

Modelling Human Neural Circuits Across Scales of Complexity

Abstract

Excitatory pyramidal cells and inhibitory interneurons expressing somatostatin (SST), parvalbumin (PV), and vasoactive intestinal peptide (VIP) form a canonical microcircuit across cortical layers 2 and 3 (L2/3). Computational modelling techniques are instrumental in unravelling the distinct functions and interactions amongst the neuron populations of this microcircuit, both under healthy and pathological conditions. Past research suggesting reduced inhibition by somatostatin-expressing interneurons plays a key role in altered inhibition associated with depression in humans has led to the creation of a robust multi-compartmental model of this L2/3 microcircuit, generated by integration of human cellular, circuit and gene expression data. However, simulations using this complex multicompartmental model are both lengthy and computationally expensive. Here, we attempt to provide a more lightweight and abstract alternative to this model. We alter an existing rate-based model of this cortical microcircuit, tuned using mouse V1 data, to reproduce the high level population firing rate activity of the multi-compartmental model as closely as possible. We recreate both models in the intermediate model description language NeuroML2 and adjust the rate-based model to replicate the behaviour of the multi-compartmental human model. When running simulations on the adjusted rate-based model, we observe an acceptable reduction in accuracy when compared to simulations run using the multicompartmental model. Hence, this rate-based model provides a lightweight and rapid alternative that successfully mimics its multi-compartmental counterpart to a sufficient extent. This allows for fast simulations of the L2/3 cortical microcircuit without requiring substantial computational resources, at the expense of a level of accuracy. As such, our model provides a more accessible and transparent alternative, thereby facilitating future investigations into the complex functions and behaviours of the human circuitry behind it.

Declaration of Contribution

The multicompartmental neural model used in this study was developed by Yao et al (2022). The rate-based model that's adjusted in this study was developed by Garcia del Molino et al (2017). The Yao et al model was converted from its original implementation in the NEURON simulator to NeuroML2 by supervisor Padraig Gleeson and colleague Sanjay Ankur. The Garcia del Molino model was rewritten in the Low Entropy Model Specification (LEMS) language, by supervisor Padraig Gleeson. A Python script to generate the Garcia del Molino model in NeuroMLlite format using the LEMS model structure was built by supervisor Padraig Gleeson.

I declare that the following represents my personal contributions. A Python script to run simulations on the rewritten Yao et al model using the NEURON simulation environment, using NeuroML's Python and Java APIs, and generate current-frequency curves and voltage traces, is my work. The manual adjustment of the parameters of the rate-based model written in LEMS, were my own work. The manual adjustment of connectivity weights in the rate-based model were my own work as well. Finally, the methodical simulations and generation of spike rates for the rate-based model under different levels of reduction of SST inhibition, is my work.

Table of Contents

Abstract.....	1
Declaration of Contribution.....	2
Introduction.....	4
Materials & Methodology.....	7
Introduction into Computational Neuroscience.....	7
Models of the Target Microcircuit.....	7
NeuroML.....	9
NeuroML Framework: Libraries & Tools.....	10
NEURON Simulation Environment.....	11
Results.....	13
Yao Et Al Multi-Compartmental Model.....	13
Garcia Del Molino Rate-Based Model.....	15
Rate-Based Model - Population Rate Parameters Adjusted.....	17
Rate-Based Model - Connectivity Weights Adjusted.....	19
Rate-Based Model - Baseline Currents Adjusted.....	20
Comparison of Models.....	21
Discussion.....	23
Conclusion.....	27
References.....	28

Introduction

Treatment-resistant major depressive disorder, commonly known as depression, is characterised by dysregulation and dysfunction of cortical circuits involved in mood regulation. Altered cortical inhibition is implicated in a variety of brain disorders, depression among them. Recent research specifically identified reduced dendritic inhibition from somatostatin (SST) interneurons as a key component in this altered inhibition (Yao, et al., 2022).

A recent study by Yao et al (2022) was able to not only establish this link mechanistically but also produce a powerful computational model of the key L2/3 cortical circuit involved in this. The microcircuit in question consists of excitatory pyramidal cells and GABAergic parvalbumin (PV), somatostatin (SST), and vasoactive intestinal peptide (VIP) expressing interneurons (three major non-overlapping classes of interneurons making up more than 80% of cortical GABAergic interneurons (Rudy, et al., 2011)).

The original model developed by Yao et al is a complex, sophisticated multicompartmental model, consisting of 1000 neurons (80% pyramidal, 5% PV, 7% SST and 8% VIP). Multicompartmental models are complex biophysically detailed representations of neurons, which divide neurons into segments or compartments, and employ mathematical equations, such as the cable equation (Rall, 1962), to describe their biophysical properties. While multicompartmental models, such as the one developed by Yao et al, faithfully emulate the behaviour of actual neurons, their complexity presents practical challenges (Izhikevich, 2004).

Particularly when referring to large networks, such complicated models, consisting of many dynamic parts - with each neuron having hundreds or even thousands of compartments - are very computationally intensive (Blundell, et al., 2018). Running simulations with models such as the one developed by Yao et al, necessitate high-performance computing infrastructure, such as clusters, to complete the simulation within any viable timeframe. This limits the accessibility of multicompartmental models to the wider research community, whilst also making it difficult to work with such models on large scales, even with a sufficiently powerful infrastructure.

To overcome these limitations, one potential alternative is the development of a simpler rate-based model that can emulate the model behaviours of the original. Rate-based neuron models are simplified representations that describe neuron population activity in terms of firing rates or average activity, rather than modelling individual neurons (Abbott and Chance, 2005). Hence, rate-based models are far less computationally demanding. However, as a consequence of their high levels of abstraction, such models cannot emulate individual neurons or even networks as accurately as multicompartmental models.

In this investigation, we develop a lightweight, rate-based variation of the Yao et al multicompartmental model meant to be less demanding on computational resources and significantly reducing runtime. We manually adjust and refine an existing rate-based model of this microcircuit, originally developed by Garcia del Molino et al, 2017, aiming to replicate simulated data from the multicompartmental model. This model consists of four population rates, as well as a matrix of connectivity weights. There are also four baseline current inputs, one to each population, as well as a modulatory current input to the VIP population (Garcia del Molino, et al., 2017), which was removed when the model was adjusted.

Prior to the investigation, the Yao et al model was faithfully reproduced in the standardised simulator-independent language NeuroML (Gleeson, et al., 2010; Sinha, et al., 2023; NeuroML Documentation, 2024), whilst the model developed by Garcia del Molino et al was reproduced in the domain-independent language LEMS (Cannon, et al., 2014). Both languages are simulator-independent, mesh well together (LEMS underlies NeuroML2) and have Python APIs (Vella, et al., 2014), making them easy to work with without sacrificing performance.

Our version of the Garcia del Molino model has four population rates. We manually fine-tune each population rate to closely mirror the firing frequency-input current curve relationship generated by the multicompartmental model. Subsequently, we alter the rate-based model's connectivity weights to match the connection strengths between the populations in the multicompartmental model. Finally, we adjust the baseline input currents of the rate-based model to replicate the behaviour of the complex model, developed by Yao et al, 2020. To assess the accuracy of the adjusted rate-based model in emulating the multicompartmental model, we simulate baseline activity and population firing rates with varying levels of reduced SST inhibition, and test how accurately they emulate the spike rates from the multicompartmental model under the same levels of SST reduction.

