Experiences from a Multi-disciplinary Course Sequence Development on Cyber and Software Automation in Neuroscience

Prasad Calyam and Satish S. Nair

Department of Electrical Engineering and Computer Science, University of Missouri-Columbia, USA
Email: {calyamp, nairs}@missouri.edu

Abstract— Neuroscientists are increasingly relying on parallel and distributed computing resources for analysis and visualization of their neuron simulations. This requires expert knowledge of programming and cyberinfrastructure configuration, which is beyond the repertoire of most neuroscience programs. This paper presents early experiences from a one-credit graduate research training course titled ECE 8001 "Software and Cyber Automation in Neuroscience" at the University of Missouri for engendering multi-disciplinary collaborations between computational neuroscience and cyberinfrastructure students and faculty. Specifically, we discuss the course organization and exemplar outcomes involving a next-generation science gateway for training novice users on exemplar neuroscience use cases that involve using tools such as NEURON and MATLAB on local as well as Neuroscience Gateway resources. We also discuss our vision towards a course sequence curriculum for graduate/undergraduate students from biological/psychological sciences and computer science/engineering to jointly build "self-service" training modules using Jupyter Notebook platforms. Thus, our efforts show how we can create scalable and sustainable cyber and software automation for fulfilling a broad set of neuroscience research and education use cases.

Index Terms— Computational Neuroscience, Cyberinfrastructure Development, Science Workflow Automation, Jupyter Notebooks, HPC Training Modules

I. INTRODUCTION

Research and training in neural science and engineering are increasingly dealing with analysis of voluminous multimodal data and involve tackling multi-disciplinary problems [1]. The NSF iNeuro report [2] summarizes related issues as: access to multi-omics data archives, to heterogeneous software and computing resources, and to multi-site interdisciplinary expertise in e.g., engineering, biology and psychology. Parallel and distributed computing resources available in the form of cyberinfrastructure (CI) resources such as Neuroscience Gateway (NSG) [3] enables ready access for analysis and visualization of large data sets from simulation experiments. However, to completely utilize the full range of CI capabilities for bold new knowledge discovery in neuroscience, neuroscientists with limited CI skills need training with the best practices of High Performance Computing (HPC) and large-scale data management. Thus, there is a need for collaborative state-of-the-art training programs and multi-disciplinary course sequences that benefit researchers and educators in the important area of neural science and engineering.

To address this need, a fruitful collaboration between computational neuroscience and CI experts has resulted in a funded-project from the National Science Foundation (OAC-1730655; 09/17 – 08/20) to develop innovative CI-based training modules to advance research in data-intensive neuroscience. Computational neuroscience researchers, undergraduate and graduate students and teachers are the three targeted communities in the project. To develop the training modules, we have initiated a first-of-a-kind course sequence starting with a 1-credit graduate research training course titled ECE 8001 "Software and Cyber Automation in Neuroscience". The training modules are being designed to result in a set of "self-service" laboratory exercises featuring prototypes of on-demand tools that can be integrated with existing CI resources (i.e., institutional CI or external CI such as NSG) and CI technologies to support novice users with limited CI skills. The 1-credit course was first offered in Fall 2017, with a follow-on in Spring 2018 as a companion to an existing graduate-level Cloud Computing course sequence. The course is continuing to be developed to cross-train students from biological/psychological sciences and computer science/engineering in an interdisciplinary setting. Based on the training material outcomes, we are in the process of deploying them broadly e.g., in an undergraduate course in neuroscience at MU, and in similar courses at our partner 2- and 4-year colleges.

In this paper, we present our early experiences from the course content organization and student training outcomes of the first 1-credit course offering over a two-semester period. The course content organization was performed under five focus areas: (i) extensive survey to prioritize CI user requirements for data-intensive neuroscience research and education use cases, (ii) user interface portal design for easy job submission to publicly available CI resources, (iii) CI templates to abstract details of workflow resource configuration and performance monitoring objectives, (iv) lab exercises development based on (i) – (iv) activity outcomes, as well as their delivery with CI tools integration and related documentation creation.

We also present exemplar/preliminary outcomes of the course students' work to demonstrate the potential impact of our course offering. Specifically, we present a students-developed lab exercise involving the development of a Neuro module using NSG CI resources. The lab exercise involves cyber and software automation via a next-generation science gateway (with novel features such as e.g., an expert chatbot) to simulate the following single cell models: spiking/bursting

cell, synaptic transmission - earthworm example, two-cell half-center oscillator model - using Python and software of their choice (e.g., NEURON, GENESIS, NEST). In addition, we present another lab exercise that involves a MATLAB-based image processing automation workflow that relates to a Zebrafish branchiomotor neuron analysis. On the CI side, the lab exercise involved providing access to HPC resources via use of RESTful web services of NSG and University of Missouri (MU) CI that allowed for tracking of job status, execution time estimates and error conditions. Further, students proposed a design for integration of frameworks such as CIPRES [4] and Jupyter Notebooks [5] for scalable and sustainable CI services to handle workflows of exemplar neuroscience use cases across local institution HPC and NSG sites. These course efforts are thus guiding our future plans to develop a multi-disciplinary course sequence that engenders integration of both neuroscience and CI concepts/tools to solve meaningful neuroscience problems with distributed HPC resources.

The reminder of the paper is organized as follows: Section II presents the needs and challenges for cross-training students in neuroscience and CI. Section III describes our course organization. Section IV presents the students-developed laboratory exercise outcome. Section V summarizes results from a user survey of the course participants. Section VI concludes the paper.

II. CROSS-TRAINING IN NEUROSCIENCE AND CYBERINFRASTRUCTURE

Computational neuroscience research has been hampered because of limited sharing of data and tools within the community. Labs pursue their individual research independently, distribute their research only via journal articles, and are forced to reinvent the same computing pipelines necessary for their analysis. Due to almost a non-existent sharing platform for computational neuroscience teaching and research, the programming is usually difficult to reproduce due to poor coding skills, lack of version control and manual implementation of certain tasks [6] [7]. Access to HPC resources and efficient handling of large datasets for large-scale processing loads is also a significant limiting factor. Reserving sufficient machine hours at HPC centers, learning the complex user interfaces to HPC data centers, and managing the submission of jobs and the retrieval of results are the major hurdles to utilize available CI resources at local institutions or external public sites [3] [8].

