LSU Computational System Biology Gateway for Education

Eroma Abeysinghe
Science Gateways Research Center
Pervasive Technology Institute
Indiana University
Bloomington, Indiana
eabeysin@iu.edu

Michal Brylinski
Department of Biological Sciences,
Louisiana State University
Baton Rouge, Louisiana
michal@brylinski.org

Marcus Christie
Science Gateways Research Center
Pervasive Technology Institute
Indiana University
Bloomington, Indiana
machrist@iu.edu

Suresh Marru
Science Gateways Research Center
Pervasive Technology Institute
Indiana University
Bloomington, Indiana
smarru@iu.edu

Marlon Pierce
Science Gateways Research Center
Pervasive Technology Institute
Indiana University
Bloomington, Indiana
marpierc@iu.edu

ABSTRACT

Science gateways are a mechanism for delivering scientific software as a service, especially when the software requires high performance computing (HPC) resources to run effectively. The existence of a science gateway eliminates the user's need to learn to work with HPC systems and to manage software installations and updates. With well-designed user interfaces, users can more quickly become effective users of scientific applications and can manage information needed for replicating, modifying, and sharing results. All of these efficiency gains enable users to focus more on their research. In addition, science gateways are being identified as an effective educational tool, a tool to be used in classroom environments as a method to get students quickly into research on domain specific questions. In the absence of a science gateway, students are likely to need a considerable time to learn to work with HPC systems, and any time spent on such will reduce their time on the actual science. This poster presents how the Louisiana State University (LSU) gateway for the Computational System Biology Group (CSBG) - (www.brylinski.org) was updated and improved to be a classroom teaching tool. This work makes extensive use of Apache Airavata's group management capabilities.

CCS CONCEPTS

• **Computing methodologies → Modeling methodologies**; **Simulation environments**; **Simulation tools**; *Model verification and validation*; • **Software and its engineering → Software as a service orchestration system**.

KEYWORDS

Apache Airavata, Bioinformatics, Science Gateway, Education

ACM Reference Format:
Eroma Abeysinghe, Michal Brylinski, Marcus Christie, Suresh Marru, and Marlon Pierce. 2019. LSU Computational System Biology Gateway for Education. In *Practice and Experience in Advanced Research Computing (PEARC '19), July 28-August 1, 2019, Chicago, IL, USA.* ACM, New York, NY, USA, 4 pages. https://doi.org/10.1145/3332186.3333259

1 INTRODUCTION

Science gateways simplify the process for gaining access to scientific application tools on HPC systems. In addition to supporting research, gateways are effective teaching tools for classroom environments. The LSU CSBG gateway was initially launched with focus on building research community around it and is now expanding its use as a teaching tool for both advanced undergraduate and graduate students in biological sciences. As part of the Extended Developer Support (EDS) services from the Science Gateways Community Institute (SGCI) [1], the principle investigator (PI) started to discuss the requirements to upgrade the gateway to be more classroom or student friendly. The LSU CSBG gateway is deployed and maintained by SciGaP.org gateway services [2] and together with the EDS consultants, PI, and the gateway team, the new gateway features were designed and implemented.

2 CSBG TOOLS

LSU CSBG offers a variety of tools for Structural Bioinformatics to support the prediction of protein structure and function from raw sequence data (Table 1).

Table 1: CSBG tools for Structural Bioinformatics

Application	Purpose
eThread	Protein structure modeling
eFindSite	Drug-binding site detection
eMatchSite	Ligand binding sites alignment
eMolFrag	Molecular fragment extraction
eFindSite^{PPI}	Protein-binding site detection
eSynth	Virtual molecular synthesis
eVolver	Protein sequence design

PEARC '19, July 28-August 1, 2019, Chicago, IL, USA
© 2019 Copyright held by the owner/author(s).
ACM ISBN 978-1-4503-7227-5/19/07.
https://doi.org/10.1145/3332186.3333259

eThread is a template-based protein structure prediction approach. Template structures are selected from the Protein Data Bank with meta-threading and machine learning. Protein models are assigned confidence scores, which correlate with the actual model quality [3],[4].

eFindSite is a ligand-binding site prediction and virtual screening algorithm detecting common ligand-binding sites in a set of evolutionarily related proteins identified with threading/fold recognition. eFindSite also performs ligand-based virtual screening against identified pockets [5], [6].

eMatchSite is a sequence order-independent algorithm to align and match ligand binding sites. eMatchSite accurately identifies those pockets binding similar compounds even in proteins with different global structures. Furthermore, it tolerates, to a large extent, structural distortions in protein models, thus experimentally solved structures are not required [7], [8].

