

Advancing Reproducible Science through Open Source Laboratory Protocols as Software

Project Information

- Applicant:
 - Luiza Zucchi Hesketh Gomes (LuHesketh)
- Mentors:
 - SIFT, LLC. (Daniel Bryce)
 - UCSD (Tim Fallon)
- Skills Required:
 - Python, VENUS(Hamilton Software), JSON, XML,
 - Domain expertise in biological & chemical laboratory processes
 - Semantic Python-WEB Technologies
- Difficulty:
 - Intermediate
- Size:
 - Large (350 hours)

Abstract

The project aims to develop a Python-to-Python conversor for LabOP-written Laboratory protocols (<https://github.com/Bioprotocols/labop>) to be executed by Hamilton Machines using PyHamilton(<https://github.com/dgretton/pyhamilton>) as a the main bridge. LabOP(Laboratory Open Protocol Language) is a Python Language that allows for Laboratory protocols to be Translated for either human or machine execution and PyHamilton is an open-source Python package that allows its user to control Hamilton liquid handling robots without its native software constraints. This project will benefit scientists by providing open-source protocols for replicating experiments through an open-source software stack. The Protocol used as model is a DNA purification protocol and its integration with LabOP came to fruition with the goal of making the protocol as reproducible as possible since the translation process can be done in different ways by different sites using distinct machines.

Introduction

Problem Statement: Advances in Biotechnology, Life Sciences and its applications were possible thanks mainly to the introduction of Automation of processes, but there are still bottlenecks that prevent the ideal Reproducibility in most Life Sciences laboratories, the biggest one being the lack of standardization of laboratory protocols, a difficult challenge to face since each laboratory has its own specific protocol for its specific conditions even if they all perform the same method in the end. Another implication is the previous knowledge to optimize the execution of the protocols since specific backgrounds in Molecular Biology, Engineering and in Programming are necessary to be able to introduce the necessary solutions as each field has its own demands. To begin tackling this issue it is interesting to take a look at what exactly a laboratory protocol is.

In a nutshell, protocols are like programs, a set of instructions that are run on machines repeatedly following a standard method. One way to improve reproducibility when tracing protocols is to develop and integrate open source programs where the protocols can be written the same way the researcher would do in the Laboratory, but translated in a way that is accessible to both humans and different kinds of machines.

Background: There are already platforms offering similar systems to help researchers organize their processes, but they are usually paid and operate on their own software, which limits the sharing of written protocols to other laboratories, since they will not have the necessary tools to execute them.

Recently, open source solutions have been developed to address the bottleneck of NOT having standard shared protocols between laboratories, but such options are still in the early stages, not being fully integrated with all laboratory automation interfaces nor with user interfaces, thus it still limits accessibility since researchers must have a background in the area to be able to operate them.

With that in mind, [LabOP](#) was created as a viable option for approaching science as software. This Python package allows laboratory protocols to be written in Python and translated to be executed by both humans and machines as well, thus solving the lack of specificity for sharing protocols between institutions. However, the software is not yet integrated into all laboratory automation and requires improvement in its integration to other softwares to guarantee maximum reproducibility.

Solution: The solution revolves around developing methods to adapt open, shareable, and reproducible protocols for the Hamilton interface for lab automation, integrating [LabOP](#) with [pyLabRobot](#) (where PyHamilton is hosted) to simplify intra- and inter-lab reproducibility and Demonstrating at 2 sites (Germany and Brazil) with two protocols (DNA purification, and plate reader calibration). Open source protocols and dataset artifacts to be used as validation tests for protocol adopters will also be provided.

Goals

By the end of the program this project aims to have a fully integrated Open-source software to specialize LabOP Python protocols for use on a Hamilton robot via PyHamilton/PyLabRobot. To test this integration two different protocols will be conceived in the LabOP language and demonstrated at two different sites in the world. One of the reasons for this strategy is to create Reproducibility Artifacts and have the research centers benefit from ready to use protocols that can be run on a Hamilton or other supported liquid handlers (manually by bench Scientist or perhaps with an OT2). Another goal is to provide a researcher-friendly software that supports protocol adaptation/creation for execution on a Hamilton machine without the need for user specialization on every single one of its specific commands.

Deliverables: The deliverables of the project will consist of phase reports on a public blog, a LabOP-based PyLabRobot specialization software for protocol writing and execution, and several data and Reproducibility artifacts that will be documented and delivered.

Future Work: Future work will expand upon this project by incorporating additional protocols to develop similar reproducibility artifacts for different protocols, as well as use the developed software to connect additional laboratory instruments to LabOP.

Implementation Plan

The project will be divided into three phases, which involve software development, data collection, and reporting results. The software development phase will focus on defining test cases and integrating LabOP and pyLabRobot. The data collection phase will involve running several protocols to collect data that can be used for assessing reproducibility. The reporting phase will package and document project artifacts for public reuse.

- **Phase 1: Test Cases and Software Development**

Description: Develop a LabOP description of an existing DNA purification protocol that will run on the Hamilton liquid handler. Develop extensions to the LabOP library to convert protocols to pyLabRobot scripts for execution on the Hamilton liquid handler.

Completion Criterion: Demonstrate DNA purification protocol conversion from a generic LabOP to a Hamilton-specific protocol that passes Hamilton software checks.

- **Phase 2: Data Collection**

Description: Run the DNA purification protocol at two sites at least twice to generate protocol, execution, and measurement data for constructing a reproducibility artifact.