Materials & Methodology

Introduction into Computational Neuroscience

Computational neuroscience is an interdisciplinary field at the intersection of neuroscience, physics and computational techniques. It involves the theoretical study of the brain, with the aim of uncovering the principles and mechanisms which underlie brain function through the use of mathematical models and computational techniques. At its core, computational or theoretical neuroscience encompasses the analysis and interpretation of experimental data, as well as the development of mathematical models elucidating the behaviour of neurons and neural circuitry (Trappenberg, 2022, pp. 3-31).

Through computational methods, researchers can extract insights from diverse experimental data sources, such as electrophysiological and behavioural studies. Meanwhile, computational modelling employs mathematical and computational techniques to simulate and investigate the behaviour and dynamics of neurons and neural networks, across various levels of complexity (Trappenberg, 2022, pp. 3-31).

By formulating mathematical equations and algorithms that can replicate neuron and network behaviours, computational models provide a framework for hypothesis testing and probing studies, whilst not requiring the use of animal studies or human volunteers. Researchers can use models to simulate the behaviour of neurons, ion channels, synapses and networks, investigate various aspects of neural processing and explore how activity patterns emerge and give rise to observed behaviours or disorders (Trappenberg, 2022, pp. 3-31).

In this investigation, we delve into two distinct computational modelling paradigms of vastly different levels of complexity: highly intricate multicompartmental models; and heavily abstracted rate-based models.

Models of the Target Microcircuit

Computational models can be of varying levels of complexity, from simple rate-based and point-neuron models to very complex biophysically detailed multicompartmental neurons (Blundell, et al., 2018). Multicompartmental models, such as the one developed by Yao et al (2022), are complex mathematical representations of neurons, that employ a vast array of equations to describe biophysical properties of neurons, such as membrane voltage, action potential, ion conductance, and ion channel densities. A core component of such models is the division of neurons into compartments, each with its set of differential equations for these properties. These compartments help form a faithful replica of the actual neuron morphology and are connected using the cable equation (Rall, 1962), which describes the propagation of electrical signals across compartments, following the direction of information flow.

Yao et al developed conductance-based multicompartmental models for each key neuron type present in the L2/3 microcircuit. Human cellular, synaptic, circuit and gene expression data (Howard, et al., 2022) were used to develop and optimise the model, with rodent data being used to supplement any lacking human data. Multi-objective optimisation was performed, using a mixture of an existing genetic algorithm (Hay, et al., 2011) and the BluePyOpt Python module (Van Geit, et al., 2016). The models themselves, as well as the ion channel models used in said models, were developed using the NEURON simulation environment (Hines, et al., 2020)). Prior to this investigation, the model had been standardised to the simulator-independent language NeuroML.

We use the rate-based model of the same microcircuit, developed by Garcia del Molino et al (2017), as the foundation for our rate-based version of Yao et al's multicompartmental model. Rate-based neuron models, or rate models, are simplified mathematical representations of neuron behaviour and dynamics. Rather than being detailed representations of individual neurons, rate models focus on the average firing rate of entire neuron populations. This makes them a powerful tool for simulating and studying large-scale networks of neurons, where modelling individual neurons becomes computationally expensive (Izhikevich, 2004).

In rate-based models, neuron populations are represented by population units with one average firing rate, as opposed to representing each neuron as a series of discrete spiking events. This of course comes with the assumption that the behaviour of the neuron populations can be described by their average firing rates. These population dynamics are usually described by one or mathematical functions.

Garcia del Molino et al's rate-based model is built using experimental data from optogenetic studies performed in the V1 of behaving mice (Pakan, et al., 2016). Populations have a different connectivity to the model developed by Yao et al (2022), most likely due to being built from rodent experiments rather than human studies, as well as the targeted brain region being different. Average population firing rates are determined by a nonlinear function of the population's input current, first introduced in Abbott and Chance (2005):

$$r_i = f(V_i) = \frac{V_i - V_{th}}{\tau(V_{th} - V_r)} \frac{1}{1 - e^{-(V_i - V_{th})/\nu}}$$

$$V_i = V_l + \left(\sum_j W_{ij} r_j + I_i + I_{bkg}^i \right) / g_l^i$$

$$\tau_r \frac{dr_i}{dt} = -r_i + f(V_i)$$

This function involves several parameters, amongst them voltage threshold (V_{th}), reset potential (V_r), leak voltage (V_l), leak conductance (g_l) and membrane time constant (τ) (Abbott and Chance, 2005; Garcia del Molino, et al., 2017). Prior to the investigation, the Garcia del Molino model was converted into LEMS (Cannon, et al., 2014) and NeuroMLlite JSON format (NeuroML Documentation, 2024).

NeuroML

Due to working with fundamentally different models, which are implemented using different simulation tools and computer languages, we decided to convert the models to NeuroML (Neuronal Markup Language) and LEMS (Low Entropy Model Specification). NeuroML is a simulation-independent modelling language used for describing complex neuronal models and simulations. It provides a standardised format for representing the electrophysical and biophysical properties of neurons and the connectivity of neural network models. Its greatest advantage is its simulator interoperability (Gleeson, et al., 2010). As a simulator-independent model description language, NeuroML is compatible with numerous simulation environments and comes with several tools and libraries that allow easy conversion from NeuroML to the simulator's native model description language (NeuroML Documentation, 2024).

NeuroML models ion channels, neurons, and neural networks using a modular, hierarchical approach. Ion channel, neuron and network models are each divided into their own NeuroML code files; neuron and neural network models include their constituent ion channel and neuron files respectively in the code, in a form of modularisation (Gleeson, et al., 2010). We are using the latest version of NeuroML, NeuroMLv2/LEMS. LEMS is a simulator-independent, domain-independent modelling language, which isn't tied to any particular domain of biology. LEMS underlies NeuroML code by defining conventional units and dimensions, the dynamics of ion channels and the structure of neurons in multicompartmental models, as well as being used in simulation files, which contain the models to be simulated and the parameters of the simulation (Cannon, et al., 2014; NeuroML Documentation, 2024).

NeuroML Framework: Libraries & Tools

NeuroMLv2/LEMS is part of a larger complex of tools and Java and Python libraries and APIs. Although NeuroML and LEMS are easier for humans to read than other model description languages due to being XML-based, declarative languages like XML are still difficult to work with. For this reason, the NeuroML/LEMS schema is wrapped by several Java and Python libraries and APIs (see Figure 1). The jLEMS library written in Java and pyLEMS library written in Python provide implementation of LEMS and the LEMS parser, and can write, load and simulate models. The libNeuroML Python library is a Python API for reading, writing and validating NeuroML code (Vella, et al., 2014; Sinha, et al., 2023).

The jNeuroML library wraps jLEMS and the NeuroML2/LEMS definitions, whilst providing additional functionality, such as the conversion from NeuroML to other simulator formats and running simulation using these simulation environments. It also has a built-in simulation environment, capable of simulating simple models without compartments. Finally, the pyNeuroML Python library wraps around jNeuroML, libNeuroML and pyLEMS, whilst providing additional helper methods (Sinha, et al., 2023; NeuroML Documentation, 2024). These libraries make it much easier to write, read and work with NeuroML as Python is a more human friendly language than XML (Vella, et al., 2014). Moreover, they automate the conversion process from NeuroML to other formats and allow the running of simulations using many other simulation environments with only Python code.