There are a number of approaches to mitigate these issues. Libraries and frameworks aim to ease the programming tasks, and in the case of `psychopy_ext` [9], one can increase code quality. A reliance on open-source software is viewed as a way to increase reproducibility and reusability [6] [7] [10]. Pipeline environments such as LONI [11] address issues of reproducibility by allowing an easy linkage of executables and the persistence of pipeline definitions. Some pipelining tools also have additional features for provenance tracking or the ability to detect redundancy in analyses, also increasing reproducibility and reusability [12]. Pipelining tools are interfaced to high-throughput computing infrastructures, but these are often local to an institution. Science Gateways have been and are being developed to provide community

access to shared CI resources and to manage technical and administrative challenges on behalf of the researchers. NSG [3] and CBrain [8] are amongst the popular CI sites that provide such services. Workflow management systems to support application pipelines are usually integrated with easy-to-use user interfaces that allow users to customize and handle their large-scale processing and data management needs [12].

Secure access of CI resources is a concern; in both NSG and CBrain, individual user accounts track usage and access, assuring that users submitting code are properly vetted. Some recent science gateway projects aim to provide access to distributed cloud resources in addition to leveraging resources from national HPC data centers. Projects such as "Science in the Cloud" (SIC) [13] use Docker containers, open source software, and automated deployment for reproducibility. Other projects such as MASSIVE provide access to specialized instrumentation in addition to access to HPC or cloud resources [10].

All these approaches are helpful, serving different audiences and purposes. However, more training efforts such as ours are needed to specifically guide novice users to better leverage distributed CI resources, and also help expert programmers to efficiently and rapidly scale their workflows. Our focus is to combine best practices in providing programming frameworks, pipeline environments and workflow management systems, and create opportunities for automation across cloud infrastructures and access to specialized instrumentation. The novelty in our efforts is in developing training modules that provide a guided experience for novice users, mediating access to HPC resources, and allowing the reuse/repurpose of prior successful job execution based CI templates. The guided experience within our next-generation science gateway design is achieved through collection of user requirement abstractions at pipeline and workflow levels in order to allow persistence of parameters and files in a generalizable way through the use of CI templates. In addition, the user requirements collection and customization of CI resource capabilities is facilitated through user interfaces with dynamic forms and context-aware chatbots. Table I shows the longer-term goal that our project aims to accomplish in the course sequence development in terms of neuro and CI modules/exercises that caters to a broad set of neuroscience research and education use cases.

III. COURSE ORGANIZATION

In this section, we present the motivations for our course sequence development based on our prior efforts, and a sample of the pent-up neuroscience use cases that could benefit from training on cyber and software automation.

A. Prior Computational Neuroscience Training Efforts

Neuroscientists have made immense progress (as illustrated in Figure 1) in characterizing neural phenomena at specific scales ranging from genomic, molecular, cellular, circuits, systems and behavioral levels, and data continues to accumulate at an astounding pace at all these levels. However, understanding how information processing integrates across levels, and ultimately generates behavior poses a huge challenge presently [14]. NSF *'Understanding the*

Examples of Neuro Modules/Exercises	Examples of CI Modules/Exercises
- Using software of their choice (NEURON, GENESIS, NEST, etc.) to develop software to simulate the following single cell models: spiking/bursting cell, synaptic transmission– earthworm example, two-cell half-center oscillator model - Using software of their choice to develop software to simulate the following network models: two-cell half-center oscillator model of fish swimming; 100-cell model of how fear is learned in the mammalian amygdala - Analysis of multiple neural signals to detect oscillations using MATLAB and other packages - Machine learning examples in neuroscience, e.g., calculation of local field potential (LFP) from neuronal spiking data - Develop simplified software modules of exercises for distribution to 2- and 4-year colleges via web; similarly develop simplified CI-enabled modules for K-12 usage	- Use of workspaces featuring virtual desktops that host tutorials with supporting software, enabling 'democratized' and easy access, including via hand-held devices 'Chaining' different software programs using web services for parameter sweep studies - Interfacing with databases such as NeuronDB to transfer and analyze distributed data sets - Providing access to HPC resources via user interfaces of NSG and MU that allow tracking of job status, execution time and error conditions - Meta data management, data sharing, and data access control mechanisms - Understanding network configurations that foster collaboration between experimental and computational neuroscientists across geographically distributed campuses

Brain initiative' [1] emphasizes the importance of computational tools in addressing this challenge through this excerpted quote - "... computational infrastructure, theory, and models that will accelerate the integration of knowledge across experimental scales from atomic to behavioral; across multiple science, engineering, and computational disciplines; and across species and lifespans."

Preliminary Results of an Integrative Education Theme: Our prior project efforts began with the process of filling the gap in effective cross-training for scientists from biological, psychological and physical sciences at the undergraduate level. Specifically, our team developed a novel interdisciplinary and integrative undergraduate course in computational neuroscience (ECE/BioSci 4580 Computational Neuroscience), with both software (computer models with interactive interface) and hardware (wet-lab) experiments. This course has been successfully cross-training undergraduates from the colleges of engineering and arts and science for the past eight years [15] [16]. We have developed the novel integrative theme "function-biology-model-math" to determine what function is being implemented, determine how biology implements it, model it in circuit terms, and finally write down the math representation of the model. In the process of this course development, we organized the content for cross-disciplinary courses that necessitate novel learning paradigms for all concerned - undergraduates, faculty, and administrators [17].
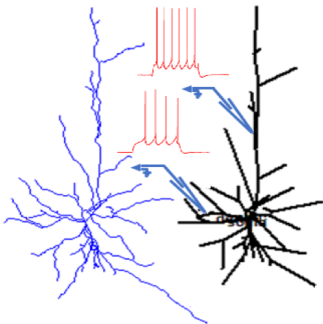

