INTERACTIVE BIOLOGY CLOUDLAB
SYSTEM ARCHITECTURE AND APPLICATIONS IN LARGE SCALE ONLINE EDUCATION

A DISSERTATION
SUBMITTED TO THE DEPARTMENT OF COMPUTER SCIENCE
AND THE COMMITTEE ON GRADUATE STUDIES
OF STANFORD UNIVERSITY
IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY

Zahid Hossain

December 2016

I certify that I have read this dissertation and that, in my opinion, it is fully adequate in scope and quality as a dissertation for the degree of Doctor of Philosophy.

(Ingmar Riedel-Kruse) Principal Adviser

I certify that I have read this dissertation and that, in my opinion, it is fully adequate in scope and quality as a dissertation for the degree of Doctor of Philosophy.

(David Dill)

I certify that I have read this dissertation and that, in my opinion, it is fully adequate in scope and quality as a dissertation for the degree of Doctor of Philosophy.

(Paulo Blikstein)

I certify that I have read this dissertation and that, in my opinion, it is fully adequate in scope and quality as a dissertation for the degree of Doctor of Philosophy.

(Michael Bernstein)

Approved for the Stanford University Committee on Graduate Studies

To my wife Sarita Hossain, and my soon to be born son Zareef Rafique Hossain

Abstract

Experimentation in life science is necessary for academia, industry and education, but many access barriers exist due to professional training, operation cost, maintenance, and safety requirements. High-throughput experimental equipment combined with cloud computing infrastructure has the potential to alleviate these access barriers through abstraction and time-sharing. Recently, a few commercial cloudlabs such as Transcriptic and Emerald Cloudlab have emerged, where highly standardized experimentation protocols are outsourced and executed semi-automatically based on the instructions provided by the users ahead of time without any means to interrupt or interact with an ongoing experiment. Therefore, a truly interactive biology cloudlab does not yet exist, whereas interaction is the key to exploration based science.

In this dissertation, we conceptualized an interactive biology cloudlab paradigm. The core of this conceptualization is a domain specific device, Biotic Processing Unit (BPU) that hosts a biological sample and allows interactive experimentation. Biological systems are particularly difficult to handle as they, unlike physical systems, may exhibit unpredictable natural variabilities. Therefore, to provide a better Quality of Service (QoS), we proposed a general method to automatically monitor an array of backend BPUs to check the underlying biological state. We implemented two different cloudlab architectures to support 1) nonreal-time chemotaxis experimentation with a slime mold, Physarum polycephalum and 2) real-time phototaxis experimentation with a single celled micro-swimmer Euglena gracilis. In the Physarum based cloudlab, users time-shared a set of pre-allocated BPUs for experiments that would last two days. In the Euglena based cloudlab, users were scheduled in a queue using an Automatic Call Distributor (ACD) system, where experiments would last ~ 1 minute. We compared these two architectures to draw general design rules, and recommended future cloudlab implementations based on the time-scale of the underlying biology.

We iteratively developed our cloudlab by deploying it in various educational settings with different interaction modalities: in graduate level university classes, middle and high school classes, and eventually through an MOOC course with over 300 students. To this end, we also implemented a modeling platform along with other HCI components to facilitate visual and data analytics to enable inquiry-based learning as mandated by the National Research Council. In conclusion, we have paved the way to make complex biology experimentation accessible to a broader audience - including researchers, citizen scientists, and learners alike - at low cost and scale.

Acknowledgments

This dissertation is a true testimony of the interdisciplinary collaboration that Stanford University fosters. I am eternally grateful to all my collaborators, for whom I have decided to dedicate a seperate acknowledgement shortly after this.

First, I would like to remember my parents, Amzad Hossain and Jahanara Begum for their eternal patience to bring me up in the face of serious financial hardship. A middle class family from Bangladesh, without social security or any retirement plans, would typically expect their only son to support the family after college. Instead, their relentless support, even in the times of despair, for my long and non-stop academic ventures were nothing short of an unrepayable sacrifice. I am extremely grateful to my sister Mariha Afroz, who have taken care of my parents while I was away from home since 2008 (starting with a MS degree from Simon Fraser University in Canada) to embark on a long academic journey.

I would like to thank my advisor Ingmar Riedel-Kruse for guiding me through this interdisciplinary Ph.D that not only involved serious engineering but also team management. Without his support, there would be no interactive biology cloudlabs. Besides, Ingmar has always been very compassionate and supportive towards me being an international student with stringent visa restrictions in the USA.

I am grateful to the Morgridge family for supporting my Ph.D through the Stanford Interdisciplinary Graduate Fellowship (SIGF). I am also grateful to NSF for funding the cloudlab project.

Lastly, and most importantly, I would like to thank my wife Sarita Hossain. Sarita is a brilliant accountant, who had excellent job prospects back in Bangladesh. But she left all of that to come support my Ph.D journey in the USA only to realize that she was unable to pursue her own career due to complex visa related restrictions. Yet, she is the one who encouraged me the most to finish this Ph.D. Though I can never repay her sacrifices, I would like to dedicate my entire Ph.D work to my beloved wife Sarita Hossain.

Relevant Publications & Collaborators

This dissertation stems from four selected publications [65, 63, 62, 64], which were written in collaboration with several co-authors (listed below) across three departments (CS, BioE and Education) and two institutions (Stanford and Northwestern University). Text appearing in this dissertation had great contributions from these co-authors. This body of work would not be possible without this interdisciplinary collaboration.

- **Stanford University, Computer Science:** Sean Choi and Stephen Koo
- **Stanford University, Bioengineering:** Alice Chung, Nathan Orloff, Xiaofan Jin, Honesty Kim, Jordan Shapiro, Chynthia Truong, Casey Litton and Ingmar Riedel-Kruse (PI)
- **Stanford University, Education:** Engin Bumbacher, Paulo Blikstein
- **Northwestern University:** Ashley Walter, Sachin Pradhan and Kemi Jona

Stephen Koo helped with the software development in Chapter 2, Casey Litton helped with both software and hardware development in Chapter 3. The middle school user study in Chapter 3 was conducted by Engin Bumbacher. Engin also helped with the data analysis of that middle school user study. Ashley Walter and Sachin Pradhan helped with running the third iLab user study from Chicago - Chapter 3.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Motivation for Biology Experimentation in the Cloud

Biological experimentation is important for academic research, industry, and education, especially with biology becoming increasingly quantitative and interdisciplinary. However, opportunities are lost because of steep access barriers due to costs, logistics, safety, and training requirements. Citizen science and crowd sourcing, in a virtual gamified environment, have already demonstrated how collective knowledge of even non-experts can make novel discoveries when wetlab logistics are lifted [11, 68, 76]. In many biological experiments, the actual wet lab steps are merely logistical, time-consuming, error-prone, and expensive due to specialized equipment needs. A computational scientist might only be interested in the final data, and have neither the expertise nor the access to the required equipment to carry out the physical experiment.

A cloud computing-like paradigm where experimental equipment is time-shared from a central location could address the above issues. On the one hand, life-science technology [72, 91, 110, 119, 13] is being continuously advanced with high-throughput experimentation equipment, but on the other, they are rarely designed to be accessed remotely and shared across multiple users concurrently. Recently, a few commercial platforms have emerged, such as Transcriptic [6] and Emerald Cloud Labs [1] that off-shore some experimentation in a centralized location where they are executed in a semi-automatic manner through a combination of batch processing and on-site scientists performing experiments. But with no opportunities for interaction, the exploration-based science that hands-on experimentation allows is lost. Thus the operation mode of the existing commercial solutions is essentially similar to outsourcing.

Interactive cloudlabs are also poised to solve significant educational challenges. Next Generation Science Standards (NGSS) [26] and National Research Council (NRC) [105] have recommended authentic inquiry-based learning in schools that require students to be familiar with true scientific practices - i.e. generate a hypothesis, run experiments, gather data, analyze data and perform modeling. The inquiry-based learning, however, is challenging to accomplish in practice. First, many schools do not have adequate wet lab facilities to afford hands-on experimentation. Second, there are strict safety concerns that exclude hazardous, yet interesting samples. Toward this, many remote non-biotech [60] labs (with real backend instruments) have been developed over the years, but they are not designed to be accessible by multiple users, and none exist for biology. Simply put, an interactive biology cloudlab neither exists for scientific research nor for education at present and the steep access barrier to biological experimentation is still there.

In this dissertation we aim to alleviate this access barrier and develop an interactive biology cloudlab by applying Human Computer Interaction (HCI) principles, but with real biology at the core. Therefore we term it more aptly as Human Biology Interaction (HBI). As a first step towards HBI, we implemented different types of biological experimentation devices, Biotic Processing Units (BPU), that allow users to execute experiments interactively and programmatically. A key insight is that many biological experiments follow strict protocols, which can be expressed in a computer programming language and consequently automated both locally or remotely. We then extended this notion further by integrating these BPUs in a centralized cloudlab - thereby time-sharing the facility with many online users at low cost. The ultimate goal here was to democratize biology experimentation through the economy of scale. We iteratively deployed and user-tested our cloudlab system by deploying in different educational settings. Education provides a very large, mostly homogenous, user base (there are over $\sim$ 15 million high-school learners in the USA alone [4]). The experimentation complexity in education, in comparison to scientific research, is modest and with the curricula being repeatedly offered, it provides our cloudlab technology with a key economic driver. Finally, we reflect upon our current implementations and conceptualize a general purpose BPU, and proposed design choices for a generalized cloudlab.

1.2 Contributions

The overarching contribution of this dissertation is the development and testing of an interactive, including a realtime-interactive, biology cloudlab technology. This was accomplished with the following specific contributions:

Conceptualized an Interactive Biology Cloudlab

Multiple online users can share a single lab facility to run experiments concurrently and interactively, thus enabling HBI. This notion is fundamentally different from the two existing commercial cloudlab offerings, Transcriptic [6] and Emerald Cloud [1], where experiments are run passively in batches, if experiments are automated at all, with no means for interaction. Towards this, we have developed two cloudlabs - one that supports non-realtime chemotaxis experimentation with a macroscopic slime mold *Physarum Polycephalum*, and the other that supports realtime phototaxis experimentation with a single celled organism *Euglena Gracilis*. Our cloudlab design scales linearly to support a massive number of users. We specifically made a business case with the Euglena-based realtime cloudlab and showed a million online users can be served every year at a modest cost and physical footprint (250 BPUs with a total 10 m^2 space requirement and < 1 cent/experiment).

Enabled Inquiry-Based Learning Within Online Education

We adopted an iterative design-based approach towards developing a cloudlab that ultimately enabled full inquiry-based learning [90] through online education at scale. We first deployed a *Physarum*-based cloudlab and user-tested it in a graduate-level university class, which successfully demonstrated that insightful discoveries could be made by even non-biology experts when wet lab logistics are lifted. With this encouraging result, we then developed a much more scalable *Euglena*-based cloudlab and tested it 1) in a university setting, with advanced data analytics, 2) in a middle school setting, in conjunction wth a simulation platform and 3) through a third-party online educational website. We then holistically combined all the components, (i.e., data analytics, simulation, live experimentation and pre-programmed experimentation) and deployed our cloudlab, for the first time, within a MOOCs course (OpenEDx, $>$ 300 students). This deployment demonstrated how complex science practices and inquiry-based learning, as mandated by NGSS and NRC, can be delivered within a single platform through the Internet and at scale.

Proposed a Generalized BPU Design

We took lessons from all our previous implementations and formalized an abstract design of an interactive experimentation device, the Biotic Processing Unit (BPU). It's a *domain-specific* device that can handle a certain class of experiments, similar to how a Graphics Processing Unit (GPU) is *domain specific* to a certain class of parallel algorithms yet interoperates with the rest of the computer system through a central CPU and memory bus. A different implementation of a BPU then handles a different experiment type. However, importantly, the design allows for both realtime and non-realtime interactions with concurrent users. Biological systems often exhibit natural and unpredictable variabilities, which makes them uniquely challenging to provide a quality of service (QoS) in comparison to physical systems. Therefore, we also discuss a general framework to automatically monitor the current biological state using *Information* theory, and how it can be used to maintain QoS in a cloudlab.

Proposed Generalized Cloudlab Architectures

Based on previous deployments, we proposed a general cloudlab architecture with user and BPU management that can be deployed in multiple scenarios at the same time using virtual partitioning - i.e. the same cloudlab can be deployed for a school in Chicago and a MOOC at the same time. We also proposed two general strategies for resource allocation and scheduling experiments among a set of backend BPUs that depends on the response time-scale of the underlying biology.

1.3 Dissertation Structure

This dissertation is based on three selected peer-reviewed publications [65, 63, 64] and one submitted manuscript [62]. We present these manuscripts *as-is* in different chapters: Chapter 2 [65], Chapter 3 [63] and Chapter 4 [62]. The supplementary references in these chapters correspond to the original published supplementary materials and therefore we omit them in this dissertation for brevity. Some parts of of Chapter 5 and Chapter 6 were discussed in the aforementioned publications but a larger part awaits future publications. The chapters are summarized as follows:

- **Chapter 2:** We designed and implemented the first interactive biology cloudlab based on the chemotaxis of a slime mold *Physarum*, where the stimulus response was non-realtime. A user study, blended within teaching a graduate-level university class, demonstrated how insightful discoveries could be made by even non-biologists when wetlab logistics were lifted.

- **Chapter 3:** Here we designed a realtime system to support phototaxis experiments with a single-celled organism, *Euglena*. We deployed this system in various educational settings to user-test and validate different modalities of experimentation and lesson delivery. In this study, we also implemented a simulation platform to juxtapose with real experimentation. In this chapter, we developed several key components that eventually led to a cloudlab that can be deployed at scale, for example, 1) auto-monitoring of the underlying biology to provide biological fault tolerance, 2) live and pre-programmed experimentation for intuitive exploration and more systematic investigations, and 3) scalability analysis for MOOC deployments, and a business case analysis to support a million users per year with a modest cost and physical footprint.

- **Chapter 4:** We deployed a Euglena based cloudlab within a MOOC mini-course for the first time, by holistically combining all the components: experimentation, data analytics, visual analytics and simulation within a single platform. Thus, for the first time, we were able to deliver an end-to-end authentic inquiry-based learning environment [105] through the Internet.

- **Chapter 5:** We generalized the notion of a *domain-specific* biology experimentation device (BPU), and discussed how underlying biology could be automatically monitored for QoS. Along these lines, we discussed a specific framework to monitor the Euglena-based system, and then discussed a more general framework to quantify any stimulus-response-based system using *information* theory.

- **Chapter 6:** Here we propose a general architecture for a cloudlab, which can also be replicated outside Biology. Through user and BPU management, a single cloudlab can be deployed in multiple scenarios concurrently using virtual partitioning. Finally, we propose two strategies for resource allocation and scheduling depending on the stimulus-response time scale of the underlying biology.

- **Chapter 7:** In this concluding chapter we summarize our work briefly and discuss many potential future research directions for Education and Learning Analytics, HCI, and extension of the cloudlab system itself toward future deployment in large-scale scientific research.

Chapter 2

Non-realtime Interactive Cloudlab

This chapter is based on the following publication [65] ...

> Zahid Hossain, Xiaofan Jin, Engin W. Bumbacher, Alice M. Chung, Stephen Koo, Jordan D. Shapiro, Cynthia Y. Truong, Sean Choi, Nathan D. Orloff, Paulo Blikstein, and Ingmar H. Riedel-Kruse. *Interactive Cloud Experimentation for Biology: An Online Education Case Study*. In **Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems**, **CHI** '15, pages 3681–3690, New York, NY, USA, 2015. ACM.

2.1 Abstract

Interacting with biological systems via experiments is important for academia, industry, and education, but access barriers exist due to training, costs, safety, logistics, and spatial separation. High-throughput equipment combined with web streaming could enable interactive biology experiments online, but no such platform currently exists. We present a cloud experimentation architecture (paralleling cloud computation), which is optimized for a class of domain-specific equipments (biotic processing units - BPU) to share and execute many experiments in parallel remotely and interactively at all time. We implemented an instance of this architecture that enables chemotactic experiments with a slime mold Physarum Polycephelum. A user study in the blended teaching and research setting of a graduate-level biophysics class demonstrated that this platform lowers the access barrier for non-biologists, enables discovery, and facilitates learning analytics. This architecture is flexible for integration with various biological specimens and equipments to facilitate scalable interactive online education, collaborations, research, and citizen science.

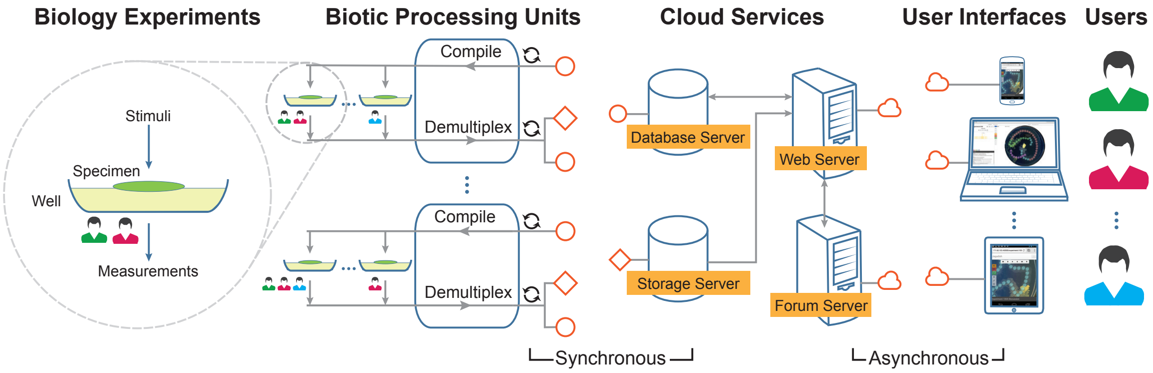


Figure 2.1: **We developed a general architecture for a *cloud experimentation* system that allows multiple users to optimally share high-throughput equipment online to run many interactive biological experiments in parallel and to collaborate**. Experiments are carried out in wells and can be shared by multiple users. Many such experiments are executed in parallel on a individual Biotic Processing Units (BPUs), multiples of which operate autonomously and synchronously with two clock cycles. One clock polls the central server for a set of currently scheduled instructions that are multiplexed from several users; the BPU then compiles and executes these instructions. On the other clock cycle, the BPU takes experimental readings (for example, imaging a well or measuring the current temperature) and demultiplexes these data for different users before sending them back to the central database (state data) and storage server (bulk data). Priority is given to one clock when both cycles overlap, based on the specific type of experimentation. Users access the system over the Internet without the need to book a time slot, and they can perform interactive experiments, i.e., change the experimental instructions multiple times throughout the course of the experiment. The web server provides the user interface for experimentation, while the forum server hosts social networking services (chat capabilities, a question and answer forum, etc.). Overall, this architecture is optimized to coordinate asynchronous user actions with synchronous equipment cycles, which enables a convenient user experience while optimally utilizing parallelized equipment at the same time.

2.2 Introduction and Motivation

Interacting with biological systems via experiments is important for academia, industry, and education, but many access barriers exist that are related to training requirements, cost, safety and logistics. Consider, for example, a computational scientist lacking the hands-on wet lab training to test her own hypotheses experimentally, in which case the final data is of primary focus while the actual act of performing the experiment is merely logistical. Similarly, access barriers arise in life-science education where traditional teaching labs are too costly or time consuming [71], or where online courses [34] do not include lab sections [19, 117]. On the other hand, citizen-science and crowd-sourcing projects have demonstrated how non-scientists can make relevant scientific contributions [33, 103], especially when users can design experiments that are centrally executed in batch by a technician [76]. Hence, enabling many more people to directly interact with microbiology in various contexts by designing and executing biology experiments while abstracting away the skills required for their actual execution would be very powerful.

In this paper we introduce the concept of *interactive cloud experimentation* for biology, which enables multiple users to execute live biology experiments over the Internet by efficiently sharing the necessary resources (Fig. 2.1) seamlessly. This abstracts away the complex logistics of experimentation and allows users to rather focus on the data analytics, which enables a broader interdisciplinary participation in life-science research and education. The notion is similar to the well-established framework of cloud computing. Ongoing advances in life-science technology [72, 91, 110, 119] are continuously pushing the boundaries of high-throughput experimental technologies with the required automation and parallelization capabilities but they are rarely designed to be accessed remotely and shared across multiple users concurrently. Our proposed *cloud experimentation* aims to enable high-throughput technologies to be shared across many users over the Internet concurrently while allowing iterative interactions, i.e. a user will be able to make changes to her experiments based on the current state of the investigation. This strategy is also critically distinct from previous remote experimentation efforts in the academia [51, 61, 82], which were primarily designed for real-time feedback control of one instrument by a single user. We are not aware of any fully automated cloud or remote labs for biology, although various educational remote labs exist in other science and engineering disciplines [9, 43, 54, 55]. Few “cloud lab" companies have emerged recently that execute biology experiments in a centralized location [6, 1, 5]. As of the writing of this manuscript, none enables users to run experiments online interactively (mailing DNA samples to be cloned and mailed back has a very different quality of interactivity than what we provide). But according to these companies’ website interactive experimentation is envisioned for the future, which argues for additional relevance of our present work.

We propose a general systems architecture for *cloud experimentation* for biology that can scale to large numbers of users and diverse applications in a cost - and logistics - effective manner (Fig. 2.1). We prototyped this architecture as a “mini- cloud," which served as a lab component for a graduate-level biophysics class and conducted a user study within both research and educational contexts. During this study, we assessed the effectiveness, general logistics, and HCI aspects of our system, particularly focusing on answering the following questions: (i) Can *cloud experimentation* be successfully integrated in education? (ii) Does it enable true open-ended research, especially by lowering access barriers for non-biologists? We then discuss limitations of the present system followed by some practical lessons learned, and scalability issues. Finally we allude on how such a system can be leveraged for learning analytics and discuss future directions. The novelty of our work was not in the individual parts but in the whole combination of existing technologies to derive the aforementioned system. The key contribution was to implement and analyze the first end-to-end use-case of a truly interactive biology cloud lab for education. This work is targeted toward a broad audience of engineers, biophysicists, educators, and learning researchers.

2.2.1 System Architecture and Biotic Processing Unit (BPU)

Biological investigations are diverse (Fig. 2.1), and unlike general purpose computing, there exists no clear basis (e.g. binary 1s and 0s) for executing all types of experiments. Therefore, we adopted a domain-specific philosophy [112] to design conceptual high-throughput hardware - *Biotic Processing Unit* (BPU) - to handle only a specific type of experiment with a specific set of instructions. Swapping out this hardware allows

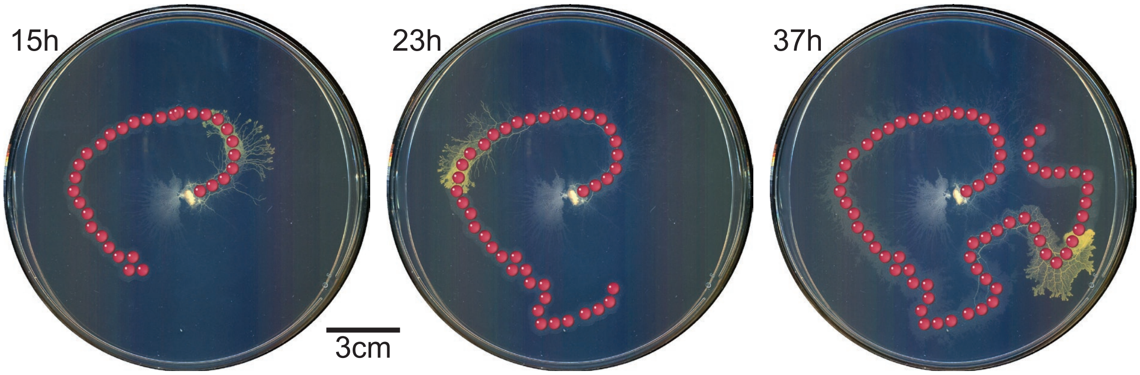


Figure 2.2: **The spatiotemporal chemotactic response of the slime mold *P. polycephalum* (yellow) to an oatmeal solution trail (red) offers a scientifically interesting experimental paradigm with high-dimensional input/output spaces.** Food trails of liquid oatmeal pipetted onto an agar surface lead to growth behavior in which *P. polycephalum* follows the trail at a speed on the scale of 1 cm/h. Once the organism encounters one or more food sources, it optimizes its branches in terms of path length and fault tolerance. During this process, the organism sends pulsating fluid flows at a frequency of approximately 0.5 min^{-1} throughout this network, achieving both mass transport and global communication. Note how *P. physarum* follows the trail through sharp turns while occasionally deviating from the trail. See SOM1 for animation. This experiment was executed automatically by the BPU implementation in Fig. 2.3.

execution of different types of experiments. The goal then is to design a general architecture of a cloud system that can exploit and integrate these hardware under a common platform using a set of protocols while maintaining some key properties: 1) *scalable*, 2) *time-shared* and 3) *available* at all times, meaning that users can access and run experiments anytime without having to book a time slot. In this section we will mainly discuss the design criteria of the key hardware component of our system – *Biotic Processing Unit* (BPU) – and how it interacts with the central servers.

At the backend, experiments are executed in multiple BPUs, and we formally define one of these as an automated hardware that houses a specific set of biological specimens in one or more isolated compartments, termed *wells* (Fig. 2.1). Each of these *wells* is shared among one or more users, who collaboratively run an independent experiment where the stimulus and measurements can be characterized by the dimensionality of the corresponding spaces. Similar works in the past [61, 82], though different engineering disciplines, have mostly shared their piece of hardware by requiring a user to book a time slot in advance. This approach is not suitable for biology as a single experiment may require an extended period. An alternative approach could be batch-processing, whereby all the experimental instructions from all the users can be aggregated and run concurrently without any further interactions with the user, although it is desirable that users are able to run experiments in several interactive cycles. We enable this interactivity by defining a time scheduling protocol that is carried out jointly by the central server and a microcontroller inside a BPU.

Time Scheduling To achieve both interactivity and concurrent execution of experiments, we let users provide instructions in discrete *blocks*, where each *block* consists of a small sequence of instructions that need to be performed at a future time. Critically, the instructions within a single *block* can be executed in any order, i.e a *block* is a declarative program [81]. A user is allowed to add *blocks* at any time as well as edit or delete any existing *block* that has not been executed yet. Total time ordering of the instructions is achieved by scheduling these *blocks* at different points on a time line. With this restriction, a BPU is then able to aggregate *blocks* from all the users that pertain to a certain time and interleave them in a way that is optimal for a batch execution by the BPU, which may involve a complex actuation sequence of various parts. We term the life time of a single experiment as an *experimental session*.

Microcontroller A microcontroller must be integrated within each BPU because the instructions, which must be domain specific like the BPU itself, need to be interpreted at the BPU site. This microcontroller may operate *synchronously* with two independent clock cycles (Fig. 2.1). The first clocks polls the central server to find appropriate instruction *blocks* from different users, interprets them, and executes them as discussed earlier. The second clock acquires the measurement data (output) from the various *wells* and demultiplexes them for different users before sending them back to the storage server (Fig 1).

Cloud Services and UI Users submit experimental instructions in *blocks*, which needs to be performed at a future time, interactively through a web application that communicates with central database server asynchronously. Instructions from several users are queued in this database server until BPUs are ready to poll in a synchronous manner as discussed earlier. This buffering aspect of the database server helps connect the asynchronous user interactions in the frontend with the synchronous BPUs at the backend (see Fig 2.1). The current state of the experiment, which resulted from executing past instructions, are relayed to the user whenever they are available. Note that the frontend UI also need to be domain specific, although not as specific as the BPU since, for example, a single UI can control a large class of experiments involving stimuli and measurements via multidimensional spatiotemporal arrays (such as chemical pipetting and time-lapse imaging, respectively). We omit discussion related to user and data management as any standard framework for these purposes can be adopted.