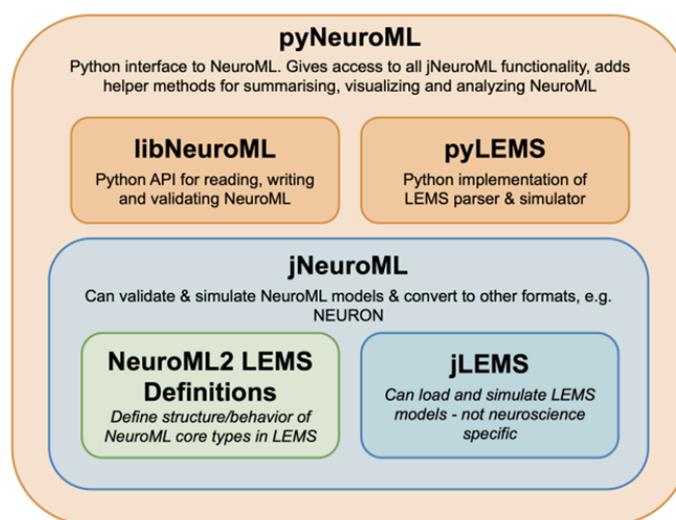


Figure 1. Python and Java libraries that make up the core NeuroML software stack, cited from Sinha, et al., 2023.

All of the above libraries were used at some point during our investigation. Aside from these, there are other tools and APIs for implementing and developing models written in NeuroML. In our investigation, we work with NeuroMLlite, a framework that builds on NeuroML2 and is in active development at the time of this report. NeuroMLlite aims to provide a high-level specification, written in JSON, that can be used to generate networks in NeuroML, as well as other formats (NeuroML Documentation, 2024). We use this framework when adjusting the rate-based model developed by Garcia del Molino et al (2017), as NeuroMLlite provides a robust GUI that allows us to observe how adjusting model parameters affects the population rates in real-time.

NEURON Simulation Environment

We mentioned earlier that the jNeuroML library has a built-in simulation environment (NeuroML Documentation). However, this environment is very simple and cannot perform simulations using multicompartmental models, such as the model developed by Yao et al, 2022. Instead, NeuroML is compatible with a diverse array of simulators through simulator interoperability (Gleeson, et al., 2010; Sinha, et al., 2023). We chose NEURON as our simulation environment of choice for running the Yao et al model, due to it being a powerful and versatile simulator, as well as it being the simulator used to develop the Yao et al model.

NEURON is a widely used tool for simulating the activity and behaviour of neural models. At its core is the NEURON simulation engine, capable of simulating detailed multicompartmental models incorporating biophysical properties, such as ion channels, synapses, and membrane capacitance. The simulation environment allows the construction of complex models through a hierarchical approach, allowing users to build models by assembling their basic components, such as compartments and synapses (Hines, et al., 2020).

Like many other simulators, NEURON has its own model description language based on C: the Hierarchical Object-Oriented Scripting Language, HOC. Being based on C, HOC is a performance-focused language, allowing simulations of complex models to be run with minimal computational resources needed. Moreover, being object-oriented, its code is modular and reusable. However, it does come with a steep learning curve and a verbose syntax, making it difficult to work with (Hines, et al., 2009; Hines, et al., 2020).

For this reason, NEURON also provides a Python wrapper library (Hines, et al., 2009), which makes NEURON much easier to work with. However, HOC is heavily tied to the NEURON simulation environment, resulting in models written in HOC having limited portability. We use NeuroML to overcome this problem, as the jNeuroML library can convert NeuroML2 and LEMS code into HOC code and run simulations using NEURON under the hood.

Results

Yao Et Al Multi-Compartmental Model

Prior to the commencing of this investigation, the multicompartmental model developed by Yao et al, 2022, and the rate-based model developed by Garcia del Molino et al, 2017, were converted to NeuroML2/LEMS (Cannon, et al., 2014) and NeuroMLlite (NeuroML Documentation, 2024) formats respectively. To ensure that the NeuroML versions are faithful representations of the models, we test the performance of both models compared to their original formats, using figures and data from the original studies (Yao, et al., 2022; Garcia del Molino, et al., 2017) as a test metric, in order to confirm that they faithfully mirror the properties of the originals.

In NeuroML2, models are structured hierarchically, with higher-level elements being made up of lower-level components, each with its own NeuroML2 description. To test how faithfully the NeuroML version of the Yao et al model reproduces the properties and behaviour of the original, we simulate models of single neurons from each population. These single-cell models have no connections to a greater network, so running simulations using these models is easier and less computationally demanding.

We use the pyNeuroML library (NeuroML Documentation, 2024) to run simulations on these single-neuron models using the NEURON simulation environment (Hines, et al., 2020). We generate voltage traces using 0.1nA stimulus current amplitude for excitatory pyramidal cells and SST interneurons, and 0.2nA stimulus current amplitude for PV and VIP interneurons. All frequency-input current curves were generated by applying step currents of 0.01nA increments within the range of -0.1nA and 0.3nA. All simulations were run at a temperature of 32°C.

As shown in Figure 2, all generated voltage traces accurately matched the voltage traces from the original Yao et al study, with PV cells having a high firing rate and pyramidal cells having a low firing rate compared to SST and VIP cells. Similarly, all generated frequency-input curves closely match their original counterparts. From this, we concluded that the NeuroML2/LEMS implementation of the L2/3 multicompartmental model accurately represents the original model at the level of individual cells. We determined confirming this to be sufficient for our investigation, as we did not intend to use the fully connected 1000-neuron model in this study.