Fig. 1.   Neuron single cell model from a public database

Software Experiments in ECE/BioSci/BE 4580 Computational Neuroscience: Students in this popular course

(campus-wide enrollments routinely exceed 30 students) explore neuroscience fundamentals from a theoretical, mathematical or experimental perspective using 6 interactive software experiments and 6 hardware experiments, of which two sample software experiments are briefly described below:

Software Experiment-1 – How do we model a passive membrane? A cell membrane can be modeled as an electrical circuit with capacitance (lipid bi-layer) and resistance (various channel proteins embedded in it) as illustrated in Figure 2. Application of Ohm's law to this simple circuit yields a first order equation, and this is used to emphasize first order dynamics and the concept of time constant. This passive cell membrane equation is 'solved' to determine the evolution of the membrane potential as a function of time with a constant current injection. After a brief review of voltage gated channels, students are challenged to predict what channels might be needed to generate an "action potential" in the membrane.

Software Experiment-2 – How does a neuron spike? This interactive lesson we developed focuses on the ionic basis of a neuron action potential, including voltage-dependent (active) and voltage-independent (passive) currents. Detailed current equations are assembled into a spiking neuron model, enabling the students to understand the model development process, in addition to parametric explorations.

Building upon these software experiments, we are continuing to develop simplified neuroscience tutorials for faculty who are involved in teaching undergraduate neuroscience curriculum around the nation, and delivering the tutorials via a popular summer workshop we have been conducting for them since 2007.

B. NeuroScience Use Cases for New Courses Development

In order to inform our new course sequence development, a survey was conducted at MU amongst multiple neuro-related disciplines to determine the desired neuroscience data-intensive analysis use cases that would benefit from cyber and software automation. Survey was sent to 50+ neuroscience researchers and educators on MU campus spread over 10 departments and 5 Colleges. Six use cases were identified from survey data analysis: Neuron Simulation, Zebrafish Branchiomotor Neuron Analysis, RNA Sequencing

for Neural Single Cell Modeling, fMRI, WCGNA, and 2 Photon Calcium Imaging. There are a few MRI pipelines such as LONI that give researchers access to cloud platform resources, but the other use cases we have identified do not appear to have developed high-performance pipelines leveraging distributed CI resources. Therefore, we believe that our cyber and software automation efforts for these use cases has the potential to create benefits impact for researchers and educators in related areas.

The Neuron Simulation use case had two model problems that are as follows: (i) a single-neuron model, and (ii) a neuron network model. Our current work has developed a simplified version of the NEURON Single-cell simulation (see Figure 1) from data collection through execution on HPC resources. The NEURON Network simulation (see Figure 2) has been partially implemented, and is being used for the new course sequence development efforts.

1) Neural Network Simulation: The current pipeline for the Neuron Network simulation on the MU Lewis cluster involves significant manual file editing. First, the user updates the model parameters contained in an excel file, composed of several tabs. This is bundled up with all the necessary execution files and sent to the cluster. Three processing steps occur on the cluster: (i) The model is generated from the parameters in the spreadsheet, (ii) The model is executed using the generated files from step 1, and (iii) The output data are checked and saved. The results can then be displayed and output to the user.
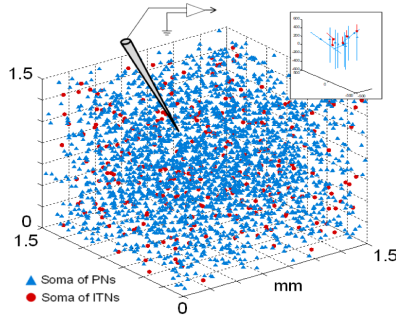

Fig. 2.   Neuron Network Model of 27,000 Neurons

2) Neuron Single Cell Simulation: This workflow is simpler in comparison to Network Neuron Simulation, however it does not need complex data inputs. This workflow needs three groups of data to run. The first set of questions include the geometry of the neuron cell and the lengths and diameters of soma, axon and dendrites. The second set of questions include the different channels in which the user would like to run his/her analysis on. The third set of questions are about the type of experiment and parameters relating to the experiment.

3) Zebrafish Branchiomotor Neuron Analysis: This use case involves zebrafish larval jaw movement analysis to demonstrate the workings of the branchiomotor neuron present in the brain circuits of vertebrates. The gape or mouth opening analysis as shown in Figure 3 is used for video microscopy to perform fast, accurate and automated image analysis for identification of the jaw movement [18]. Using Region-of-Interest (ROI) information and a custom MATLAB-based video annotation tool, gape frequency is

extracted from video sequences comprising of thousand frames of different individuals. Based on the number of experiments, duration and other properties of each file, the storage requirements can reach up to hundreds of terabytes to successfully explore just two different ages and 10 genotypes of the specimen. The large storage requirement to conduct meaningful experimentation is a strong motivation for the development of cloud transformation solution and was developed as a lab exercise during the second course offering.

4) RNA Sequencing for Neural Single Cell Modeling: RNA sequencing (RNA-Seq), or whole transcriptome sequencing, uses next-generation sequencing (NGS) to reveal the amount of RNA in a biological sample, to reveal the changing cellular transcriptome. Specifically, RNA-Seq enables viewing alternative gene spliced transcripts, post-transcriptional modifications, gene fusion, mutations/SNPs and changes in gene expression over time, or differences in gene expression in different groups or treatments. A workflow for this use case has been setup that requires distributed CI resources and a portal for experimentation with different parameter combinations.

IV. Exemplar Course Outcomes

The first offering of the new 1-credit course as a companion to an existing graduate-level Cloud Computing course so far has resulted in a next-generation science gateway (i.e., our NEURON Science Gateway) architecture for Neuron simulations that is described in detail in Section IV-A. It has also resulted in use cases gathering via a pent-up needs survey to identify the needs of researchers and educators. The students decided to choose the Neuron Simulation use case for showcasing an end-to-end user interface demonstration of a next-generation science gateway, as described in detail in Section IV-B. Lastly, the students have conducted an extensive literature survey to start identifying the specifications for a federated multi-omics cloud analytics 'self-service' training platform. As explained in Section IV-D, the students explored the idea of using open source, and easy-to-use CI templates that can be *reused* and *re-purposed* for a variety of use cases.