Practical Implementation of the Architecture

Biology In order to assess the practical utility of this cloud architecture, we implemented a mini-cloud system for educational purposes that allowed students in a graduate-level biophysics class to execute open-ended biology experiments like real scientists. We named our system *Jagadish* after the Bengali polymath Sir Jagadish Chandra Bose who had worked extensively in both wireless communication [24, 42] and biology [23]. In this section we discuss our implementation at a broader level while we welcome the reader to see the appendix and supplementary material for all the details (Appendix A). This implementation also permitted instructors to perform learning analytics [12, 15, 18]. We selected the chemotactic response of the slime mold *Physarum polycephalum* (Fig. 2.2, SOM1) as our experimental paradigm. *P. polycephalum* is a single-celled, multi-nuclei,

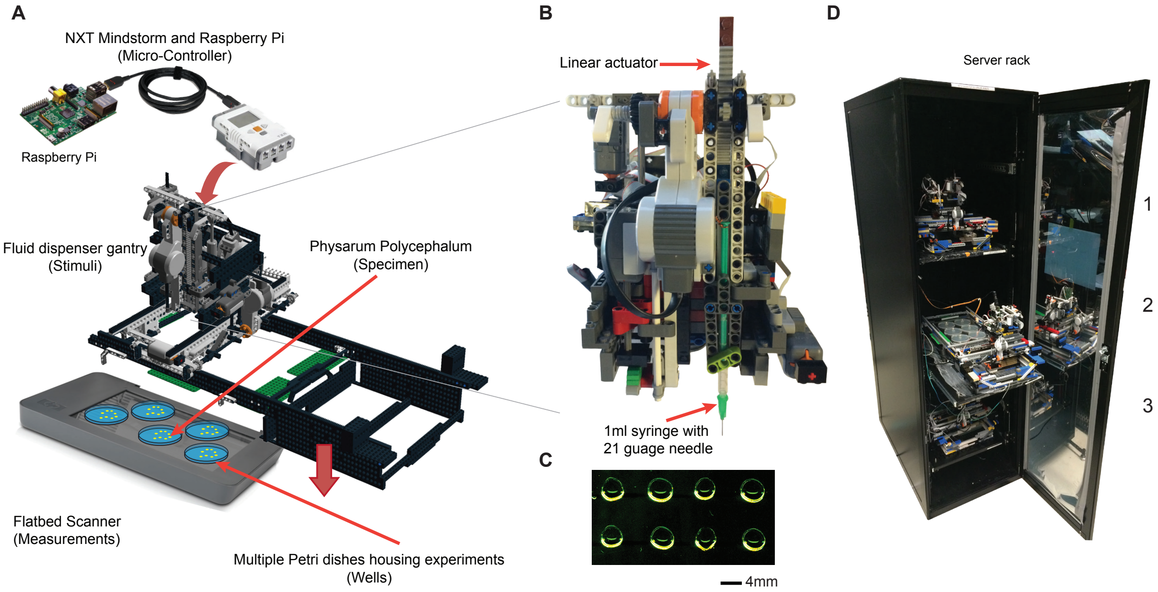


Figure 2.3: **We developed an instance of a BPU for automated fluid handling and imaging to execute versatile biology experiments.** (A) A Lego-based gantry for positioning a pipette (stimuli) in the x, y, and z planes is placed atop a flatbed scanner (measurements). Biological materials are housed in standard plasticware (*wells*) placed on this scanner. A Raspberry Pi device combined with an NXT robotics kit controller forms the microcontroller component of the BPU, communicating with the central server, compiling instructions on the fly, and demultiplexing the raw scanned image data for appropriate users (Materials and Methods). The small size and the built-in WiFi capabilities of Raspberry Pi also enable wireless operation of the BPU inside a controlled environment. (B) The Lego-actuated pipettor is made from a standard syringe that reliably pipettes volumes on the order of 10 μL. (C) Example of eight successive water droplets dispensed and imaged by the BPU. (D) A standard computer server rack provides housing for three BPUs (1-3).

cytoplasmic organism that forms active and dynamic tube networks to search for food [7, 8, 84, 107, 108, 122]. These interesting macroscopic growth and foraging phenotypes represent both multicellular behavior and development [86], inspiring many questions for further investigation in areas ranging from basic biology and biophysics to abstract computation [7].

Prototype BPU To automate these experiments with *P. polycephalum*, we built a BPU that carries out automated liquid handling and imaging tasks to support multiple experiments inside standard Petri dishes (Fig. 2.3A; SOM2 and Methods for details). In this system, each Petri dish represents a *well* that houses a single experiment that can be shared by multiple online users. The input space in this case is the spatiotemporal dispensing of liquid oatmeal solution, which prompts the chemotactic response of *P. polycephalum*, while the output space is primarily a time-lapse sequence of images that captures these responses. The oatmeal solution is dispensed by a motorized gantry, which we prototyped using a Lego NXT robotics kit, that positions a liquid pipettor (Fig. 2.3B) on top of a regular flatbed scanner. Standard plasticware and Petri dishes containing various liquids and biological materials are placed on this scanner, which carries out time-lapse imaging of the specimens from below. One BPU fits six Petri dishes (90 mm in diameter) or five standard rectangular plastic wells (85 mm x 127 mm). Imaging rate and resolution were set to 6 images/h at 300 DPI. A Lego NXT robotics kit combined with a Raspberry PI mini-computer board [3] served as the controller for this BPU. For the backend, we stacked three such BPUs into an enclosed regular computer server rack (Fig. 2.3D). Multiple hardware and software fail-safe mechanisms were implemented to ensure reliability and long-term durability for a 10-week deployment in a classroom.

Backend Servers We implemented a scalable backend server (Fig. 2.4A) to connect these BPUs (Fig. 2.3) to the Internet. The backend server system consists of a web server, a database server, a chat server (to allow discussion among collaborators and communication with the system administrator), and a storage server (primarily for bulk data, such as time-lapse images captured by the scanner). The entire software stack is open-source. Each BPU, which houses multiple experiments concurrently, polls the database server synchronously at a regular interval of 10 min before compiling and executing all pending pipetting instructions from several users in parallel.

User Interactions A system admin would start an experimental session by preparing Petri dishes (6 to 18 depending on the size of the dishes) that are inoculated by *P. polycephalum* at the center unless a special initial condition was specified by a student ahead of time. For example, towards the end of the course, one student requested the system admin to start his experiments with multiple tiny isolated pieces of *P. polycephalum* instead of one at the center (discussed in details later). Experiments pertaining to a single student were distributed across multiple BPUs to ensure there are other experiments to continue with in an event of a catastrophic BPU failure. Students were notified through emails along with secret keys, unique for each student, once all experiments were loaded. A student could then access her experiments using the given key, which she could optionally change later, from her account's homepage (see Fig 2.4A). This experimental session would last two to three days in which time there would be no further manual intervention. During this time students were able to manipulate and investigate the states of their experiments concurrently through a web based UI (discussed below) at any time and place without having to book a time slot. All experimental data were archived when the session expires and students were able to investigate these later at any time using the same UI.

We developed the frontend UI as a cross-platform web application (Fig. 2.4, SOM2-4 for details). After selecting an experiment from the dashboard (Fig. 2.4A), the user is directed to the web interface (Fig. 2.4B), which is essentially a time-lapse movie player showing the selected Petri dish. The user can now select various UI elements (Fig. 2.4C) to play back and investigate the collected time-lapse data interactively (Fig. 2.4D) as well as program new *blocks* of instructions based on the current state of the investigation. All experimental instructions are *visually programmed*, i.e., liquid stimuli are drawn as desired output pattern directly onto the time-lapse images, where stimuli can be dispensed as single drop at a time or as a trail of fluid (Fig. 2.4B): a *temporal brush* leads to a trail that grows incrementally over time instead of dispensing all of the liquid at

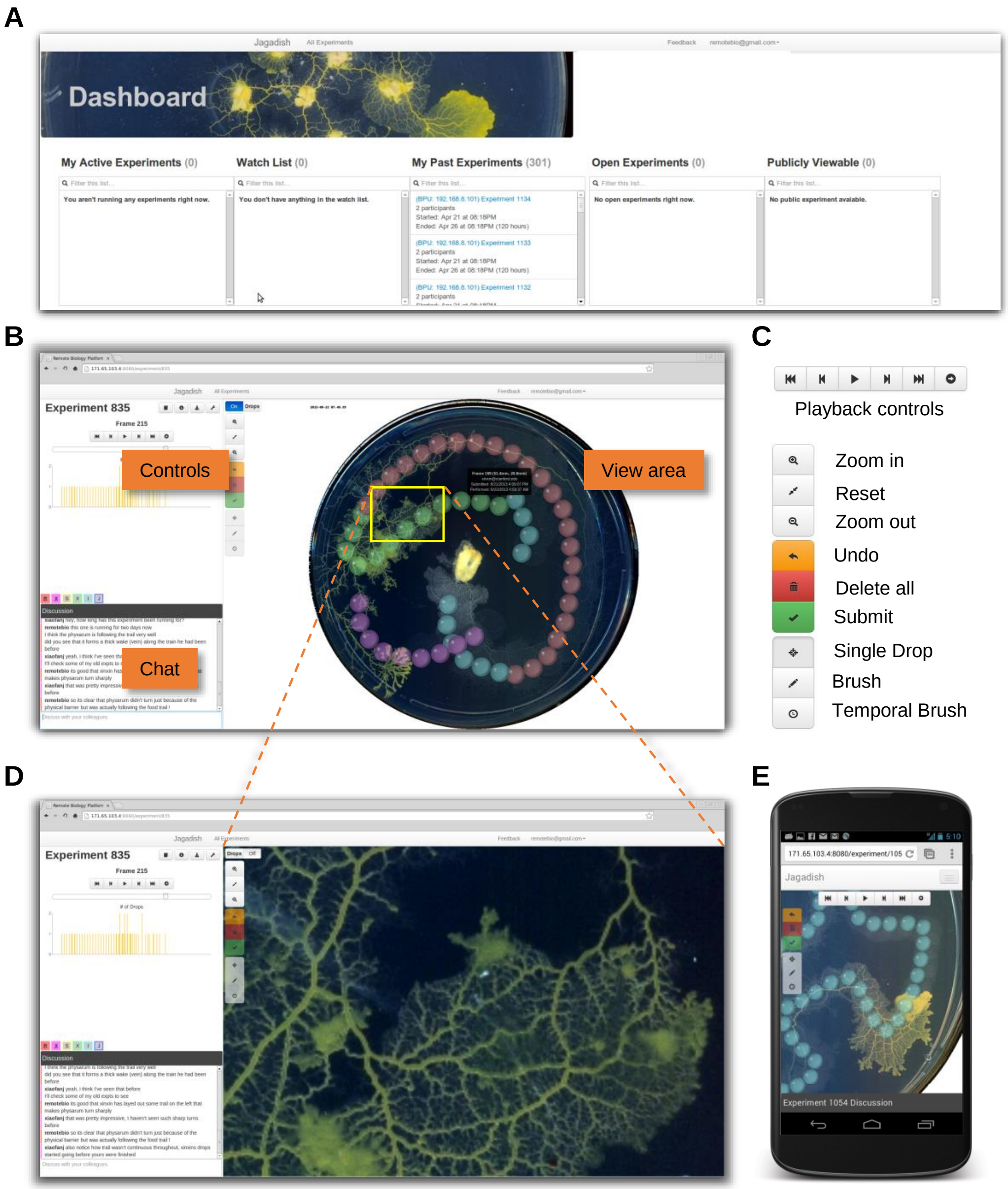

A
Jagadish
All Experiments
Feedback
remotebio@gmail.com
Dashboard
My Active Experiments (0)
Watch List (0)
My Past Experiments (301)
Open Experiments (0)
Publicly Viewable (0)
You aren't running any experiments right now.
You don't have anything in the watch list.
(BPU: 192.168.8.101) Experiment 1134
2 participants
Started: Apr 21 at 08:18PM
Ended: Apr 26 at 08:18PM (120 hours)
(BPU: 192.168.8.101) Experiment 1133
2 participants
Started: Apr 21 at 08:18PM
Ended: Apr 26 at 08:18PM (120 hours)
(BPU: 192.168.8.101) Experiment 1132
2 participants
No open experiments right now.
No public experiment available.
B
Experiment 835
Frame 215
Controls
View area
Chat
Discussion
C
Playback controls
Zoom in
Reset
Zoom out
Undo
Delete all
Submit
Single Drop
Brush
Temporal Brush
D
Experiment 835
Frame 215
Discussion
E
171.65.103.4:8080/experiment/105
Jagadish
Experiment 1054 Discussion

Figure 2.4: **The frontend web-based UI allows the user to program instructions through visual programming, to analyze experimental data remotely, and to collaborate with other users.** (A) The dashboard view displays five lists of experiments, enabling easy navigation through all past and currently running experiments owned by the user as well as access to experiments made open or publicly viewable by other users. (B) The desktop interface allows the user to perform experiments (e.g., program timing and position food drop placements), browse and magnify the data, and chat with other users. This layout was optimized for experiments with *P. polycephalum* (Fig. 2.2). (C) The relevant user interaction control buttons from (B). (D) The user can zoom-in and pan on high-resolution images provided by the server on demand (in contrast to a simple digital zoom). (E) Mobile UI with a customized layout for small displays. See SOM2-4 for more detail.

once. Being a web application, our UI runs on most systems, including mobile phones (Fig. 2.4E, SOM4). We implemented adaptive streaming of time-lapse images to account for slow Internet speed and display size. Finally, we also added a live chat capability and a collaborative editor (similar to Google Docs) to allow discussion and collaborative experimentation among multiple users. In this implementation we allowed users to optionally set their own experiments to be viewable publicly, i.e. other users can look at the results but not necessarily modify anything.

2.3 User Study

We evaluated our implementation of a *cloud experimentation* system in the light of its utility in education and research (mostly qualitatively as it is the first of its kind) by integrating it into an interdisciplinary graduate-level biophysics course titled, “Biophysics of Multicellular Systems and Amorphous Computing." The course objective was to learn modeling approaches for understanding multicellular biological pattern formation. During this user study we automatically logged every user interaction (i.e., position and timing of mouse click as well as any other data entry) in the backend database server upon full written consent of the students. The frontend UI sent a small packet of ping data to the backend server every second as long as the experiment page was in the foreground or the user interacted with the system in some way, allowing us to compute session times more reliably. Users were able to interact with 30 UI elements; we logged each element according to the timestamp, IP address, browser type, experiment identification number, image sequence visible on the screen, and exact viewing geometry (window size, image size, zoom level, center of image on the window). These logs enabled us to reconstruct log-in activity as a movie based on a set of very compressed data (see Fig 2.7C).

We carried out three one-on-one interviews with each student: an initial interview at the end of week 2 (out of 10 weeks), an interview in the middle of the term (week 5), and an exit interview during week 10. We also collected written feedback and bug reports with every homework. During the second interview, we learned that the students wanted a system feature that allowed them to view each others’ experimental data, but not necessarily interact with it. We immediately implemented this critical design feature.

2.3.1 Cloud Experimentation for Education

From an educational point of view we were particularly interested in whether this platform could serve as a wet-lab component to enrich a theory course. Eleven online experimentation sessions were conducted over 10 weeks (Fig. 2.5A). Each session lasted 2-3 days, during which each student ran 2-6 experiments concurrently; the system was robust throughout the course. Four graduate students enrolled from different backgrounds: two from bioengineering, one from electrical engineering, and one from applied physics. The latter two students had no prior biology wet-lab experience. This small student population allowed us to follow each student closely (aided by the data-logging capabilities of the system itself), to collect feedback multiple times during the course, and to conduct post-interviews. This provided us with an in-depth understanding of how, in essence, such a system could aid education and research without having to deal with complexities due to scale in an initial deployment.

The system was integrated into this course in three main phases: familiarization and guided homework (2 weeks), hypothesis iteration (4 weeks), and final project (2 weeks). Students spent the first two weeks exploring and becoming accustomed to *P. polycephalum* and the UI, ran experiments and measured the growth rate and fractal dimension of *P. polycephalum* (Fig. 2.5B) [70] . All students reported that they found the UI to be fairly intuitive and simple, with the exceptions of a few minor bugs and confusing UI elements that were fixed immediately. During the second phase, students were asked to develop and test experimental hypotheses, for examples, whether *P. polycephalum* can be made to split into two parts, or whether it can distinguish between different-sized food sources (Fig. 2.5A, SOM5a-c). During the final project phase, students worked in pairs on one of these hypothesis with the goal of bringing quantitative experimental data and biophysical modeling together; we will discuss one of these projects in depth in the next section (Fig. 2.6).

Student feedback indicated that compared to conventional labs, this *cloud experimentation* platform lowered the threshold of entry to biology experimentation in three major ways. First, it empowered non-biologists to perform real experiments without concerns about wet-lab training and safety. For example, the electrical engineering and the physics student respectively stated: *"It was a matter of playing around"* and *"...if you really require me to take one month to train for it, then I would probably just skip that [class]."* All students reported that their initial system contact was easy yet unstructured and exploratory while they worked through the guided homework (Fig. 2.5B). These initial playful interactions led to more systematic self-driven explorations that gave rise to different qualitative hypotheses. Second, the system abstracted away all of the wet-lab details and allowed the students to concentrate on experimental strategies and data analysis; as expressed here: *"When I worked in a wetlab, I would have to prepare a whole bunch of agar plates, ...cleaning a whole bunch of stuff, ...lots of like chemical mixing so you get the right concentration. .... that's pretty time consuming stuff that is sort of logistical, so it's nice to not have to do it."* Third, the system provided a critical convenience by allowing students to remain continually engaged with their experiments from any place at any time: *"To place a droplet every 30 minutes, you would have to be up 24 hours, you could not even take a nap."* The logged user interaction data confirmed that students ran experiments through mobile phones while on the move and sometimes even past midnight.

A pressing question in educational research is whether computer simulations can substitute for real experiments [16, 19, 47, 71, 92, 111, 126]. In interviews, the students expressed a clear preference for the latter. First, real systems have implicit narratives attached to them through which students can be more appreciative of and connect more naturally to the system; knowing that something is real changes student motivation. As one student expressed *"Well, it's real, that's why it is so exciting! And so, then you're genuinely interested in asking more questions. So, I think you learn a lot more."* Second, real systems promote open discovery, especially in the context of biology. This context can also lead to unexpected experimental observations (see next section, Fig. 2.6A), which would not be possible in a virtual environment (e.g., PhET [92]). Ultimately, each instructional medium (conventional lab, virtual lab, online lab) has its own benefits [111], whereas combined or hybrid approaches normally achieve better learning gains [19, 38].

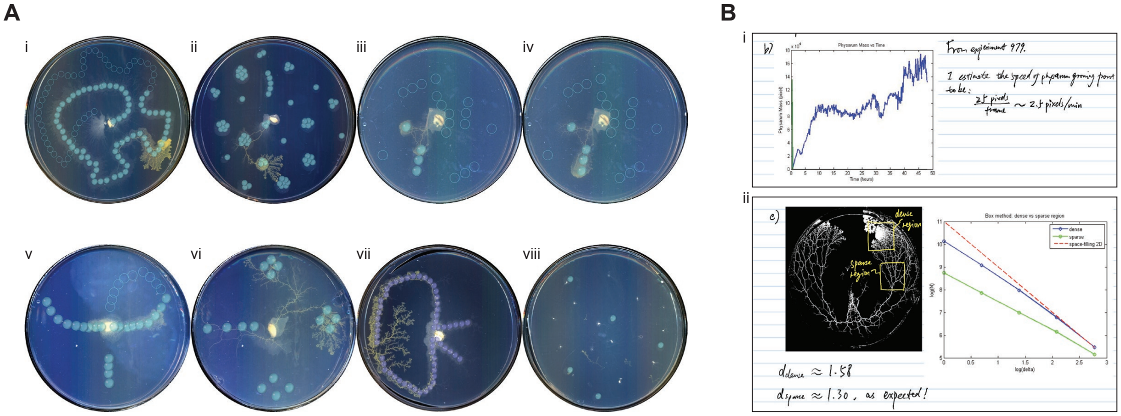


Figure 2.5: **Integrating our *cloud experimentation* system into a lecture-based course on the biophysics of multicellular systems empowered biology experimentation by abstracting away hands-on skills and logistics.** (A) Examples of experiments carried out by students throughout the course. The students progressed from initial open-ended, playful exploration of broad behavioral aspects, to developing more specific hypotheses about the observed behaviors, to setting up experiments to test these hypotheses. (i) How well does *P. polycephalum* follow a path and make sharp turns? (ii) How does *P. polycephalum* travel between islands of food? (iii-iv) How does *P. polycephalum* decide where to go when presented with different numbers of food droplets (in this case, three versus one)? (v-viii) Changes in the experiments carried out by one student throughout the course. See SOM5a-c. (B) Sample homework from students. (i) Quantification of apparent area (growth) of *P. polycephalum* over time. (ii) Computing the fractal dimension from binarized *P. polycephalum* images via the box-counting method.

Hence, an exciting future research area is how to achieve optimal synergy between *cloud experimentation* and existing educational media [19].

2.3.2 Cloud Experimentation for Research

Can *cloud experimentation* enable key aspects of the scientific process and lead to genuine scientific advancements? Our evidence suggests a positive answer: The applied physics student who had no prior wet-lab or biology experience, went through multiple hypotheses and exploratory phases throughout the course (Fig. 2.5Av-viii). During this process, he observed that *P. polycephalum* often does not stop growing even when all food stimuli are depleted (Fig. 2.5Avii). This observation led him to run a controlled experiment with two Petri dishes, one with and one without food. He then made the unexpected observation that a tiny isolated fragment of *P. polycephalum* moved across the dish in a “worm"-like, random, self-avoiding path. The student emailed a screen shot (Fig. 2.6A) to the system administrator with the text *“... I’ve attached the picture of an example of what I mean here - you probably did it by accident last time but it gave birth to a worm-like small strain of physarum moving around which is really interesting and potential more suitable for modeling!"* During the remainder of the course this student (together with another student) employed several more rounds of experimenting and modeling to explore this initial observation and to understand how the

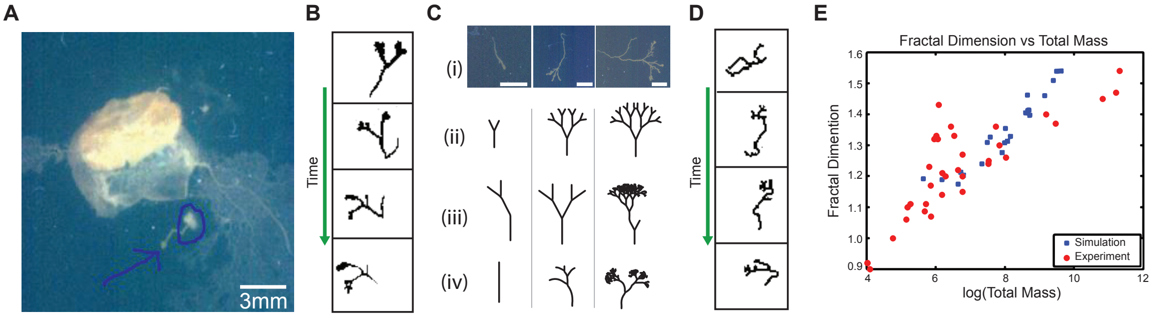


Figure 2.6: **Chance observation by one student and subsequent iterative biophysical model development and experimental testing represent genuine discovery and the scientific method.** (A) Email attachment by student sent to the system administrator illustrating the observation that small pieces of *P. polycephalum* behave like a “worm" rather than a fractal (arrow and circle drawn by student). (B) Experimental time sequence of medium sized *P. polycephalum* demonstrating branching dynamics and morphology. (C) The modeling approach taken by this student (jointly with another student) as part of their final project. (i) *P. polycephalum* data at slightly different scales. The student was particularly struck by his observation that organisms with smaller masses had fewer branches, which seemingly went against the notion of “self-similarity across scales" in fractals that had been discussed earlier in the course. Scale bars, 3 mm. (ii-iv) Models developed by students of increasing mathematical complexity and visual correspondence with experimental data: (ii) Static symmetric bifurcation model. (iii) Static random bifurcation model. (iv) Dynamic growth-retraction model that also includes conservation of total mass. (D) Time sequence of model (iv) for medium-sized fragments captures realistic branching dynamics and morphologies (compare to B). See SOM6. (E) In order to go beyond a visual comparison, the students compared the fractal dimension of the *P. polycephalum* fragments and total mass between the model (blue) and the experimental data (red), and obtained reasonable quantitative agreement.

branching dynamics and morphology of the organism changes over time and with the organism’s size (mass) (Fig. 2.6B-D). The student asked the system administrator to setup experiments with smaller fragments of *P. polycephalum* of varying size. The administrator fulfilled this request (Fig. 2.5Aviii), which went beyond the experimental paradigm intended for the course. The student was particularly struck by his observation that organisms with smaller masses had fewer branches (Fig. 2.6Ci), which seemingly went against the notion of “self-similarity across scales" in fractals [70] that had been discussed earlier in the course. The students’ models iterated from symmetric (Fig. 2.6Cii) to probabilistic (Fig. 2.6Ciii) branching models, and eventually developed a dynamic model with several phenomenological rules (Fig. 2.6Civ), such as the model would conserve mass and branch tips would grow as long as there is mass available, otherwise growth stalls and tips shrinks to the last branching point when stalled for too long, thereby providing mass for other growing tips. This model generated visually realistic dynamic morphologies with multiple simultaneously expanding and retracting branches (Fig. 2.6D). The students also noted limitations in their model, for example that the ratio of retraction and outgrowth rates did not match with the experimental data. In order to go beyond a visual comparison, the students also compared the fractal dimension of *P. polycephalum* fragments versus total mass (Fig. 2.6D, SOM6) between the model and the experimental data, and obtained quantitative agreement [19].

To assess the novelty of these findings, including modeling, we conducted a literature search that failed to uncover any direct reports of this behavior (see Appendix for details on this literature search). While we suspect that the mass-dependent morphology of *P. polycephalum* may be known [8, 46], no publication has discussed this feature or reported systematic investigations of its dynamics. The students will publish a more detailed analysis of their model separately. This case study aptly demonstrates that the logistic abstraction offered by *cloud experimentation* enables individuals without biology training to perform meaningful experiments. Critically, our system empowers users to iterate computational models side-by-side with experiments. This achievement matches design principles where educational researchers have recently advocated to “selectively expose" students only to learning-relevant features, while technical aspects are hidden unless needed to accomplish the learning goal [17].

2.4 Limitations

Any cloud experimentation system is necessarily limited to a specific sub-set of possible biology experiments, due to the biological material present and the restricted state and measurement space of the automation. In other words, the equivalence of Turing completeness for biology experimentation is not obvious. For example, in our implementation of the BPU, the user was restricted to study *physarum* chemotaxis to oatmeal solutions, and there was no mechanism for providing an alternative stimulus such as shining light patterns to study phototaxis. Furthermore, while backend servicing is inherent to all cloud systems, biological systems would typically require extra care. For example, *physarum* experiments were loaded manually at the beginning of an experimental session but in principle this type of bottleneck can be mitigated by automating the loading process itself as BPUs are domain specific.