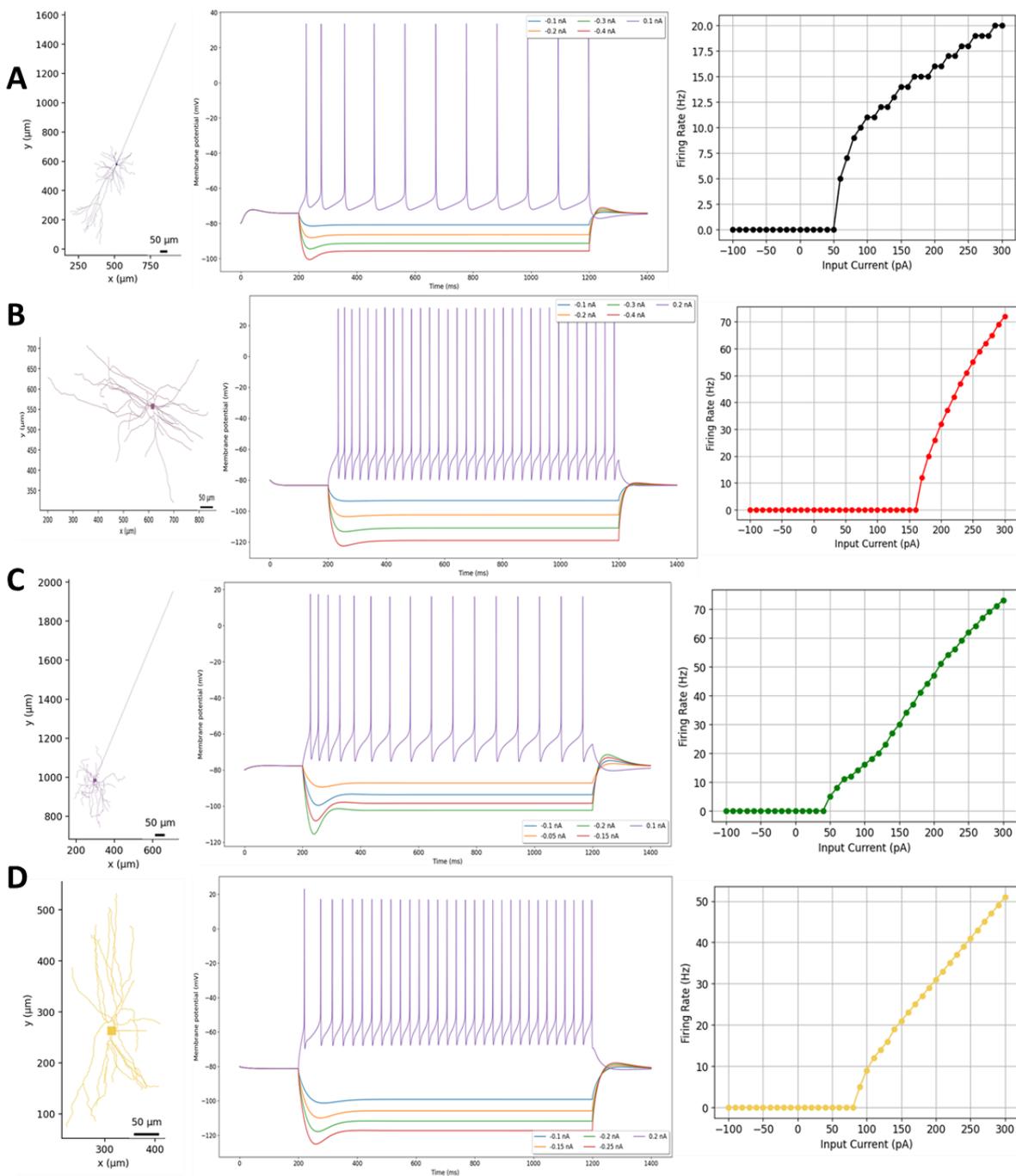


Figure 2. NeuroML version faithfully replicates properties of the multicompartmental model of the human L2/3 cortical microcircuit.

(A) Left: L2/3 pyramidal neuron morphology reconstructed in 2D. Center: Voltage response of pyramidal cell model to depolarizing and hyperpolarizing step currents. Right: Firing frequency-input current curve shows relationship between current input and firing rate of pyramidal cell model. (B, C & D) Same as the above, but for PV, SST and VIP interneuron models.

All figures were generated using the plot function of the pyNeuroML API (NeuroML Documentation, 2024)

Garcia Del Molino Rate-Based Model

We used a rate-based model developed by Garcia del Molino et al (2017) as the foundation for our rate-based model of the human L2/3 microcircuit. We selected this model because it represents the same microcircuit and exhibits similar properties and behaviour to the multicompartmental model. Although the rate-based model was generated using data from the mouse V1 (Pakan, et al., 2016) rather than the human L2/3, the microcircuit models are still very similar in cell composition and connectivity. Moreover, the firing rates of the cell models are already relatively similar. As such, adapting the rate-based model to reproduce the complex model's behaviour is a relatively simple process.

Before this investigation, the individual cells of the rate-based model were described in LEMS, while the network was described in NeuroMLlite JSON format. Each population was derived from a foundational component, which was built on top of the Abbott and Chance (2005) non-linear function for calculating average population firing rates in rate-based models. This function has several parameters, which differ between the four rate populations. Aside from the populations, the model also includes four baseline currents inputs, one for each population, and a modulatory current input to the VIP population. The connections between populations are defined by a matrix of connection weights (see figure 3B).

To test if the NeuroMLlite version of the rate-based model replicates the population rates and other properties of the original model, we simulate the model and compare the recorded population rates to those generated from the original. Simulations were running using two different baselines, each with different initial firing rates and baseline current inputs. NeuroMLlite was used to run simulations using the built-in jNeuroML simulator (NeuroML Documentation, 2024). We used the jNeuroML simulator because the models are rate-based and have no compartments, so it is more efficient to use the built-in simulation environment for such a simple model.

We found that the generated rate-time graphs matched those of the model in the original study, replicating several key behaviours such as the disinhibition regime in simulations with low baseline activity and the response reversal regime in simulations with high baseline activity, two emerging phenomena on which the study by Garcia del Molino et al (2017) focuses on. We also observed this from the generated tuning curves, which similarly matched those generated from the original model. From these results, we concluded that the NeuroMLlite version of the model faithfully represents the original rate-based model developed by Garcia del Molino et al, 2017, and as such can be used as a foundation for a rate-based model that replicates the behaviours of the multicompartmental model developed by Yao et al, 2020.

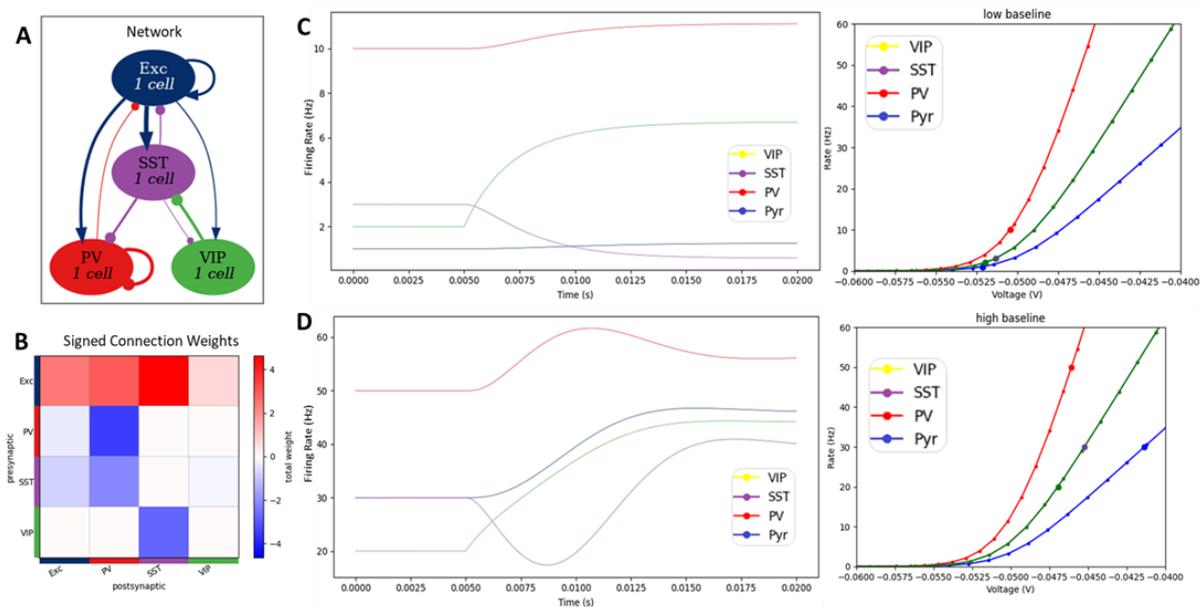


Figure 3. NeuroMLlite version faithfully replicates properties of the rate-based model of mouse V1 microcircuit.