In the second semester, the expanded focus for our next-generation science gateway involved investigating the existing state-of-the-art CI technologies used in NSG and other large-scale training platforms such as e.g., CIPRES [4] workflow manager and Jupyter Notebooks [5]. Our goal was to integrate these technologies within our science gateway development to experiment with capabilities that launch on-demand computational neuroscience simulations for educational and research purposes in a scalable and seamless manner on local institutional HPC resources or at NSG. Together with the companion cloud computing class, students have developed and deployed cyber and software automation solution designs that involve tools such as NEURON and MATLAB for the use cases detailed in Section III-B.

From course sequence organization perspective, the first two course offerings have provided us confidence to initiate the development of a new 3-credit course that addresses neuroscience and CI modules/exercises listed in Table I at an undergraduate level. This new course will prepare students from multiple disciplines at both the graduate and undergraduate levels to continue with the advanced CI courses on
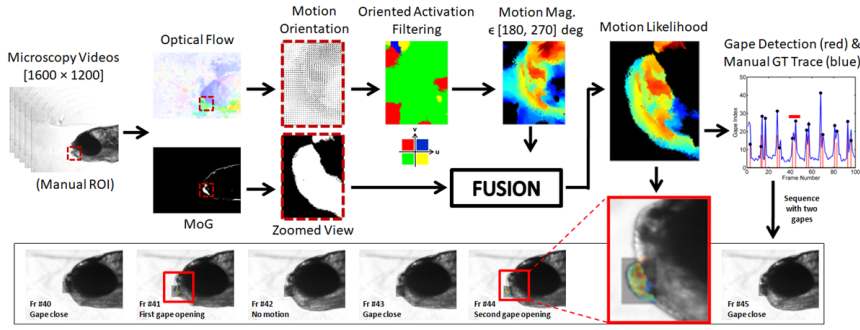
Fig. 3. Pipeline for high-throughput image analytics for Zebrafish larval jaw movement and gape frequency analysis using motion estimation and fusion.
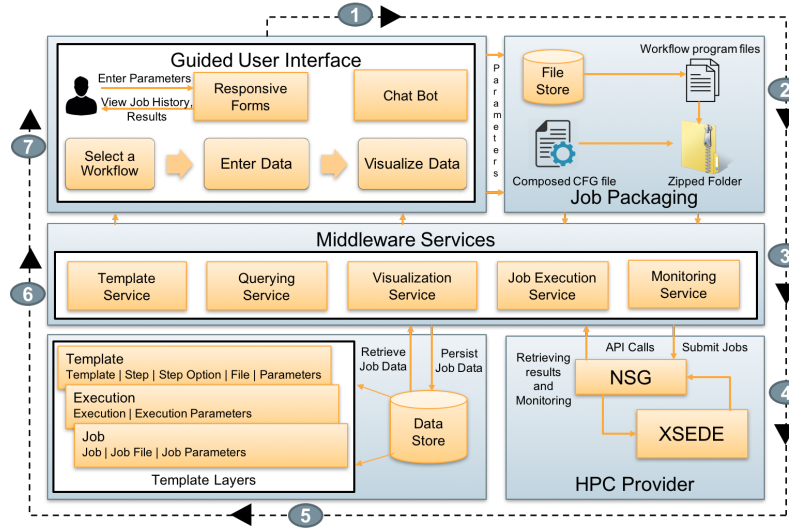

Fig. 4. Architecture of our science gateway developed by graduate students during the first course offering.

Cloud Computing and Big Data Analytics offered by one of the co-authors. Thus, we hope to create a natural pipeline for training in advanced CI principles and design/deployment targeted to solve the challenges within data-intensive neuroscience. As a matter of fact, we believe that the same training concepts could be extended/applied to address challenges in other similar fields such as Bioinformatics or Imaging with data-intensive workflows requiring distributed CI resources.

A. Science Gateway Architecture

Community web portals (i.e., science gateways) to build simulation workflows have attained moderate success in neuroscience. Typically, it requires in-depth expert knowledge to provide the pipeline scripts and input files in addition to having advanced programming skills and a detailed understanding of the pipeline. To overcome these shortcomings, the students proposed and implemented a Science Gateway architecture as shown in Figure 4, which currently is composed of three logical components: the *Guided User Interface* module, the *Middleware Services* module, and the *Workflow Manager* module. The initial users were assumed to be members of the neuroscience community at MU, who include both novice and expert researchers that would like to leverage both MU CI and any other external community CI resources. To support novice users, the students designed

the modules to be customizable and deployable for predefined pipelines on distributed CI resources at NSG. The parameters for the computation pipelines are represented as a neuroscience job "template" to denote the specifications for the workflow steps, options, inputs and outputs. The users have the flexibility to define multiple new templates, or re-purpose previous ones based on their need. The goal for such a plug-and-play approach was to relieve both novice and expert neuroscience researchers from the burden of CI configurations and complex simulation programming.

