Neuroscience Gateway Development

Authors: Erik Guetz, Shrivathsav Seshan, Vasu Vikram
Mentors: Amit Majumdar, Subhashini Sivagnanam
Collaborated With: Anita Bandrowski

Abstract

Using Python and the Neuroscience Gateway (NSG) Portal, we were able to efficiently run neuron simulations on 50+ models hosted on ModelDB and the Neuroscience Information Framework (NIF) as well as analyze the outputs of these simulations to verify the integrity of these models. With frequent updates and patches being implemented on the Trestles cluster, periodically checking whether the functionality of the existing 887 models remains is required. After compiling our simulation data, we began to see that many of the models launch various Graphical User Interfaces (GUIs) and plotting programs (Fig 1) that are not compatible with the Trestles Cluster. Twelve of the models that we tested successfully produced readable and savable outputs.

Introduction:

The goals of this project were to devise a way to provide feedback for ModelDB regarding the integrity of the models that are currently hosted, isolate models that work "as is" on Trestles, and make the necessary code fixes to those that don't work "as is".

- What is the NSG Portal?

The NSG Portal is a web tool that allows anyone around the world to easily set the parameters of and run neuronal simulations on Trestles.

- What is ModelDB?

ModelDB is a database of computational neuron simulations that is included in NIF's corpus of neuroscience data.

- What is the NEURON Software Package?

The Neuron software package allows users to build or code their own neuron models and to run neuron simulations.