eMolFrag is an automated method to extract molecular fragments from compound libraries. Fragments are categorized as core fragments (bricks) and flexible linkers. These chemical building blocks can subsequently be used to construct virtual screening libraries for targeted drug discovery [9].

eFindSite^{PPI} is a program to detect protein binding sites and residues with meta-threading. eFindSite^{PPI} also predicts interfacial geometry and specific interactions stabilizing protein-protein complexes, such as hydrogen bonds, salt bridges, aromatic and hydrophobic interactions [10],[11].

eSynth is an automated method to synthesize virtual compounds by reconnecting building blocks obtained by fragmenting parent molecules. The synthesis process employs an exhaustive graph-based search algorithm following connectivity patterns extracted from the input compounds. The primary application of eSynth is the rapid construction of virtual screening libraries for targeted drug discovery [12].

eVolver is an optimization engine evolving protein sequences to stabilize the respective structures by a variety of potentials, which are compatible with those commonly used in protein threading. The scoring function implemented in eVolver combines several energy terms, a burial potential, secondary structure preferences, a distant-dependent contact potential, sequence profiles and anti-bunching restraints [13], [14].

3 CSBG USAGE IN CLASSROOM

The LSU CSBG gateway is deployed and developed for bioinformatics research work. While the gateway was being used for research, the PI wanted to start using the gateway in the classroom as well. A single gateway to be used by advance researchers as well as students required several layers of configurational as well as code level changes. The gateway team and the PI discussed what are the requirements that needed to be available in order to bring novice students to use the gateway. Those features were implemented, tested and made available for the gateway to configure as needed.

Three tools are currently used in classroom: eThread, eFindSite, and eSimDock. Students taking project-oriented courses, Biophysics of Macromolecules for undergraduate students and Computational Biology for graduate students, complete their projects using the gateway. These projects consist of three tasks. First, structure models of known drug targets are constructed with eThread, which is followed by drug binding site prediction with eFindSite. The last task is to dock a drug molecule with eSimDock to the predicted binding site.

An example of the completed project is shown in Figure 1. This 3D structure model of a ligand-protein complex was built by an undergraduate student from the amino acid sequence of a protein target and the chemical structure of a ligand using our gateways. The PI is currently developing additional projects, including protein design with eVolver, prediction of drug side-effects with eMatchSite, and fragment-based drug discovery with eMolFrag and eSynth. These new projects will be available to students in Fall 2019.

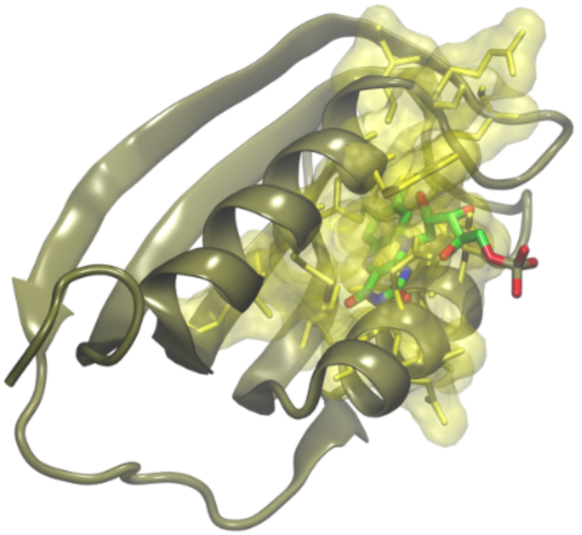


Figure 1: Example of a protein structure modeled by eThread (solid ribbons) with a ligand docked by eSimDock (color sticks) to a putative binding site predicted by eFindSite (transparent surface).

4 GATEWAY FOR EDUCATION WITH SCIGAP SERVICES

4.1 Gateway Setup

The initial LSU CSBG gateway was developed and deployed with the support from SGCI EDS consultations. The gateway is hosted and managed by the multi tenanted SciGaP [2] gateway platforms with storage facilities for gateway user files. The SciGaP platform uses Apache Airavata [15] middleware for computational job construction, job submission to HPCs and to manage job data files, inputs and outputs.

Apart from the primary services such as user identity management, accounts, authorization, and access to multiple high performance computer (HPC) resources from campus and national resource providers, the gateway had specific requirements in order to make it 'classroom friendly'. The web client used for the user interactions with the middleware is developed using python based

Django Framework and together with Apache Airavata middleware both are hosted by the SciGaP servers at Indiana University. The current gateway and middleware system layout is depicted in Figure 2.