2.5 Discussion and Conclusion

The primary goal for this project was to understand the effectiveness and potential of *cloud experimentation* for biology, and we share some practical lessons for future implementations: (1) Building and utilizing such cloud systems requires integrating diverse expertise ranging from biology, mechatronics, database, web interfaces, education, and more. A modularized approached with well-defined and minimal I/O interfaces between modules enables parallel development and upgrading of individual components. Parallel modules (such as multiple BPUs) provide overall robustness against component failure, and we propose distributing experiments from the same user across multiple BPUs. (2) Constructing and maintaining the BPU and it’s biological content is the most challenging of those modules as specimens must be stable and responsive to defined stimuli over a long time, while all other components (electronics, web servers, etc.) are straightforward by today’s standards. Thus, we recommend to first identify a suitable BPU and experimental specimen, assess and test their robustness and logistics towards the desired application, and only then implement any of the other components. (3) Any (online) experiment has limitations, but users will likely request additional features. We

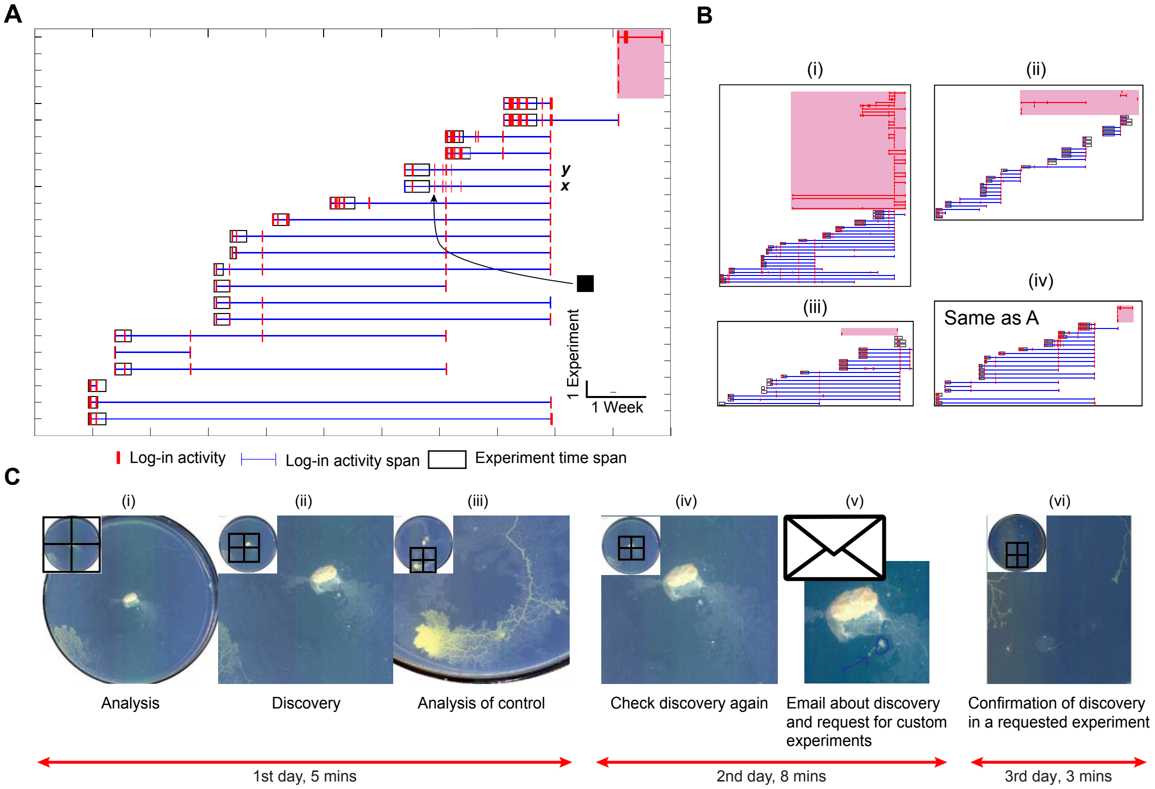


Figure 2.7: **Rich user interaction data can be mined with visualization for student assessment and learning research.** (A) The coarsest level of visualization in which the interaction history (experimentation and data analysis) of a single student is shown over the entire course in chronological order. Red ticks, log-in activities; horizontal blue line, first to last log-in activities for a particular experiment; pink background, log-in activities on experiments belonging to other students. (B) Comparison of user history for all four students revealed substantial differences in user history (Biv is the same as A). (C) Complete visual reconstruction of what a student did and saw on the UI enabled us to investigate how discoveries (Fig. 2.6) came about (see full movie in supplementary material SOM7). (i-ii) Student zooming in when the small-scale behavior of *P. polycephalum* was first seen (experiment marked with an **x** in (A); session is marked with a black box). (iii) Immediate switching to concurrently running a control experiment (marked **y** in (A)) and zooming indicates that the student is searching the available data for similar behavior. (iv) Logging back into the original experiment for re-analysis followed by (v) emailing the system administrator with an attached image. (vi) The student notices similar behavior in the custom setup experiment on day 3, confirming discovery. See SOM7 for full animation.

suggest that the developers and system administrators aim to be (reasonably) flexible, especially while these cloud systems are under development, and the UI should allow to request extra features. For example, we allowed the users to request different placements and sizes of *P. polycephalum* seeds on the dish by emailing a sketched image to the administrator. (4) There are many open questions regarding what constitutes an optimal UI for performing experiments online. For example, during our study users were online only for very short bursts of time, rendering our chat ineffective for inter-user communication; an online Q/A forum, similar to Piazza or Stackoverflow [116], would have been a better choice. Hence critical UI features should be user tested well in advance, while a few non-critical ones can be beta-tested during the actual usage.

The proposed *cloud experimentation* architecture is scalable as BPUs are independent of each other and more BPUs can be added easily on demand, while failure of one does not effect any other. For example, In our implementation, we had a single BPU failure during the 10 weeks user study, which we were able to recover within 2 hours while the rest of the system was still live. Even though BPUs are domain specific, one can easily run different types of experiments that happen to fall within the state space of a BPU. For example, we employed the same prototype BPU to run gene induction experiments on 24-well plates where users could program instructions using a simple scripting language (data not shown).

Tools for Learning analytics [98] come for free with *cloud experimentation* as we were able to track every student's activities in full detail as demonstrated in Figure 2.7. An instructor could potentially use these activity logs to visualize emerging patterns, for example Figure 2.7A,B reveals that one student was very keen at looking back to this previous results as well as others' result before starting his own experiments. Figure 2.7C demonstrates how we were able to reconstruct the exact sequence of actions made by a student in form of a video from just logged data that gave us an intriguing window into the moment the physics student made his first "worm like" observation. Thus the application of *cloud experimentation* in learning analytics is clear and we intend to provide a detailed study in a separate paper.

Wide adoption of *cloud experimentation* depends on the availability of suitable BPUs and interest groups of early adopters (most likely educators). Much existing (automated) life-science equipments do not lend themselves directly as BPU, since stimuli and measurement are often not integrated within one machine (such as separated liquid-handling robots [72] and motorized microscopes [91]), or equipment control is closed during a run (such as real-time PCR machines [123], which do integrate stimulation and measurement, but where protocols cannot be altered during a run). BPUs can be developed at the high-end professional level as well as at the do-it-yourself scale. The presented Lego BPU is itself functional as a mini-cloud, and is supported by increasingly low-cost robotics, such as those used for 3D printers. The do-it-yourself and open source communities have made significant contributions to larger development efforts [2]. We ultimately envision horizontal evolution of BPUs to address diverse applications, and that great value is placed on designs that are simple, easy to assemble, and modular, which over time will lead to higher-throughput and lower-cost systems. For universality, BPUs ultimately need to be coupled (e.g. output of one BPU is fed as input to another), and a lot of future work will hinge on accomplishing this in a scalable manner. Since many biological applications occur at sub-mm scales, even massively parallel BPUs in the future would require only small footprints and researchers have already demonstrated how micro-organisms could be interacted with remotely in form of *Biotic Games*, housed in small micro-fluidic chambers [95]. Reasonably strict protocols for BPUs and UIs will support this development. We speculate that early adoption will come from educational applications, with primary life-science research following later, since educational experiments can be less sophisticated, include a much larger and homogeneous user base, and no novel discoveries must be made during the use of the platform. For the foreseeable future, these cloud systems may remain local (such as within a school), although an educational school supply company could offer a central service similar to that offered by some remote labs in engineering education.

In summary, we have developed a system architecture for biology *cloud experimentation* that is optimized for sharing parallelized high-throughput equipments among many users over the Internet in a *scalable* manner. The main distinguishing features compared to previous work in other engineering disciplines [9, 43, 51, 54, 55, 61] are that all experiments are *interactive* and are *available* all the time to the users seamlessly, while they executed in a high-throughput manner using time-sharing at the backend. Our key contribution was to successfully implement this architecture and analyze its utility for applications in education and true biological discovery. We also discussed future directions for further development along this line and deployment scenarios along with possibilities in learning research. BPU building instructions and open source software are included (SOM8; SOD1; SOS1) for implementation and further development.

Chapter 3

Realtime Interactive Cloudlab

This chapter is based on the following publication [63] ...

> Zahid Hossain, Engin W. Bumbacher, Alice M. Chung, Honesty Kim, Casey Litton, Ashley D. Walter, Walter, Kemi Jona, Paulo Blikstein, and Ingmar H. Riedel-Kruse. *Interactive and Scalable Biology Cloud Experimentation for Scientific Inquiry and Education.* **Nature Biotechnology**, Accepted 4th Oct,2016. [In Press].

3.1 Abstract

We developed a biology cloud experimentation platform that is realtime interactive, fully automated, low-cost, and scalable to millions of users. A cluster of back-end instruments (biotic processing units, BPUs) enable distributed and fault-tolerant phototaxis experiments with micro-swimmers (Euglena). This online research paradigm provides access to scientific experimentation for learners and researchers alike as demonstrated by educational use cases in college and middle-school.

3.2 Introduction

Many access barriers to life-science experimentation exist for academic and commercial research, mainly due to professional training needs, cost of equipment purchase and operation, and safety considerations [104]. The computational cloud and time-sharing paradigms [35, 48] have recently inspired the development and deployment of biology cloud experimentation labs for research, such as commercial platforms that can execute experiments semi-automatically [59, 104] and EteRNA provides experimental feedback for citizen scientists [76]. However, these platforms still face limitations, such as relying on batch processing with no opportunity for realtime interaction while the experiment is running, hindering the exploration that hands-on experimentation allows, and long experimental turnaround taking days.

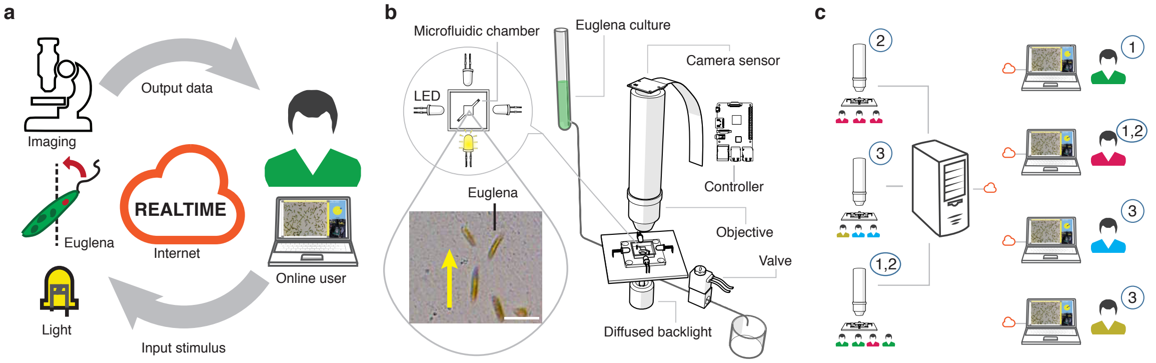


Figure 3.1: **We developed a biology cloud experimentation platform that scales cost-effectively to large user numbers and versatile applications.** (a) The experimental paradigm allows online users to send light stimuli to biological substrates, e.g., phototactic Euglena cells, and observe the response in real time. (b) The back-end hardware consists of a webcam microscope targeted at a microfluidic chip positioned between four LEDs. The chip contains Euglena that can be replenished automatically from an upstream reservoir via an electronic valve. This stimulus-biology-sensor module includes its own micro-controller and is conceptualized as a Biotic Processing Unit (BPU). Scale bar, $50\mu m$. (c) An array of BPUs is monitored and managed by a central server. Both users and BPUs belong to different groups (circled numbers), and users are routed to the appropriate BPU (same group) optimizing for wait time and BPU quality (Supplemental Text S2, Fig. S5).

Cloud labs are also poised to help solve significant educational challenges. Familiarity with advanced scientific practices and "authentic inquiry" [31, 90, 102] are imperative for K-12 and college education ("Next Generation Science Standards" (NGSS) [102, 26], but are difficult to achieve in real-world classrooms given logistics and cost [31, 105]. In addition to traditional physical hands-on labs, virtual and remote labs (VRLs) have been successfully deployed recently, with each modality having its distinct advantages given educational goals and situational context [38, 60, 120, 22, 99]. Physical remote labs for life science education are comparably underdeveloped [60], in large part due to the associated logistics of specimen handling. We previously developed, demonstrated and deployed the first educational biology cloud lab with slime mold chemotaxis experiments [65], which was suited for non-realtime interactions, but did not scale cost-effectively given back-end logistics and turnaround time.

Here we conceptualized, implemented, and validated a biology cloud experimentation platform (Fig. 3.1) that (1) enables the types of inquiry mandated for professional science and educational purposes; (2) has a low entry barrier and can be used even at the middle-school level; (3) is realtime interactive; (4) has a fast result turnaround time within minutes; (5) is fault tolerant against biological variability and failure; (6) scales to millions of users worldwide from a design as well as economic viewpoint; (7) has a large exploration and discovery space; and (8) generalizes to many other experiment types.

3.3 Interactive biology experimentation online

Our cloud platform focuses on the photoresponsive behavior of Euglena Gracilis, a single-celled organism 50 Âţm in length. While swimming forward, it rolls and wobbles around its long axis to scan all directions for light with its single "eye spot" (Fig. 3.1a). Euglena are commonly used in hands-on biology education [80, 83, 32, 77] and are relevant for basic research [67, 96, 97]; food, chemical, and fuel production [74]; and as biosensors [87].

Experiments are executed on a cluster of Biotic Processing Units (BPUs) [65], instruments that combine sensors, biological material, actuators, and a micro-computer (Fig. 3.1b). Each BPU consists of a webcam microscope containing a microfluidic chip with four attached light-emitting diodes (LEDs) that provide directional light stimuli to Euglena (Movie S1), which are cultured in reservoirs and supplied to the microfluidic chips via automated valves as needed. The micro-computer controls the LEDs, streams live video, post-processes data, and communicates with the central server (Supplemental Text S2; Fig. S5). We adopted the task scheduling concepts of High-Performance Computing (HPC) [44] to design the central server. This server assigns BPUs and remote users according to a non-exclusive group allocation policy (Fig. 3.1c; Section 6, Supplemental Text S2.4,), handles distinct BPU types, routes experiments to the best-suited BPU, and optimizes wait time through load balancing.

Via a web interface, users choose a specific BPU or are auto-routed (Fig. 3.2a) to execute experiments in realtime live mode or in asynchronous batch mode (Movie S2). The live mode user interface (Fig. 3.2b) employs a virtual analog joystick to control intensities of four LEDs (Fig. 3.1b) to induce directional light stimuli to Euglena; two live video streams show the microscopic Euglena responses and the macroscopic LED actuation. In batch mode (Fig. 3.2c), the user designs and uploads a program that contains instructions for time sequences of LED intensities. The back-end server automatically tracks shape and motion of all motile cells and overlays these data on the captured videos (Fig. 3.2d, Supplemental Text S2.5). The video, stimulus, and track data are stored for future download and analysis.

Live mode enables open-ended, realtime interactive exploration of Euglena biophysics followed by quantitative substantiation in batch mode. A user can test Euglena's response to changes in light direction and intensity and then observe variability among traces (Fig. 3.2d). The prevalent behavior is negative phototaxis, but localized tumbling and changes in cell morphology are also observable (Movie S3). We characterized the system by executing periodic light on/off experiments in batch mode, measuring the time constants for cell alignment with light $\tau_1 = (6.7 \pm 2.4s, N = 6, \text{mean} \pm \text{std throughout})$ and subsequent off-light orientation decay $\tau_2 = (9.9 \pm 2.6s)$ (Fig. 3.2e). We defined responsiveness to quantify how well Euglena aligned with light after 15 s of light exposure (scale 0-1, for random to perfect alignment, Supplemental Text S2.2). This responsiveness score also depends on light intensity that exhibits Hill-type characteristic (Fig. 3.2f). Hence, experimenters can investigate Euglena's response to changes in light direction and intensity on the time scale of seconds, study Euglena long-term behavior over weeks (Fig. 3.3), and record and download this data for offline analysis (Movie S2).

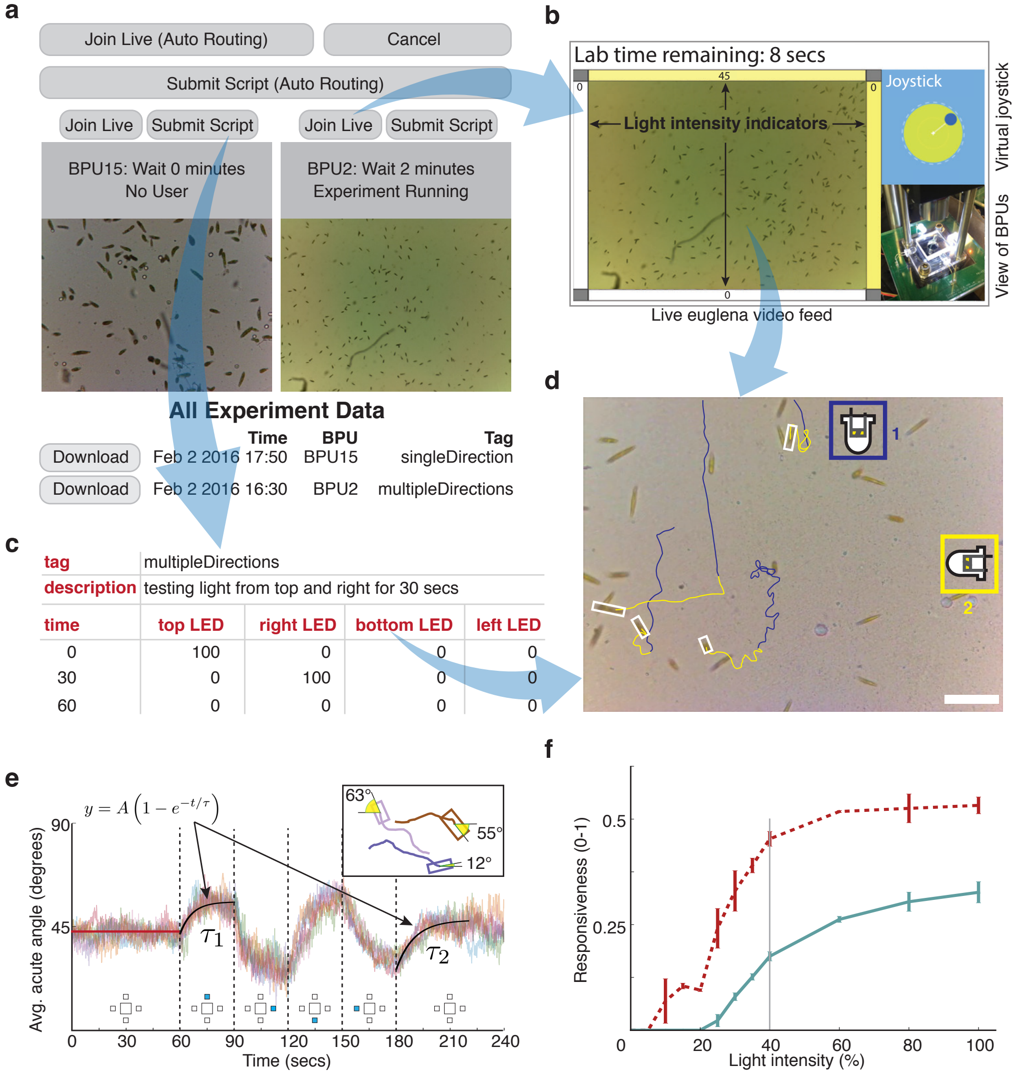
3.4 Robust, cost-effective, and dynamic scaling of BPU clusters

To make this BPU cluster cost-effectively scalable and tolerant against failures in hardware, software, and “bioware” (Fig. S5), we extended HPC to include biology by auto-monitoring its state: The system submits batch experiments to each BPU every hour (Fig. 3.2e) to measure three variables: cell density, motility, and light responsiveness. Population density and responsiveness monitored over 10 days can be stable (Fig. 3.3a), undergo micro-ecological fluctuations (Fig. 3.3b), and be susceptible to external ambient light cycles (Fig. 3.3c). This biological variability emphasizes a key challenge of any cloud lab, i.e., to consistently provide a pre-specified experimentation experience. User testing revealed that responsiveness above ~ 0.4 was easily recognizable (Supplemental Text S2.2), providing a quantitative target for good BPU performance. (Even lower responsiveness highlights interesting and noticeable Euglena behavior (Fig. 3.2f)). BPUs not meeting specifications can often be recovered by automated flushing (Fig. 3.3b); organisms and chips are replaced every ~ 4 week, leading to a maintenance effort of ~ 10 min/week/BPU (Supplemental Text S3.2). Under current maintenance protocols, individual BPU performance was good $\sim 61\%$ of the time, and continued experimentation does not decrease BPU performance or Euglena responsiveness (Supplemental Text S3.1). Each BPU can handle >100,000 experiments/year for ~ 1 cent/experiment (~ 4 min/experiment; setup and maintenance cost of $\sim$ \$1000/BPU/year), with negligible wait time for randomly accessing users. Dynamic addition of BPUs ("hot swapping”, Fig. 3.3d) or queuing of batch experiments increases throughput (Supplemental Text S3.3). Running a cluster with six BPUs guarantees the availability of at least one good BPU 99.5% of the time at an average availability of 3.6 BPUs, with users automatically routed to good BPUs.

3.5 Educational use cases

We evaluated the platform in three educational contexts encompassing and illustrating various aspects of future usage in education and even research (Fig. 3.4). We primarily assessed (1) whether the technology works robustly, (2) whether it can be operated even by middle-school students, and (3) whether it achieves the key elements of best laboratory practice as described in America’s Lab Report [102], i.e., integration into the flow of instruction, alignment with process and content learning goals, and enablement of student to engage in reflection and discussion. The cloud lab was always embedded into regular instruction and scaffolded along the main phases of the inquiry cycle [90]; more details on study design and outcomes are provided in Supplemental Text S4.

First, we studied whether university students taking a professor-led theory based biophysics class could successfully carry out experiments and sophisticated quantitative data analysis from home in a self-paced manner (Fig. 3.4a,b, Movie S2). Over 14 days, ten students, working individually, completed a homework project focusing on concepts regarding micro-swimmers, diffusion, and low Reynolds number hydrodynamics [93]. Using the live mode (Fig. 3.2d), students explored Euglena light-response behavior and made cells swim along geometric paths (Fig. 3.4a). Students were able to self-discover semi-quantitative relationships, e.g., reporting that the “fraction of Euglena participating in the directed motion seems to increase as you



All Experiment Data

	Time	BPU	Tag
Download	Feb 2 2016 17:50	BPU15	singleDirection
Download	Feb 2 2016 16:30	BPU2	multipleDirections

tag	multipleDirections
description	testing light from top and right for 30 secs

time	top LED	right LED	bottom LED	left LED
0	100	0	0	0
30	0	100	0	0
60	0	0	0	0

Figure 3.2: **The cloud lab enables biophysics experiments with time scales from seconds to minutes in live and batch modes.** (a) Schematic of landing webpage presenting choice of available BPUs, choice of interactive (“Join Live”) or batch mode (“Submit Script”), and previously recorded experimental data (“Download”). (b) The live mode provides a virtual joystick to control the intensities of the four LEDs. Euglena response is fed back in real time through a live video stream; a secondary live video stream shows the BPUs in action. (c) An example of preprogrammed instructions for batch mode. (d) An example of an experimental result in response to a light stimulus sequence from top to right (blue, yellow) (see a script in c) showing Euglena swimming traces extracted automatically. Scale bar, $100\mu m$. (e) Response and relaxation time constants of Euglena to align with light direction at the maximum light intensity (orientation measured in acute angles $0-90°$; see inset: 0 and 90° correspond to perfect alignment). (f) Population level orientation responsiveness (0-1, from random to perfect response) to varying light intensities; procedure as in e. Results for two BPUs highlights biological variability when cultures are in different states; vertical bar marks approximate transition towards saturation (3 trials performed for each of the 10 data points in each of the two curves, error bars are 1 stddev).

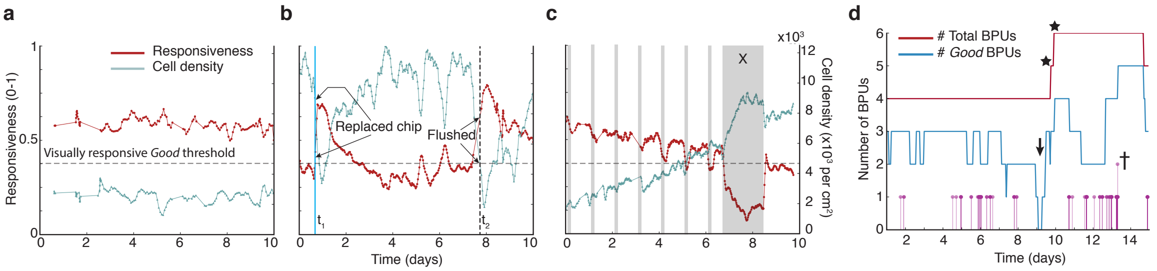


Figure 3.3: **Auto-monitoring framework and BPU multiplexing enable scaling and robustness of the cloud lab over weeks and demonstrate the potential for micro-ecological long-term studies.** (a) Example of ideal BPU performance over 10 days. All BPUs are auto-monitored by a central server that runs automated experiments hourly, measuring cell density and light responses (protocol as in Fig. 3.2e). (b) An example of BPU where chip and organisms were renewed (t_1), followed by slow deterioration over one week; subsequent flushing (t_2) recovered cell density and responsiveness. (c) Example influence of ambient external light on a BPU. Lights were shut off at night (gray bar) leading to periodic cellular responses, or the whole setup was shielded from external light (X) leading to a more pronounced effect followed by recovery. (d) Performance summary of BPU cluster over 14 days (deployment during user study Fig. 3.4a-c), the number of total BPUs (red) versus good BPUs (blue). Note performance dip (arrow), followed by self-recovery; BPUs were added dynamically to the cluster ($*$). Cluster capacity was $\sim 10,000$ experiments/2 weeks (area under the blue curve); only 116 experiments were executed [area of pink lines indicating student activity; note occurrence of the simultaneous demand for 2 BPUs (dagger)]. Even during peak demand, availability was never exceeded. A 5-h time-weighted filter was applied to data in a-c. (BPU cluster was run continuously over more than 2.5 months, where between 3 and 6 BPUs were online all the time; BPU availability and goodness as described in the text.)

hold the joystick longer, and depending on the intensity of the light.” They performed back-of-the-envelope analyses regarding Euglena size ($\sim$ 50m), speed ($\sim$ 50m/s), and drag and propulsion forces ($\sim$ 10pN) [93], experimentally confirming theoretical lecture content. Students then analyzed self-generated large-scale batch data (Fig. 3.2c, typically hundreds of auto-tracked cell traces in a 1-min movie; Movie S2) in MatLab, testing two hypotheses: (1) Do Euglena behave like passive Brownian particles? 90% found the expected relationship of root mean square displacement versus time was violated, and the apparent diffusion coefficients were too high given cell size (student example: expected: D$\sim$ 0.01m^2/s; measured D$\sim$ $2,000$ m^2/s; Fig. 3.4b left). (2) Does the population-average velocity differ between dark and light conditions? 60% reported that cells slowed when the light was on, 10% reported that cells sped up, and the others found no significant differences (student example: 26 ± 12 m/s (N=389) vs. 13 ± 10 m/s (N=431), off vs. on, respectively; Fig. 3.4b right). A decrease in velocity for increased light is expected [96], but results may vary given experimental conditions. These results demonstrate that one-minute experiments provide students with hundreds of auto-traced cells supporting sophisticated statistical analysis. The logged data revealed that students accessed the system at their own convenience at day and night and engaged in different modes of experimentation, from “playful” (as self-described by multiple students) to more systematic testing of one or multiple light directions and intensities. Student’s feedback and the fact that they each ran 11 ± 6 experiments (three were sufficed for the assignment) indicates that the platform affords ease of experimentation and incentivizes self-driven exploration. Students’ feedback also captured many items that motivated this project, including ease of exploration and gaining intuition (30%); ease of obtaining and analyzing a large batch dataset (30%); and minimal manual labor, need for technical understanding, and logistic effort, allowing focus on thinking (50%). Examples of feedback include “It was fun to play around with real organisms ... didn’t require thinking about the set up”; “Playing for a few minutes gave me some intuition”; “Text mode allows more detailed and controlled tests”; “Very little rote labor time, spent most time thinking!”

Second, we studied whether real-time experimentation could be integrated into middle-school classroom settings and whether it could be combined with simulation-based platforms to support sophisticated model exploration practices as prescribed by the NGSS [27, 36](Fig. 3.4c,d, Movie S4). During a 50-minute class period, 27 students (7th and 8th grade, three classes) working in pairs executed the following activities. In one class all pairs ran their own live experiments, in two classes the live experiment was projected to the front wall and operated by one student while the whole class discussed and suggested joystick movements. This generated the hypothesis that Euglena move away from light. Then, student pairs tested this hypothesis by measuring the percentage of Euglena cells moving away from light in previously recorded movies. The entire class discussed possible mechanisms by which Euglena may perceive and respond to light. Student pairs then engaged with a 3D biophysics modeling environment (Fig. 4c) in which a Euglena cell was represented as cuboid surging along and rolling around its long axis. This model had three user-defined parameters (Surge, Coupling, Roll) with the instantaneous pitch velocity being proportionally coupled to the light amount entering through one body side. Depending on parameter choices, the model mimics many light-responsive behaviors, including positive and negative phototaxis, straight travel versus meandering swimming paths, and even chaotic behavior (Movie S4). Students explored the function of these three parameters by iterating among self-chosen parameter configurations and then running and stimulating their model through joystick operations, with the overall goal of matching a pre-recorded swimming path. Students ran 19±4 simulation experiments; all students found fitting parameter configurations [21]. Cluster analysis of the activity logs (Fig. 3.4d) suggests three dominant strategies of students' model exploration: (1) systematic change of one parameter at a time followed by exactly one test experiment (40%); (2) alternating between multiple cycles in this systematic stage, followed by extended experimentation with a fixed parameter configuration (30%); and (3) unstructured transition between changing zero, one, or multiple parameters simultaneously (30%). These patterns are consistent with the literature on students' productive model explorations [78]. Students engaged in generative and productive discussions, that led to content-aligned discoveries such as the roll parameter being required for the cell to "sees in every direction" or methodological discussion about how the real Euglena differs from its model. Post-tests revealed that students learned the concept of Euglena phototaxis (90% correct) and engaged in scientific argumentation.