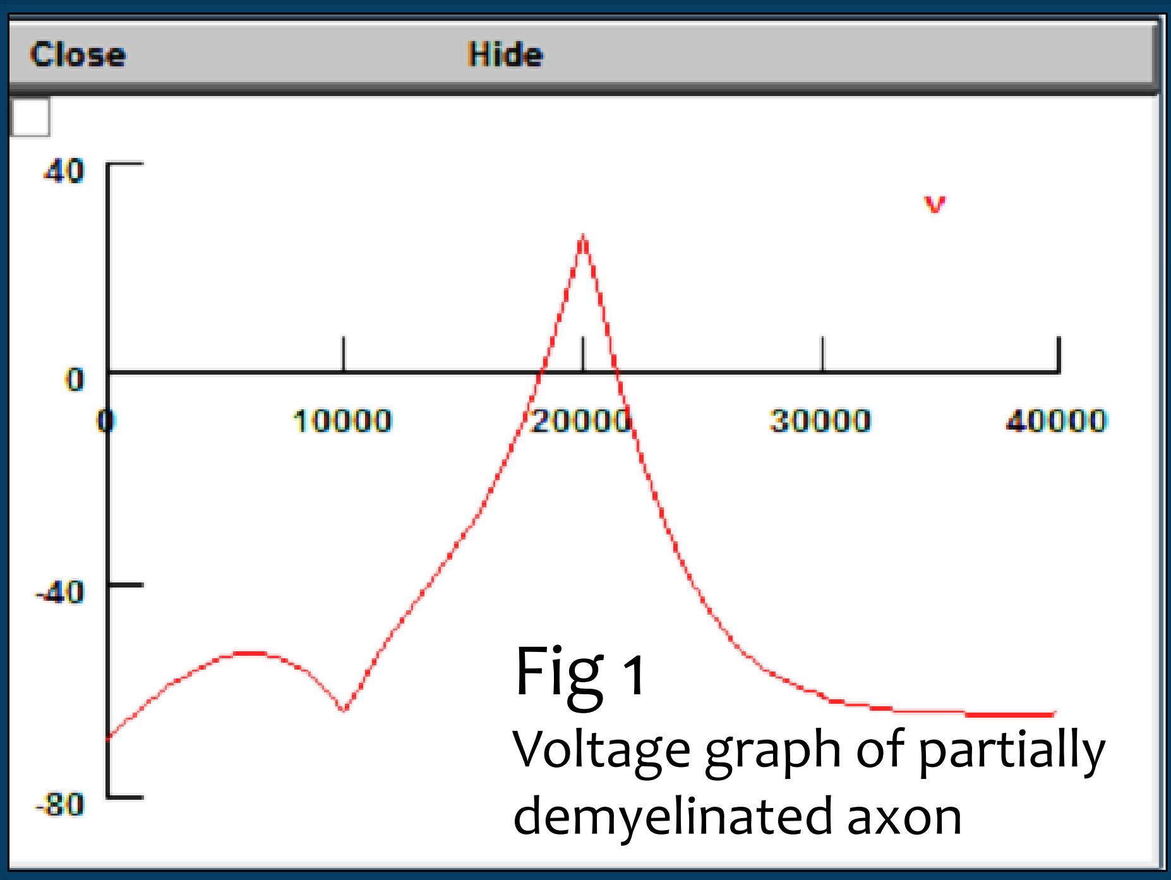
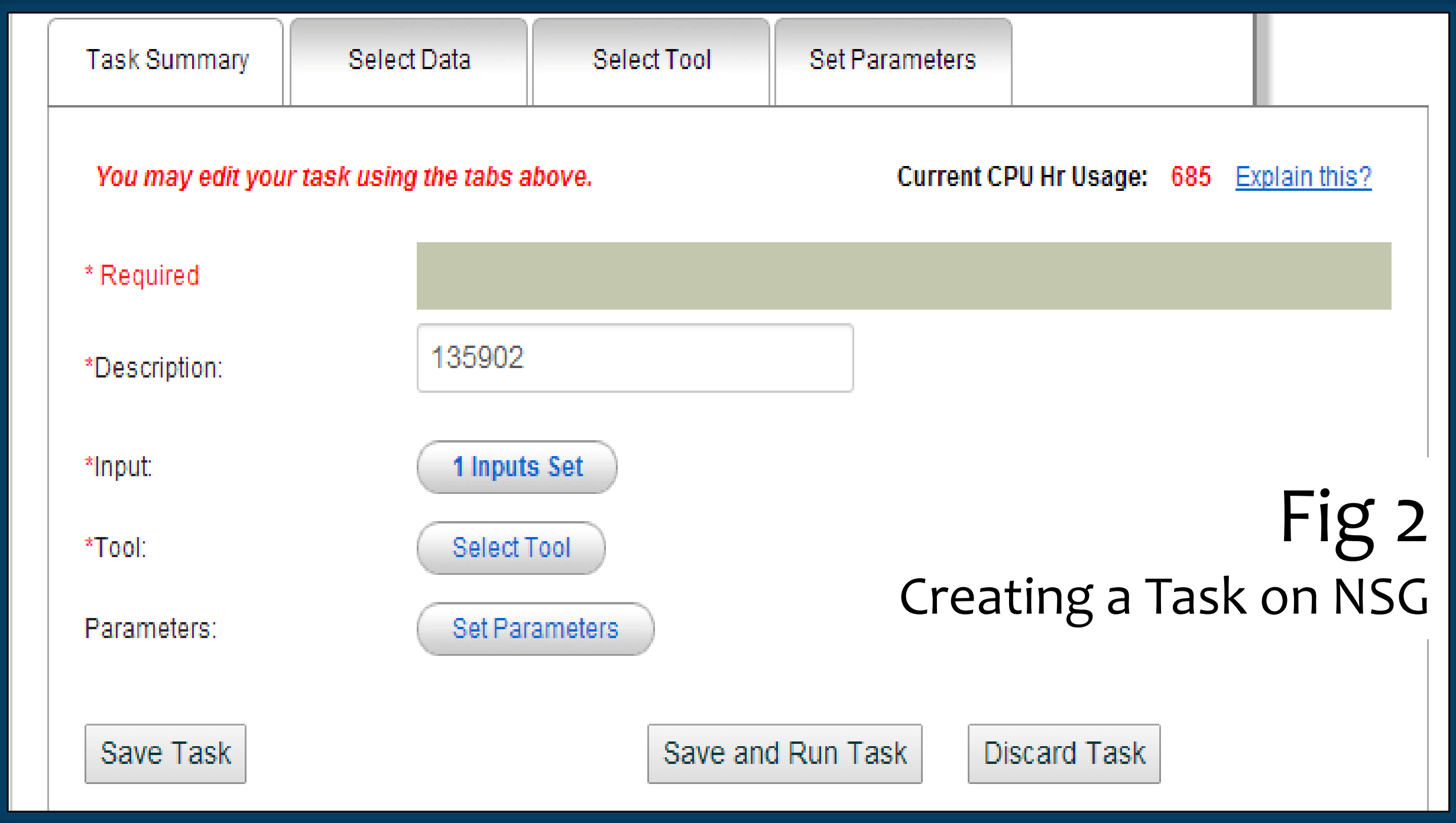


Fig 1
Voltage graph of partially demyelinated axon



You may edit your task using the tabs above. Current CPU Hr Usage: 685 Explain this?