Completion Criterion: Create metadata and data needed for developing a reproducibility artifact in the next phase.

- **Phase 3: Data Preparation and Dissemination**

Description: Develop a dataset that includes data from Phase 2 and test cases developed in Phase 1 that can be used to reproduce the data. Develop a reproducibility assessment to determine whether additional test runs reproduce the results represented by the data.

Completion Criterion: Make reproducibility artifacts (dataset and test cases) publicly available with documentation.

Project timeline

The project is expected to take 350 in total over the course of 12 weeks, approximately 30 hours per week. For more details on the general steps see Figure 1.

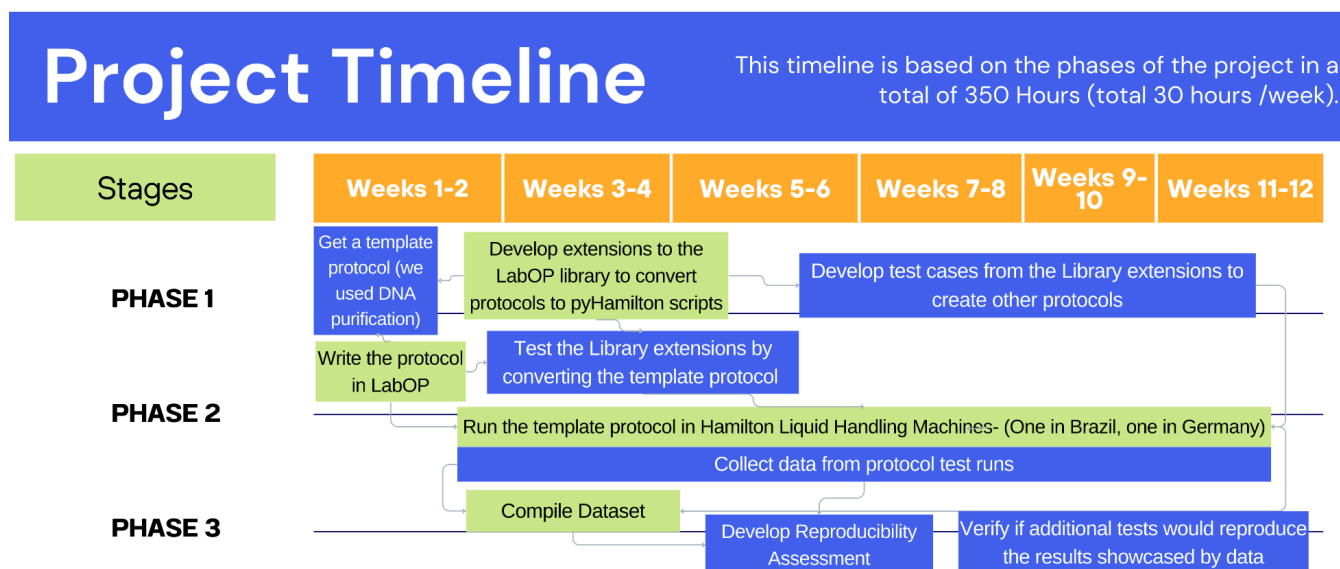


Figure 1: Project timeline representing the steps of each Phase. The data collection phase is expected to be longer than the others due to the tests being run in Distinct locations (Germany, Brazil).

References

- LabOP: <https://github.com/bioprotocols/labop>
- pyLabRobot: <https://github.com/PyLabRobot/pylabrobot>
- PyHamilton: <https://github.com/dgretton/pyhamilton>
- B. Bartley, J. Beal, M. Rogers, D. Bryce, R. P. Goldman, B. Keller, P. Lee, V. Biggers, J. Nowak, and M. Weston, "Building an Open Representation for Biological Protocols," *ACM Journal on Emerging Technologies in Computing Special Issue on BioFoundries and Cloud Laboratories*, in review.

- Emma J Chory, Dana W Gretton, Erika A DeBenedictis, Kevin M Esvelt. "Flexible open-source automation for robotic bioengineering." bioRxiv 2020.04.14.041368; doi: <https://doi.org/10.1101/2020.04.14.041368>, in review.

Biographical Information

- Name: Luiza Zucchi Hesketh Gomes (LuHesketh)
- University: Campinas State University
- Program: BSc Licenciatura Ciências Biológicas(4 year/ 8 semester)
- Expected graduation year: 2024
- Timezone: Brazil(GMT+5:45)
- Website: [linkedin.com/in/luiza-zucchi-hesketh-0020631a5](https://www.linkedin.com/in/luiza-zucchi-hesketh-0020631a5)
- Github: <https://github.com/LuHesketh>

I am an eighth-semester Bsc Biological Sciences student from São Paulo, Brazil. I have been interested in life sciences and technology since my early years of school.

I have acquired good programming knowledge by learning technologies like HTML, CSS, JSON, Python and Figma. I have built a few projects with the technologies I learned and they are listed on my website and on Github.

Recently I've been appointed as Executive Director of the Brazilian Association of Synthetic Biology and I've been actively seeking to improve collaborations between laboratories by sharing their common protocols for genetic engineering and molecular biology processes, and I believe that the best way to achieve that is to implement a common language for laboratory protocol design that could be used to represent & execute such protocols across diverse laboratories. Whether it is by a machine or a person(or even both), a laboratory protocol should be Straightforward and understandable, that's why I'm working with LabOP to write protocols for Hamilton liquid-handling robots.