(A) Microcircuit connectivity graph, “1 cell” means one population rate. (B) Signed connection weight matrix of rate-based model connections. (C) and (D) Transient firing rate dynamics under onset of modulatory VIP current at 0.005s (left) and population rate tuning curves (right) reproduced under low baseline (top) and high baseline (bottom) conditions respectively.

All figures were generated using the GUI provided by NeuroMLlite (NeuroML Documentation, 2024).

Rate-Based Model - Population Rate Parameters Adjusted

With both models tested against their original version, we begin adjusting the rate-based model developed by Garcia del Molino et al (2017) to reproduce the properties and behaviour of the multicompartmental model. We first remove the VIP modulatory input current, as there was no such modulatory current in the original study by Yao et al (2022). As such, removing it would make it impossible to test the performance of the rate-based model against its multicompartmental equivalent.

Before we adapt the connectivity of the model, the properties of individual rate-based populations must replicate that of their multicompartmental neuron counterparts. As such, we temporarily remove all connectivity between the population rates. We do this so that the firing rate of each population will be caused by the population alone, with no external influences from the other rate populations. Here, we do not alter the population firing rate (Abbott and Chance, 2005). Instead, we manually adjusted the parameters of each population rate, through a process of trial and error, to replicate the properties of individual multicompartmental cells as closely as possible.

We generate firing rate-baseline current curves and compare them to the firing frequency-input current curves generated during the aforementioned testing of the multicompartmental model. The rate-current curves were generated by applying step currents of 0.01nA increments within the range of -0.1nA and 0.3nA, just as was done for the complex models. We employ these graphs to iteratively refine the parameters of each population rate, aiming to replicate the firing rate-input current relationship of the original neuron models developed in the study by Yao et al, as closely as possible.

This is challenging, as the relationship between firing rate and input current in rate-based models is linear after the threshold is reached. However, this relationship has far more complexity in multicompartmental models, making replicating this relationship in rate-based models difficult. In this model, this is especially the case for the excitatory pyramidal cells, as illustrated in Figure 4B. Here, we observe a sharp increase in firing rate to 10Hz after achieving the threshold, followed by a minimal further increase in firing rate. This non-linear relationship is particularly difficult to model using rate models, hence why the pyramidal population rate fails to faithfully reproduce the rate-current curve.

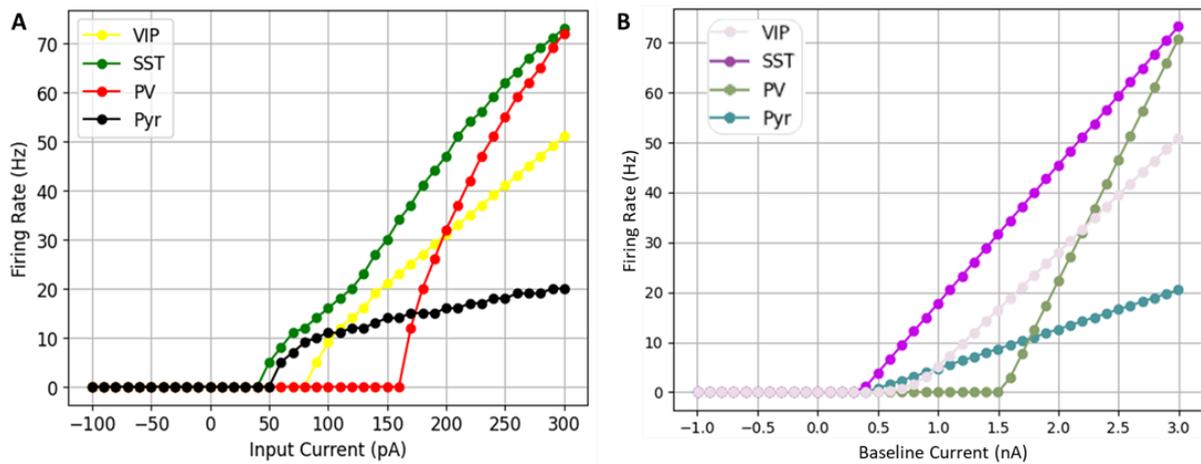


Figure 4. Adjusted population rates emulate firing frequency-input current relationship of their multicompartmental neuron counterparts

(A) Combined firing frequency-input current curve of the L2/3 multicompartmental model. (B) Combined firing frequency-input current curve of rate-based version of the L2/3 multicompartmental model. Adjusted SST, PV and VIP interneuron population rates of the rate-based model successfully emulate the frequency-input relationship of their multicompartmental counterparts. The pyramidal population rate struggles to emulate the non-linear frequency-input relationship of multicompartmental pyramidal cell models.

Figures 4A and 4B were generated using the plot function of the pyNeuroML API and the the GUI provided by NeuroMLlite respectively (NeuroML Documentation, 2024).

Although the behaviour of the multicompartmental cell models wasn't perfectly replicated, we successfully reproduce the relationship of said cell models between their firing rates and the input current to a satisfactory extent. Despite some minor deviations, the rate-based populations effectively emulate the key properties of the multicompartmental cells. Thus, we conclude that the individual population rates encapsulate the fundamental characteristics of their multicompartmental counterparts.

Rate-Based Model - Connectivity Weights Adjusted

With the individual population rates adjusted to match the behaviour of their multicompartmental counterparts, we next focus on adjusting the higher-level components of the rate-based model, specifically the baseline currents and the connectivity between populations. The connectivity weights must be representative of the connection strengths between the populations in the multicompartmental model. As such, the original connectivity weight matrix from the model developed by Garcia del Molino et al (2017), must be replaced by a weight matrix representative of the connection strengths in the complex model.

Connection strength in the multicompartmental model is complex, with each cell having multiple projections to other cells, each projection having multiple connections to other cells, each connection having multiple synapses and each synapse having a different weight. On top of that, each cell population has multiple cells, with a total of 1000 cells over the four populations (Yao, et al., 2020). A connection exists when a presynaptic cell projects to one or more postsynaptic cells. In the end, we measured the strength of each projection with the following equation: $n * w * G / pop$, where n is the number of connections in a projection, w is the average weight per connection, G is the average synaptic conductance and pop is the postsynaptic population size.

This equation gives us a comprehensive understanding of the strength of each projection relative to other projections between populations. As such, we use it to construct a connection weight matrix for the rate-based model. Because the model's population size decreases from 1000 neurons in the multicompartmental model to 4 population rates in the rate-based model, all connection weights in the matrix are reduced by a scale factor of 0.01, as shown in Figure 5B.