In order to replicate the above NSG web services functionality seamlessly at our local MU Lewis cluster environment, the students have repurposed the CIPRES [4] science gateway framework. With CIPRES, a simple browser interface is available to manage user accounts, manage job data, share tools and allow access with machine-specific parameters to remote HPC resources at sites including NSG and XSEDE sites. Given CIPRES is compatible with our Lewis cluster scheduler, its integration as shown in Figure 5 has substantially reduced our efforts to design and develop workflow management mechanisms to handle large scale jobs (such as those pertaining to e.g., NEURON code) relating to both research and education. This way the students have started to bridge the gaps in leveraging "bring your own hardware"
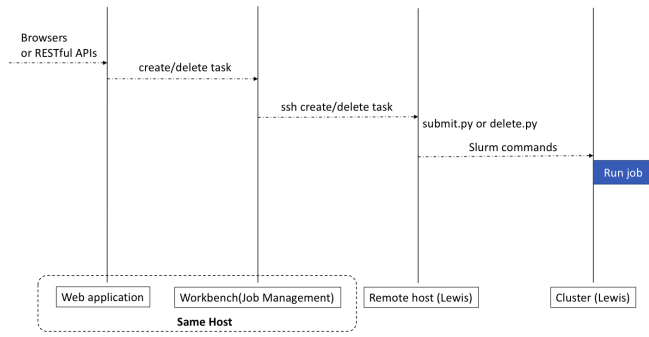
Fig. 5. Interaction between various components of science gateway and a remote HPC resource site such as MU Lewis cluster.

scenarios to integrate local institution HPC resources with other nationally accessible HPC resources.

B. Guided User Interface for Neuroscience Simulations

To enable the neural network simulation (outlined in Section III-B.1) with cyber and software automation, we presented the students the opportunity to design and develop a NEURON Science Gateway user interface shown in Figure 6. The students approached this problem as follows. First, they had an assumption that the necessary data to define the model will be elicited from the researcher through a guided user interface. There is an option to skip pre-processing and provide the two necessary files to the main program. If the option to upload the files is not chosen, the user will specify the number of neuron types in the model, the number of each type, and the percent directed connectivity between each type, including to the same type. This produces a connectivity matrix definition about how neurons are connected. A weight matrix is also defined using random weights on the connections. An expert user in this case may specify the number of nodes, the number of tasks, the number of CPUs per task, the memory per core, and the maximum batch time. For a novice user, the guided user interface makes recommendations based on previous job execution history knowledge.

To enable the neural single cell simulation (outlined in Section III-B.2) with cyber and software automation, the students considered a workflow where the values for the parameters of the first two steps are hard-coded and hence do not require user input. Only the information about the third set of questions is elicited i.e., the type of experiment the user wants to run. Depending on the type of experiment chosen, the user is asked to enter the values for the parameters needed for that experiment. Once the required parameters are collected, a `cfg` file is created and it is zipped with the other required files. This zipped folder is then handled by consumption module which invokes the `curl` commands provided by NSG and runs the workflow on the HPC resources provided by XSEDE.

The guided user interfaces for both the Neuron Simulations were driven by a set of responsive/dynamic forms and a context-aware chatbot (shown in Figure 6). The users are guided through the steps of selecting a workflow, entering the parameters, and visualizing the data after the job execution is complete. The context-aware chatbot responds to the user

questions about the workflow, and helps in creation of a `cfg` file containing the gathered parameters. Users are able to navigate through the user interface screens of a neuroscience workflow while simultaneously interacting with the chatbot. All files necessary for the execution of the given workflow are retrieved from the file store and zipped with the `cfg` file and are passed to the Workflow Manager module. The Middleware Services module sets up a job monitoring process, and also creates a record of successful job completions as a template by interacting with a MySQL relational database that stores the template information. The abstract template for a workflow is combined with the parameters provided by the user and persisted as a job template candidate for future use. If a user request in future matches one of the existing template specifications, that matching template is retrieved automatically for *reuse* i.e., for repeated executions with the same parameters, or *re-purpose* i.e., to adapt the values of some or all parameters as a new job.

C. Jupyter notebooks for Neuroscience Simulations

Scientific simulations on HPC resources typically involve non-interactive and batch execution processes using high-capacity cores, memory and networks spanning several hours to days. Such processes are inflexible when modifying the input/output data type and incorporating rich media. In fact, this utilitarian approach is optimized for raw performance, which leaves limited scope for human ease of interactive use.

To solve this issue and facilitate multiple users to concurrently run interactive lab exercises, the students experimented with the Jupyter Notebook framework [5]. Jupyter Notebooks are documents which contain computer code and rich-text elements (e.g., equations, figures, weblinks) that enables the user (i.e., an instructor or student) to inspect the state of an execution in real time. This server-client application has the flexibility to edit and execute the "notebook" on a local desktop without Internet access or via web-access via a remote server. These features allows the simulations to be modified and communicated in an understandable and reproducible manner, and can be used to remotely deliver instructional material to a large student audience.

There are two possible ways that are suggested in [19] to use the Jupyter Notebooks framework in a science gateway environment. The first simplistic option is to use it as another package as part of the gateway technology stack. The user can invoke the respective notebooks on a need-basis from the existing templates to deploy it on a compute node. Although this is may be easier to incorporate with an existing science gateway, this design solution cannot utilize the complete capabilities of its interactive features. The other more radical shift is to deploy the Jupyter Notebooks themselves as the science gateway interface. Some work in this direction is in progress in many national-level HPC sites [19] and has been deployed at the Minnesota Supercomputing Institute [20]. One of the main disadvantages of this option pertains to the inherent scaling issues, which the students plan to further investigate in the next course offering as an ongoing effort.