Third, we studied whether this cloud lab could be operated and curated through existing third-party educational content management systems that would allow for wider dissemination (Fig. 3.4e,f, Movie S5), and whether the batch mode feature would be suitable for middle-school. We chose the iLabStudio.org platform [58], which enables teachers to create personalized lesson content around online physics and chemistry experiments and to manage student progress. We implemented a general API and a corresponding iLab batch interface (Fig. 3.4e). During two 50-minute class periods on successive days, 34 students working individually or in pairs (8th grade, two classes) carried out the following activities. Students watched pre-recorded videos of interactive experiments and then engaged in an open classroom discussion generating hypotheses of how Euglena would react to student generated light stimuli. Students responded with "moving to the light" (60%)", moving "away from light" (20%), or described more complex behaviors (20%); some provided an explanation, such as the "need for photosynthesis" or that the "light might cook them" (both are correct depending on light intensity). To test their hypotheses, students then designed and ran batch experiments (29 total), i.e., entering intensity, duration, and direction of light stimulus. The chosen stimulus sequences revealed versatile experimental designs, including systematic variation of light direction or intensity, testing of multiple variables in sequence, and seemingly less structured designs (Fig. 3.4f). Students justified their rationales from "raise the intensity" to "put random numbers." We characterized 60% of the designs as sufficiently systematic to test for the influence of light intensity, direction, or both. Students and teacher discussed experimental designs and results as they were delivered sequentially from the experimental queue. Based on their own data, students reported moving to the light (25%); away from light (45%); and no directional response (30%). These heterogeneous results arose in part because some students did not choose high enough light intensity levels to induce noticeable negative phototaxis. When students afterward considered how to improve their experimental designs; 50% suggested investigating the effect of light intensity more closely. When asked "What did you think of this experiment?", 85% expressed liking it, 30% explicitly mentioning Euglena or living organisms.

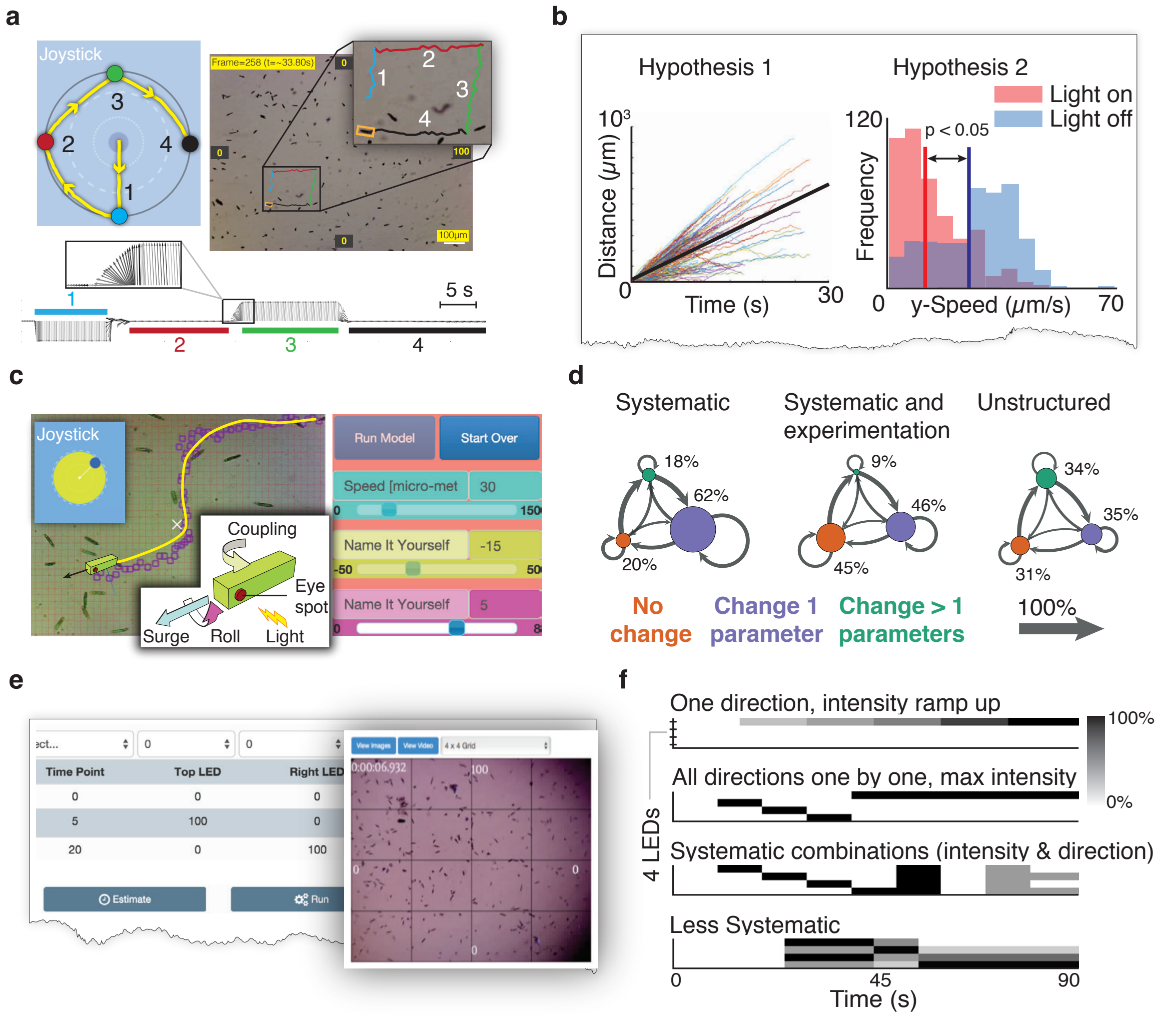
From these use cases, we conclude that this platform ran robustly and that we successfully deployed an experimentation paradigm that did not exist in the classroom before, i.e., realtime interaction with microscopic cells on the seconds timescale, which additionally supported complex, quantitative data analysis and modeling. This should be contrasted with current instructional standards and school lab practices, i.e., passive and qualitative observation of living cells under a microscope; fixed slide samples, videos, or pictures being even more common; in most sophisticated and rare scenario students observe the population level aggregation of Euglena in a petri dish under external light over the course of 15-30 minutes [83]. Ideally, five to ten live and batch mode experiments are combined to enable initial free form exploration followed by controlled experimental design. We note that new opportunities for mining of educational datasets emerge ("learning analytics") [21, 52] as logging user activity data on such platforms is easier and more scalable than in traditional physical labs. For example, revealing differences in student strategy and systematicity (Figs. 3.4d,f) is useful for instructors to help their students and also for educational research in general. The user numbers in our studies are too small to draw more specific conclusions, but this work only marks the beginning of future extended design-based research and wider dissemination [41].

3.6 Discussion

The experiment throughput and cost of this platform scale to serve massive user numbers and diverse curricular demands from middle-school to college and MOOCs [56]. There are > 15 million high-school students in the USA alone [4], and hundreds of millions in developing countries or remote locations could access such platforms via increasingly ubiquitous smartphones [89]. We estimate that providing lesson plans similarly to Study 1 (Fig. 3.4a,b) to one million users per year could be achieved with ~ 250 BPUs, a modest back-end footprint of $\sim 10\text{m}^2$ and a regular 1Gb/s internet connectivity; cloud lab access for all students in a class at ~ 1 cent/experiment would cost instructors less than one living Euglena sample (Supplementary Text S3.4).

This technology also has significant potential for primary life science research. It already supports complex investigations of micro-swimmers (Fig. 3.2e,f) and micro-ecology (Fig. 3.3) as of current interest to the biophysics community [53, 96, 97]. Image data is information rich, e.g., unexpectedly we captured cell-division events (Movie S3); combined with a rich stimulus space many phenomena can be identified and systematically studied. Due to its domain specific design [113], this platform is expandable beyond Euglena and light stimuli to a general class of increasingly automated and low-cost/high-throughput experiments, such as involving valve switching in microfluidic devices [13] and cloud chemistry [106]. Theoreticians carrying out their own investigations, and large-scale citizen science [76] is within reach.

In conclusion, we demonstrated a new online access and scientific inquiry paradigm that turns observational microbiology into an interactive experience. This enables (1) interacting with living cells in realtime, (2) complex microscopic inquiry practices (3) learning analytics for life science experimentation, and (4) improved in-class-time use, logistics, costs, and safety. The key technical contribution was to extend the distributed computing concept to include unreliable biological specimen to maintain Quality of Service. This approach makes complex biological experiments and modern biotechnology accessible and interactive to multiple,



Speed [micro-met 30

Name It Yourself -15

Name It Yourself 5

Time Point	Top LED	Right LED
0	0	0
5	100	0
20	0	100

Figure 3.4: **User studies in middle-school and college settings demonstrate utility of platform for face-to-face and online education.** (a) University students performed exploratory joystick-based experiments from home, gaining intuition about Euglena's phototactic behavior by making it swim in geometric trajectories such as a rectangle. (b) Automatically generated large-scale data (hundreds of cells) using batch mode allowed students to test two hypotheses: (left) Are Euglena active particles or just undergo passive Brownian motion? (right) Does the population-averaged swimming speed depend on light conditions (light off, blue, N=389; light on, orange, N=431)? Student performed Kolmogorov-Smirnov test that rejected the null hypothesis, i.e. the distribution are the same, at 95% confidence level. (c) Middle-school students engaged in modeling Euglena phototaxis after in-class joystick experimentation. Euglena is modeled as cuboid surging and rolling around its long axis; pitch velocity is coupled proportionally to light amount entering through one side. Three sliders enable setting surge, roll, and coupling followed by virtual experimentation with a joystick (equivalent to Fig. 4a) to explore the model's 3D behavior. (d) Clusters (silhouette score=0.47 in 0-1 range) of student approaches to model exploration (5, 4, 4 student pairs, respectively). (e) Middle-school students designed batch experiments via the third-party platform interface iLab and analyzed generated movies. (f) Examples of student's main experimental design categories. (Number of students for each study: see text.)

currently underserved audiences, e.g., students, teachers, scientists, and the general public. Although the needs for education and research are not identical, they may synergistically drive technology development and its economics. All code and BPU designs are released open source (Supplemental Text S7) enabling wider dissemination and development, and we invite the life-science community to adapt their protocols and technology to make them interactive and available online.

Chapter 4

Large Scale Online Education Deployment

This chapter is based on the following submission [62] ...

> Zahid Hossain, Engin W. Bumbacher, Alison Brauneis, Monica Diaz, Andy Saltarelli, Paulo Blikstein, and Ingmar H. Riedel-Kruse. *Biology Remote Lab Enables Authentic Inquiry-Based Learning at MOOC Scale*. In **CHI '16** *[Under Review]*. ACM, 2016.

4.1 Abstract

Online learning technologies, e.g. MOOCs, promise to deliver quality education at low cost and scale. Incorporating authentic scientific inquiry and real lab experimentation online is an unsolved challenge, especially for the life-sciences. Here we report on the iterative design and large-scale deployment of an open online course with a remote biology lab that engaged remote learners (>300) in the scientific practices of experimentation, modeling and data analysis to investigate phototaxis of a microorganism. We demonstrate (1) the robustness and scalability of the online lab technology (>2,300 experiments run), (2) the design principles and synergistic integration of multiple UI and learning activities and suitable data formats to facilitate inquiry based learning, (3) design features that leverages the natural variability of real biology experiments to instigate authentic inquiry. Platform and course content are now suited for large-scale adaptation in formal education, i.e., K-12 and college; furthermore, the utility for (citizen) science is suggested.

4.2 Introduction

Delivering education through digital technologies, particularly online platforms, has many potential benefits, such as cost-effective scaling of instructional content, user-driven autonomous learning, and more sophisticated methods for automated learner feedback, such as the ones found in intelligent tutors. One major unsolved challenge for distance learning is the proper integration of tools to enable and support authentic inquiry tasks involving real experimentation work [115]. The computer-based inquiry has mainly been supported by interactive simulations, and a few remote labs have been used at smaller scales to enable real experimentation, primarily for engineering education. However, integration of experimentation, especially for the life sciences, is technologically and pedagogically challenging, in particular at larger scales. We are aware of only one massive open online course (MOOC) that systematically deploys real labs (a lab for electronics, in that case) [39] and it remains unclear how to optimize user interface design and scaffolding of content to achieve the type of complex inquiry-based learning mandated by new national standards such as the Next Generation Science Standards of the United States[79, 100].

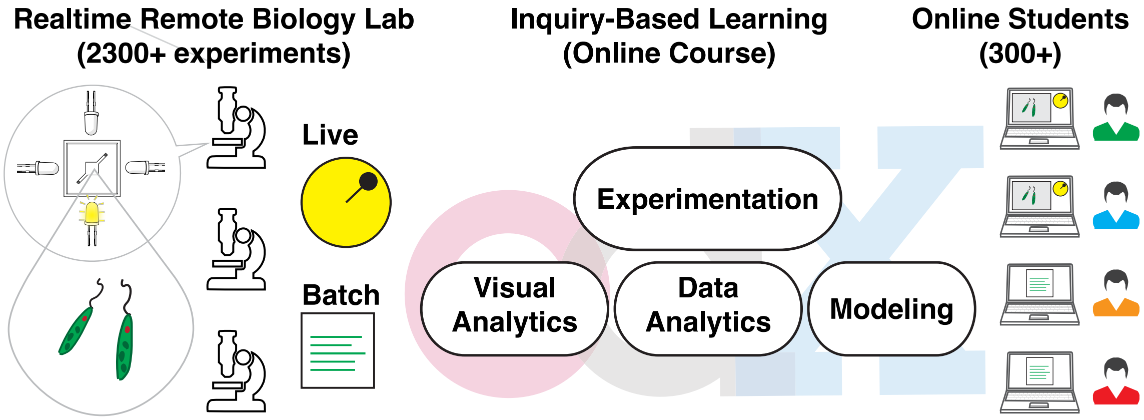


Figure 4.1: **Integration of a scalable biology remote lab into a MOOC (>2,300 experiments run by >300 students).** We explored the general affordances and design rules of online experimentation science labs to enable inquiry-based learning.

Previously, Hossain et al. [63] developed realtime and interactive technology for a biology remote lab that focused on phototaxis of Euglena gracilis, a single-celled organism (see Fig. 4.3). This lab was successfully deployed for educational settings in teacher-mediated classes in both K-12 and university settings. These initial studies demonstrated the potential for this technology for authentic inquiry learning in classrooms [31] that combines experimentation with modeling and complex data-processing practices, in ways that were previously not possible in biology education. Specifically, the realtime interactive lab enabled students to remotely stimulate microorganisms in real time, which constitutes a departure from the conventional passive observation through microscopes. The technology promised cost-effective scaling of real experimentation labs. However, the user studies in this work tested the different interaction modalities (experimentation, data analysis, and modeling) only in isolation, at a much smaller scale, and in physical classrooms.

The main contribution of this work is to demonstrate how an online biology experimentation lab can be deployed at "MOOC scale" (hundreds to thousands of students) in a single course that supports authentic inquiry-based learning [90] by coherently combining experimentation, modeling, and data analysis, in line with recommendations for lab-based curricula by the National Research Council (NRC) [105]. This work includes the following sub-contributions: (i) We developed a new version of the technology for the biology remote lab and implemented [63] it to be significantly more fault-tolerant, and MOOC-ready with more than double the throughput. (ii) We integrated multiple stand-alone user interface modules and learning activities (experimentation, visual analytics, modeling, and data analysis) and identified user-friendly yet powerful data-handling formats that enabled seamless integration of these activities; together, these tools enabled key features of a full scientific inquiry cycle [90] orchestrated through an online course-delivery platform. (iii) We unraveled specific design features to deal with non-deterministic and noisy experiments encountered with biology substrates, scaffolding course structure such that biological variability becomes a feature leading the learner to deeper understanding and fostering self-motivated exploration.

4.3 Related and Previous Work

The potential benefits of lab work in science education are well explored in contexts of traditional hands-on labs. The America's Lab Report by the NRC [105] identified seven learning goals for which lab work is essential: (1) enhancing mastery of the subject matter, (2) developing scientific reasoning, (3) understanding the complexity and ambiguity of empirical work, (4) developing practical skills (5) understanding the nature of science; (6) cultivating interest in science and interest in learning science, and (7) developing teamwork abilities. Labs have been used in particular in the context of inquiry-based learning, defined by Pedaste et al. [90] as *"an educational strategy in which students follow methods and practices similar to those of professional scientists in order to construct knowledge."* Studies on the effectiveness of instructional approaches for science education have also shown the benefits of inquiry-based approaches compared to passive methods [30, 121, 90].

However, it is challenging to foster such authentic inquiry-based learning with open online course (OOC) platforms that are traditionally built for video-based instruction. Interactive computer simulations could be well-integrated, but they are inherently restricted with respect to the behaviors of the real phenomenon they can reproduce [45]. Remote labs [60] however, which have some of the logistical benefits of computer simulations and of the data richness of real labs, are promising candidates for authentic inquiry-based learning in online settings. To date, most of these labs are situated in physics and electronics education [60], and not designed to be accessed concurrently by hundreds or thousands of students. Remote labs for life sciences, in particular, are underdeveloped due to the complex need for handling biological specimens; recently a successful deployment was demonstrated by Hossain et al. [64]. On the other hand, successful approaches to inquiry-based learning in online settings need well-designed scaffolds built into the course to support students both with respect to engaging in the inquiry tasks [114] and the motivation required to persist through the course [121]. Key issues in the deployment of MOOCs have been the lack of students' motivation while completing activities that are largely based on passive video-based learning and isolation [124].

Hossain et al. [63] previously developed a biology remote lab and deployed it in small-scale educational settings. This online lab was interactive in realtime, fully automated, low-cost, and scalable by design. Online

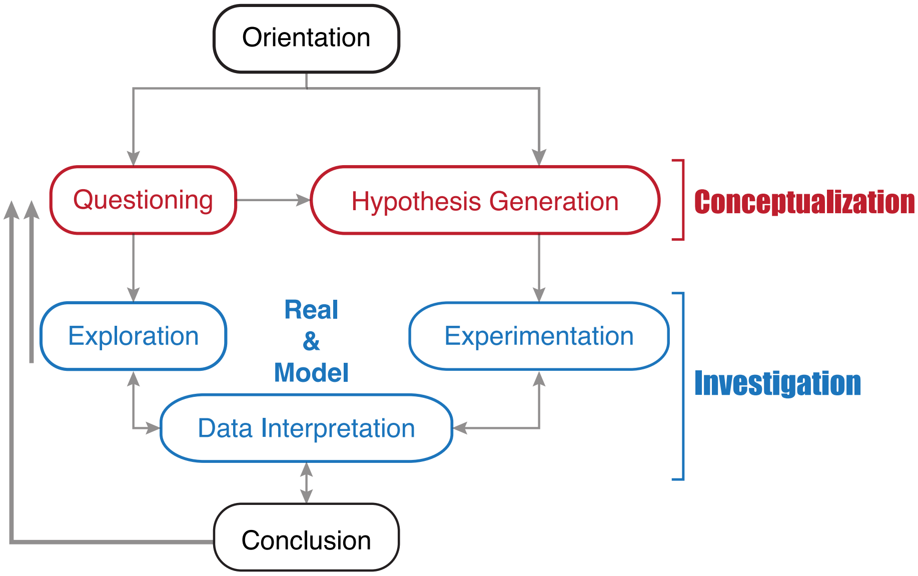


Figure 4.2: **The course takes students through the different phases and sub-phases of a full inquiry cycle in an integrated manner.** We adapted the schematic from Pedaste et al. [90] with emphasis on exploration and experimentation both with real specimen or models.

users ran interactive experiments on the phototactic microorganism Euglena gracilis to learn how single-celled organisms sense and react to environmental stimuli such as light. Borrowing concepts from cloud computing implementation, the platform architecture load-balances concurrent experimental tasks with a cluster of back-end instruments (biotic processing units, BPUs) in a distributed and fault-tolerant manner; each BPU can run $\sim 100,000$ experiments per year for $<$ \$0.01 per experiment. A modeling interface was implemented that enabled students to alter biophysical parameters in virtual Euglena to explore cellular responses to light and then compare that data to real Euglena. And although this platform was successfully deployed to support college-level biophysics projects as well as experimentation and modeling in middle schools, the individual components were never holistically combined for one learner group - let alone in a completely online learning setting.

4.4 System Improvements

We extended the design and implementation of Hossain [63] with end-to-end asynchronous principles by first completely decoupling the main components: 1) BPU operations, 2) data post-processing with automatic cell tracking and data exports, 3) scheduling, and 4) web servers 5) data storage. This allowed us to implement a non-blocking asynchronous communication protocol between components, which not only more than doubled the overall throughput, but also made the system linearly scalable and significantly more fault tolerant. Consequently, system failures are mostly opaque to the end users. This adoption of cloud computing like architecture with custom scheduling and monitoring protocols on the real biological specimen, housed inside BPUs [63] (compute nodes), was the key to constructing a cost effective and robust remote lab that allowed concurrent access by many users to real experimentation over the Internet.

4.5 Course Goals and Design

In this work we implemented an online course through an MOOC platform called OpenEDx with three overarching course goals in mind: 1) deliver a "full inquiry cycle" based learning platform online, 2) teach students about the phototaxis of a microorganism *Euglena Gracilis*, and 3) train students with various UIs, modeling and data analytics tools to deliver a set of *laboratory experiences* [105]. We elaborate on these goals in the following:

First, we sought to integrate, within a single online course, the components of the inquiry-based learning as defined by the NRC [105]: (1) posing questions and formulating testable hypotheses (2) designing and carrying out investigations (3) using tools to make observations, gather and analyze data (4) building, evaluating, testing or verifying explanatory models in light of empirical data (5) interpreting and communicating results. In order to coordinate and contextualize these components, we based our course design on a recently synthesized model of the inquiry cycle by Pedaste et al. [90] (Fig. 4.2).

Second, we chose, as the content learning goals of this course, the phototactic response of the single celled organism Euglena [40] (Fig. 4.3) as it exemplifies the general taxis principle applying to many cell types and stimuli, such as Euglena gravitaxis [75] and bacterial chemotaxis [14]. Euglena have an eye spot that senses light coming from one direction only; they also have a long flagellum that allows them to swim forward and rotate around their axes. The photoreceptor is directly coupled to the beating pattern of the flagellum. Appropriate coupling of strength and directionality allows the cell to stably swim toward or away from the light. However, not all cells behave exactly the same: these microscopic cells exhibit variability and individuality, as do all biological organisms. Handling this biological noise and variability merits particular consideration, as this variability can prevent a consistent user experience. However, dealing with real experimental data with natural variability can be a very productive learning experience [19], and was identified as a key laboratory experience by NRC [105, 118].

Third, as part of process learning goals, students must be able to use various instruments and UI tools at ease to execute experiments (free form exploration as well as controlled), explore models, collect experimental data, and infer results via visual and data analytics. All these tools need to be centralized in a way that reduces switching-cost between phases and sub-phases of the inquiry cycle (Fig. 4.2), thus providing a seamless set of *laboratory experiences* [105].

In the light of these three goals, we implemented a course in the OpenEDx platform (Table 4.1) with 7 units. Each unit confronts students with a new *scientific skill* (task) (Fig 4.2), new *biological content* (Fig. 4.3), and a new *scientific instrument* (HCI technology) (Fig. 4.4). Units were designed to take 20-60 min each, for a total course time of $\sim$ 4h. The 6th unit encourages students to postulate a testable hypothesis and voluntarily

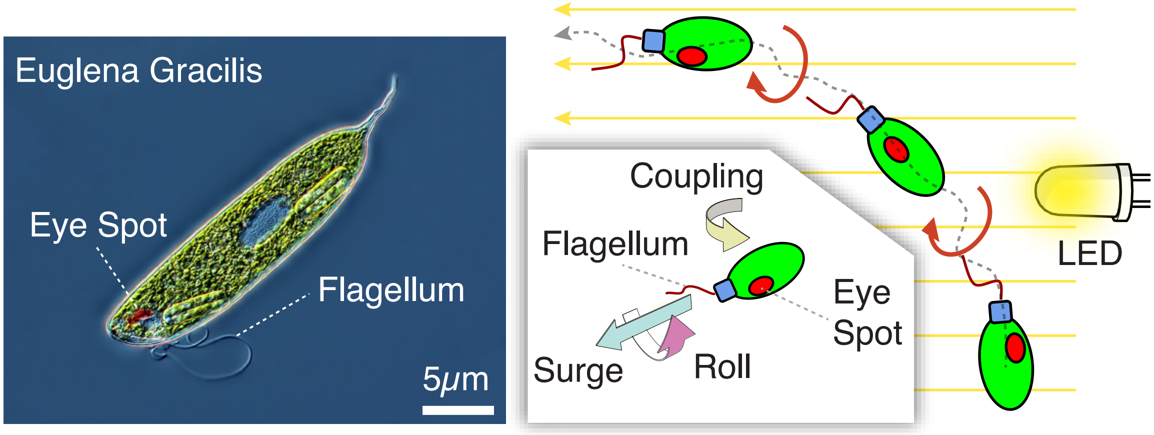


Figure 4.3: **Content learning goal of the course**: Euglena Gracilis (left) is a single-celled organism that performs negative phototaxis, orienting and moving away from light, by rolling around its longitudinal axes via a feedback coupling between the eye spot and the flagellum. On the right, we depict a schematic of an Euglena model.

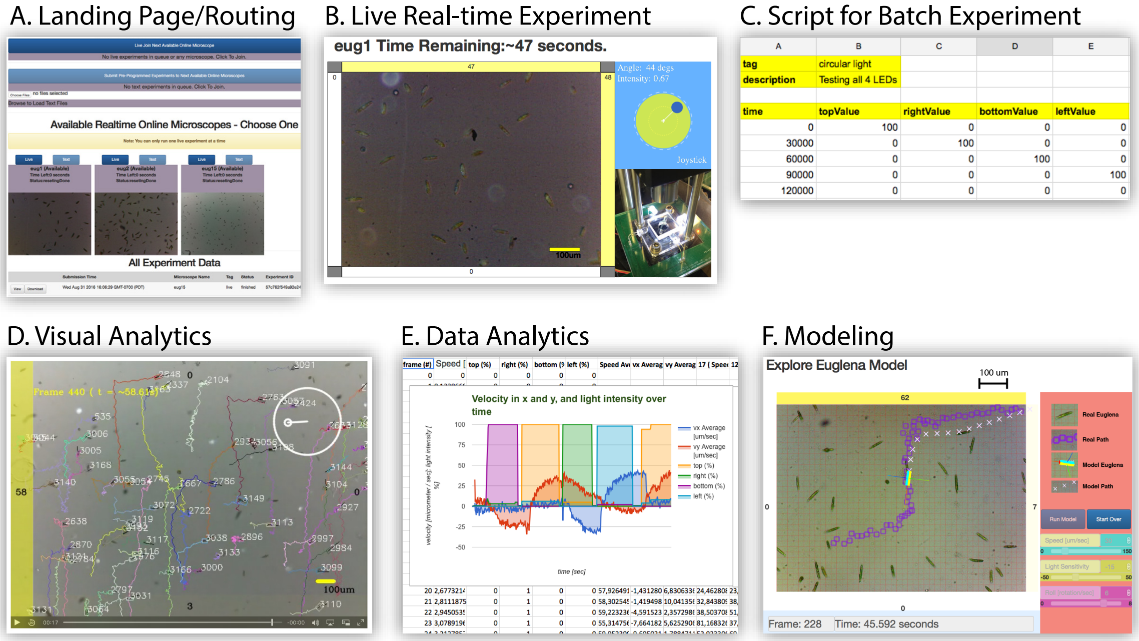
pursue a self-guided research by going through the whole inquiry cycle on their own. The 7th unit provides a summary and collects students' feedback. This course layout was selected after multiple rounds of iterative user studies and prototyping in order to optimize the progressive increase in complexity for the learner.

4.6 User Interface Design Considerations (HCI)

In this section we will discuss various UI design choices we made at three different user-interaction phases: 1) experimentation, 2) post-processing and data interpretation and 3) modeling. The UI in each of these phases were created as a *design probe* [50] with specific goals, while the UI design were guided by iterative development and pilot studies, mostly conducted in our previous work [63]. The pilot studies involved feedback from students in a graduate level biophysics class, middle/high school students, and semi-structured interviews with 7-12 bioengineering graduate students/postdocs on campus.