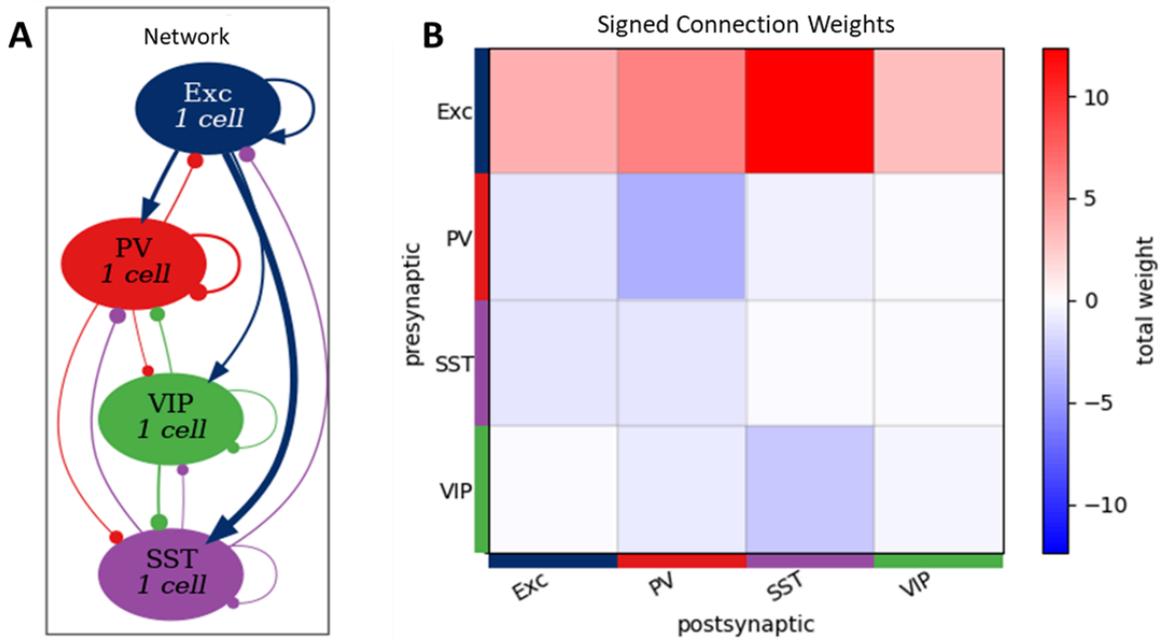


Figure 5. Connectivity weight matrix is adjusted to replicate the connectivity of the multicompartmental model of the L2/3 microcircuit.

(A) Microcircuit connectivity graph reconstructed from new connectivity matrix.

(B) Signed connection weight matrix constructed using new connection weights.

All figures were generated using the GUI provided by NeuroMLlite (NeuroML Documentation, 2024).

Rate-Based Model - Baseline Currents Adjusted

As the core of the network has now been adapted to emulate its multicompartmental counterpart, we next adjust the baseline current inputs. In neuron and neural network models, baseline currents model the intrinsic or spontaneous activity of the neurons in the absence of an external stimulus. They arise from various sources, such as leak channels or background activity. As such, baseline currents represent the external environment of neurons, albeit in a simplified manner.

For the behaviour of the rate-based model to emulate that of its multicompartmental counterpart, this external environment must also be adjusted. As such, we manually adjust the baseline current inputs to each population, through a methodical process of trial and error, to reproduce the firing rates generated by the multicompartmental model. We simulate the rate-based model using the built-in jNeuroML simulator (NeuroML Documentation, 2024), under resting state conditions. We generate the firing rates of all populations in the rate-based model and compare them to those generated by its multicompartmental counterpart in the study by Yao et al, 2022. We use these graphs to iteratively refine the baseline currents, in order to replicate the neuron firing activity displayed by the multicompartmental model with a high degree of accuracy.

We successfully reproduce the spike rate dynamics of the complex model, as shown in Figure 6. Notably, all firing rates simulated by the rate-based model fall within $\pm 0.2\text{Hz}$ of the spike rates recorded in the original study by Yao et al (2022). From this error margin, we determine that the rate-based model can successfully emulate the firing activity of its multicompartmental counterpart under standard physiological conditions.

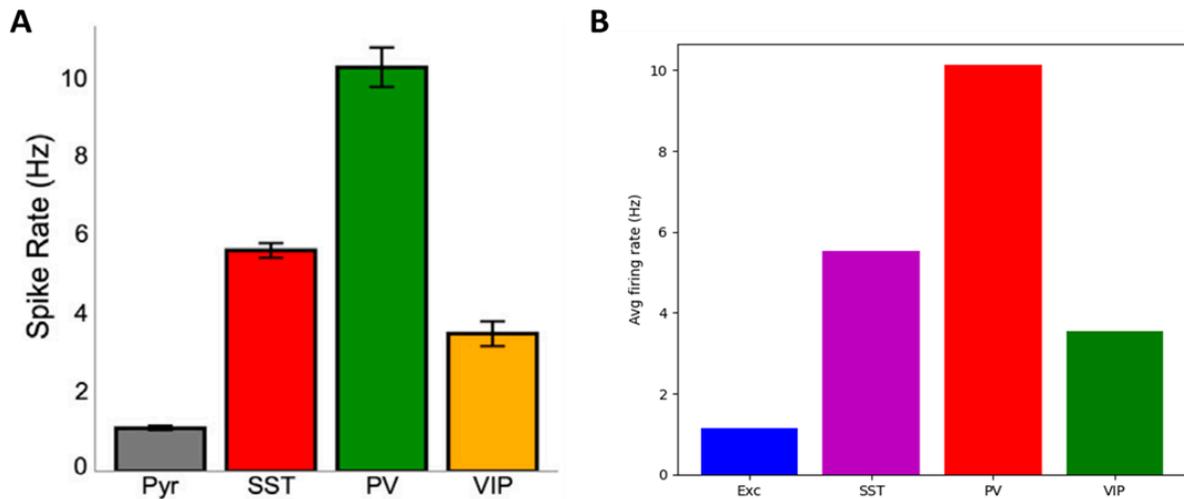


Figure 6. Adjusted rate-based model successfully replicates population firing rates under standard, healthy conditions.

(A) Population spiking rates of L2/3 multicompartmental model, cited from Yao, et al., 2022.

(B) Population firing rates of rate-based populations match the spiking rates of their multicompartmental counterparts within a standard error of ± 0.2 Hz.

Figure 6B was generated using the Python API provided by NeuroMLlite (NeuroML Documentation, 2024).

Comparison of Models

The rate-based model has been adapted to successfully emulate key behaviours of its multicompartmental counterpart, namely population firing rates, under regular conditions. However, its ability to accurately reproduce these behaviours outside of such conditions must also be tested. Therefore, we employ the rate-based model to replicate data derived from simulations on the multicompartmental model, under reduced SST interneuron inhibition.

This method of testing model performance is inherently flawed, as the reduction of SST inhibition in the study conducted by Yao et al, 2022, consists of both synaptic and tonic inhibition. Tonic inhibition involves the inhibition of individual neurons, making it harder for them to reach the action potential threshold, and as such can't be easily replicated in a rate-based model, which models entire populations. To overcome this, during our analysis, we focus on whether the relationship between the firing rates and the level of SST reduction is preserved, rather than the actual data values being replicated.

Although reproducing tonic inhibition in a rate-based model isn't practical, we can still model synaptic inhibition of SST interneurons. We do this by reducing the connection strengths of projections from the SST population to other populations by 0%, 20%, 40%, 60%, 80% and 100% scale factors. These represent the reduction of SST activity by 100%, 80%, 60%, 40%, 20% and 0% scale factors respectively. Using these scale factors, we simulate the rate-based model and plot bar charts plotting excitatory pyramidal cell firing rates under all levels of reduction of SST connectivity strength, as shown in Figure 7A. We repeat this for SST, PV and VIP interneurons, although only for a 40% reduction of SST activity (60% reduction of projection strengths).