In our current science gateway environment, the students have developed "self-service" capable Jupyter Notebook instances for the Neuron Simulation and Zebrafish Branchiometer Neuron Analysis use cases detailed in Section III-B. As shown in Figure 7, the setup involves a user login via
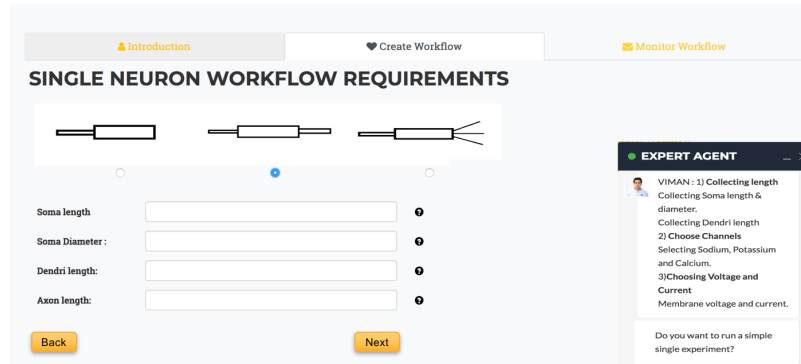
Fig. 6. Guided User Interface within our science gateway - Dynamic Forms and context-aware Chatbot
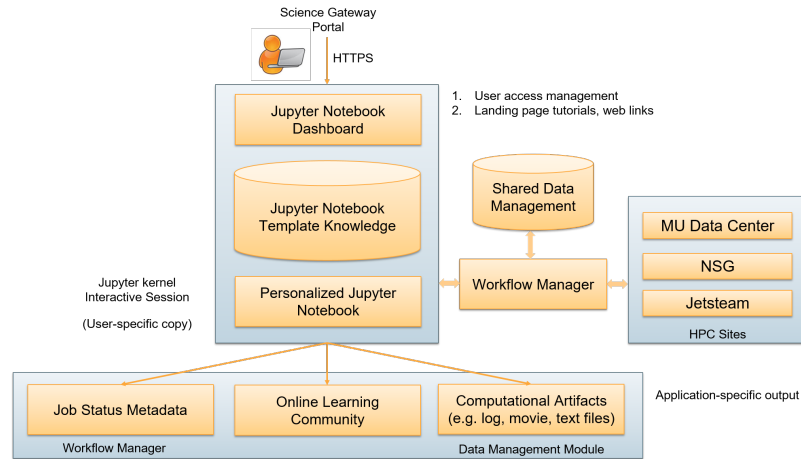

Fig. 7. Educational learning material delivery with the Jupyter Notebooks with job submission provision to multiple HPC sites.
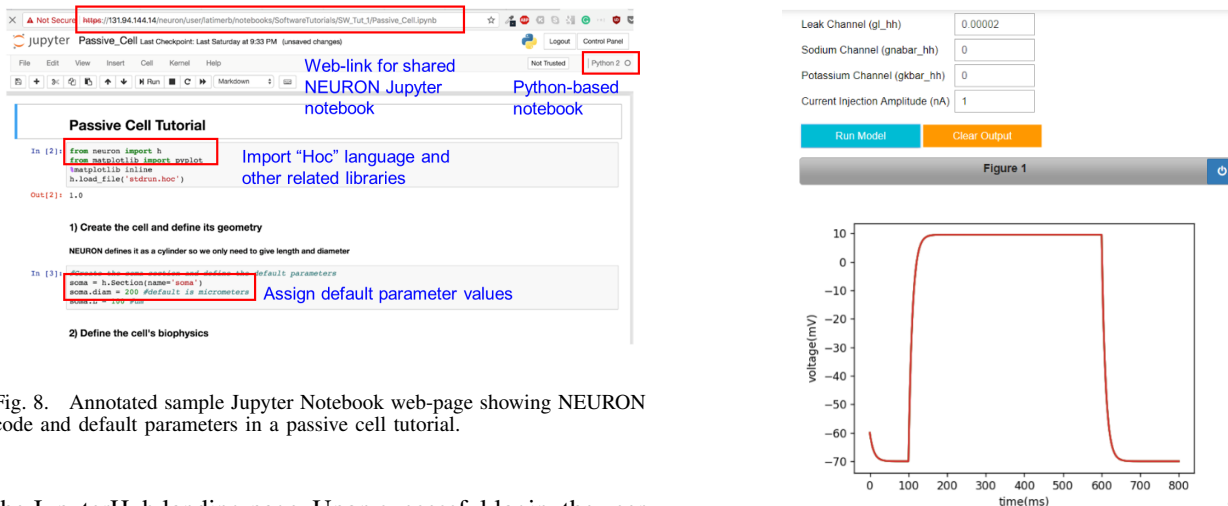

Fig. 8. Annotated sample Jupyter Notebook web-page showing NEURON code and default parameters in a passive cell tutorial.

the JupyterHub landing page. Upon successful login, the user will be guided to customize and share his/her personalized notebooks on the dashboard, thereby enabling collaboration on the learning material within an online learning community. The current components setup also provides regular job status updates which is specially useful for long-running job related experiments. The setup also assumes that reliable and scalable import/export of computational artifacts (e.g., log, text, movie, CSV files) that are important for the user will be handled by external web services such as e.g., the Globus Online research data management services.

By obfuscating the underlying Python code, the

Fig. 9. Jupyter Notebook executing NEURON simulation for instructional purposes.

NEURON-based notebooks allow the non-programmers to focus on experimenting with the parameters for Neuron Simulations as shown via the interfaces of Figure 8 and Figure 9. This approach also enables the instructors to focus on the learning material and teaching objectives during the class, instead of addressing trivial issues such as environment configuration, credentials for software access,
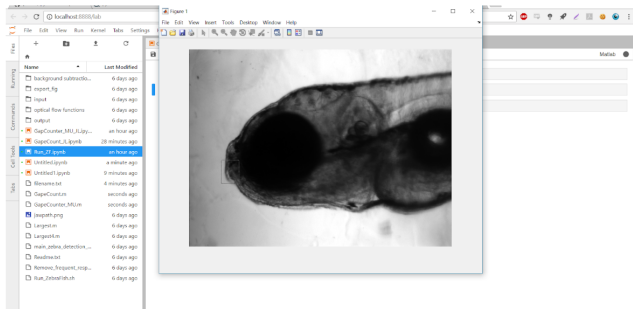
Fig. 10. Jupyter notebook demonstrating input data import for the Zebrafish use-case

or other installation problems. A similar approach was used to create the Jupyter notebook interfaces for the MATLAB-based video annotation tool such as the e.g., shown in Figure 10 for the Zebrafish Branchiometer Neuron Analysis use case. Two main functionality relating to zebrafish video microscopy were integrated within a single document: (a) cloud transformation from desktop to import very large input data, and (b) parameter experimentation to modify the image filter properties. Thus, such a design of the lab exercise learning module allows a biologist who is a novice HPC user to easily conduct and customize high-throughput experimentation with remote HPC resources.

