleski, J., and Grillner, S. (2009). Simple cellular and network control principles govern complex patterns of motor behavior. *Proc. Natl. Acad. Sci. U.S.A.* 106, 20027–20032. doi: 10.1073/pnas.0906722106
- Kozlov, A., Kotaleski, J. H., Aurell, E., Grillner, S., and Lansner, A. (2001). Modeling of substance P and 5-HT induced synaptic plasticity in the lamprey spinal CPG: consequences for network pattern generation. *J. Comput. Neurosci.* 11, 183–200. doi: 10.1023/A:1012806018730
- Lahiri, A. K., and Bevan, M. D. (2020). Dopaminergic transmission rapidly and persistently enhances excitability of D1 receptor-expressing striatal projection neurons. *Neuron* 106, 277.e6–290.e6. doi: 10.1016/j.neuron.2020.01.028
- Lansner, A., Hellgren Kotaleski, J., and Grillner, S. (1998). Modeling of the spinal neuronal circuitry underlying locomotion in a lower vertebrate. *Ann. New York Acad. Sci.* 860, 239–249. doi: 10.1111/j.1749-6632.1998.tb09053.x

- Leopold, A. V., Shcherbakova, D. M., and Verkhusha, V. V. (2019). Fluorescent biosensors for neurotransmission and neuromodulation: engineering and applications. *Front. Cellular Neurosci.* 13:474. doi: 10.3389/fncel.2019.00474
- Lindroos, R., Dorst, M. C., Du, K., Filipović, M., Keller, D., Ketzef, M., et al. (2018). Basal ganglia neuromodulation over multiple temporal and structural scales—simulations of direct pathway MSNs investigate the fast onset of dopaminergic effects and predict the role of Kv4.2. *Front. Neural Circuits* 12:3. doi: 10.3389/fncir.2018.00003
- Lindroos, R., and Hellgren Kotaleski, J. (2021). Predicting complex spikes in striatal projection neurons of the direct pathway following neuromodulation by acetylcholine and dopamine. *Eur. J. Neurosci.* 53, 2117–2134. doi: 10.1111/ejn.14891
- Lv, X., Dickerson, J. W., Rook, J. M., Lindsley, C. W., Conn, P. J., and Xiang, Z. (2017). M1 muscarinic activation induces long-lasting increase in intrinsic excitability of striatal projection neurons. *Neuropharmacology* 118, 209–222. doi: 10.1016/j.neuropharm.2017.03.017
- Maltese, M., March, J. R., Bashaw, A. G., and Tritsch, N. X. (2021). Dopamine differentially modulates the size of projection neuron ensembles in the intact and dopamine-depleted striatum. *ELife* 10, 1–24. doi: 10.7554/eLife.68041
- McKinley, J. W., Shi, Z., Kawikova, I., Hur, M., Bamford, I. J., Sudarsana Devi, S. P., et al. (2019). Dopamine deficiency reduces striatal cholinergic interneuron function in models of Parkinson's disease. *Neuron* 103, 1056.e6–1072.e6. doi: 10.1016/j.neuron.2019.06.013
- Nadim, F., and Bucher, D. (2014). Neuromodulation of neurons and synapses. *Curr. Opin. Neurobiol.* 29, 48–56. doi: 10.1016/j.conb.2014.05.003
- Oh, M., Zhao, S., Matveev, V., and Nadim, F. (2012). Neuromodulatory changes in short-term synaptic dynamics may be mediated by two distinct mechanisms of presynaptic calcium entry. *J. Computational Neurosci.* 33, 573–585. doi: 10.1007/s10827-012-0402-z
- Park, J. Y., and Spruston, N. (2012). Synergistic actions of metabotropic acetylcholine and glutamate receptors on the excitability of hippocampal CA1 pyramidal neurons. *J. Neurosci.* 32, 6081–6091. doi: 10.1523/JNEUROSCI.6519-11.2012
- Perrin, E., and Venance, L. (2019). Bridging the gap between striatal plasticity and learning. *Curr. Opin. Neurobiol.* 54, 104–112. doi: 10.1016/j.conb.2018.09.007
- Planert, H., Berger, T. K., and Silberberg, G. (2013). Membrane properties of striatal direct and indirect pathway neurons in mouse and rat slices and their modulation by dopamine. *PLoS One* 8:e57054. doi: 10.1371/journal.pone.0057054
- Surmeier, D. J., Graves, S. M., and Shen, W. (2014). Dopaminergic modulation of striatal networks in health and Parkinson's disease. *Curr. Opin. Neurobiol.* 29, 109–117. doi: 10.1016/j.conb.2014.07.008
- Swensen, A. M., and Marder, E. (2000). Multiple peptides converge to activate the same voltage-dependent current in a central pattern-generating circuit. *J. Neurosci.* 20, 6752–6759. doi: 10.1523/jneurosci.20-18-06752.2000
- Threlfell, S., Lalic, T., Platt, N. J., Jennings, K. A., Deisseroth, K., and Cragg, S. J. (2012). Striatal dopamine release is triggered by synchronized activity in cholinergic interneurons. *Neuron* 75, 58–64. doi: 10.1016/j.neuron.2012.04.038
- Van Geit, W., Gevaert, M., Chindemi, G., Rössert, C., Courcol, J. D., Muller, E. B., et al. (2016). BluePyOpt: leveraging open source software and cloud infrastructure to optimise model parameters in neuroscience. *Front. Neuroinform.* 10:17. doi: 10.3389/fninf.2016.00017
- Xu, H., Perez, S., Cornil, A., Detraux, B., Prokin, I., Cui, Y., et al. (2018). Dopamine–endocannabinoid interactions mediate spike-timing-dependent potentiation in the striatum. *Nat. Commun.* 9:4118. doi: 10.1038/s41467-018-06409-5
- Zucca, S., Zucca, A., Nakano, T., Aoki, S., and Wickens, J. (2018). Pauses in cholinergic interneuron firing exert an inhibitory control on striatal output in vivo. *Elife* 7:e32510. doi: 10.7554/eLife.32510

Conflict of Interest: The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Publisher's Note: All claims expressed in this article are solely those of the authors and do not necessarily represent those of their affiliated organizations, or those of the publisher, the editors and the reviewers. Any product that may be evaluated in this article, or claim that may be made by its manufacturer, is not guaranteed or endorsed by the publisher.

Copyright © 2021 Frost Nylen, Hjorth, Grillner and Hellgren Kotaleski. This is an open-access article distributed under the terms of the Creative Commons Attribution License (CC BY). The use, distribution or reproduction in other forums is permitted, provided the original author(s) and the copyright owner(s) are credited and that the original publication in this journal is cited, in accordance with accepted academic practice. No use, distribution or reproduction is permitted which does not comply with these terms.