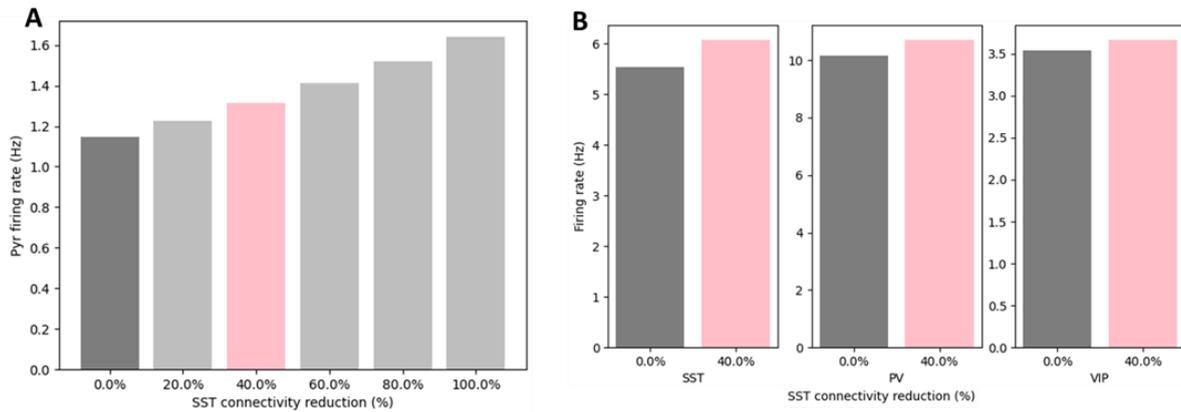


Figure 7. Population firing rates of the adjusted rate-based model tested under reduced levels of SST inhibition.

(A) Pyramidal population firing rate simulated under varied levels of reduction of SST projection strengths. The linear relationship is successfully replicated, but individual increases are smaller than expected.

(B) Same as the above, but for SST, PV and VIP population rates, under standard conditions and 40% reduction of SST projection strengths.

All figures were generated using the Python API provided by NeuroMLlite (NeuroML Documentation, 2024).

From these graphs, shown in Figure 7, we can identify a clear increase in the firing rates of all four populations, as an immediate consequence of the reduction of connection strengths from the SST population. This matches the behaviour of the multicompartmental cell models (Yao, et al., 2022), although there is less of an increase than expected. Moreover, the relationship between pyramidal cell firing rates and the reduction of SST projection strength is mostly linear, accurately emulating the relationship between pyramidal cells and SST activity reduction in the multicompartmental model (Yao, et al., 2022). Once again though, the increase in firing rate as SST activity reduction increases, is much smaller than expected.

Discussion

In this investigation, using an existing model developed by Garcia del Molino et al (2017) as a foundation, we developed a rate-based model capable of emulating the key properties and model behaviours of a multicompartmental model of the human L2/3 cortical microcircuit, developed by Yao et al (2022). However, the model is not without several limitations and there are still plenty of areas of improvement.

Overall, the rate-based model has proved capable of replicating neuronal behaviour and firing rates of the multicompartmental model under normal physiological conditions. Moreover, it accurately emulates the overall behaviour of its complex counterpart under abnormal or pathological conditions, such as under the reduction of SST projection strengths. However, its ability to accurately reproduce data simulated by the model developed by Yao et al, 2022, outside of a typical physiological environment, remains to be seen. On top of that, while the model can successfully reproduce the firing rate behaviours of the entire model, discrepancies arise when it comes to individual populations. As such, although the model can accurately reproduce the key behaviours of the model, it fails to reproduce all the nuances of the multicompartmental model in our tests.

From this, we can only partially confirm that our model can accurately emulate its multicompartmental counterpart, as the results of our investigation are relatively mixed. Firstly, as mentioned earlier, the individual population rates cannot accurately replicate the mostly non-linear relationship between the firing rate of neurons and the input current stimulus that is observed in simulations of individual cells of the multicompartmental model. This is an inherent problem of the abstract nature of rate-based models, which represent population firing rates as mathematical equations, in a linear relationship between firing rate and input current strength. Although this is an expected problem, it does not change that, although the model can replicate their overall behaviour, it fails to fully emulate certain nuances of individual neurons of its multicompartmental counterpart. This raises the following question: have the key properties of individual multicompartmental neurons been successfully replicated?

To add to this, there is uncertainty over the overall accuracy of the model. The adapted connectivity matrix and refined baseline currents allow successful replication of the spike rates generated by the multicompartmental model under standard conditions. However, when testing the model's firing rates under pathological conditions, specifically the reduction of the inhibitory activity of SST interneurons, the abstract nature of the rate-based model becomes a problem again. In the study by Yao et al, 2022, the inhibition of SST activity is the result of synaptic and tonic inhibition of SST interneurons. However, tonic inhibition is a general inhibition of the biophysical and electrophysiological activity of individual cells. The rate-model does not include a tonic inhibition to all cell populations, and so this could only be approximated by the reduction of projection strengths. Although we mitigate this issue by focusing on reproducing the overall trend between the population firing rates and the degree of SST activity reduction, this test remains incomplete, as we remain unsure of the accuracy of the generated spike rate data. As such, we only partially confirm the accuracy of our model in replicating the key behaviours and properties of its multicompartmental counterpart.

Aside from the aforementioned issues, there is also uncertainty about the methods used in fitting the rate-based model, as some of them are relatively crude. This is especially the case with the adjustment of the baseline currents and parameters of the population rates, as these were refined manually through a process of trial and error. Although this method isn't flawed, it is also not the most optimal option. An alternative would be the use of machine learning techniques, such as the genetic algorithm used to optimise the individual multicompartmental neurons in the study by Yao et al, 2022, which would result in a much more refined, and most likely more precise, set of parameters and baseline currents for the rate-based model.

Another relatively crude method used is the scale factor applied to the connection strengths of the model, to account for the differences in population sizes between the multicompartmental model (1000 neurons) and its rate-based equivalent (4 population rates). However, the population sizes are not equal, with the pyramidal neuron population being far larger in the complex model, and so they do not decrease by the same level. We use a scale factor of 0.01, which reduces the connection strength by a factor of 100, for the sake of simplicity. This, however, may mean that the connectivity strengths are inaccurate and that the baseline currents are potentially overfitted to compensate for this discrepancy. A potential improvement could involve calculating the ratio of how much each population decreased in size and utilising it to derive scale factors unique to each population.

Overall, we conclude that there are many limitations with both the methods used and the model itself. As such, there are plenty of opportunities for improvements and further refinement of the rate-based model. First of all, the lack of evidence regarding whether or not the model can replicate the behaviour of its multicompartmental counterpart needs to be addressed. One way to do this is to replicate, or if that's not possible mimic, the tonic inhibition of the SST population. To achieve this, the effect tonic inhibition alone has on the firing rate of SST cells in the multicompartmental model must be measured and determined. This can then be reproduced in the rate-based model by either iteratively adjusting the baseline currents to reproduce the effects of tonic inhibition; or directly reducing the firing rate of the SST population by the amount of reduction tonic inhibition would cause. The latter can be done by altering the LEMS code describing the behaviour of the model's population rates, as described in the study of Abbott and Chance (2005), to also include a parameter that reduces the final firing rate by a certain percentage.

A more practical approach would be to simulate, under abnormal or pathological conditions, the multicompartmental model developed by Yao et al (2022), most likely using powerful computing infrastructure such as clusters. The complex model would be simulated with only the projection strength of the SST population reduced. This allows us to test whether the rate-based model can replicate the data generated by its multicompartmental counterpart under reduced activity of SST interneurons. Furthermore, this test can be repeated with different neuron populations, and even by enhancing population activity rather than reducing it, to gain a more comprehensive understanding of the model's ability to replicate the firing activity of its multicompartmental counterpart.