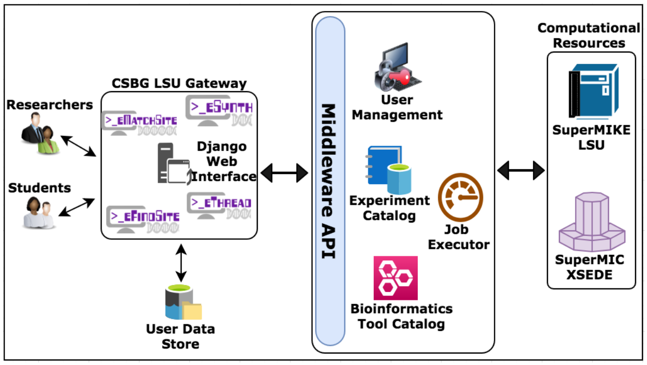


Figure 2: CSBG LSU Gateway with Apache Airavata

4.2 Gateway Users & Groups

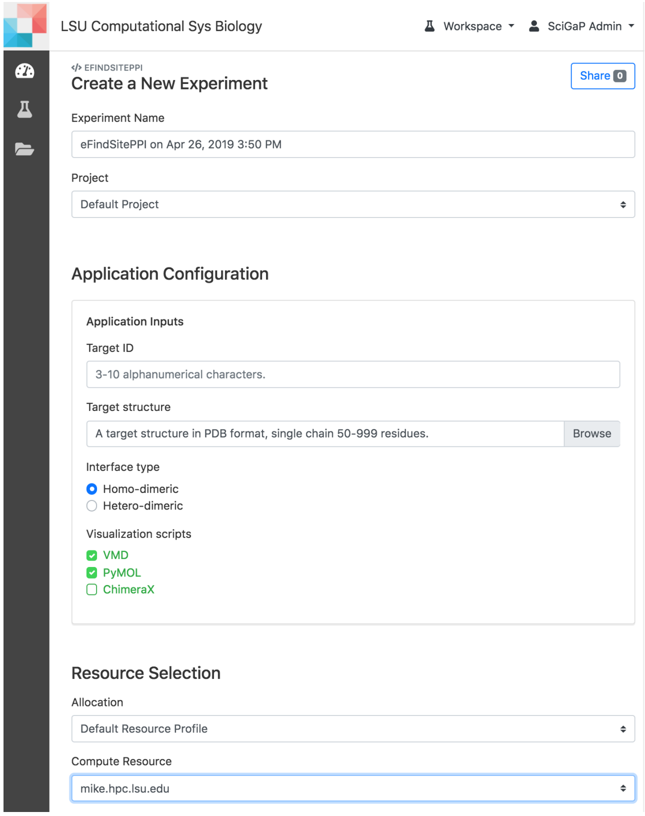
In order to enable the gateway for educational use, it needed to be able to support multiple classrooms at a given time, with each classroom having to be able to use specific application tools. The gateway is implemented in a such way where each classroom can be represented by a 'Group' within the gateway. We use Apache Airavata's Sharing Service [16] to associate application tools and HPC resource allocations with a group for each classroom.

For a classroom, a representative, teacher assistant (TA) can be named as the administrator to manage students within this gateway group. The appointed classroom administrator can add students to the group and give them the proper role within the group to use the gateway. Students can share their work within their group as well as with their fellow gateway users. The students or researchers can create their own user groups, mainly for data and experiment sharing purposes. In order to be added to a group, the students or the gateway user needs to create a gateway account. The gateway account creation can be done locally to the gateway or by using existing institute logins through federated authentication system, CILogon [17]. The SciGaP gateways user authorization and authentication is managed by Keycloak [18], an open source identity and access management system.

4.3 CSBG Tool Configuration

With Apache Airavata's Django-based gateway, administrators can control application tool access within the gateway. Gateway administrators can add tools for the gateway users. Through the sharing feature, the application tools can be exposed only to the intended user groups. When users log in to the gateway, they only see the tools that are shared with them in the 'Workspace'. The 'Workspace' is the space that users could launch 'Experiments' to submit jobs to HPC resources. The 'Experiment' is the metadata properties required for the job execution in remote HPC resource [19].