D. CI Template Layers

Templates are key to providing a generalizable service to our users for reuse and re-purpose scenarios. As shown in Figure 4, there are three abstraction layers of templates which the students implemented in a MySQL relational database.

The highest abstraction layer (i.e., the Template Layer) generalizes a pipeline. The Template table contains the use case information for the pipeline configuration. For instance, students implemented information portions of the NEURON Single Cell and NEURON Network use cases. There are Steps in the pipeline, which contain one or more exclusive Step Options. For example, the NEURON Single Cell use case has modeling and visualization steps. For the modeling step, user can have two options: current injection and synapse. Each Step Option may have different input and output Parameters and Files associated with it. For example, the current injection option outputs a `SimpleCurrentInjection.cfg` file and has input parameters for $amplitude$, $delay$, dt, $duration$, $tstop$, and v_init. The middle abstraction level (i.e., the Job Layer) has tables for the Job, which is associated with a Step Option, and the Job Parameter and Job File, which are associated with the Parameter and File entries for that Step Option. The job has a $name$ and is associated with the user who created the job, and the set of three job tables together persist the choices made, and the parameters supplied by the user. Finally, the third level of abstraction (i.e., the Execution Layer) will persist $execution\ time$, $performance\ metrics$, $execution\ success\ or\ failure$, and any $error\ messages$ that are returned. Additional tables could be created to contain detailed step execution and performance data, if needed.

To enable reuse and re-purpose of the templates, successful job completion information is persisted in the database with associated parameters and a user-supplied job name. Thus, templates can be reused in the future by e.g., querying on the creator information of a job. Users can select any previously run jobs and run them again as illustrated in Figure 11. The user is also given an option to edit the parameters used previously and can create a new instance of the same template. This process can be repeated multiple times, and each time an old instance can be used or it can be slightly altered to run a new instance. This allows for multiple experiments to be run easily - perhaps to establish norms or to narrow the effect of changing a single parameter. In comparison, to re-purpose a template to execute a job step that may have same or different steps, a user can use the parameters gathered for one template (e.g., the Neuron Single Cell Simulation) to be re-purposed in another template (e.g., the Neuron Network Simulation).

V. USERS SURVEY

We conducted a user survey targeting various stakeholder teams involved in the course delivery over the two semester period: (i) Application User Team - Neuron/network simulations, (ii) Application User Team - MATLAB-based image processing automation, (iii) CI Engineer Team - distributed CI resource management using Lewis and NSG, and (iv) Science Gateway Portal Developer Team - Jupyter Notebooks and CIPRES configuration. Each of the users where asked to: (a) Describe what they intended to learn about either CI or Neuroscience or both in the course, and (b) Provide 1 or 2 learning objectives related to either the application perspective (e.g., How Neuroscience and CI can benefit from each other?), or from the CI perspective (e.g., How there are new HPC cloud challenges with dynamic data-driven science?), or from the science gateway portal perspective (e.g., How to create repeatable and scalable workflows to benefit application users and effectively/efficiently use available CI resources?). Users also were asked to summarize how the interdisciplinary course work and problem-solving based project work have influenced their degree goals, and expanded the understanding of their respective fields, as well as the merits/challenges of working in student/faculty teams. Lastly, they were asked to provide any suggestions for the instructors on how to improve the 1-credit course to cross-train students in future offerings.

Key take aways from the survey responses are as follows: The interdisciplinary project outcomes were more beneficial to application and CI users when the application users took the lead and provided regular feedback on aspects of CI development that were useful. The CI users in some cases were able to get the application users excited on the possibilities for scalability and automation after prototype demos were created. All the students felt that the interdisciplinary project experience was more interesting and valuable to their careers than project in a specific area; this is because it enabled them to not only pursue advanced neuroscience and CI concepts learning of technical skills for real world problems, but also learn soft skills on understanding terminology and working with multi-disciplinary expertise related to different areas. Particularly, the CI students realized that they need additional training and survey science concepts understanding to effec-

Fig. 11. Job history - Retrieval of previous Jobs for reuse

tively gather science gateway requirements from application users, and keep up with their evolving needs.

Both the CI and neuroscience groups recognized the need for CI templates to reuse and repurpose for satisfying various neuroscience use cases. They also agreed that having CI engineering around application-centered user needs (versus matching needs with just a catalog of available hardware resource capabilities) would produce more innovative outcomes. The application groups found value in creation of end-to-end exemplars for various use cases in order to provide feedback on additional features, as well as to share the benefits of CI integration with their colleagues. The instructors were given feedback to improve future coursework with more planning to allow students/faculty to interact more with each other closely in the interdisciplinary projects. Also, students wanted to study case studies where interdisciplinary collaborations have succeeded in science gateway projects (e.g., the CIPRES project, Globus Online project). They were asking for more opportunities to understand system level aspects and unique application requirements for CI around automation and scalability.

VI. CONCLUSION

This paper presented early experiences from a new 1-credit multi-disciplinary graduate research training course i.e., ECE 8001 "Software and Cyber Automation in Neuroscience" at the University of Missouri. This course was offered over two semesters for targeted cross training students from biological/psychological sciences and computer science/engineering for improving CI capabilities for neuroscience research and education use cases. The course organization in the first semester helped student groups to explore CI resources and integrate them via the design and development of a next-generation science gateway i.e., the NEURON Science Gateway. Their prototype was aimed to support novice and expert users interested in pursuing neuroscience teaching and research, however with minimum needed prior HPC expertise. The second course offering expanded the multidisciplinary student collaborative efforts to develop "self-service" training modules for neuroscience research and education use cases using open frameworks such as the CIPRES and the Jupyter Notebooks. This has created new opportunities for our next-generation science gateway to provide higher-scale capabilities with local institution resources as well as remote HPC resources via seamless/repeatable workflow management processes.