4.6.1 Experimentation UI

The central server distributes incoming experiments among a set of backend BPUs by taking into account the current BPU health (auto-monitored [63]) and queue wait-time. The experiments could then be executed either in a real-time *live* interactive playful mode (Fig. 4.4B), or in a *batch* mode where the experimental instructions would be submitted as a pre-programmed script (Fig. 4.4C). In the following we will discuss the various UI design choices that are relevant to experiment execution.



A	B	C	D	E
tag	circular light			
description	Testing all 4 LEDs			
time	topValue	rightValue	bottomValue	leftValue
0	100	0	0	0
30000	0	100	0	0
60000	0	0	100	0
90000	0	0	0	100
120000	0	0	0	0

Figure 4.4: **HCI modules to deliver procedural learning goals of the course in an unsupervised MOOC setting.** (A) Landing page to route students among a suite of online microscopes. Students can either choose to get auto-routed to the best microscope or choose a specific one. (B) Realtime Euglena biology lab in *live* interactive mode. (C) An experiment script in CSV format for the *batch* mode. (D) A playback movie viewer for visual analytics via automatically tracked Euglena cells. (E) Google Sheets application for data analytics, including statistical analysis and graphing of Euglena traces. (F) Modeling applet simulating Euglena overlaid on pre-recorded video.

Biological Variability and BPU Selection

Biological samples often undergo natural variability - e.g. cells dying, culture going through circadian rhythm or the sample being contaminated etc - which makes remote automation uniquely challenging compared to physical systems. Therefore, it is of paramount importance that the biological state of every BPU is auto-monitored by the central system (see Chapter 5) and the incoming experiments are auto-routed to the best available for repeatability. However, this introduces a bias, especially when such cloudlabs are deployed for educational purposes where one of the key learning goals might also be to demonstrate this biological variability. The goal of the design is therefore to automatically mitigate this biological variability to provide a better Quality of Service (QoS) and also allow the flexibility to expose this variability whenever deemed appropriate.

Towards this, we developed a dashboard style landing page (Fig. 4.4A) that shows all the available BPUs with their current thumbnail views. From here a student could either ask the system to automatically route her experiment to the best available BPU, or choose a specific one. Later we utilized these UI features by scaffolding the course lesson to demonstrate both repeatability and natural variability (see Section 4.9, Unit 2), which is an unique strength of a real biology cloudlab.

In the future we plan to add more BPU health related information next to the BPU thumbnails. A health ratings (stars or numerical), and a visualization of BPU health trend over a short historical time-frame would inform students better about the expected variability ahead of time. For example, a student might intentionally want to run an experiment on an unhealthy BPU, where the Euglena is not as responsive to observe what really happens at that biological state.

Live Experimentation

Live interactive experimentation is crucial for gathering an initial intuitive understanding of the phototaxis phenomena. First, Euglena phototaxis is non-linear (Fig. 3.2f) with respect to the light intensity and only activates when the light intensity is $> 40\%$. Second, a Euglena swarm takes ~ 7 secs to align to a light stimulus. From the third user study of our previous work [63] (see Section 3.5), which was devoid of the *live* interactive mode, we learned that students face difficulty in observing phototaxis without having a sense of these light and time scales. Therefore, the goal of the design is to instigate an initial understanding of negative phototaxis behavior of Euglena with a sense of time, length and light scales, in a playful manner.

We implemented the *live* UI with components that are essential to convey these different scales (Fig. 4.4B). The joystick interface provides relative light intensity information while the live microscope video stream is overlayed with time and length scales, which is not only a mandatory practice in biology but already provides an intuitive understanding of Euglena velocities. However, certain lighting combinations were not possible with a joystick UI e.g. shining left and the right LEDs at the same time. But note that the main purpose of this *live* UI was to expose a means to intuitively understand phototaxis in a playful game like manner, while a more controlled experimentation would be substantiated in the *batch* mode.

Due to the second and third user studies of Section 3.5, we also learned that students initially misunderstood light directions as the LEDs are not visible in the microscope view. We suspect that this misunderstanding might have stemmed from a cognitive bias to expect Euglena to move towards light (positive phototaxis) as Euglena is known to be a photosynthetic organism (i.e. it lives on light). Although Euglena does exhibit positive phototaxis under a very specific condition over longer period of time, it is not easy to observe that with bright LEDs in a short experiment. Therefore, we added light indicators (as bars with light intensity percentage overlaid) on the sides of the live video stream that mitigated the confusion in the future studies. It is tempting to implement a much more explicit visualization of LEDs but care must be taken; for example, drawing an animated light bulb on the sides might result in a wrong impressions that light is emanating from a point source and radiating out. Whereas, in reality the visible section of the microfluidic chamber is much smaller compared to the physical size of the LED and therefore light can be considered entering the sides of the chamber uniformly, which is depicted more accurately with a bar.

Several studies with remote labs [29, 28, 66, 85] have demonstrated that students are greatly motivated when they feel part of a real lab environment, especially when the lab is accessed remotely over virtual interfaces. One effective strategy is to expose the underlying workings of the real lab visually using an external camera, if even that camera plays no role in the actual experimentation. We therefore added an external camera view into the BPUs (Fig. 4.4B) and students were able to watch LEDs going on/off as they move their joystick, which also provided a sense of the video lag in the microscope view as an useful side-effect.

Batch Experimentation

In the *batch* mode, students would submit pre-programmed experiments (potentially multiple experiments in one shot) to be executed offline asynchronously. Once an intuitive understanding of the light and time scales of Euglena phototaxis is understood from the *live* mode, *batch* mode could be used to test hypotheses repeatedly with clean timing and light data. Programming is much more expressive than the joystick UI as all lighting conditions could be tested. For example, students could turn on all four LEDs and study Euglena aggregation due to artificial pressure. Furthermore, *batch* programming was particularly useful when the internet speed was too low for live video streaming. Here, the goal of the design was to make programming as simple as possible and accessible to a diverse audience - middle-school to graduate level students.

We tested three different data structure based programming for the time and light sequences: 1) XML, 2) JSON and 3) CSV. Through different pilot studies [63], we concluded that CSV is the most accessible format, as it was the simplest and students could use Google Spreadsheet (free) to write programs in a more natural row-column format (Fig. 4.4C). We also provided two extra fields in the CSV, to add a *tag* and a *description* to the experiment so that students could easily find the corresponding data when the experiment is complete. In addition, middle/high school students are already familiar with Google Spreadsheet as it's used in classrooms. Simplicity and accessibility of the programming format is particularly important for a MOOCs deployment where the student cohort is diverse and prone to dropping out.

4.6.2 Post-Processing and Data Interpretation UI

Our system automatically extracted information from the captured time-lapse image data and exported them into a 1) video based visualizations for visual analytics, and 2) numerical data for data analytics. Biological data is often image based and information rich, but extracting meaningful information may require application of complex computer vision algorithms. Providing only raw data and leaving the information extraction exercise to the students (especially in MOOCs with diverse unpredictable student cohort) may defeat the purpose of the cloudlab even after all the hard work. Therefore, we extracted these information automatically at the service side, while the data and UI design goals were to make data interpretation as easy as possible.

Visual Analytics

We developed the visual analytics tool iteratively by first implementing a simple grid overlay on the captured video as shown in Figure 3.4e. Students would play the video back and forth and count the change of Euglena population in each grid block to assess phototaxis. Due to the third user study in Chapter 3 [63], we realized that this simple visual analytics was deficient in capturing subtle phototactic behavior, which can be hard to observe for untrained students. On the other hand, we also realized that a powerful visualization could be a key to engage students in MOOCs. The goal of this design was to allow students to measure all the necessary quantities, or atleast guesstimate, right off of the visualization instead of downloading and running complex analysis on numerical data.

We developed tools in our backend server to track every Euglena cell automatically over time and composed a video based visualization (see Fig. 4.4D). This visualization of tracked Euglena cells greatly improved how one perceived phototaxis, even when its subtle, due to the emergent behavior of the Euglena swarm. Furthermore, the wiggly motion of Euglena, due to the rotation about its own body axis, became evidently clear. This rotation plays a vital role in the phototaxis mechanism and thus its very important for students to be able to see this in reality. This video had additional components: joystick positions (for *live* experiments) with the corresponding light indicators with intensity percentages, a time scale and a length scale (this video can be viewed directly on the website). Every cell has an ID tag that can be cross-referenced with the numerical data discussed in the next subsection. With these information, a student is already able to measure several quantities by simple inspection, e.g. size of Euglena, velocity due to light on/off, response lag time by inspecting when the tracks start to bend, rotational speed of Euglena by counting peaks of the wiggly motion and etc.

Numerical Data

For statistical analysis and controlled experimentation, we also provided all the tracked Euglena data in numerical format. The design goal here was to export all numerical data in a way that is approachable to a diverse audience (from middle school to graduate level students).

A single one-minute experiment can have track information for 300-1000 cells and therefore can quickly become cumbersome to process. In our previous studies (Chapter 3, [63]) we exported these data in JSON format and provided corresponding APIs for Matlab and Python for higher level operations, i.e. select Euglena tracks within a time frame, compute velocity components within the selection etc. But, we quickly realized challenges: Matlab is not free, and assuming Python knowledge heavily shrinks potential MOOC audience, especially as we were also targeting middle school students. Therefore, in this MOOC deployment, we exported all data in MS Excel *xlsx* format (see Fig. 4.4E). Firstly, the row-column format is easy to understand and manipulate, secondly MS Excel is relatively ubiquitous, and thirdly even Google Spreadsheet (free) can import *xlsx* file and both Google Spreadsheet and MS Excel already have advanced statistical and graphical functions. In many middle/high schools, students already use Google Spreadsheets for their science classes. Beside raw data we also exported derived data such as point velocity estimations, and everything is converted to SI units by taking into microscope magnifications and other machine parameters into account. However, the row-column format is not as expressive as JSON combined with an API, e.g. not all plots could be generated from Excel with a simple row-column selection and a click of a button. Therefore, we exported extra *xlsx* files with the necessary data transformation to facilitate certain parts of our MOOC lesson and this customized data export could vary from one lesson to another. We found this strategy of exporting rather extra *xlsx* files to be overall more accessible to students than exporting a single JSON with a programming API.

4.6.3 Modeling UI

We iteratively developed a modeling interface that provides a mechanistic explanation of the phototaxis phenomena. This model is similar to he one deployed in the second user study of Chapter 3: an Euglena is approximated with a 3D cuboid that has three parameters, *surge* (forward velocity), *roll* (rotation about its long axis) and *coupling* (sensitivity to light direction). The underlying principle is that Euglena has a single eye-spot that can detect light from only one direction. The cell moves forward (*surge*) while rotating about its own axis to scan light from all directions and steer itself accordingly (based on the *coupling*). This model was overlaid on top of a real Euglena path, and the exercise for the student was to explore the parameters to match the model with the real Euglena path. The goal of the design was to simplify our interface based on the lessons learned from earlier pilot studies and allow students to focus more deeply on the role of these three parameters in phototaxis.

In the second user study of Chapter 3, students not only had to explore the three parameters, but also had to play with a joystick to apply simulated light stimulus. This added an extra layer of complexity as students had to get the joystick sequence right along with those three parameters. A controlled study [25] revealed that without the joystick, students were significantly more systematic with the parameter exploration, e.g. change one parameter at a time with fewer repetition. In the same pilot study, another exercise for the student was also to give those parameter a meaningful name (based on what they observed the parameter did) because the actual names were initially hidden. In this MOOC deployment, we simplified this modeling interface by replacing joystick with a pre-defined light sequence and revelead the parameter names (see Fig. 4.4F). We also added length scale and legends to describe visual components related to the model. With this modeling UI in combination with the real experimentation, we opened up an opportunity to study bifocal [19] modeling in the context of biology education.

4.7 Synergistic Integration of all UI components

We holistically combined all the UI described in the above subsection under the same platform to provide an end-to-end inquiry-based learning (see Fig. 4.2) platform over the Internet at scale. A student could login to a single website, run a series of real experiments either in *live* mode or *batch* mode while testing various hypotheses, interpret data either visually or by numerical analysis, and explore model parameters to understand the underlying phototaxis principles. Through iterative development, we made careful UI design choices in every step to primarily make sure our system was easy to use, consistent with biology practices (always providing scales bars), provides better QoS through auto-routing as well as flexible enough to expose biological variability, accessible to a broader audience due to automatic extraction of data and use of data formats that are easy to handle even for a middle school student. Hence, our system allows easy, yet rich experimentation, i.e. the system has *“a low floor and a high ceiling”* [94].

4.8 Research design and Course Implementation

Unit #	Scientific Skill (procedural learning) task	Biology (content learning)	Technology
1	Passive observation	Euglena are single cells	Online microscope
2	Active experimentation	Euglena respond to light	Real-time interactive & batch experiments
3	Visual analytics	Euglena roll around their axis while swimming	Post-experiment video analysis
4	Exploration and conceptualization of models; parameter fitting	From structure to function: feedback loop between eye spot and flagellum	Modeling Applet
5	Data processing; interpreting graphs	Euglena speed does not change (much) when light is turned on	Google Sheets
6	Self-guided project; own hypothesis; experiment	Following your own observation	Revisiting all tools
7		Summary, revisit, transfer to other contexts	

Table 4.1: **Final course layout.** Note that each unit essentially introduces an online tool ("instrument"), and engages students in a different inquiry practice with a new bit of biological content.

We ran an iterative design-based research [10] study in which we recruited different student groups while refining the remote lab technology, student interfaces, and course content. First, we implemented a simpler course in the MOOC platform OpenEDx that included the live lab, data download and processing in MATLAB, and multiple modeling activities. Based on this first experience, we implemented various improvements, including switching to Google Sheets from MATLAB (in order to make it more approachable for online students), simplifying the modeling environment to a single activity, and limiting the course layout to focus on the minimal biology content necessary to provide proper motivation for the inquiry cycle. We then iteratively tested and updated this course with $1-3$ students at a time for a total of ~ 20 students, leading to the course layout described in Table 4.1. When we launched the course publicly, a new course session lasting 5-6 days was offered each successive week, and we made only minor updates after each session. We concluded the study with focus interviews of teachers who took this course and expressed interest to use it in their own teaching.

Nine week-long sessions were offered, and all analyses reported in this paper are based on the first six sessions. Students were recruited via monthly Open edX newsletters; additionally, ~ 300 teachers were contacted directly. A total of 993 students signed up, distributed among the six sessions $(97, 259, 296, 157, 76, 48)$, of which $325 (35\%)$ started the course. The completion rate is $\sim 51\%$ which was estimated by the number of students who answered questions in unit 1 compared to unit 7. Students came from 46 countries, mostly from the USA (42%) with a median age of 32 (IRQ=19), 47% of whom were female. Students took 3.5 ± 1.1h to finish the course, with each of the seven units taking ~ 30 mins to complete, except unit 5, which took ~ 1h.

4.9 Online course student study results

In the following we will present the results for each course unit, with a particular focus on the HCI components.

Technical performance of the biology cloud lab A total of $\sim 2,300$ experiments were executed, with ~ 7 experiments on average per student. During these 6 weeks, students hardly experienced any wait-time (median: 4.8 secs, IRQ: 1.55 secs, which is within the system loss time for routing) for *live* experimentation as the demand for the online microscopes was always below the supply, except for a single incident (Fig. 4.5). Due to the worldwide accessibility, the usage of the remote lab was effectively amortized throughout a day, 7 days per week (Fig. 4.5) though we observed peak traffic typically in the middle of the night. Net capacity wise, our remote lab was underutilized and could have executed ~ 10 times as many experiments (and thus students), which is double the capacity from Hossain [63]. The manual effort to maintain the entire microscope cluster on the back-end was < 1 h/week to exchange organisms and microfluidic chips.

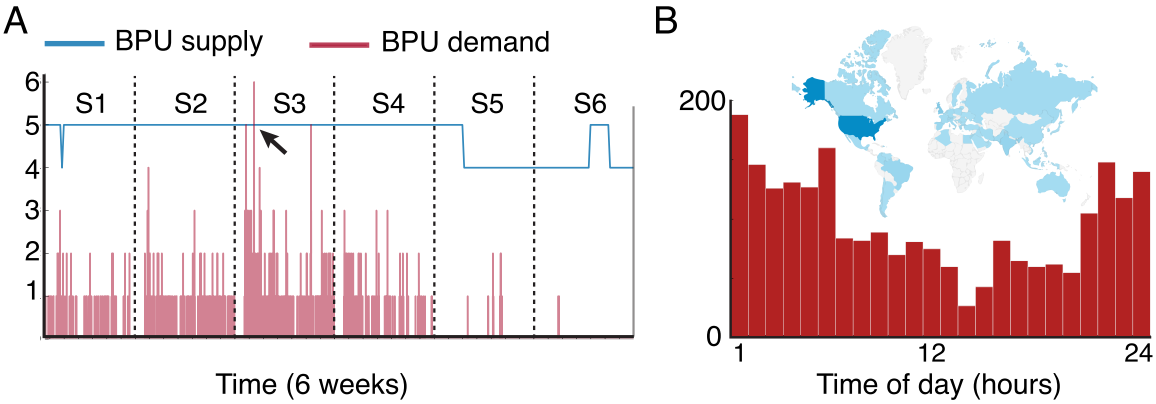


Figure 4.5: **Long term usage of the remote lab**: (A) BPU (online microscope) demand during the first 6 weeks (S1,S2 and so forth) was well below the supply at all time except for a singular incident (marked with arrow) when a student had to wait for her turn. (B) System access pattern over 24h period during 6 weeks. The traffic was mostly amortized over the entire day with peak activity seen during the middle of the night. The inset shows the density of the traffic sources on a map.

Unit 1: Observation (online microscope) After a pre-survey, the first unit introduced the student to a set of online microscopes (Fig. 4.4A). Students answered questions about the quality of live streaming and download speed. 40% percent of students reported "really nice" (likely due to internet speed); another 45% reported that it was "at least reasonable." We noticed that students with internet speed of > 50 Mbps typically never complained about the quality of streaming though some students even with 5Mbps seem to have succeeded in the course. Students then identified Euglena they had observed in the online microscope in pictures showing various microorganisms.

Unit 2: Experimentation (interactive and scripted) Unit 2 introduced students to the joystick on the online microscope, explained how the set-up worked, and prompted them to try it and report their observations. Without any specific prompt (question: "what do you see"), 83% (N=163) of students reported Euglena responded to light, among which $62\%, 10\%, 7\%$ and 4% recognized negative phototaxis, positive phototaxis, both phototaxis or *"spinning"* without linear motion, respectively. These observations indicated that the Euglena light responses in the experiments were clear enough for the majority (negative phototaxis is the expected dominant response). There were various reasons for students not giving the expected answer. Some students used too low light stimulus or did not wait long enough for Euglena to respond, whereas some students self-selected a microscope with either too few cells on the screen or cells that were in a state of too much light sensitivity, with a higher likelihood to just spin on the spot. Note that, in this unit, we explicitly asked students to repeat their experiment on different manually chosen BPUs (including BPUs with different magnifications) and describe if they see any variability.

We were interested in assessing the types of experiments run by students before and during unit 2. For every experiment, we collapsed the joystick positions over time into a single image by convolving a Gaussian distribution on every joystick position and performed hierarchical clustering on 1321 experiments ($\sim 72\%$ of all *live* experiments that had sufficient mouse movements) from the first six sessions. A judicial cut-off in the hierarchy revealed six dominant clusters (silhouette score: 0.53 in $0-1$ range) (see inlets in Fig. 4.6). The corresponding cluster centers correspond to experiments that either focused on directional responses (C4, C5, C6) or on responses to varying light direction (C1) or intensity (C3). A Locally Linear Embedding (LLE) analysis with two components (Fig. 4.6) reveals the general spread and subtle variations among the experiments. Note that in many experiments students tested intermediate light values (Fig. 4.6C2 and C3) as opposed to full intensity.

We also introduced the batch mode beginning with session 3 as an optional activity, which, in combination with watching a video after the online observation, successfully offset problems with internet connectivity. However, far fewer *batch* experiments (69 *batch* experiments compared to 2255 *live* in the first 6 sessions) were run than live experiments.

Unit 3: Visual analytics/Qualitative data interpretation Unit 3 instructed students how to view, download, and analyze their data (Fig. 4.4D). At this stage, the measurements were solely based on visual analytics, i.e. directly off the movie with overlaid visualizations of tracked Euglena cells, timer, and a scale bar. Students were asked to explain why Euglena might exhibit a wobbling, meandering motion that is typically noticeable in the traces (Fig. 4.7). 50% of the students hypothesized that it may allow for detection of light direction (which we considered being correct); other answers included better maneuverability or faster escape from predators. When asked to estimate the forward speed and the wobbling frequency, 67% and 15% of students answered correctly on the first attempt, respectively, with the latter question reflecting a more challenging

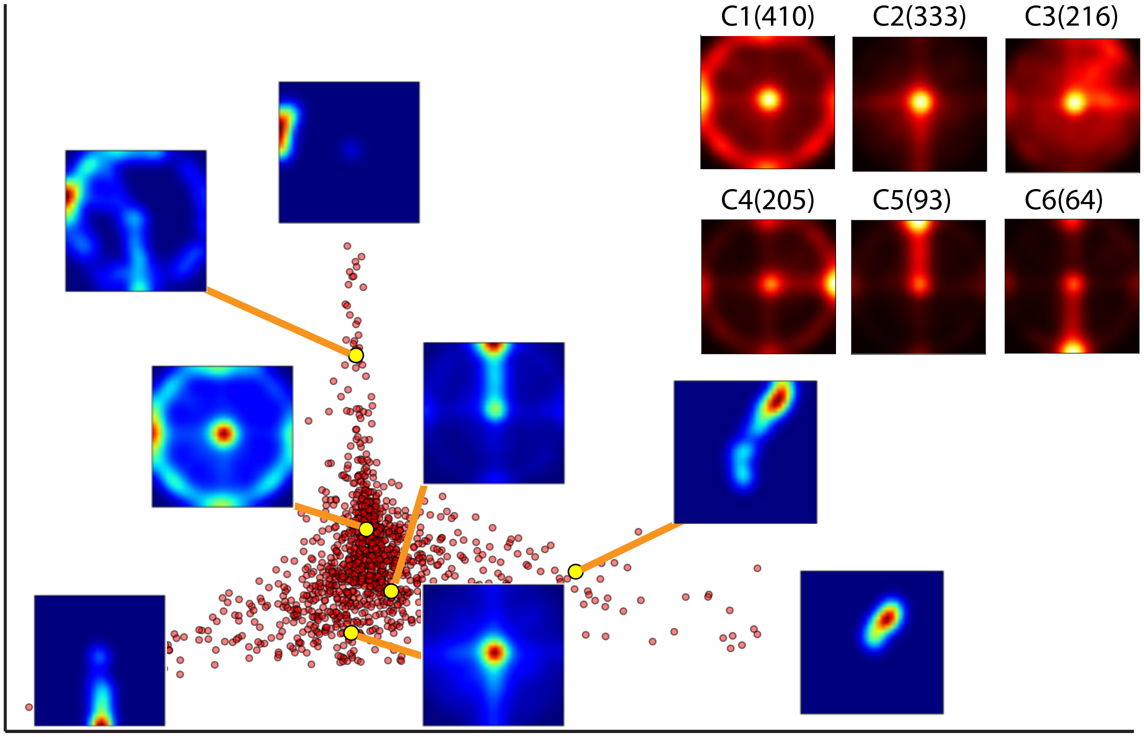


Figure 4.6: **Typical *live* joystick experiment run by the students and a Locally Linear Embedding analysis shows the varieties among all the experiments run before unit 3 (N=1321)**. The inset shows the six dominant clusters extracted using hierarchical clustering (silhouette score of 0.53 in 0-1).

concept. The ultimate reported median linear speed was 60μm/sec (IQR=69.2, N=148), and the median roll speed was 0.5rev/sec (IQR=0.5, N=147), which are within known ranges for Euglena. Students appreciated the benefit of such visual analytics, i.e the traces of Euglena overlaid on experimental video, with a median feedback score of 9 (highest score *"very useful"*, IQR=3, N=52). These observations and analyses primed the students for the modeling exercises in the next unit.

Unit 4: Model exploration and evaluation Unit 4 introduced students to more structural details regarding the single eye spot and flagellum (Fig. 4.3), providing insight into the mechanism by which Euglena rotate around their axes. Students then engaged with the modeling environment (Fig. 4.4E), where they were prompted to find the best the parameter values that fit the model to the real Euglena and then explore how to accomplish positive phototaxis.

We found that 48% (N=77) students were successful in the fitting tasks, i.e. found a parameter set that led to a closely matching swimming path. We identified four strategies that students attempted to explore the parameter space from $N = 1531$ modeling experiments. For this, we converted a sequence of modeling experiments of a given student to a single Markov transition probability matrix with three states: one, two, and

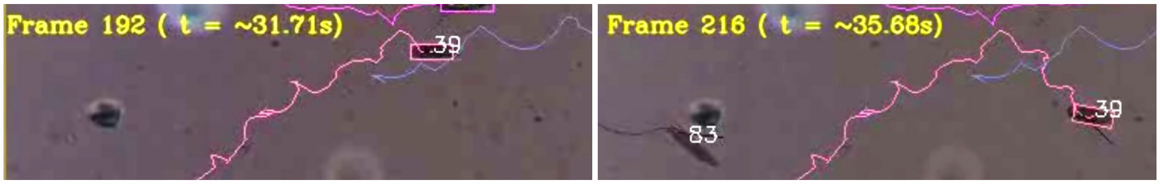


Figure 4.7: **Post-experiment movie analysis and back-of-the-envelope measurements.** For example, counting the peaks in the meandering path within a time frame already gives the rotational speed of a Euglena cell about its long axis.

three parameter changes between successive experiments (repetition of the same experiment was discounted). We then extracted four dominant clusters (silhouette score 0.58 in $0-1$ range) within these transition matrices using hierarchical clustering (Fig. 4.8). Cluster 1 and 2 (Fig. 4.8) reveals the most efficient strategy as student in these clusters ran only 14 and 14.5 (median) experiments respectively before posting a near optimal solutions. Students in these clusters predominantly switched to changing 1 parameter only followed by multiple parameters at a time. Whereas in cluster 4, which proved to be the most inefficient strategy, students ran 25 (median) experiment for the same task and the key difference was that students predominantly switched to changing 2 parameters followed by 3 at a time. Finally, students in cluster 3 mostly changed 2 parameters at a time and required 17 experiments for the task. We didn't notice any significant differences in the number of repeated experiments across clusters. This observation is consistent with the small scale study by Hossain [63].

Next, students were asked to go back to the real experiment after the modeling and report whether they noted any differences. The 59 students who volunteered to answer gave in average 1.5 differences (SD = 0.7). 69% of students noted that the behavior of real Euglena changed depending on the light intensities, but not the model behavior: some real Euglena spun in one place (40%), wobbled differently (19%), seemingly increased in forward or rotational speed (15%), or moved towards the light (8%). 31% of students noted variations in real Euglena behavior both within a single cell and in the population. Finally, 15% of students noted that real Euglena lagged in their reaction to light, whereas the reaction of the model was always instantaneous. These observations show that the students recognized subtle yet significant differences that went beyond what was explicitly discussed in the instructions. We argue that these differences became obvious mainly due to the juxtaposition of the model and real Euglena.