Since the gateway PI is using the gateway for teaching, the gateway team had to implement advanced application input properties and validations to minimize erroneous inputs. Reducing the errors in inputs saves time and HPC resources from waste. Science gateway application input basic types are either string, number or file uploads. To extend these basic types, several new ways are introduced for gateway administrators to use when configuring their applications. Using option selections, check boxes and lists the users are provided the inputs they could select, and for number inputs the administrator can specify a range, whether decimals are allowed, and similar restrictions. (Figure 3). The gateway also supports cascading input lists, so users will only be able to select from the provided lists, thus reducing the chances of making mistakes. The gateway also provides validating input file types, which validate the extensions to be of the pre-specified types in order to make sure users upload the correct files. All the validations are done at the web portal side to minimize any delays.



</> EFINDSITEPPI

Create a New Experiment

Experiment Name

eFindSitePPI on Apr 26, 2019 3:50 PM

Project

Default Project

Application Configuration

Application Inputs

Target ID

3-10 alphanumerical characters.

Target structure

A target structure in PDB format, single chain 50-999 residues.

Interface type

● Homo-dimeric
○ Hetero-dimeric

Visualization scripts

☑ VMD
☑ PyMOL
☐ ChimeraX

Resource Selection

Allocation

Default Resource Profile

Compute Resource

mike.hpc.lsu.edu

Figure 3: eFindSite^{PPI} Experiment Creation - Option selections, Checkboxes and String inputs with Character length validations

4.4 Group Resource Profile

In order to facilitate multiple user groups and to configure the level of resources that are available for each, a Group Resource Profile (GRP) data model was added to Airavata's Registry Service, with API methods to manage them. The GRP is a configuration of compute resource availability in terms of nodes, CPUs and walltime for a group. Multiple GRPs can exist in the gateway. Using user interfaces in the Django portal, gateway administrators can create them and add multiple HPC resources or configure in other ways as needed. The gateway administrator can create as many SSH keys/credentials using the gateway credential store [20] for these GRPs and assign them for secure communications with compute and storage resources. The GRPs then can be shared with user groups, and those GRPs will be available for users to select at experiment creation. A single user can belong to multiple user groups.

4.5 Gateway Theme and Branding

In previous PHP-based Airavata gateways, the gateway theme and branding were managed through a git repository, and updates to the gateway content required knowledge of git as well as considerable time. With the Django-based gateway, this process has been changed to use the Wagtail Content Management System (CMS) [21] to manage the gateway theme. Wagtail CMS provides in-place methods for administrators to directly update and review content changes prior to publishing them. When the gateway is used in classroom, the content will need frequent updates to provide information to users. Having it done through a CMS with its editors to add or update content is time saving. Gateway administrators can delegate the task to teaching assistants to add the relevant content for their classrooms.

5 FUTURE WORK

We will continuously improve the LSU CSBG Gateway in order to expand the research and student user community. Our goal is to make all tools developed by CSBG available for the research and education communities through the gateway. We plan to introduce the gateway to other institutes and their classroom environments as well as to focus on post processing and visualization tool integration.

6 CONCLUSIONS

The gateway PI and his team at LSU CSBG will continue to improve their tools and make them available for the community of both advanced researchers and students through the gateway. The users can use the latest tools through the gateway without going through the trouble of compiling the tools, updating when changes take place and also without worrying about HPCs availability for computations. Each user group will have their own space within the gateway, share their work and continue as they progress through each layer of research from student to advanced researcher.

ACKNOWLEDGMENTS

The Louisiana State University - Computational System Biology Group Science Gateway was implemented with the support from the Extended Developer Support of Science Gateways Community Institute supported by NSF award #1547611. Development of the Apache Airavata used to develop the science gateway is supported by NSF award #1339774.

REFERENCES

[1] Nancy Wilkins-Diehr, Michael Zentner, HUBzero Platform, Marlon Pierce, Maytal Dahan, Katherine Lawrence, Linda Hayden, and Nayiri Mullinix. The science gateways community institute at two years. In *Proc. Pract. Experience Adv. Res. Comput.*, pages 53–1, 2018.

[2] Marlon Pierce, Suresh Marru, Eroma Abeysinghe, Sudhakar Pamidighantam, Marcus Christie, and Dimuthu Wannipurage. Supporting science gateways using apache airavata and scigap services. In *Proceedings of the Practice and Experience on Advanced Research Computing*, page 99. ACM, 2018.

[3] Michal Brylinski and Daswanth Lingam. ethread: a highly optimized machine learning-based approach to meta-threading and the modeling of protein tertiary structures. *PLoS One*, 7(11):e50200, 2012.

[4] Michal Brylinski. Unleashing the power of meta-threading for evolution/structure-based function inference of proteins. *Frontiers in genetics*, 4:118, 2013.