The two course offerings outcomes have provided a good foundation for planning a future course sequence that can provide training modules/exercises spanning CI and neuroscience concepts to both graduate and undergraduate students. To fully realize the vision of our planned course sequence development, there are several possible directions in which we seek to expand. The templates for the NEURON Single Cell and NEURON Network use cases need to be tested with more number of users, and refactored to match a broader set of user requirements. Development of pipelines and the corresponding parameter collection dynamic forms and chatbot guidance logic needs to be pursued for the other use cases such as fMRI, WGCNA, and 2-Photon Calcium Imaging. To expand on the next-generation science gateway architecture development, secure and reliable integration of MU CI resources including the Lewis cluster especially for data management and access control need to be explored. In addition, the training modules/exercises that are emerging from the course work of the students as Jupyter notebooks need to be enhanced with usability studies in a systematic manner using survey methods. Further, additional features such as federated user authentication/authorization, notebook versioning, image data import/export services, etc. can help with wider deployment and training to foster use by teachers and students at other institutions.

ACKNOWLEDGMENT

REFERENCES

[1] NSF (2016). Understanding the BRAIN, `https://www.nsf.gov/news/special_reports/brain/`.

[2] NSF (2015). iNeuro Conference Report, `https://mdcune.psych.ucla.edu/modules/ineuro/reports/iNeuro_ConferenceReport_20150804.pdf`

[3] S. Sivagnanam, A. Majumdar, K. Yoshimoto, V. Astakhov, A. Bandrowski, M. Martone and N. T. Carnevale, "Introducing the Neuroscience Gateway," 2013. `https://www.nsgportal.org`

[4] M. Miller, W. Pfeiffer, T. Schwartz, "The CIPRES science gateway: a community resource for phylogenetic analyses", *Proc. of the ACM TeraGrid Conference*, 2011.

[5] T. Kluyver, B. Ragan-Kelley, F. Prez, B. Granger, M. Bussonnier, J. Frederic, K. Kelley, J. Hamrick, J. Grout, S. Corlay, P. Ivanov, "Jupyter Notebooks-a publishing format for reproducible computational workflows", *ELPUB, pp. 87-90*, 2016.

[6] J. Freeman, "Open source tools for large-scale neuroscience," *Current Opinion in Neurobiology, vol. 32, pp. 156-163*, 2015.

[7] J. Kubilius, "A framework for streamlining research workflow in neuroscience and psychology," *Frontiers in Neuroinformatics, vol. 7*, 2014.

[8] T. Sherif, P. Rioux, M.-E. Rousseau, N. Kassis, N. Beck, R. Adalat, S. Das, T. Glatard, and A. C. Evans, "CBRAIN: a web-based, distributed computing platform for collaborative neuroimaging research," *Frontiers in Neuroinformatics, vol. 8*, 2014.

[9] "psychopy_ext: A framework for a rapid reproducible design, analysis and plotting of experiments in behavioral neuroscience and psychology". `https://github.com/qbilius/psychopy_ext`

[10] W. J. Goscinski, P. Mcintosh, et. al., "The multi-modal Australian ScienceS Imaging and Visualization Environment (MASSIVE) high performance computing infrastructure: applications in neuroscience and neuroinformatics research," *Frontiers in Neuroinformatics, vol. 8*, 2014.

[11] D. E. Rex, J. Q. Ma, and A. W. Toga, "The LONI Pipeline Processing Environment", *NeuroImage, vol. 19, no. 3, pp. 1033-1048*, 2003.

[12] M. Atkinson, S. Gesing, J. Montagnat, and I. Taylor, "Scientific workflows: Past, present and future," *Future Generation Computer Systems, vol. 75, pp. 216-227*, 2017.

[13] G. Kiar, K. J. Gorgolewski, D. Kleissas, W. G. Roncal, B. Litt, B. Wandell, R. A. Poldrack, M. Wiener, R. J. Vogelstein, R. Burns, and J. T. Vogelstein, "Science in the cloud (SIC): A use case in MRI connectomics," *GigaScience, vol. 6, no. 5, pp. 1-10*, July 2017.

[14] Y. Liu, S. M. Khan, J. Wang, M. Rynge, Y. Zhang, S. Zeng, S. Chen, J. Maldonado, B. Valliyodan, P. Calyam, M. Merchant, H. Nguyen, D. Xu, and T. Joshi, "PGen: Large-Scale Pegasus Workflow for Genomic Variation Analysis in SoyKB", *BMC Bioinformatics 17 (Suppl 13):337, DOI: 10.1186/s12859-016-1227-y*, 2016.

[15] B. Latimer, D. Bergin, V. Guntu, D. Schulz, S. Nair, "Integrating model-based approaches into a neuroscience curriculum – An interdisciplinary neuroscience course in engineering", *IEEE Transactions on Education*, 2018. DOI: 10.1109/TE.2018.2859411

[16] B. Latimer, D. Bergin, D. Schulz, S. Nair, "Open source software tools for teaching neuroscience", *Journal of Undergraduate Neuroscience Education*, 16(3):A197-A202, 2018.

[17] A. Majumdar, S. Sivagnanam, N.T. Carnevale, K. Yoshimoto, P. Gleeson, A. Quintana, and R. A. Silver, "Neuroscience Gateway - Cyberinfrastructure Providing Supercomputing Resources for Large Scale Computational Neuroscience Research", *Frontiers in Neuroinformatics*, 2016.

[18] Y. M. Kassim, N. M. Al-Shakarji, E. Asante, A. Chandrasekhar, and K. Palaniappan, "Dissecting Branchiomotor Neuron Circuits in Zebrafish Toward High-Throughput Automated Analysis of Jaw Movements," *IEEE International Symposium on Biomedical Imaging (ISBI), Washington DC, USA*, 2018.

[19] A. Zonca, and R. S. Sinkovits, "Deploying Jupyter Notebooks at scale on XSEDE resources for Science Gateways and workshops," *In Proceedings of the Practice and Experience in Advanced Research Computing (PEARC), USA*, 2018.

[20] M. Milligan, "Interactive HPC Gateways with Jupyter and Jupyterhub," *In Proceedings of the Practice and Experience in Advanced Research Computing (PEARC), USA*, 2017.