Both methods can offer valuable insight into our rate-based model's ability to replicate the behaviour and properties of the multicompartmental model developed by Yao et al, 2022. However, they are outside the scope of this investigation.

Conclusion

To summarise, our investigation leveraged the rate-based model of Garcia del Molino et al, 2017, to develop a rate-based model that can emulate the multicompartmental model of a human L2/3 microcircuit, developed by Yao et al, 2022. Overall, our model successfully reproduces the firing rate behaviour of its multicompartmental counterpart under standard physiological conditions and manages to emulate the general trend of the complex model firing rate behaviour under reduced SST inhibitory activity, used to assess the model under pathological conditions. However, the model's capability to accurately emulate the multicompartmental model's key properties and behaviours under abnormal or pathological conditions is still unknown. Furthermore, while the model's population rates successfully replicate the general behaviour of individual multicompartmental neurons, they fail to accurately emulate all the behaviours and properties of individual neurons, especially the more nuanced behaviours. Nevertheless, this model is still capable of emulating the overall behaviour and key properties of the multicompartmental model developed by Yao et al.

Moving forward in the future, this model can be used as an easily accessible and transparent tool for investigating the behaviour and properties of the human L2/3 cortical microcircuit, without requiring high-performance computing architecture, such as clusters. However, it should be noted that using this inherently abstract model for investigating the behaviour of individual cells or even populations is not recommended. Finally, it's important to highlight that rate-based models are on the opposite side of the scale of complexity compared to multicompartmental models. This means that rate-based models are inherently in how accurately they can emulate the behaviour of a much more complex multicompartmental model. Therefore, future efforts could focus on developing a point-neuron model to reproduce the behaviour of the model developed by Yao et al, 2022. This is because point-neuron models have greater complexity than rate-based models and can more accurately reproduce the many properties of multicompartmental models, whilst still being simpler and computationally less intensive, not requiring powerful computing infrastructure to simulate (Izhikevich, 2004).

References

Abbott, L.F., Chance, F.S., 2005. Drivers and modulators from push-pull and balanced synaptic input. *Progress in Brain Research*, 149, pp. 147–155.

Blundell, I., Brette, R., Cleland, T., Close, T., Coca, D., Davison, A., Diaz-Pier, S., Musoles, C.F., Gleeson, P., Goodman, D.F.M., Hines, M., Hopkins, M.W., Kumbhar, P., Lester, D.R., Marin, B., Morrison, A., Müller, E., Nowotny, T., Peyser, A., Plotnikov, D., Richmond, P., Rowley, A., Rumpe, B., Stimberg, M., Stokes, A.B., Tomkins, A., Trensch, G., Woodman, M., Epler, J.M., 2018. Code Generation in Computational Neuroscience: A Review of Tools and Techniques. *Frontiers*, 12(68).

Cannon, R.C., Gleeson, P., Crook, S., Ganapathy, G., Marin, B., Piasini, E., Silver, R.A., 2014. LEMS: a language for expressing complex biological models in concise and hierarchical form and its use in underpinning NeuroML 2. *Frontiers in Neuroinformatics*, 8(79).

Garcia del Molino, L.C., Yang, G.R., Mejias, J.F., Wang, X.J., 2017. Paradoxical response reversal of top-down modulation in cortical circuits with three interneuron types. *eLife*, 6.

Gleeson, P., Crook, S., Cannon, R.C., Hines, M.L., Billings, G.O., Farinella, M., Morse, T.M., Davison, A.P., Ray, S., Bhalla, U.S., Barnes, S.R., Dimitrova, Y.D., Silver, R.A., 2010. NeuroML: a language for describing data driven models of neurons and networks with a high degree of biological detail. *PLoS Computational Biology*, 6(6).

Hay, E., Hill, S., Schurmann, F., Markram, H., and Segev, I., (2011). Models of neocortical layer 5b pyramidal cells capturing a wide range of dendritic and perisomatic active properties. *PLoS Computational Biology*, 7(7).

Hines, M., Carnevale, T., McDougal, R.A. (2020). NEURON Simulation Environment. In: Jaeger, D., Jung, R. (eds) *Encyclopedia of Computational Neuroscience*. Springer, New York, NY.

Hines, M., Davison, A.P., Muller, E., 2009. NEURON and Python. *Frontiers in neuroinformatics*, 3(1).

Howard, D., Chameh, H.M., Guet-McCreight, A., Hsiao, H. A., Vuong, M., Seo, Y. S., Shah, P., Nigam, A., Chen, Y., Davie, M., Hay, E., Valiante, T. A., Tripathy, S. J., 2022. An in vitro whole-cell electrophysiology dataset of human cortical neurons. *GigaScience*, 11.

Izhikevich, E.M., 2004. Which model to use for cortical spiking neurons? *IEEE transactions on neural networks*, 15(5), pp. 1063-1070.

NeuroML Editorial Board, 2024. *NeuroML – NeuroML Documentation*. [online] Available at: <https://docs.neuroml.org> [Accessed 4 March 2024].

Pakan, J.M.P., Lowe, S.C., Dylida, E., Keemink, S.W., Currie, S.P., Coutts, C.A., Rochefort, N.L., 2016. Behavioral-state modulation of inhibition is context-dependent and cell type specific in mouse visual cortex. *eLife*, 5.

Rall, W., 1962. Theory of physiological properties of dendrites. *Annals of the New York Academy of Sciences*, 96(4), pp. 1071-1092.

Rudy, B., Fishell, G., Lee, S., Hjerling-Leffler, J., 2011. Three groups of interneurons account for nearly 100% of neocortical GABAergic neurons. *Developmental Neurobiology*, 71(1), pp. 45–61.

Sinha, A., Gleeson, P., Marin, B., Dura-Bernal, S., Panagiotou, S., Crook, S., Cantarelli, M., Cannon, R.C., Davison, A.P., Gurnani, H., Silver, A., 2023. The NeuroML ecosystem for standardized multi-scale modeling in neuroscience. *bioRxiv*.

Trappenberg, T.P., 2022. *Fundamentals of Computational Neuroscience*. 3rd ed. Oxford: Oxford Academic.

Van Geit, W., Gevaert, M., Chindemi, G., Rössert, C., Courcol, J.-D., Muller, E.B., Schurmann, F., Segev, I., and Markram, H. (2016). BluePyOpt: leveraging open-source software and cloud infrastructure to optimise model parameters in neuroscience. *Frontiers in Neuroinformatics*. 10(17).

Vella, M., Cannon, R.C., Crook, S., Davison, A.P., Ganapathy G, Robinson HPC, Silver RA, Gleeson P, 2014. LibNeuroML and PyLEMS: using Python to combine procedural and declarative modeling approaches in computational neuroscience. *Frontiers in Neuroinformatics*, 8(38).

Yao, H.K., Guet-McCreight, A., Mazza, F., Chameh, H.M., Prevot, T.D., Griffiths, J.D., Tripathy, S.J., Valiante, T.A., Sibille, E., Hay, E., 2022. Reduced inhibition in depression impairs stimulus processing in human cortical microcircuits. *CellPress*, 38(2).