[5] Michal Brylinski and Wei P Feinstein. efindsite: improved prediction of ligand binding sites in protein models using meta-threading, machine learning and auxiliary ligands. *Journal of computer-aided molecular design*, 27(6):551–567, 2013.

[6] Wei P Feinstein and Michal Brylinski. efindsite: Enhanced fingerprint-based virtual screening against predicted ligand binding sites in protein models. *Molecular informatics*, 33(2):135–150, 2014.

[7] Michal Brylinski. ematchsite: sequence order-independent structure alignments of ligand binding pockets in protein models. *PLoS computational biology*, 10(9):e1003829, 2014.

[8] Michal Brylinski. Local alignment of ligand binding sites in proteins for polypharmacology and drug repositioning. *Protein Function Prediction: Methods and Protocols*, pages 109–122, 2017.

[9] Tairan Liu, Misagh Naderi, Chris Alvin, Supratik Mukhopadhyay, and Michal Brylinski. Break down in order to build up: decomposing small molecules for fragment-based drug design with e molfrag. *Journal of chemical information and modeling*, 57(4):627–631, 2017.

[10] Surabhi Maheshwari and Michal Brylinski. Prediction of protein–protein interaction sites from weakly homologous template structures using meta-threading and machine learning. *Journal of Molecular Recognition*, 28(1):35–48, 2015.

[11] Surabhi Maheshwari and Michal Brylinski. Template-based identification of proteinâĂŞprotein interfaces using efindsiteppi. *Methods*, 93:64–71, 2016. Computational protein function predictions.

[12] Misagh Naderi, Chris Alvin, Yun Ding, Supratik Mukhopadhyay, and journal=âĂIJJournal of Cheminformatics Brylinski, MichalâĂİ. A graph-based approach to construct target-focused libraries for virtual screening. 8(1):14, Mar 2016.

[13] Michal Brylinski. e volver: an optimization engine for evolving protein sequences to stabilize the respective structures. *BMC Research Notes*, 6(1):303, Jul 2013.

[14] Michal Brylinski. The utility of artificially evolved sequences in protein threading and fold recognition. *Journal of Theoretical Biology*, 328:77 – 88, 2013.

[15] Suresh Marru, Lahiru Gunathilake, Chathura Herath, Patanachai Tangchaisin, Marlon Pierce, Chris Mattmann, Raminder Singh, Thilina Gunarathne, Eran Chinthaka, Ross Gardler, et al. Apache airavata: a framework for distributed applications and computational workflows. In *Proceedings of the 2011 ACM workshop on Gateway computing environments*, pages 21–28. ACM, 2011.

[16] Supun Nakandala, Suresh Marru, Marlon Piece, Sudhakar Pamidighantam, Kenneth Yoshimoto, Terri Schwartz, Subhashini Sivagnanam, Amit Majumdar, and Mark A Miller. Apache airavata sharing service: A tool for enabling user collaboration in science gateways. In *Proceedings of the Practice and Experience in Advanced Research Computing 2017 on Sustainability, Success and Impact*, page 20. ACM, 2017.

[17] Jim Basney, Terry Fleury, and Jeff Gaynor. Cilogon: A federated x. 509 certification authority for cyberinfrastructure logon. *Concurrency and Computation: Practice and Experience*, 26(13):2225–2239, 2014.

[18] Marcus A Christie, Anuj Bhandar, Supun Nakandala, Suresh Marru, Eroma Abeysinghe, Sudhakar Pamidighantam, and Marlon E Pierce. Using keycloak for gateway authentication and authorization. 2017.

[19] Marlon Pierce, Suresh Marru, Borries Demeler, Raminderjeet Singh, and Gary Gorbet. The apache airavata application programming interface: Overview and evaluation with the ultrascan science gateway. In *Proceedings of the 9th Gateway Computing Environments Workshop*, GCE âĂŹ14, pages 25–29, Piscataway, NJ, USA, 2014. IEEE Press.

[20] Thejaka Amila Kanewala, Suresh Marru, Jim Basney, and Marlon Pierce. A credential store for multi-tenant science gateways. In *Cluster, Cloud and Grid Computing (CCGrid), 2014 14th IEEE/ACM International Symposium on*, pages 445–454. IEEE, 2014.

[21] Stephen Paul Adithela, Marcus Christie, Suresh Marru, and Marlon Pierce. Django content management system evaluation and integration with apache airavata. PEARC âĂŹ18, pages 86:1–86:4, New York, NY, USA, 2018. ACM.