* Required

*Description: 135902

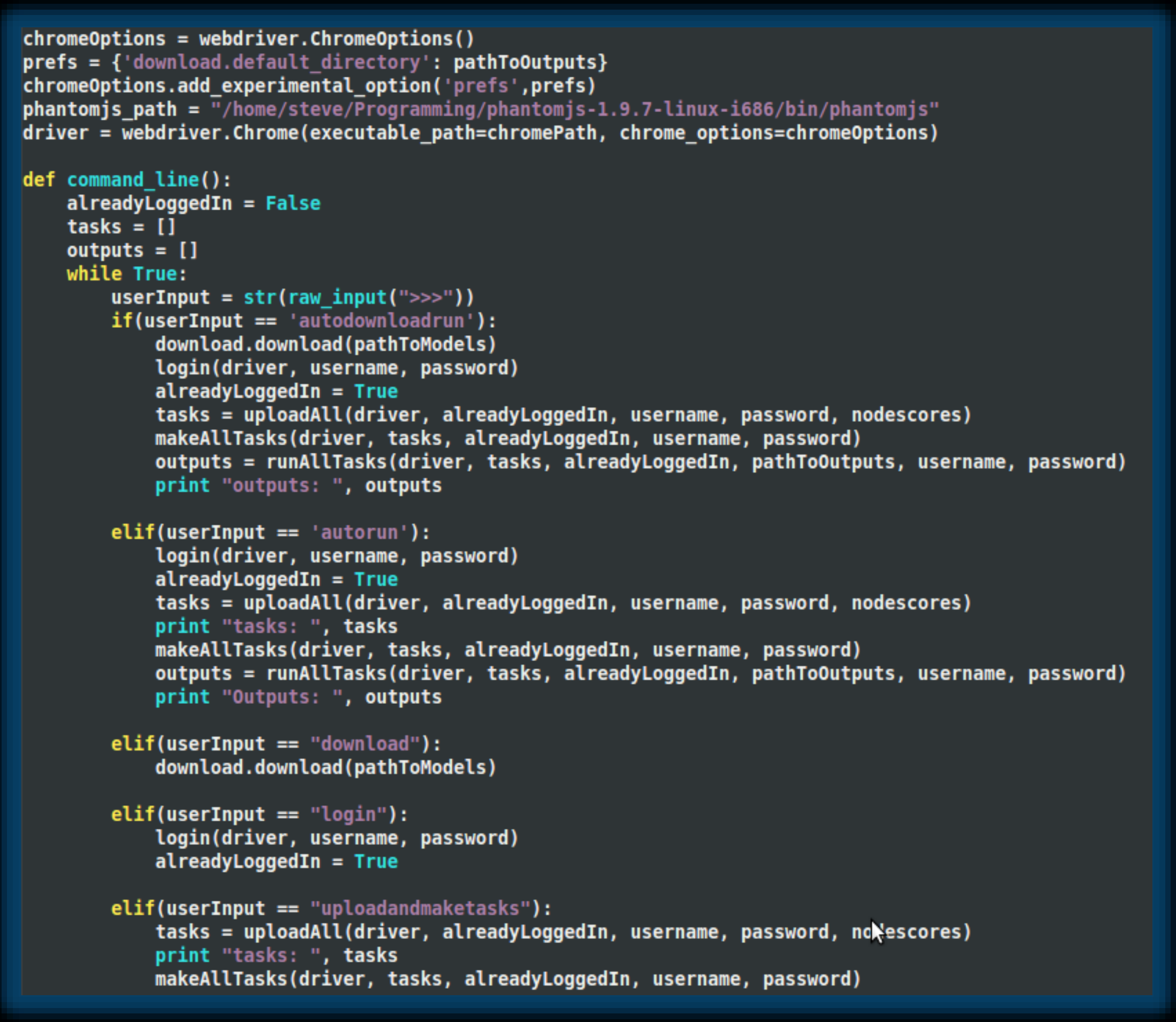
*Input: 1 Inputs Set

*Tool: Select Tool

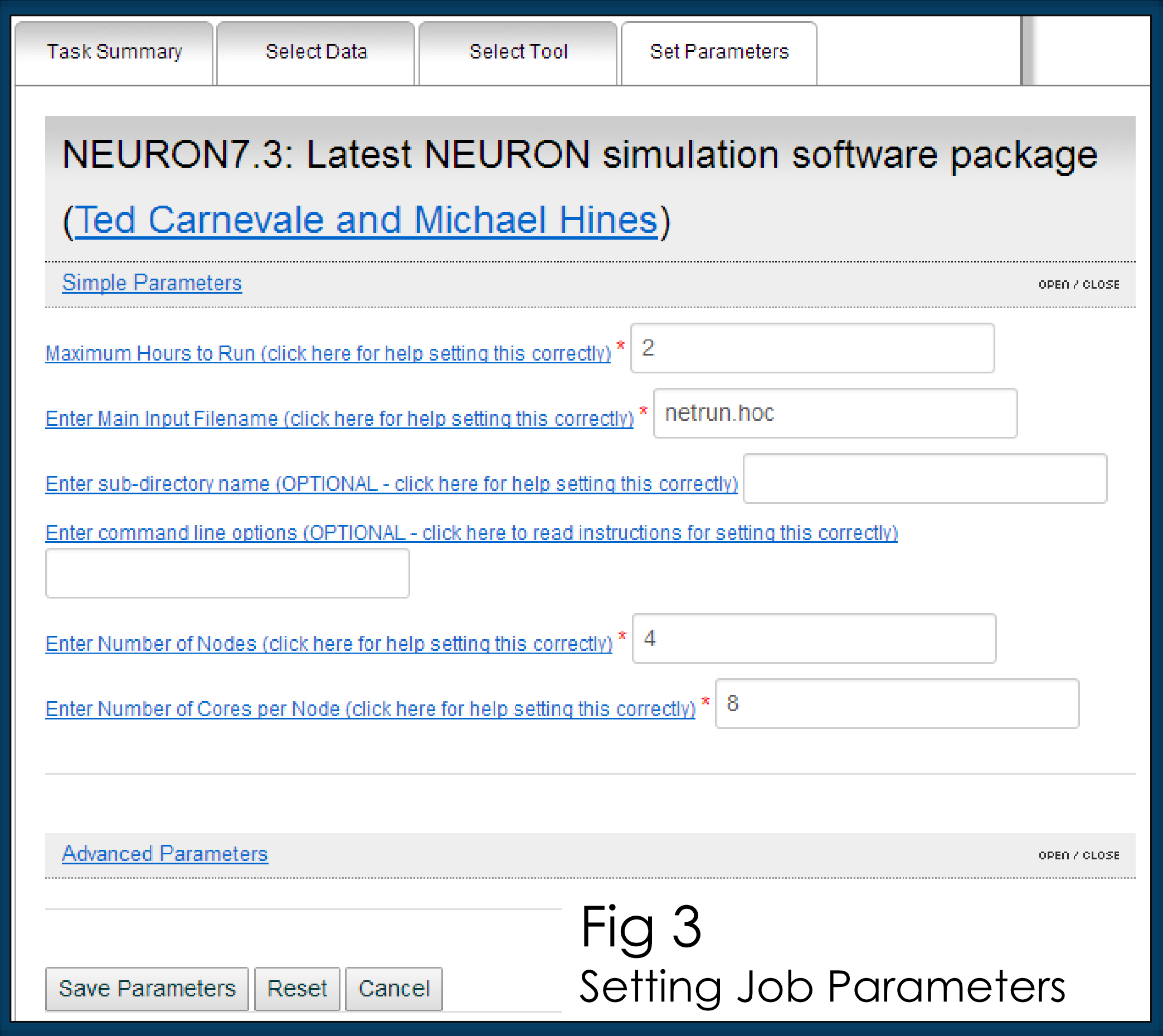
Parameters: Set Parameters

Save Task | Save and Run Task | Discard Task

Fig 2
Creating a Task on NSG



```
chromeOptions = webdriver.ChromeOptions()
prefs = {'download.default_directory': pathToOutputs}
chromeOptions.add_experimental_option('prefs',prefs)
phantomjs_path = "/home/steve/Programming/phantomjs-1.9.7-linux-i686/bin/phantomjs"
driver = webdriver.Chrome(executable_path=chromePath, chrome_options=chromeOptions)

def command_line():
    alreadyLoggedIn = False
    tasks = []
    outputs = []
    while True:
        userInput = str(raw_input(">>>"))
        if(userInput == 'autodownloadrun'):
            download.download(pathToModels)
            login(driver, username, password)
            alreadyLoggedIn = True
            tasks = uploadAll(driver, alreadyLoggedIn, username, password, nodescores)
            makeAllTasks(driver, tasks, alreadyLoggedIn, username, password)
            outputs = runAllTasks(driver, tasks, alreadyLoggedIn, pathToOutputs, username, password)
            print "outputs: ", outputs

        elif(userInput == 'autorun'):
            login(driver, username, password)
            alreadyLoggedIn = True
            tasks = uploadAll(driver, alreadyLoggedIn, username, password, nodescores)
            print "tasks: ", tasks
            makeAllTasks(driver, tasks, alreadyLoggedIn, username, password)
            outputs = runAllTasks(driver, tasks, alreadyLoggedIn, pathToOutputs, username, password)
            print "Outputs: ", outputs

        elif(userInput == "download"):
            download.download(pathToModels)

        elif(userInput == "login"):
            login(driver, username, password)
            alreadyLoggedIn = True

        elif(userInput == "uploadandmaketasks"):
            tasks = uploadAll(driver, alreadyLoggedIn, username, password, nodescores)
            print "tasks: ", tasks
            makeAllTasks(driver, tasks, alreadyLoggedIn, username, password)
```



NEURON7.3: Latest NEURON simulation software package (Ted Carnevale and Michael Hines)

Simple Parameters OPEN / CLOSE

Maximum Hours to Run (click here for help setting this correctly) * 2

Enter Main Input Filename (click here for help setting this correctly) * netrun.hoc

Enter sub-directory name (OPTIONAL - click here for help setting this correctly)

Enter command line options (OPTIONAL - click here to read instructions for setting this correctly)