Unit 5: Quantitative data processing and analysis In Unit 5, students were asked to import their data into Google Sheets and perform large-scale statistical analyses. This unit taught the full process of data loading, plotting, and labeling. students first worked through a highly scaffolded example to analyze the Euglena speed with Light-On versus Light-Off (which has at best a weak effect). They were then asked to perform a similar analysis on their own, such as to determine graphically whether the velocity vectors changed with directional light stimulus from the LED, using the decomposition of the velocity into its cardinal directions along the x and y directions. Depending on the direction of the light origin, one velocity component should be zero, on average, and the other either negative or positive (Fig. 4.9). Students had to run a new set of experiments for

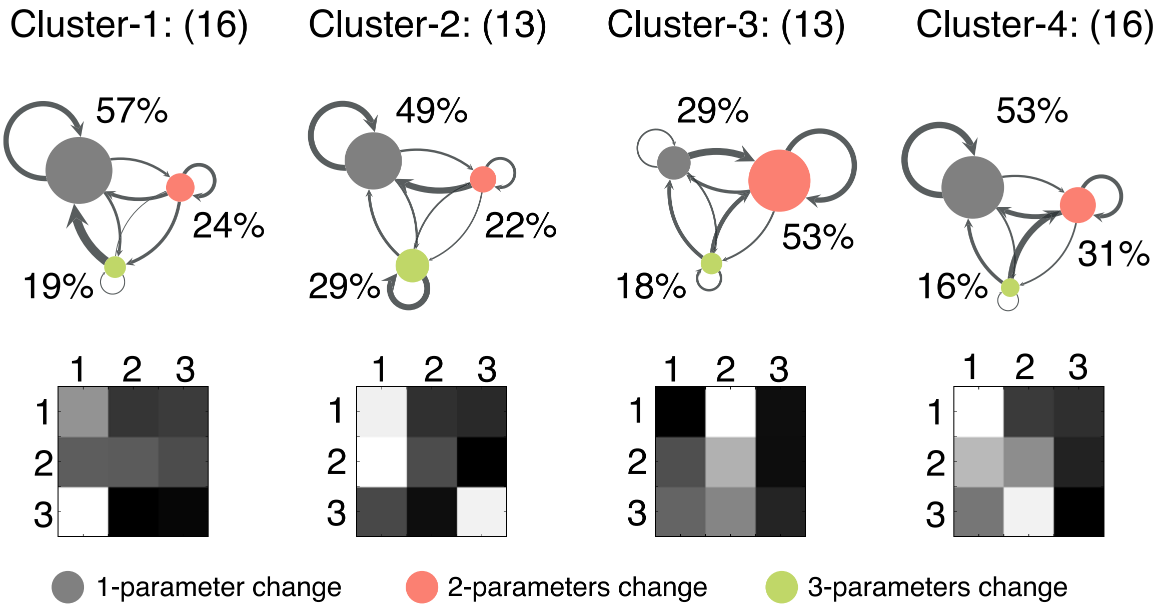


Figure 4.8: **Cluster analysis of parameter fitting strategies for a Euglena model**. Clusters are arranged from the most efficient to the least efficient strategy (left to right) in terms of how many modeling experiments students ran (median: 14, 14.5, 17 and 25 respectively) before posting a near optimal solution. The number in parenthesis represent students count while the percentages are net probabilities of the states.

this analysis.

Google Sheets is a viable option for data analysis and interpretation and we asked students to voluntarily provide us with a link to their work. 44 students provided us with their graphs and here we randomly picked analyzed 14 from session 3. We concluded that 6 students ran the appropriate experiments and produced proper graphs, meaning that the stimulus protocol was adequate for the question to be answered, the experimental data were usable (there were enough cells and long enough stimuli were used), the graphs displayed the correct data, and the labels and formatting were correct. 8 students had proper graphs but not appropriate stimulus protocols, and 8 students had the opposite. This unit took the bulk of the students' time, on average 1h, indicating that the unit was longer and more complex to complete than the others.

Unit 6: Open and self-guided investigations In the final activity unit, students were led to carry out a self-guided research activity, where they proceeded through the main parts of the inquiry cycle while applying all or most of the previously used tools. We prompted students to make an observation (specifically one that had not been stated by the course material previously), transform this observation into a testable hypothesis with experimental designs. Students were then encouraged (optionally) to pursue the actual experimentation and draw conclusions. 21 students attempted to pursue this optional research activity while 15 of them did it meaningfully with 8 students even providing us with their Google Sheet links.

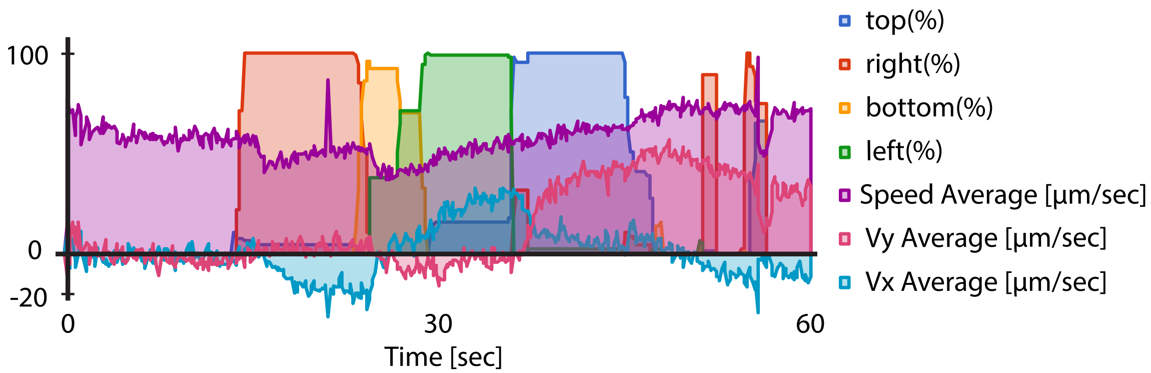


Figure 4.9: **Graphing examples by a student**. The average Euglena speed (purple) remains stable, while the mean velocity components in x and y clearly are either +ve/-ve or 0 depending on the light direction.

Students made several meaningful observations. Some observations were formulated as a causal clause ("It seems that when two Euglenas crush, their velocities change."), others were more observational ("Some spin like a gyroscope and others roll. They seem so random."). Interestingly, some of these observations have been published only in the recent literature, e.g., "Many Euglena appear to take at least 2 seconds to move when exposed to sudden intense light.", an effect described as transient freezing [88]. This suggests that students were not simply repeating what they have learned but also potentially generating new knowledge and that they could even make relevant discoveries in the future.

Students formulated many testable hypotheses. We did not explicitly tell them that a quantitative test should always have a dependent and an independent variable. 65 students formulated hypotheses and 42 of them were based on dependency of one variable on another: "The fewer the Euglena in the container, the faster they respond to the light."; "It looks like the rotational speed increases when the light intensity increases." When we analyzed all of the meaningful hypotheses based on what "If the independent variable is (increased, decreased, changed), then the dependent variable will (increase, decrease, change)." We identified more than 10 classes of both independent and dependent variables: (Independent: Light intensity (on/off, threshold), light direction (two vs. one side), exposure time/minimal time of illumination, cell size, Euglena density, Euglena crashing into each other, different online microscopes;) (Dependent: Aggregation, stay at one place, directed movement, spinning, spinning frequency, rotational speed (frequency), speed, response time, delay, frequency of cell-cell touching, behavior, behavioral transition, synchronization, activation, interaction between Euglena). Other strategies did not fall into these categories, such as testing for a correlation between the mean and standard deviation of the speed. Well over 100 hypotheses could be generated and tested with the platform, as constrained by the stimulus and observation space, which opens a large possibility space for learners to carry out versatile and self-driven inquiry projects. This potential is further enabled as imaging data contain a great deal of information, including the size and shape of individual cells.

Out of 21 students who voluntarily pursued actual experimentation to test their hypotheses, here we look at two particularly successful examples closely. These two students completed all the different phases of the self-guided investigation. One student observed that Euglena only reacted at light levels of 50% and higher and decided to test the percentage of Euglena moving away from the light in response to increasing light levels. The student programmed an experiment in *batch* mode in which light levels systematically increased in steps (Fig. 4.10A) and reported: “As the light level increased, movement across the y-axis doubled, whereas the x-axis stayed consistent.” The other student observed that Euglena were not always responsive to light and hypothesized that the cells were desensitized or “exhausted” by the stimulus, especially after repeated stimulation. The student then designed and ran a batch experiment with a single light-on step (Fig. 4.10B) and found that the magnitude of velocity away from the light source increased over time but did not desensitize or exhaust the Euglena. The student disproved his hypothesis but correctly noted that an experimental setup that allows running experiments for longer than one minute might help answer the question better.

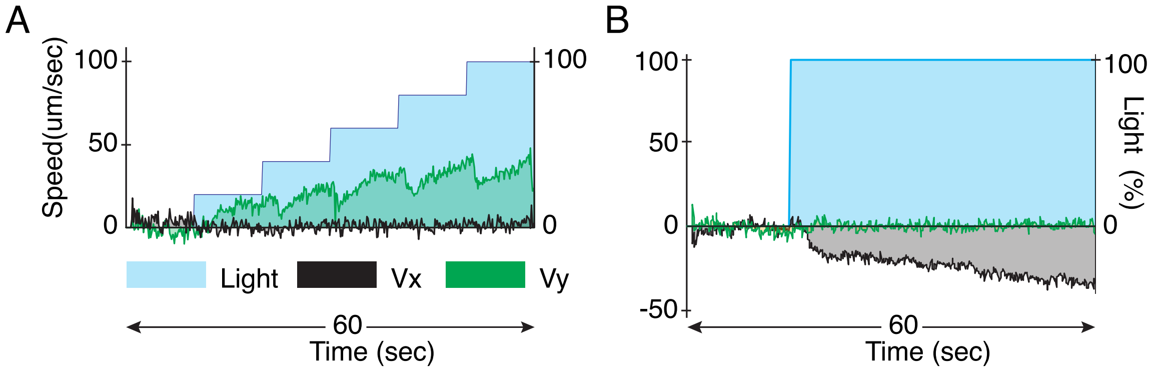
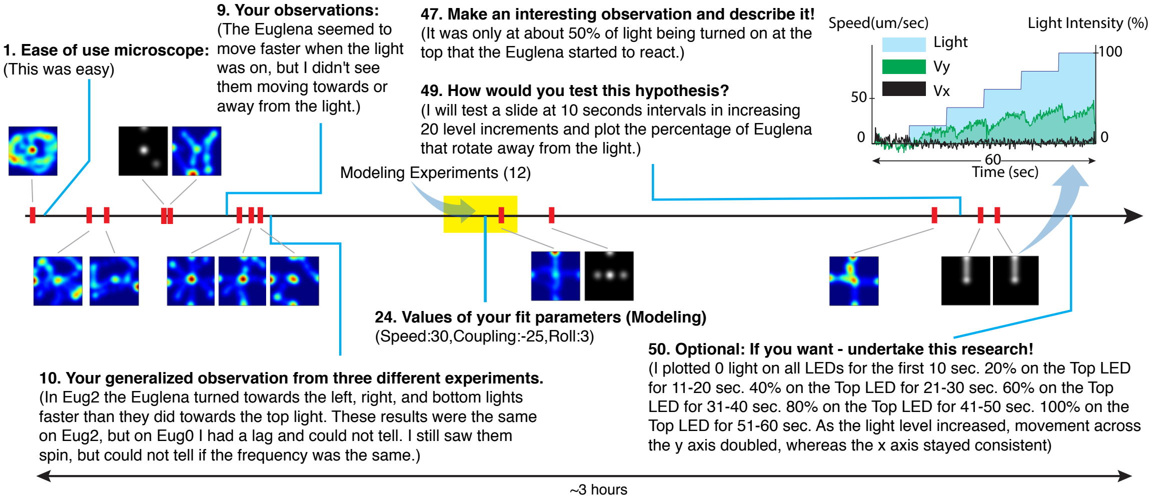


Figure 4.10: **Self-guided student projects**. A) Testing the strength of Euglena response (green) in response to increasing light levels (blue step trace). B) Testing whether Euglena desensitizes after prolonged light stimulus

Hence this unit demonstrates that the platform allows students to engage in self-driven investigations, make their own observations, formulate hypotheses, and design experimental strategies. Hence the platform has a large exploration space, i.e., a “high ceiling” [94].

Unit 7: Summary; reflections; learning The final unit summarized the course content for the students and asked students several questions. When asked about their learning experiences, students self-assessed to have learned “somewhat more than expected” (3.9 ± 1.0, scale of 1-5, N=29). Students also expressed that like to pursue this topic further “very much" (5.8 ± 1.1, scale of 1-7, N=24), and they learned about how microorganism interact with their environment “moderately” (2.9 ± 0.7, scale of 1-4, N=29).

We further asked 4 questions both before and after the course to assess students’ attitudes towards science. N=15 students voluntarily responded to both the pre and the post-survey (this question introduced only in session 4 and onwards). Answers were on a scale of 1-9 (not at all - totally): “Science is interesting”(pre=$7.5 \pm 1.4 \rightarrow$ post=7.7 ± 1.8); “I know what it is like to be a scientist” ($7.7 \pm 1.3 \rightarrow 7.4 \pm 1.7$); “Ordinary people can be scientists” ($8.7 \pm 0.6 \rightarrow 7.5 \pm 2.0, p < 0.05$); and “I can imagine myself as a scientist” ($6.7 \pm 1.8 \rightarrow 8.5 \pm 1.3, p < 0.005$). The answers to the first two questions were at a high level and did not change



1. Ease of use microscope:
(This was easy)

9. Your observations:
(The Euglena seemed to move faster when the light was on, but I didn't see them moving towards or away from the light.)

47. Make an interesting observation and describe it!
(It was only at about 50% of light being turned on at the top that the Euglena started to react.)

49. How would you test this hypothesis?
(I will test a slide at 10 seconds intervals in increasing 20 level increments and plot the percentage of Euglena that rotate away from the light.)

24. Values of your fit parameters (Modeling)
(Speed:30,Coupling:-25,Roll:3)

10. Your generalized observation from three different experiments.
(In Eug2 the Euglena turned towards the left, right, and bottom lights faster than they did towards the top light. These results were the same on Eug2, but on Eug0 I had a lag and could not tell. I still saw them spin, but could not tell if the frequency was the same.)

50. Optional: If you want - undertake this research!
(I plotted 0 light on all LEDs for the first 10 sec. 20% on the Top LED for 11-20 sec. 40% on the Top LED for 21-30 sec. 60% on the Top LED for 31-40 sec. 80% on the Top LED for 41-50 sec. 100% on the Top LED for 51-60 sec. As the light level increased, movement across the y axis doubled, whereas the x axis stayed consistent)

Figure 4.11: **Case study of the activity timeline of one student in the course (same student as in Fig. 4.10A)**. Numbers refer the task number in the course (out of ~71 tasks total). The images depicts the joystick pattern of the experiment being run at that point; grayscale images depicts *batch* experiment.

significantly. The third and fourth questions changed significantly in a way that suggests students considered science to be more challenging, and hence to require more skill/ability than initially thought, yet that they might see themselves more capable of being a scientist than before the course.

Whole Course Analysis: A Case Study There were a variety of ways students approached this course. Here, in order to gain longitudinal insight into the overall course and possible student activities, we analyzed one of the successful students as a case study (Fig. 4.11). This student is high-school teacher who ran 13 experiments (9 *live* and 4 *batch*) in order to complete the course in ∼ 3h. She ran 5 experiments, including one *batch*, before responding to the question "Your Observation" where she noted that Euglena moves faster upon light stimulus but the direction of motion was not clear. Later she executed 3 more experiments on different online microscopes to experience biological and system variability. By then, the response to light was clear but the direction of Euglena motion was not as obvious yet. She then ran 3 more experiments before formulating her hypothesis that "Euglena respond when light intensity is above the threshold of 50%." To test this hypothesis, she ran 2 more carefully designed *batch* experiments in which she ramped up the light intensity (top LED only) by 20% every 10s starting from 0%. She analyzed the experimental data, which revealed that Euglena swarms velocity along the vertical direction increased with the increasing light intensity while the horizontal component remained constant around 0. This observation confirms that Euglena exhibits phototaxis, which is dependent on light intensity. We looked at her videos and confirmed that she observed negative phototaxis all along; she moved her virtual joystick way too quickly in her initial experiments to watch the effect visually.

Students' Feedback Students liked the course and rated it as interesting and having the appropriate level of difficulty. All rated their overall experience between "very" and "extremely positive" (6.3 ± 0.6 on a 1-7 scale, N=34); difficulty was between "neutral" to "somewhat easy" (4.6 ± 1.1 on a 1-7 scale, N=31); guidance level was leaning towards "right amount" (2.8 ± 1.1 on a 1-5 scale, N=34). Students ranked the various lab activities (scale of 1-9, N=52), with "being able to stimulate cells with light in realtime" (8.4 ± 1.5), "modeling" (8.4 ± 1.2) and "download your own data and process and graph" (8.2 ± 1.2) among the most interesting.

The post-course student feedback about what they liked captures the key features we intended to reach with this online lab course: (1) Value of interactive and remote microscopy "The way we could conduct experiments remotely was very cool!"); (2) Performing deep scientific inquiry ("Feeling like I was part of real research"; "developing a scientific approach to study things"); (3) Learning biological content ("how I was able to see differences [sic] in the behavior of the Euglena"; "Learning new things about a microorganism"); (4) Synergistic integration of different activities and HCI instruments ("Highly interactive methods of using microscopes, movies, spreadsheets instead of dull passive theory on the characteristics of Euglena."); (5) Appropriate course design, content, and length ("Good amount of material for a short course."; "I liked the emphasis placed on the scientific method."; "I was able to prove myself I was going trough an investigation and how it was going from the easy things to some challenging"; (6) Lowering access barriers ("Being from develop country we dont have microscope or all the lab equpiments, this facility has provoked my passion to go for higher studies ...[sic]"; "I was able to show my child a microscope / microorganisms"); (7) Playfulness, fun, motivation, personalization, and feeling ownership ("The course helps make biology fun to learn", I liked playing around with the online microscope. It was fun looking for phenomenon on your own!"; "the way this course has been designed itself is motivating to get through the contents", "The ability to make my own experiments."); (8) Advancement beyond what current MOOCs can deliver ("I like the fact you are using online learning in a different way than most courses.", "It broke this limitation of MOOC courses that they were focused on theoretical [sic] lessons and not the ones requiring laboratory activities"; "I have paid some money in Coursera's lessons and i can say that this was the most interesting lesson i have followed.")

Students pointed out existing limitations and suggested future improvements: (1) extension of experiments (other specimens and organisms; other stimuli beyond white light including chemicals; zoom in and out; better microscope for visualizing the flagellum; longer sessions at the microscope; (2) more explanation and guidance on the data graphing and interpretations; (3) technical improvements such ability to download individual data files instead of a larger compressed file, or the ability to communicate with other students.

A central question is whether the effort to provide a real, interactive lab is justified compared to using interactive computer simulations as emulations of real systems. Currently, "interactivity" does not exist, for the most part, for microbiology informal educational contexts, as microscopy constitutes only passive observation. We asked the students for their opinions about the value of real-life experiments over computer simulations as we provided both within the same platform. N=39 students voluntarily responded and a majority $\sim 72\%$ expressed argument in favor of the real lab. Among them $\sim 36\%$ explicitly mentioned about how simulations may be inadequate at capturing fine details while a real lab provides ground truth data; "Yes, there should be a real microscope as it is impossible to guarantee that the behaviors of the simulation to be 100% natural/realistic. Using simulations rather than real cells could possibly mean that some unique phenomenon are not discovered." The other $\sim$ 36% discussed the inherent fun and motivating factor that a real lab ushers; "... using a real microscope is more exciting than using a computer simulation. Being excited is a better motivator for doing the course." The remaining students were ambivalent and thought a simulation was adequate for the purpose of this course.

An adapted version of this course may be particularly suitable for middle- and high-school science classes, as well as college. We identified and extracted feedback from teachers (K-12 and college, N=12). In general, teachers found the system to be powerful in fostering scientific inquiry due to the blending of real biology experimentation with data analysis and modeling; furthermore, filling a current gap regarding the Next Generation Science Standards NGSS[36]. Two teachers explicitly expressed interest in integrating this platform into their high-school biology classes the coming school year: "I believe that the emphasis on modeling, design, and quantitative analysis would be extremely helpful to AP Biology students, and I would love to try this in my classes." "I think that this would be a great thing to use with my students ... the thinking and feeling of being a scientist would be powerful for them."

4.10 Discussion

In summary, we successfully deployed an open online course with an integrated biology lab in a scalable manner. Students could engage in the core activities of scientific inquiry while interacting with living cells, which goes significantly beyond current educational practices of passive observation through a microscope or using computer simulations or animations; instead the lab automation and ease of data collection and analysis leads to easier logistics and extended lab time for students when working from home. The inherent capabilities for collecting automated learner data and using learning analytics techniques, and the different interaction modalities within the same platform open up interesting research avenues for researchers in education and HCI. In particular, users' course performance and feedback point to the positive potential of the platform. We draw the following conclusions and suggest future work:

First, we extended the implementation of Hossain [63] to significantly improve system throughput and fault-tolerance that was critical to a large scale deployments. During the six weeks the course was offered, the system operated far below its maximum capacity of 20,000+ experiments/week, and we encountered only four maintenance issues, e.g., changing the microfluidic chip and the Euglena culture of individual BPUs, each taking $\sim$ 20 min to resolve in a hot-swappable manner while the system remained operational. It is important to realize that this adaptation of cloud computing-like architecture was the key to enable a biology remote lab at MOOC scale.

Second, the user-friendly yet powerful data-handling formats and software interfaces made it possible for students to engage in the full scientific inquiry cycle [90], and especially using the same, self-generated data sources. The current course focused on model evaluation through parameter exploration as opposed to building models from scratch, but apart from that, the course covered all necessary components of productive laboratory experiences described by the NRC [105]. There are other systems that combine certain inquiry phases, like for example by integrating computer simulations with data analysis tools (e.g. CODAP [109]), or real experimentation with modeling tools (e.g. bifocal modeling [19]), but to our knowledge there is no other tool for biology that combines data analysis, modeling and real experimentation as coherently as ours, especially making it possible remotely and at a low cost per experiment.

Third, through iterative deployments, we converged on key design features (technology and courseware) to not only keep the non-deterministic and noisy behavior (which is inherent to all biological systems) under control but actually exploit its educational value. The importance of variability in the study of biological phenomena justifies the use of a real remote lab, but students cannot be overwhelmed by the noisy data, and the lab activities need to be well-scaffolded to avoid confusion. To that end, on the technology side, we had an automated quality assessment system to ensure a minimum level of reactivity of the cell in order to increase the signal-to-noise ratio. On the course design side, we scaffolded the activities carefully to make sure that data interpretations were correct, yet students recognized and appreciated the variability and noise in experimental data as meaningful for authentic inquiry in biology. Specifically, (1) we provided ample opportunities and encouraged students to repeat experiments on different setups throughout the course, and the fact that they could perform multiple experiments in just minutes, with minimal menial lab work, was quite helpful towards that goal; (2) we provided various prompts for what users should see and/or what stimulus they should use; it is recommended to only give users guidance after they have already tried (and potentially failed) on their own [101], and (3) throughout the course, we prompted students to consider the various aspects of variability, e.g., comparing real world data with ideal-behaving systems such as models, so they appreciate and recognize variability in signals. In the following we summarize key UI design rules (in the light of Section 4.6) that are also applicable to other cloudlabs beyond biology, and especially for large scale deployments with broader student pool:

1. **Provide flexible UI**: to support different experimentation modalities, e.g. we provided UIs for *live*, *batch* and *modeling* experiments.

2. **Utilize natural variability**: a key strength of a real cloudlab is in it's ability to expose natural variability of the underlying biology. While the system should mitigate variability to provide repeatable results, but it should also be flexible enough to expose variability whenever needed for educational purposes.

3. **Use simple but expressive data output and analysis formats**: Data extraction from biological images can be prohibitively complex. Therefore a cloudlab, especially for large scale deployment, should extract as much data as possible automatically. It is always tempting to using data formats that are easy to manipulate with programming APIs or specialized software. But doing so heavily narrows down the potential student pool. Therefore, its advisable to use simple formats that students are already familiar with, e.g. Excel or Google Spreadsheets.

4. **Provide visual analytics tools**: Provide visual analytics tool as much as possible to explore the data

without having to use complex numerical analysis or specialized software. This keeps the lesson simple, yet engaging, which is important to keep up the motivations of online students in MOOCs.

5. **Inform users about data quality**: Due to natural variability, experimental results may fluctuate. Therefore it is imperative that the system provides the BPU health status (responsiveness, density, motility 5) and other system parameters to inform users about the quality of the collected experimental data.

4.11 Future work

There are a number of important avenues for future research and development. (1) Refining and testing course content for specific relevant learner groups, such as middle and high-school biology, ultimately paving the way for usage by potentially several thousands or millions of students. (2) Include other relevant scientific practices such as collaborative teamwork or model building (rather than parameter exploration) activities. (3) Have participants do more complex projects all the way to geographically-distributed team projects. (4) Explore the potential for citizen science, or even let professional scientists work on the platform. (5) Utilize these platforms for deeper analysis using learning analytics to aid instructors and educational researchers. (6) Extend the platform to other experiment types (other light colors, other organisms, totally different microbiology experiments). (7) Update of BPU performance protocol, such as automatic LED brightness adjustment for optimal negative phototaxis response and feedback to users on “current instrument quality.”

4.12 Conclusions

In conclusion, we successfully deployed a real remote biology lab that enables authentic inquiry-based learning, for life science in an online learning environment at MOOC scale, with the following implications: (1) Applying the framework of cloud computing to biology experimentation lab robustly scales to large user numbers, which is beyond simply putting many microscopes online. We achieved significant cost optimizations, reliability, and economies of scale. (2) The integrated set of user interfaces realizes the NGSS view of the full inquiry, which is not possible with simulations or other existing solutions. In addition, for implementation in real educational settings, it is crucial to have integrated technological solutions that maximize instruction and experimentation time and reduce menial lab work and technical problems, which our platform affords. (3) The high-dimensional discovery space together with positive user responses regarding their scientific self-efficacy suggests the opportunity to not just "massify" science labs, but to actually democratize complex scientific practices. This technology could arguably be adapted to K-12 education for millions of users annually in the US alone, filling an unmet need as mandated by the NGSS[36].

Chapter 5

Biotic Processing Unit (BPU)

Equivalence to Turing completeness for Biology experimentation is not obvious, i.e. there is no known set of operations that is complete to carry out all biology experimentations. Therefore, we adopted a Domain Specific design philosophy to define a biology experimentation device, Biotic Processing Unit (BPU). A single BPU is designed to carry out a specific genre of experiment and operate on only certain types of biological specimens. However, in some cases, a different type of experiment or specimen could also be mapped within the operational space of the same BPU with little to no modifications. In this dissertation, it is important to note that our ultimate goal is to build a larger cloud computing like system with several instances of a BPU working as a distributed system. Therefore, a lot of the design principles are based on simplicity and minimality so that BPUs can be easily replicated cheaply at mass, is fault tolerant, and allows for high data and experimentation throughput.

5.1 BPU Components

Conceptually, a BPU (see Fig. 5.1) is composed of four layers of abstraction that are discussed below:

Bioware

A BPU houses biological specimen in one or more *wells* that are logically separated from one another. Physical separation may not be necessary; e.g. a single large culture of *Euglena* could be logically subdivided into a 2D grid, where each grid box constitutes a *well*. In this case, Euglena from one *well* could swim into another, but this cross-contamination is irrelevant, for example, in the phototaxis experimentation. Each *well* supports a separate experiment which could be shared across multiple *users* to allow collaboration, i.e. a BPU can run as many concurrent experiments as there are *wells*. In addition, every well in the BPU maybe fed from the same biological culture, depicted conceptually by a single biology input source, *Bio-Input*. However, every experiment may produce biological output in its own output sink, *Bio-Output*. Note that, in the case of

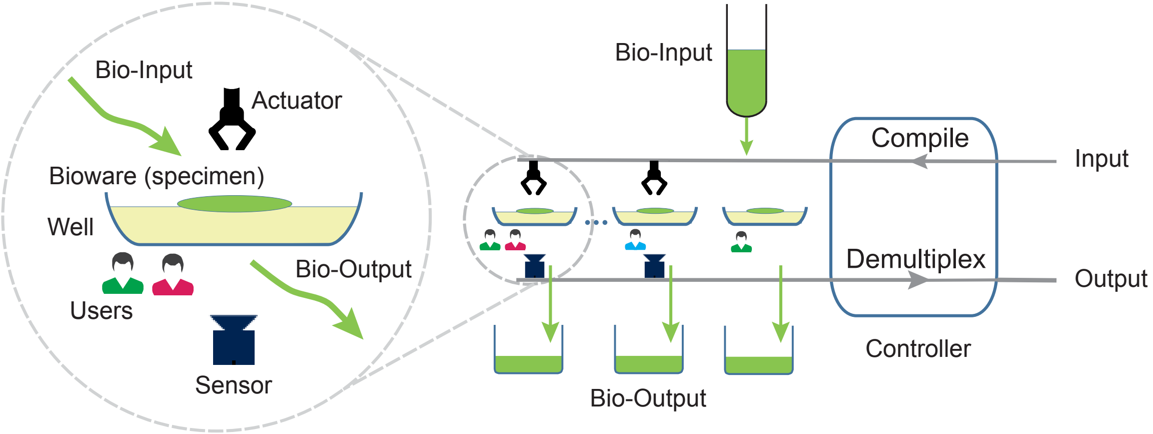


Figure 5.1: **Schematics of a conceptual Biotic Processing Unit (BPU)** that is composed of one or more *wells*, each housing the biological specimen, a *bio-input* (source of the specimen) and a *bio-output* (material output of an experiment) for each well. Together they form the (*bioware*) layer. There may be fewer *actuators* and *sensors* than that of *wells*, in which case, each *actuator* and *sensor* serve multiple *wells*. The *controller* compiles instructions from a digital *input* and de-multiplexes *sensor* readings (for each *well*) to produce digital *output* data. An experiment in a *well*, thus the entire BPU, may be shared across multiple users.

multiple organisms - i.e. population dynamics of multiple strains of bacteria, and maybe with separate physical sources of bacteria - every well gets seeded from all the cultures. That is, logically, there is still one input source, i.e. the bacterial cultures in combination. Philosophically, every *well* in a given BPU should see the exact same biological cultures and culture variabilities could be accomplished through multiple BPUs. This becomes particularly important when we measure the quality of a BPU (see Section 5.3 and 6.5) by running a phantom experiment in one of the *wells*, and especially when the nature of the biology is not *elastic* (see later in Section 6.4). This collection of *wells* with biological specimen, *bio-input* and *bio-output* form the *bioware* layer.

Actuators

External stimuli are applied on the biological specimen through *actuators*, which could be mechanical (e.g. liquid handling or pipetting) or electro-chemical in nature (e.g. modifying antibiotic or nutrient concentration in a bacterial culture, electric field across a culture of paramecium, microfluidic valves etc). Depending on the time-scale of the stimulus-response of the underlying biology (e.g. realtime response vs slow non-realtime responses to stimulus) a BPU can have fewer actuators than *wells* (see Chapter 2).

Sensors

Current experimental state in the *wells* are measured by the *sensor* layer that ultimately produces digital data as *output*. Optical Density (OD) readers, thermometers, flow cytometers, microscopy (imaging), etc and various combinations are examples of components that form the *sensor* layer. Imaging is a very common and information rich sensor for a lot of biology experimentation. Like *Actuators*, there can be fewer *sensors* than *wells* depending on the experimentation time-scale.

Controller and BPU Operation Model

The *controller* is essentially a microcomputer that drives the BPU. Experimental instructions (potentially from multiple users) are fed in as time ordered blocks of codes, expressed in some Domain Specific Language, that are compiled only at the controller site. The central cloud system, which stores and routes these instructions to the appropriate BPUs according to a schedule, is oblivious of the actual interpretation of the code, except for the time ordering of the the code block and metadata related to resource allocation, e.g the BPU identifier when a user wishes to use a particular BPU instead of getting automatically assigned to one (see Section 6.4). A single experiment is then a chain of such blocks of codes, where each block is a *declarative program* [81], i.e. instructions within each block has no time ordering. The total time ordering is accomplished by scheduling these blocks at different points on a timeline (Fig. 5.2). With this setup, the *controller* can interleave instructions from all the concurrent users' code *blocks* and operate the *actuators* in an optimal manner. This scheme is particularly advantageous in BPUs that has one mechanical *actuator* (*pippetor*) for many *wells* (see Chapter 2), in which case the *actuator* is time-shared across multiple experiments and users. In order to make an experimentation interactive, a BPU is not required to have the entire chain of instruction blocks at once except for the ones that are required only at the present time. Therefore these blocks can be streamed into the BPU, unexecuted blocks can be modified or deleted at a later time, or new blocks can be supplied on the go.

Furthermore, the *controller* acquires experimental data from the *sensors*, demultiplexes the data for each *well* (e.g. split the image for each *well* after scanning a single image for all the wells in one shot) before sending them back to the central cloud system through the digital output. The actuation and sensing may work on two different clocks, which does not necessarily need to be synchronized. Note that the BPU has two sets of input-output: 1) digital that can be easily interfaced with any computer system, and 2) biological (*bio-input* and *bio-output*) that maybe be controlled internally or by user instructions. For the purpose of easy and cheap replication of BPU, while maintaining higher overall experimental throughput, we perform all higher order data processing (e.g. computer vision based image processing on time-lapse data) in the central cloud system opposed to at the *controller* site. This allows the *controller* to be implemented using low powered devices such as Arduino or Raspberry Pi boards. In short, the key operational philosophy of the *controller* is to execute instructions and relay back experimental data *as quickly as it can* without much processing.

For robust functioning, especially in the context of many instances of BPU in a cloud system, automatic fault detection and correction is critical for any BPU. At the minimum, BPUs should be able to report error situations, halt operations while the cloud system can then draw attention for manual intervention. Extra maintenance sensor components may be installed to detect hardware malfunctioning automatically, e.g. in Chapter 2, we added several mechanical sensors to make sure the pipettor gantry is not derailed and the pipettor tip returned to home position after dispensing. In some cases, these malfunctions can be corrected automatically, e.g. in Chapter 2, *controller* would try resetting the gantry, which also involved re-calibration

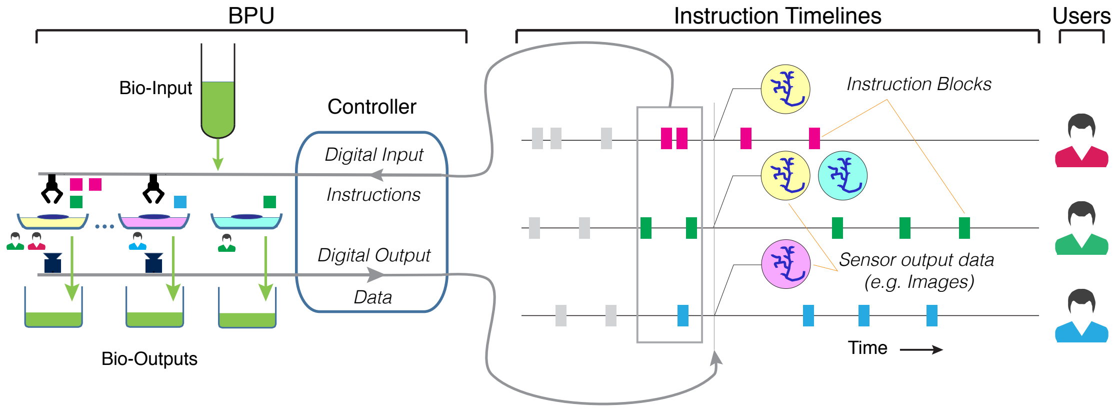


Figure 5.2: **BPU operation model**. Experimental instructions are specified as *declarative program blocks* on individual timelines of users (color coded to match the corresponding user), who maybe time-sharing a single BPU over multiple *wells*. The BPU discretizes time and fetches instruction *blocks* from all the time-sharing users (the tall rectangle encompassing instruction blocks across timelines) compiles and interleaves these instructions to perform actions on the experiments optimally (depicted by color coded squares). New blocks maybe added in the unexecuted side of the timeline and existing blocks maybe be modified or removed. The BPU itself or user instructions may control the *bio-input* and *bio-output* directly depending on the nature of the experimentation. *Sensors* read experimental state, which is fed back to the users after being demultiplexed by the *controller*. The *sensor* and *actuator* layers do not necessarily have to operate on the same clock frequency.

procedures, and retry pending actions a number of times before finally giving up and report errors. These automatic correction procedures dramatically improved the robustness of the overall cloud system developed in Chapter 2.

5.2 Experimentation Timescale and BPU Design Considerations

Biology experiments can be diverse and span timescales ranging from realtime (phototaxis of Euglena, see Chapter 3), non-realtime (chemotaxis of Physarum, see Chapter 2) to even taking days (bacterial population dynamics with bottlenecks). While the overall throughput of a cloud system could be increased by deploying several BPUs in parallel but the design of each is largely determined by the stimulus-response timescale of the underlying biology. For non-realtime responses, it's sufficient to have far fewer *actuators* and *sensors* than the *wells*, as a single *actuator* or a *sensor* can serve multiple *wells*. On the other hand, for realtime responses, the implementation could be taken to the limit where every *well* is connected to a dedicated *actuator-sensor* pair, at which point one may even consider constructing much simpler BPUs with a single *well-actuator-sensor* combination. In this case, code blocks may contain exactly one instruction and can be streamed in (implicitly time-ordered) while output data can be streamed out for a fully interactive experimentation session (see Chapter 3).

We implemented three different types of BPUs so far and they are compared in Table 5.1. The first BPU

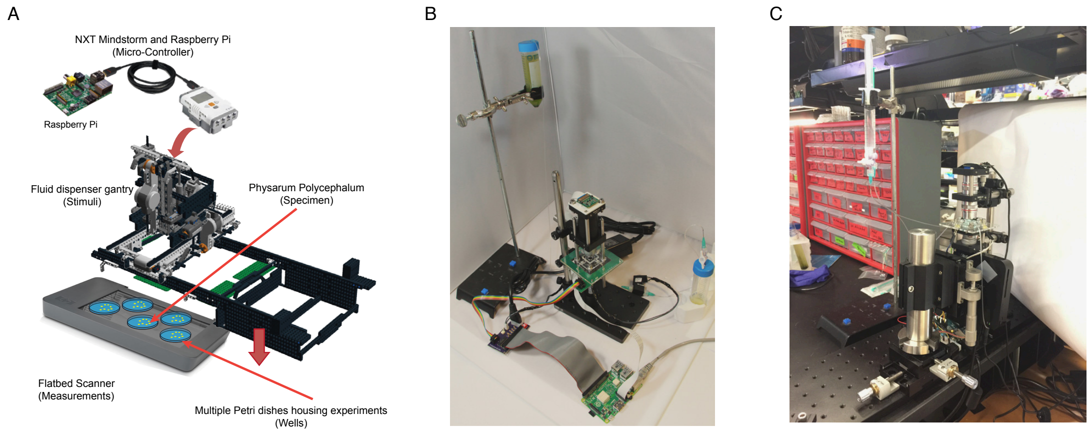


Figure 5.3: **We have implemented three different BPUs with different configurations of *wells* and *actuators*.** A) A liquid handling based BPU that has a single pipettor based actuator, and 6 Petri dishes as *wells* placed atop a flatbed scanner, which was the time-lapse imaging *sensor*. The pipettor would dispense oatmeal solution on the Petri dishes, which housed cultures of *Physarum*, in batches according to user instructions. A Raspberry Pi board in combination with a Lego NXT Mindstorm unit would form the controller unit. B) This is a single *well*, single *actuator* BPU, which would house a culture of Euglena in a microfluidic chamber (the *well*). Four LEDs (the actuator) would be used in combinations to illuminate the chamber from the four sides with varying intensities, thus providing both directional and intensity based light stimulus. A vial contains extra Euglena culture (bio-input) that could be flushed into the chamber using gravity by opening a valve and eventually collects in a downstream reservoir (bio-output). A Raspberry Pi in combination with a driver circuit board forms the controller unit. C) This is a multi *well*, multi *actuator* BPU, which could also be transformed into a single *well*, two *actuator* based BPU. The construction is similar to (B) and houses the same organism Euglena in a single microfluidic chamber except intricate light patterns can also be projected from below using a light projector. The light projectors provide spatially varying light stimulus with a range of frequencies that allow us to spatially subdivide the chamber into smaller logical *wells*, each having its own light pattern source in the projector space (multi *actuator*). If the four LEDs are used, which are global to the entire microfluidic chip, the BPU turns into a single well, with two actuators (projector and four LEDs) setup.

(Figure 5.3A) was based on the chemotaxis of a macroscopic slime mold, *Physarum Polycephalum*, whose response timescale was in the order of minutes, i.e. takes ∼10 minutes to notice any visible responses given the imaging resolution and the nature of the experimentation itself. The BPU in this case (see Chapter 2) had 6 *wells*, thus supporting 6 experiments with multiple users, and a single pipettor (*actuator*) to dispense a trail of oatmeal for Physarum to follow. Due to the non-realtime nature of the underlying biology, the BPU would poll the central cloud system to collect appropriate code blocks from several users every 10 minute before executing all the pipetting tasks in one shot. A flat-bed scanner (see Figure 2.3) would take time-lapse images of all the 6 *wells* every 10 minutes.

The second BPU (Figure 5.3B) was based on the phototaxis of a single celled microswimmer, *Euglena Gracilis*, which exhibits realtime responses to light directions in a fraction of a second for individual cells and ∼7 secs for an an entire swarm to align with a given light direction. This BPU had a single *well*, i.e. microfludic chamber with Euglena culture, with a dedicated *actuator* (4 directional LEDs along 4 sides of

Experiment Type	**Timescale**	**Wells**	**Actuators**	**Duration**	**Data Rate (B/s/exp)**
(A) Physarum	non-realtime ($\sim$10 mins updates)	6	1	2 days	~ 750
(B) Euglena (with directional LED)	realtime (10 frames/sec)	1	1	1 minute	$\sim 20\times10^3$
(C) Euglena (with directional LED & projector)	realtime (10 frames/sec)	N	N	1-5 minutes	$\sim 20\times10^3$

Table 5.1: **Characterization of three different BPU types we have implemented**. The letter in the first column corresponds to the subpanels of Figure 5.3. (B/s/exp: Bytes per second per experiment).

a microfluidic device, see Chapter 3) and a dedicated *sensor* (webcam microscope). A typical experiments would last for 1 minute during which time a user can interact with the Euglena in realtime by streaming in instructions with a joystick interface and watch responses via a live streamed video output (see Chapter 3). In this case, the *controller* would simply relay streaming data in and out of the *actuator* and *sensor*.

The third BPU type (Figure 5.3C) is also based on Euglena phototaxis, except the light stimuli is provided more locally on the 2D surface of the microfluidic chamber by a projector that shines patterned light from underneath the microfluidic chamber. This allowed us to virtually partition the chamber area into multiple smaller *wells*. A large image with patterns drawn on it is used to project the light. This BPU effectively has as many *actuators* as there are *wells* because the projected patterns can be drawn concurrently by several users. Another interesting variant of this BPU could be constructed by adding the directional 4 LEDs along with the projector. With this setup, Euglena motion could be guided more efficiently with combinations of both local and global light stimuli. However this variant has only one *well* because the 4 LEDs are global to the entire chip, but with two *actuators*.

5.3 Quantification of Biological State

It is imperative to automatically monitor and quantify a BPU's underlying biological state, which may go through natural variations, e.g circadian rhythm, population decline, growth, contamination and etc. This natural variability, often unpredictable, makes biology experimentation systems uniquely challenging from the physical ones. Importantly, this biological state is later taken into account while automatically routing online users to the best possible BPUs (see Chapter 6) and ultimately providing biological fault-tolerance. In this section, we will briefly discuss the quantification scheme implemented in Chapter 3 and then follow up with a more general framework that could be potentially applied to other types of stimulus-response based experiments.

Euglena Phototaxis Response Quantification

In Chapter 3, we measured the state of a Euglena culture across three different variables: *cell density*, *cell activity* (how much the cells moved between successive video frames) and *response*. The *cell density* and *cell activity* can be estimated easily by simply detecting cells using computer vision and analyzing successive image differences. The *response* is computed as a score by running an automated experiment (similar to Fig 3.2e), where each LED is turned on at full brightness for 30 secs in a clockwise sequence starting from the top LED. The orientation (in acute angle with respect to the horizontal axis, with 0 degrees corresponding to a Euglena aligning its length of the cell with the horizontal axis, see inset of Fig. 3.2e) of the detected Euglenas in all the video frames are measured during the last 15 secs. If for example, the top/bottom LED is on, we measure the fraction of Euglenas that are oriented vertically, i.e. the acute angle is greater than 45°, otherwise horizontally (acute angle $< 45^\circ$) in the case of left/right LED. This way, the fraction depicts the probability of finding a Euglena that is responding to the light. It is fair to assume that these probabilities are independent as Euglena state from previous lighting condition is restored within a couple of seconds. Therefore, we compute the joint probability of response for all 4 LEDs and compare it with random response, in which case the expected probability is 0.5, to arrive at a score $R \in [0,1]$ as the following, where p_i is the fraction Euglenas responding to the light i:

$$\begin{aligned} R &= \max\left(0, \frac{1}{4}\log_2\left(\frac{\prod_{i=1}^{4} p_i}{0.5^4}\right)\right) \\ &= \max\left(0, 1 + \frac{1}{4}\sum_{i=1}^{4}\log_2 p_i\right) \end{aligned}$$

Alternative schemes with more sophisticated cell tracking is also possible but any quantification with the following properties is sufficient:

1. **Positive**: positivity is desired to avoid unwanted side-effects during routing (Chapter 6.1), where *response* is weighted with other variables and system parameters. Besides, negative response philosophically means there is actually response but in opposite direction.

2. **Monotonically increasing**: the numerical values are positive and increases with increasing response quality.

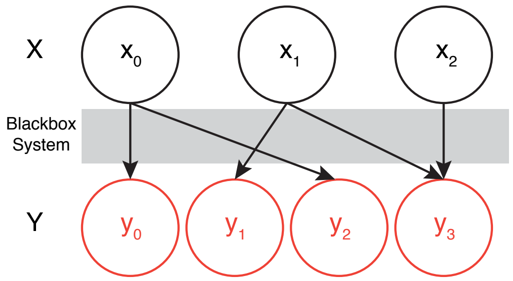
3. **Bounded**: The upper bound is known and thus the score can be scaled to $[0,1]$.

4. **Stable**: arrives at a similar numerical value when experiments are repeated around the same time.

Therefore, we chose a simpler scheme, based on Euglena orientations only, that satisfies the above conditions without heavy computational complexity.

A General Framework for Quantifying Stimulus-Response

A different biology and BPU setup might require a different *response* quantification scheme altogether. In this section, we propose one general framework, based on *Mutual Information* [37] that is applicable to any stimulus-response based biological setup.



	y_0	y_1	y_2	y_3	P(X)
x_0	0.25	0	0.083	0	0.33
x_1	0	0.21	0	0.125	0.33
x_2	0	0	0	0.33	0.33
P(Y)	0.25	0.21	0.083	0.455	

$$I[X;Y] \sim 1.2 \text{ bits}$$

Figure 5.4: **A contrived example to demonstrate how to compute *Mutual Information***. Here we assumed we had full control over the input and therefore uniformly sampled the input space while the blackbox system reponded with a random variable Y with certain probabilities as made up in the table. For this example we used $\log_2$. Note that if Y were completely determined by some function of X, then in that case $I[X;Y] = H[X] = \log_2 3 \sim 1.58$. Comparing to this, the contrived blackbox system happens to have strong stimulus-response dependance.

Mutual Information is a measure of how one random variable is related to another in general. In this section, we will only concern ourselves with the discrete form of *Mutual Information*. Thus, given a discrete joint probability distribution of two random variables X and Y, *Mutual Information* (I) is given by:

$$\begin{aligned} I[X;Y] &= \sum_{x\in X, y\in Y} P[x,y] \log \frac{P[x,y]}{P[x]P[y]}, \quad \text{Note, } \lim_{v\to 0} v \log v = 0 \\ &= H[Y] - H[Y|X], \quad \text{where, } H[Z] = -\sum_{z\in Z} P[z] \log P[z] \quad \text{is the entropy measure} \end{aligned}$$

A key property of *Mutual Information* is $I[X;Y] \geq 0$, and for log base of 2 the unit is *bits*. It is important to note that *Mutual Information* captures dependencies between two random variables more generally than simple correlation-coefficients, which only captures linear relationships. For example, if X and Y are independant of each other, then $I[X;Y] = 0$ (because $P[x,y] = P[x]P[x]$), i.e. knowing one gives "no information" about the other. On the other hand, if y is given by some deterministic function of x ($y = f(x)$), then $H(Y|X) = 0$ and consequently $I[X;Y] = H[Y] \geq 0$. In Figure 5.4 we demonstrate how to compute *mutual information* of a contrived blackbox system. Here the output random variable Y is observed in response to an input random variable X.

In this framework, the first step is to discretize both input stimuli and output response spaces. In a BPU there are several choices and resolutions of discretization for a given biology setup, and each choice will yield a different entropy measure and hence mutual information. However, for our purposes, a reasonable choice that is able to differentiate the *"goodness"* of the state of underlying biology is sufficient. In the following we will discuss two possible discretizations of the systems we have developed in Chapter 2 and Chapter 3.

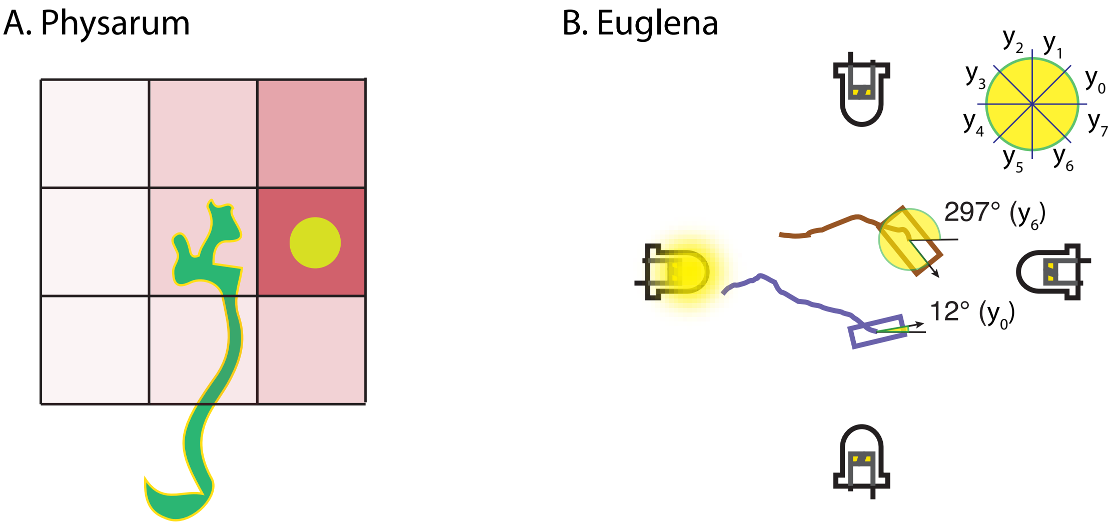


Figure 5.5: **Discretization of input and output stimuli-response spaces of two different BPUs.** A) Physarum based system in Chapter 2. The input space is spanned by the 9 possible states from the 9 food dispensing positions (food present vs not present). The output space is also spanned by 9 states determined by which grid block the physarum biomass moves into within a pre-determined time frame. In this figure, we denote food being dispensed in one of the grid blocks by a yellow circle. Shades of red depict the relative probability of the Physarum moving into the grid blocks within the next time frame. B) Euglena based system in Chapter 3. The input space is defined by the 4 LEDs and thus 16 possible states with on/off, which could also be trimmed down to only 8 directional ones. The output space could be constructed the heading angles of the cells (as shown in the inset with 8 bins). In this figure we depict two Euglena cells are swimming away from the lit left LED.

Physarum System (Chapter 2) A 3×3 virtual square grid is placed near any growing front of Physarum and the initial biomass distribution in each of the 9 blocks noted (Fig. 5.5A). These 9 blocks correspond to both input and output states. The input food stimulus is dispensed in any of these 9 blocks randomly and the biomass distribution of Physarum in all the 9 blocks is measured after a time period. The positive differences between the final and initial biomass distribution are attributed to the response due to the food stimulus. This difference is converted to a conditional probability distribution of the 9 states given the input food block, which is eventually added to the joint probability distribution table (see Fig. 5.4). Note that, if the Physarum does not move at all, this conditional distribution would be all 0 and nothing will be added to the joint probability, yielding in no *Mutual Information* as expected. Lastly, we know Physarum typically grows at 1cm/h, thus with the given spatial discretization induced by the virtual square grid, the second biomass readings can be taken after a predetermined timeframe.

Euglena System (Chapter 3) With 4 LEDs, the input space has 16 states (based on full brightness on/off) but for simplicity, one could either use 4 or 8 directional ones (Fig. 5.4). The output space could be defined by discretizing the heading angle of Euglena cells, which can be sampled ~ 7 secs after the light stimulus is applied (see Fig. 3.2e for the Euglena swarm alignment time-scale). The sampling could be performed over a time window, during which time, every Euglena cell on every video frame could be used to update the joint probability distribution (Fig. 5.4).

Even though the *Mutual Information* based framework can be applied generally to any stimulus-response based system to quantify the “goodness” of a BPU, it requires lots of stimulus-response trials to construct the joint probability distribution. Therefore, a judicial design of the state space with fewer states might be desirable for practical purposes.

Chapter 6

Cloudlab Architecture

6.1 Introduction

A cloudlab system is composed of a backend central server system that manages an array of identical BPUs (see Figure 6.1). In this chapter we discuss the general principles we adopted regarding user management and data handling, which resembles a distributed systems (see Figure 6.1), and then discuss two separate designs of scheduling/routing depending on the nature of the underlying biology. As a reminder, a key assumption made in this dissertation is that BPUs are essentially independent and do not communicate with each other directly. That is, the implementation follows a *master/workers* pattern, the *master* being the *schedular* and the *workers* being the BPUs.

6.2 User and BPU management

Users are assigned to groups, similar to UNIX groups. BPU visibility (which BPUs a user can access) and capabilities (which parts of the instrumentation a user can operate, what kind of output data is produced in the post-processing and which data is visible to the user) are defined by these groups.

Formally, we have a set of groups $\mathscr{G} = \{g_0, g_1, \ldots\}$, users $\mathscr{U} = \{u_0, u_1, \ldots\}$ and BPUs, $\mathscr{B} = \{b_0, b_1, \ldots\}$. Capabilities can be defined as set of flags $\mathscr{C} = \{c_0, c_1 \ldots\}$ and only tied to groups, i.e. each group is associated with some capabilities by $C(g_i) \subseteq \mathscr{C}$. Here, group g_0 is considered an admin group with full capabilities, i.e $C(g_0) = \mathscr{C}$. Each user and BPU belongs to one or more groups as denoted by $G(x) \subseteq \mathscr{G}$ where x can be either a BPU (b) or a user (u). Under these definitions, a user is only able to see a set of BPUs that have atleast one common group and we will call this set $CandidateBPUs(u) = \{b : G(b) \cap G(u) \neq \emptyset\}$. However, the capabilities of the user will be determined by combining the capabilities of the all the groups that this user belongs to $C(u_i) = \bigcup_{g \in G_u(u)} C(g)$ instead of basing it off of the $CandidateBPUs$. This is because, the user maybe member of some group that enables some capabilities, but none of the candidate BPUs may be member

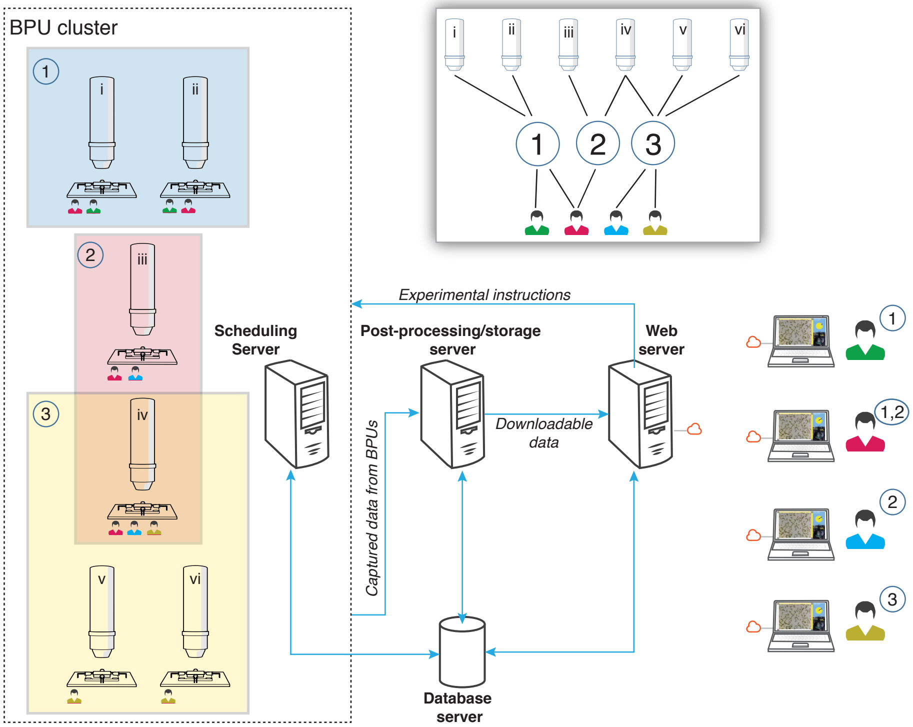


Figure 6.1: **Distributed software architecture for the cloudlab.** A cluster of BPUs is partitioned into groups, with BPU iv being shared with group 2 and 3 for example. Users may belong to multiple groups and eventually access BPUs according to the group membership, as illustrated in the inset. The architecture contains 4 main layers, 1) Scheduling server, which is the heart of the system that manages scheduling, resource allocation, and auto-monitoring. 2) Post-processing/Storage server that computes on experimental data, asynchronously, after all, data is acquired by the BPUs. 3) Web server connects communicates with the end users and provides the top level application for the users to interact with. 5) Database server not only holds static information but also the dynamic information such as the experiment queue of the scheduling server for persistence. While most of the design remains the same, the implementation of the scheduling server eventually depends on the time-scale, duration, and elasticity of the biology.

of this group, i.e. $g \in G(u)$ but $g \notin \bigcup_{b \in CandidateBPUs(u)} G(b)$. Albeit, the admin user, say u_0, belongs to all the groups by default and thus have all capabilities and full visibility of all BPUs. See Fig. 6.1 where BPUs are partitioned by group, with BPU iv being shared with group 2 and 3, and the users (color coded) see these BPUs accordingly. Therefore, we could dynamically partition (with possibly sharing BPUs across multiple groups of users) an array of backend BPUs for concurrent deployments, e.g. for a school in Chicago and a MOOCs at the same time.