Enter Number of Nodes (click here for help setting this correctly) * 4

Enter Number of Cores per Node (click here for help setting this correctly) * 8

Advanced Parameters OPEN / CLOSE

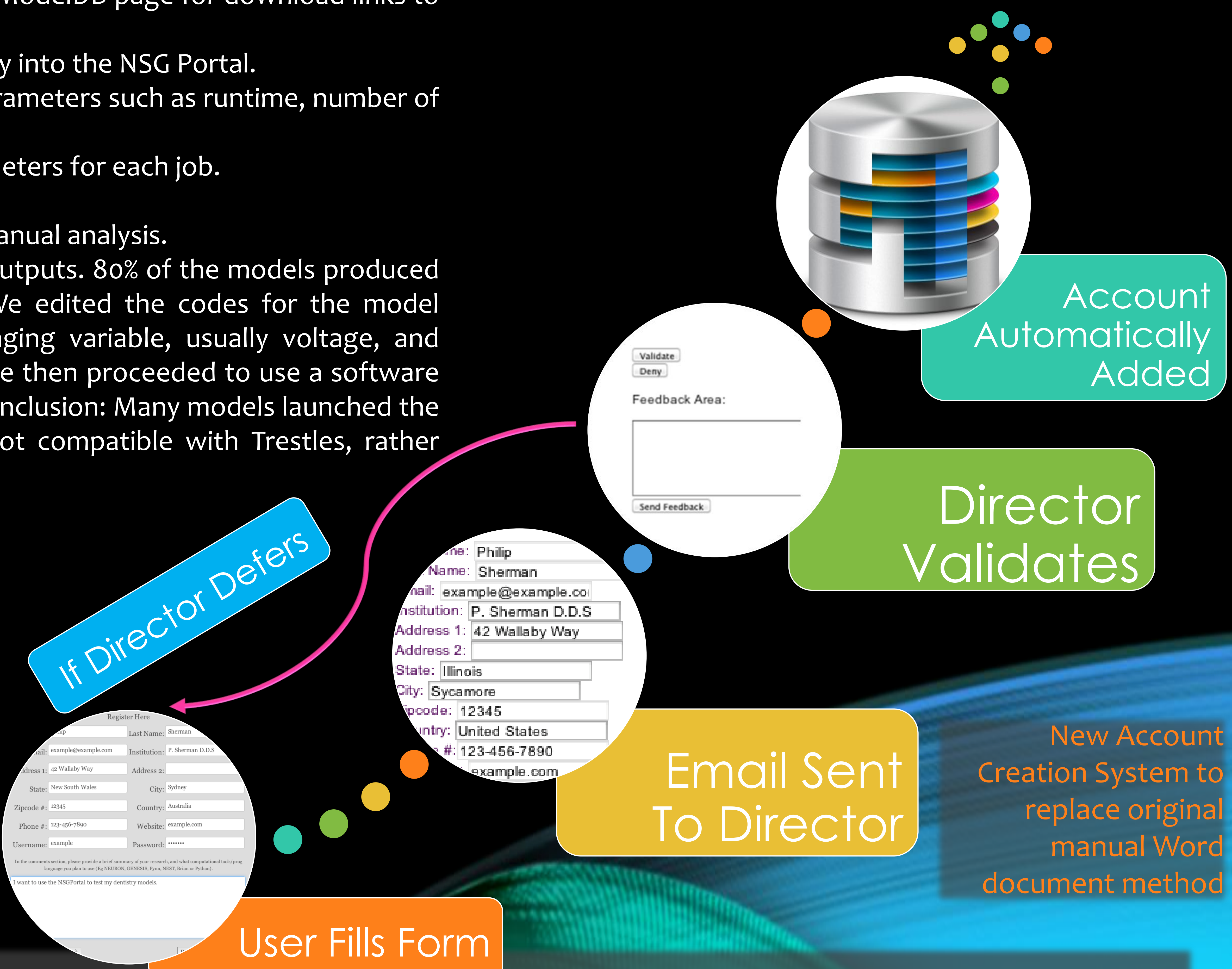
Save Parameters | Reset | Cancel

Fig 3
Setting Job Parameters

Story

Our python automation scripts were able to:
1. Scrape the NIF Integrated Models page and the ModelDB page for download links to neuron model zip files.
2. Upload all the model files from a certain directory into the NSG Portal.
3. Create a task for each file with user specified parameters such as runtime, number of nodes, cores, and main input file.
4. Parse existing CSV files containing default parameters for each job.
5. Run the jobs through the portal.
6. Download the output files from the portal for manual analysis.

We compared our outputs with NIF specified outputs. 80% of the models produced no new files or contained errors in the code. We edited the codes for the model authored by Alle et al (2009) to take the changing variable, usually voltage, and produced a new file with the newly made data. We then proceeded to use a software called VPython to visualize the data on a graph. Conclusion: Many models launched the NEURON GUI and produced graphical outputs not compatible with Trestles, rather than desired file outputs.



Register Here

Last Name: Sherman
Email: example@example.com | Institution: P. Sherman D.D.S
Address 1: 42 Wallaby Way | Address 2:
State: New South Wales | City: Sydney
Zipcode #: 12345 | Country: Australia
Phone #: 123-456-7890 | Website: example.com
Username: example | Password: •••••••

In the comments section, please provide a brief summary of your research, and what computational tools/prog language you plan to use (Eg NEURON, GENESIS, Pynn, NEST, Brian or Python).

I want to use the NSGPortal to test my dentistry models.

Philip
Name: Sherman
Email: example@example.com
Institution: P. Sherman D.D.S
Address 1: 42 Wallaby Way
Address 2:
State: Illinois
City: Sycamore
Zipcode: 12345
Country: United States
Phone #: 123-456-7890
example.com

New Account Creation System to replace original manual Word document method

To the NSG Portal we added:

- Automatic Account Creation System
- User Feedback Form
- Improved FAQ and Help Pages

Summary and Future Work:

Our work represents the first step in completely formatting and parallelizing all the models on ModelDB to work with Trestles. Corpuses like NIF and ModelDB and similar neuronal databases on Yale University's SenseLab are valuable to computational neuroscientists, but without compatibility with supercomputers like Trestles, these models are limited in their potential. In the future, with thorough recordkeeping of simulation outputs, we can systematically edit the code of each model on ModelDB for supercomputer compatibility, thus allowing neuroscientists complete access to large scale computing resources.