6.3 Software System Components

The cloudlab software system is composed similarly to a standard webservice based system, with four main components as follows (see Fig. 6.1):

1. **Web server**: provides a mean for users to submit (*batch*) or interactively (*live*) execute an experiment on the array of the backend BPUs. The experimentation UI is *domain specific* to the nature of the underlying biology (see Fig. 2.4 and 4.4). Furthermore, the webserver should also provide means to download experimental data at will and additionally could allow data exploration within the website itself without downloading the full package. For example, in Fig. 4.4 users were able to visualize Euglena tracks with other overlaid information on the website without downloading the full data package.

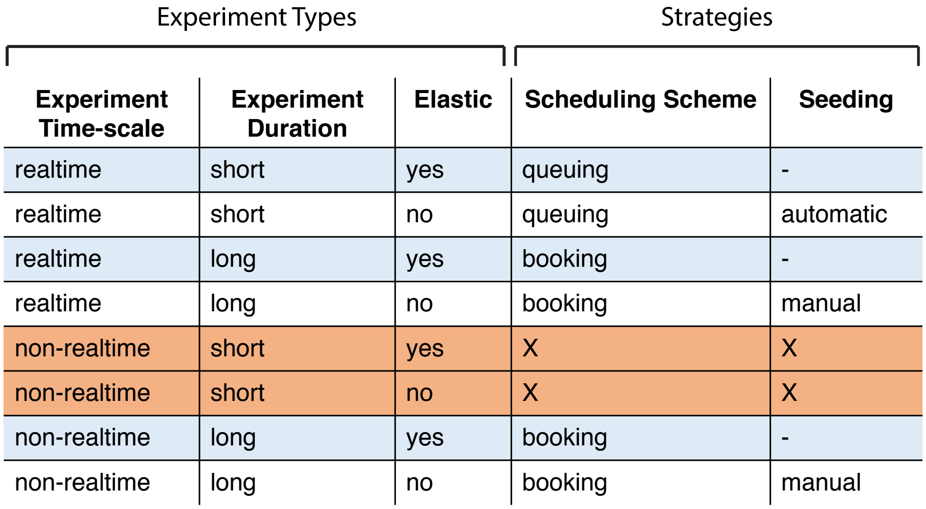
2. **Scheduling Server**: this is the heart of the system that schedules users, routes experiments to the appropriate BPU (resource allocation), schedule instructions to the BPU, and automonitors the BPU for the quality of the hosted biology. The scheduler is implemented as an independent webservice so that any frontend webserver can communicate with it, for example, a third party webserver that intends to use our cloudlab for experimentations only. There are several choices of scheduler design, which depends on the biology experimentation being supported and we will discuss this in more details shortly (Section 6.4).

3. **Post-processing and Storage server**: BPUs are primarily data acquisition devices, while all data processing e.g. autodetecting and tracking cells are performed on this server asynchronously and stored for later downloads. Having a separate post-processing server allows our BPUs to be low-powered, cheaper and simpler thus making them easily replicable. User capability flags (see above $C(u)$) are checked here to decide what data and how they need to be processed. We will later see that data-processing for BPU auto-monitoring is also performed here. While the data communication and storage from the BPUs could be standardized easily, the actual processing module should be implemented in a way that it can be subclassed and extended for different BPU types on a different cloudlab deployment.

4. **Database server**: beside the standard role of any database server, we implemented all the queues of the scheduling server in the database for persistence. So that, even if the scheduling server fails, the queues could be restored easily and operation resumed.

6.4 Resource Allocation, Scheduling, and Seeding

Resource allocation and scheduling largely depends on the time-scale (realtime or non-realtime) and the overall duration (takes minutes to complete or takes hours or even days) of a single experiment. On the other hand, whether experiments need to be automatically re-seeded or a manual procedure for this is sufficient depends on the *elasticity* of the biological specimen (whether the biology returns to its rest state after an experiment is



Experiment Types			Strategies	
Experiment Time-scale	**Experiment Duration**	**Elastic**	**Scheduling Scheme**	**Seeding**
realtime	short	yes	queuing	-
realtime	short	no	queuing	automatic
realtime	long	yes	booking	-
realtime	long	no	booking	manual
non-realtime	short	yes	X	X
non-realtime	short	no	X	X
non-realtime	long	yes	booking	-
non-realtime	long	no	booking	manual

Figure 6.2: **Scheduling and seeding strategies for different experiment types.** Experiments that are non-realtime are unlikely to have any meaningful experimentation with short durations and therefore omitted.

over). For example, our Euglena system (Chapter 3) was *elastic* (Euglena swarm would take $\sim 7-10$ secs to return to its rest state of random swimming pattern), whereas the Physarum system (Chapter 2) was not.

Figure 6.2 enumerates combinations of experiment types across three binary parameters *time-scale*, *duration* and *elasticity*. To clarify, experiments that exhibits observable responses in the order of sub-seconds is considered *realtime*. *Short* duration experiments are those that finish within minutes whereas the *long* experiments takes hours or even days. Out of the 8 combinations, two are ommitted because with a *non-realtime* response its unlikely to have a meaningful *short* experiment.

In this dissertation we have developed two systems that lie on the opposite ends of this three-parameter spectrum (see Fig. 6.2). Though most of the system components of Fig. 6.1 and the definition of the BPU (see Chapter 5) remains the same, the resource allocation strategies and scheduling were different that mostly stem from the differences in the duration of experiments. In the following, we discuss these strategies in details.

BPU Allocation by Stripping and Timeslot Booking: Physarum System

Physarum chemotaxis was *non-realtime*, the duration of each experiment would be 2-3 days (*long*) and the nature of the biology was *inelastic*, i.e, the setup had to be re-seeded after every experiment was over. For this setup, users scheduled themselves on a calendar by *booking* a timeslot and request for a number of concurrent experiments ahead of time. Physarum plates would be seeded according to users' prior request (Physarum placement and possible arrangement of oatmeal flakes would be drawn on an image and submitted ahead of time). The *scheduling server* would allocate the BPU *wells* for the current set of experiments using a *stripping* strategy, i.e. experiments from a single user were allocated on different BPUs before reusing one. This reduces the chances of one user losing all her experiments in an event of catastrophic BPU failure. All the experiments would then be started in a batch (twice every week with 3 BPUs, and a total of 18 experiment slots) and users were notified through emails. Each experiment supported multiple users (see Chapter 5), and we utilized this by assigning pre-declared collaborators to a single experiment. In this case, a group of collaborators would be considered as a single user from the resource allocation perspective. During the 2-3 days of experimentations, users would interactively submit instructions in *blocks* of *declarative programs*, pre-program ahead of time, and watch responses in ~ 10 minutes intervals as described earlier in Section 5.1. The system throughput, in this case, was bounded by the duration of the experimentation and therefore the re-seeding of the biology for the next experimental session did not need immediate automation, and was mostly handled manually.

During experimentation, the *scheduling server* would discretize time (~ 10 minutes interval), group relevant instruction blocks from all the users for the current time bin by BPUs and dispatch. Note that the entire instruction chain of a single user is not sent to the BPU and therefore new instruction *blocks* can be added, unexecuted *blocks* removed or modified interactively later.

Automatic Call Distributor (ACD) Based Routing and Queuing: Euglena System

Euglena phototaxis, in complete contrast to the Physarum, was *realtime*, experiments were of *short* duration and the biology was *elastic*. BPUs in this case was implemented with a single *actuator* (4 LEDs) and single *well* (microfluidic chamber) configuration. A typical experiment would last only a minute and therefore, we implemented a queuing based scheduling with Automatic Call Distributor (ACD)[73] based routing for resource allocation. Users would wait in a queue, similar to a call-center, while the *scheduling server* would route the experiments, on a First-Come-First-Serve (FCFS) basis, to the best available BPUs, which was determined by weighing the current biological state (*response*, *cell density*, and *cell activity*, and the estimated wait-time (dynamic). Note that the system also allowed users to specify a BPU instead of getting automatically routed. Otherwise, the queuing was of type *M/D/c* according to the *queuing theory* taxonomy (M: random Poisson process for the incoming experiments, D: fixed time spent on each BPU, c: an array of c number of BPUs). However, closed form solutions of *M/D/c* queues [49] are typically not applicable as the real scenario was quite dynamic with *live* mode users dropping out and BPUs not necessarily identical in terms of quality. Therefore, modeling and simulation were required to compute system characteristics. For example, in the case of the system developed in Chapter 4 with 6 BPUs, simulation revealed that it could handle over $30,000$ experiments per week with 90% chance that the experiments would land on BPUs with clear visual Euglena responses (see Appendix B.2 for more details).

The *scheduling server* internally maintained a separate queue for each BPU, which were dynamically updated and re-shuffled, i.e. experiments from one queue could be sent to another whenever deemed appropriate. For example, the system would continuously ping the BPUs to check for status and whenever a BPU failed or a new BPU is discovered, the queues were re-shuffled for an optimal resource allocation. Experiments were either submitted as pre-programmed scripts (*batch* mode) or interacted in realtime (*live* mode) with a joystick. *Live* mode users were handled differently as a user leaving pre-maturely while waiting in the queue, would waste valuable experimentation time. So the scheduler would ping the webserver and wait for a brief period of time (asynchronously from other queues) to check the presence of the user just prior to sending her *live* experiment to the BPU.

6.5 Auto Monitoring Framework

In this section we discuss auto-monitoring mostly in reference to the Euglena system (Chapter 3, 4). The scheduling server automatically monitors the BPUs by spawning phantom *batch* experiments using the same user-management and experimentation framework as other experiments. For this, the system has a set of bot users - *responseUser, densityUser, activityUser* for measuring *response, cell density* and *cell activity* respectively. These bot users belong to a special group called *automonitor*, which includes all the BPUs by default. From time to time, the bot users create pre-defined experiments (*bach* mode) and submit them to the queuing framework with all the other experiments from normal users. The *post-processing* server has special extendable modules for each bot user to compute the respective BPU quality and eventually updates the database with the current value. In this case, however, all the quality measures described above could have been measured by a single bot user and thus a single experiment, but the key is that the design allows for multiple bots. For *inelastic* biology, which needs to be re-seeded after every experiment, the automonitoring framework essentially measures the quality of the *bio-input* culture. Having a single *bio-input* (as described in Section 5.1) allows us to run auto monitoring to utilize any *well* as a proxy for the entire BPU.

6.6 Discussion

Out of the 6 combinations of the three-parameter space stimulus-response based experimentation (Fig. 6.2), we implemented 2 as described earlier. The biology in 2 of the remaining are *elastic*, for which we only need to know the recovery time-scale to insert waiting periods between successive experiments. Re-seeding is only necessary in the event of a catastrophe, e.g. death or contamination and therefore immediate automation is not necessary. The other 2 combinations exhibit *realtime* responses and *inelastic*, but one has short duration experimentation, which requires automatic re-seeding mechanism otherwise vital throughput is lost. Finally we recommend, ACD based scheduling (*queuing*) for *short* duration experiment and *time-slot booking* for *long* experiments.

In this dissertation, we did not concern ourselves with security and malicious users, but capping usage is important from an operational perspective for the ACD based scheduling. We implemented this by primarily putting a cap on the total experimentation time of all the experiments currently in the queue by a single user. Experiments that are too small are automatically rejected by the *scheduling server*, and these together implicitly caps the total number of experiments as well prevents very long experiments. On the other hand, experimentation duration is set by the backend system in the *time-slot booking* scheme.

Chapter 7

Conclusions

In this dissertation we have introduced the concept of a *domain-specific* biological experimentation device (BPU) and implemented two different cloudlab architectures - for realtime (Euglena phototaxis) and non-realtime (Physarum chemotaxis) experimentations - that scales linearly and can handle a large volume of users and internet traffic. The natural variability, usually unpredictable, makes biological experimentation systems uniquely challenging compared to non-biological ones. We laid out a general framework for automatic monitoring of the underlying biology and factored that into our routing scheme to provide biological fault-tolerance and ultimately better QoS. Our proposed framework can be replicated even for cloudlabs that exist in other engineering disciplines [60], where labs are mostly oriented towards single users and single devices.

We deployed and assessed the affordance of our system in two different types of educational modalities; 1) in school classrooms as an experimentation tool in the presence of a teacher, and 2) as a standalone platform to support a self-paced online biology class similar to MOOCs. The system requirements and the resulting affordance in these two environments are quite distinct. For example, in a teacher mediated school environment, delivering inquiry-based learning is challenging within the short span of a typical class ($\sim$ 50 minutes) due to logistic overheads and limited resources. In addition, there may be serious safety concerns. A biology cloudlab, on the other hand, provides a practical means to deliver this inquiry-based learning efficiently within a limited time span and without any logistic or safety concerns; students may even interact with pathogens. Interestingly, a classroom could even be flipped by assigning real experimentations as homework as cloudlabs are accessible from anywhere anytime. These activities were not even possible in todays classroom. On the other hand, online education, such as MOOCs, currently has no means of running real biology experiments and hence courses are mostly passive video based lectures. Our cloudlab enabled real experimentation at MOOCs scale for the first time, thus turning a passive read-only educational delivery to a read-write one, i.e. students are able to interact with real organisms. However, a key differences between these teacher mediated classroom vs online education is that in the latter students are separated by timezones, which allows us to support vast number of students with only a few backend BPUs. Secondly, the student body in the online education tends to be much more diverse and prone to dropping out and therefore careful design choices are necessary to keep experimentation interfaces simple yet expressive and engaging. In this dissertation, we do not intend to involve ourselves in the unresolved discourse surrounding the efficacy and success of online education. Rather, we provide a technology, for the first time, that enables inquiry-based learning with complex scientific practices within a single platform through the Internet, which could be used in school classrooms and online education, such as MOOCs, alike.

We foresee that education, with more than 15 million high-school students in the USA, and more due to MOOCs, will be an important driving force for the development of the biology cloudlab technology. The curriculum is usually offered repeatedly, allowing the technology to be developed iteratively and tested with a massive number of users, while both education and biology cloudlab benefit greatly from this symbiosis. Furthermore, this cloudlab provides a cost-effective and practical means to implement inquiry-based learning [90] and ultimately accomplish the visions of the Next Generation Science standards [26] and National Research Council [105]. Therefore, we also adopted an iterative design process to develop our cloudlabs by user-testing it in various educational settings before deploying a MOOC at a much larger scale (see Chapter 4).

Learning Analytics, with real experimentations and natural data logging capabilities of any cloud system, now has a unique opportunity to take a peek into how learners explore biological experiments (see Chapter 2) that typically have many interesting natural variabilities. Learning outcomes can be thoroughly investigated, e.g. in the context of bifocal modeling [19], when real experiments are juxtaposed with modeling (see Chapter 3). Toward this, several studies have indicated that the combination of reality (with real variabilities and noise) and modeling (typically clean data) is more beneficial for learning content than either one in isolation [38, 20, 125]. Moreover, there are indications that students typically explore experimentations in novel ways when data is shared with other students. This could be further investigated in a quantifiable manner by implementing data-sharing capabilities in the application layer of our cloudlab.

A biological cloudlab opens up many interesting HCI avenues, e.g. interacting with a real-time system and non-realtime systems are fundamentally different and require carefully designed UI. Some experimentation benefits from visual programming (see Chapter 2) while others may benefit from textual descriptions (see Chapter 3). Biotic games [95, 32, 69] is another interesting application of BPUs which has been shown to foster interest in biology in a playful manner through gamification. So far, most of the Biotic games have been implemented with BPUs operating in isolation, but a cloudlab system is easily foreseeable where games are implemented on the top UI layer. However, ethical concerns arise while handling live organisms in playful manners and researchers have already addressed some of these issues [57], but more will likely arise with different organisms and experimentation modalities. The future direction along this line may open up new modalities of gamification and crowdsourcing of complex exploratory-based experimentation, similar to EteRNA [76] or FoldIt [33], but within the context of real biology.

A key future direction towards enabling scientific research in a centralized location may involve extension of the system architecture (Chapter 6 and 5) that utilizes the *bio-input* and *bio-output* of the BPU to make a cascaded system, i.e. *bio-output* of one BPU is fed as the *bio-input* of another BPU. In such scenarios, we envision multiple homogenous clusters of BPUs, where each cluster handles a specific type of experimentation at scale, while the *bio-outputs* of one cluster is connected to the *bio-inputs* of another cluster. Initially, these connections could be implemented logically by manually shuttling materials, but automation of these processes with explicit connections will be imminent, opening up new areas of operations research.

In conclusion, we foresee that the iterative development and deployment of biology cloudlabs in the educational context, besides benefiting education greatly, will facilitate the development of individual BPU clusters (one experiment type at a time). Perhaps not all experiments can be done this way, but the key is that much of standard biological experimentations can be done much more cost-effectively and without complex logistic concerns. Ultimately, this will usher a future that will truly enable many biological experiments for scientific research purposes to be performed in centralized facilities. We already see the beginning of this trend through companies like Transcriptic [6], Emerald Cloud [1], etc. With that, we look forward to a future that fosters interdisciplinary participation and democratization of biology experimentation through cloudlabs.

Appendix A

Supplementary of Chapter 2, 3 and 4

In this dissertation, we ommit supplementary material that are already included in the original publications as listed below:

- Chapter 2: Hossain et. al [65]
- Chapter 3: Hossain et. al [63]
- Chapter 4: Hossain et. al [62]

Appendix B

BPU Responsiveness and Capacity

B.1 User Study to Find Responsiveness Threshold for Good BPUs

We noted that the responsiveness of a Euglena culture becomes visually obvious when the score is > 0.375 (Table B.1), which conservatively corresponds to ~ 0.4. We ran a crowd-sourced study on Amazon Mechanical Turk with 99 participants from the Internet to arrive at a BPU score threshold of 1.5 at which the responsiveness of the Euglena is reasonably obvious to general users. In this study, each participant evaluated 28 video clips, each of which was 30 secs long, and where the participants had to indicate the direction of Euglena motion. Four video clips with obvious directions were interspersed with attention control questions. Eventually, responses from 76 participants were considered after filtering for participants who reported low video quality due to poor bandwidth or who failed to answer more than 1 attention control questions. As shown in Table B.1, the success of Euglena motion detection by users unfamiliar with the platform is in excess of 80% when the BPU score is above ~ 0.4.

Zoom Level	BPU Score (0-1)	Incorrect	Correct	% Correct
4X	0.0 – 0.125	81	138	63%
	0.125 – 0.25	56	137	71%
	0.25 – 0.375	39	130	77%
	> **0.375**	**45**	**203**	**82**%
10X	0.0 – 0.25	81	138	64.5%
	0.25 – 0.375	56	137	72.1%
	0.375 – 0.5	**39**	**130**	**87.0**%
	0.5 – 0.625	28	188	89.5%
	> 0.625	45	203	89.9%

Table B.1: **Crowd-sourced identification of Euglena responses for different BPU scores (0-1)**

B.2 System Capacity with 6 BPUs

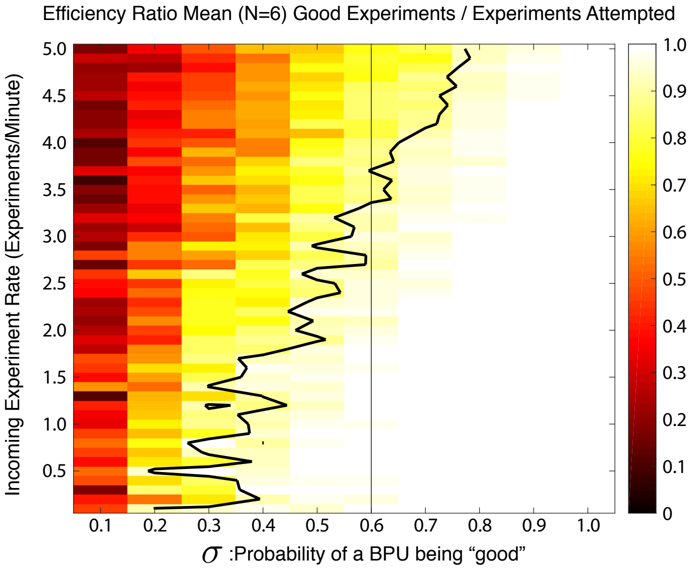


Figure B.1: **Phase space of system efficiency** - number of experiments landing on "good" (Euglena response > 0.4) BPUs / total experiments submitted) - with two parameters: probability of each BPU being "good" vs Incoming experiment rate (per minute). The black contour traces out the 90% efficiency boundary.

The queuing scheme deployed in Chapter 3 and 4 followed an M/D/c model [49], but *live* users would often drop out and BPUs were hardly identical with response qualities evolving over time. Therefore, we ran a simulation with 6 BPUs, modeled with Markov processes of two states ("good" = response level above 0.4 B.1, and "bad") with a steady state probability of $\sigma = 60\%$ of being "good". This 60% was derived from monitoring BPUs over 3 months. Note that with only 6 BPUs the probability of finding at least half of the BPUs good is over 60%. The simulation would attempt to route an incoming experiment to an available "good" BPU first and it was assumed that *live* users would not wait for more than 5 minutes (a parameter that does not, however, alter the final throughput we derive here). Fig B.1 shows the phase-space of system efficiency (number of experiments landing on good BPUs / total number of experiments submitted) from the simulation of 7 days ($N = 6$ trials) with a constant incoming experiment rate (Poisson process) vs the steady probability of a single BPU being good on the horizontal axis. With an acceptable threshold of 90% of the experiments landing on good BPUs, we find that our system with 6 BPUs could deliver a constant throughput of ~3 experiments/minute, where each experiment was 1 minute long. We also accounted for auto-monitoring in our simulation. For $\sigma > 0.5$ the contour trends linearly, and throughput roughly corroborates to a simple linear relationship $\frac{n\sigma}{T}$, where n is the total number of BPUs in the system, and T is the duration of each experiment in minutes. At this rate of 3 experiments/minute, the median wait-time was 15 seconds (IQR:2 minutes).

Bibliography

[1] Emerald Cloud Laboratory. http://emeraldcloudlab.com (last accessed: 19th Dec 2014).

[2] OSHWA: Open Source Hardware Association. http://www.oshwa.org (last accessed: 10th July 2014).

[3] Raspberry Pi. http://www.raspberrypi.org (last accessed: 10th July 2014).

[4] School Enrollment by Sex and Level. Technical Report Statistical Abstract of the United States: 2012 (131st Edition). Table 226, U.S. Census Bureau.

[5] Synthego. http://www.synthego.com (last accessed: 19th Dec 2014).

[6] Transcriptic. https://www.transcriptic.com (last accessed: 19th Dec 2014).

[7] Andrew Adamatzky. *Physarum machines: computers from slime mould*, volume 74. World Scientific, 2010.

[8] Karen Alim, Gabriel Amselem, François Peaudecerf, Michael P Brenner, and Anne Pringle. Random network peristalsis in physarum polycephalum organizes fluid flows across an individual. *Proceedings of the National Academy of Sciences*, 110(33):13306–13311, 2013.

[9] G R Alves, J M Ferreira, D Muler, H Erbe, N Hine, J B M Alves, C E Pereira, L Chiang, O Herrera, and E Sucar. 2005 IEEE Conference on Emerging Technologies and Factory Automation. In *2005 IEEE Conference on Emerging Technologies and Factory Automation*, pages 1023–1030. IEEE, 2005.

[10] Terry Anderson and Julie Shattuck. Design-based research a decade of progress in education research? *Educational researcher*, 41(1):16–25, 2012.

[11] Jeff Anderson-Lee, Eli Fisker, Vineet Kosaraju, Michelle Wu, Justin Kong, Jeehyung Lee, Minjae Lee, Mathew Zada, Adrien Treuille, and Rhiju Das. Principles for predicting rna secondary structure design difficulty. *Journal of molecular biology*, 428(5):748–757, 2016.

[12] Ryan SJD Baker and Kalina Yacef. The state of educational data mining in 2009: A review and future visions. *JEDM-Journal of Educational Data Mining*, 1(1):3–17, 2009.

[13] Frederick K Balagadd'e, Lingchong You, Carl L Hansen, Frances H Arnold, and Stephen R Quake. Long-term monitoring of bacteria undergoing programmed population control in a microchemostat. *Science*, 309(5731):137–140, 2005.

[14] Howard C Berg. Chemotaxis in bacteria. *Annual review of biophysics and bioengineering*, 4(1):119–136, 1975.

[15] Berland, Matthew and Baker, Ryan and Blikstein, Paulo. Learning Analytics in Constructivist, Inquiry-based learning environments. *Technology, Knowledge, and Learning*, 19(1-2):205–220, 2014.

[16] Paulo Blikstein. Connecting the Science Classroom and Tangible Interfaces: the Bifocal Modeling Framework. In *Learning in the Disciplines: Proceedings of the 9th International Conference of the Learning Sciences*, University of Illinois at Chicago, Chicago, IL, 2010.

[17] Paulo Blikstein. Gears of Our Childhood: Constructionist Toolkits, Robotics, and Physical Computing, Past and Future. In *Proceedings of the 12th International Conference on Interaction Design and Children*, IDC '13, pages 173–182, New York, NY, USA, 2013. ACM.

[18] Paulo Blikstein. Multimodal learning analytics. In *Proceedings of the Third International Conference on Learning Analytics and Knowledge*, pages 102–106. ACM, 2013.

[19] Paulo Blikstein. Bifocal Modeling: Comparing Physical and Computational Models Linked in Real Time. In *Playful Learning Interfaces*, pages 317–352, Netherlands, 2014. Springer.

[20] Paulo Blikstein, Tamar Fuhrmann, and Shima Salehi. Using the Bifocal Modeling Framework to Resolve "Discrepant Events" Between Physical Experiments and Virtual Models in Biology. In *Journal of Science Education and Technology*, volume 25, pages 513–526. Springer, 2016.

[21] Paulo Blikstein, Marcelo Worsley, Chris Piech, Mehran Sahami, Steven Cooper, and Daphne Koller. Programming pluralism: Using learning analytics to detect patterns in the learning of computer programming. *Journal of the Learning Sciences*, 23(4):561–599, 2014.

[22] Mads T Bonde, Guido Makransky, Jakob Wandall, Mette V Larsen, Mikkel Morsing, Hanne Jarmer, and Morten O A Sommer. Improving biotech education through gamified laboratory simulations. *Nat Biotech*, 32(7):694–697, July 2014.

[23] Jagadis Chandra Bose. *Comparative electro-physiology: a physico-physiological study*. Longmans, Green, and Co., 1907.

[24] Jagadis Chunder Bose. On the determination of the wave-length of electric radiation by diffraction grating. *Proceedings of the Royal Society of London*, 60(359-367):167–178, 1896.

[25] Engin Bumbacher, Zahid Hossain, Ingmar Riedel-Kruse, and Paulo Blikstein. Where the Rubber Meets the Road: The Impact of the Interface Design on Model Exploration in Science Inquiry. In *Proceedings of the 12th International Conference of Learing Sciences*, volume 2, pages 1277–1278, Singapore, 2016. ISLS.

[26] Rodger W Bybee. The next generation of science standards: Implications for biology education. *The American Biology Teacher*, 74(8):542–549, 2012.

[27] Rodger W Bybee. The next generation science standards and the life sciences. *Science & Children*, 50(6):7–14, 2013.

[28] Marco Casini, Domenico Prattichizzo, and Antonio Vicino. The automatic control telelab: a user-friendly interface for distance learning. *IEEE Transactions on Education*, 46(2):252–257, 2003.

[29] SH Chen, R Chen, V Ramakrishnan, SY Hu, YKCC Zhuang, CC Ko, and Ben M Chen. Development of remote laboratory experimentation through internet. In *Proceedings of the 1999 IEEE Hong Kong Symposium on Robotics and Control*, volume 2, pages 756–760. Hong Kong, 1999.

[30] Michelene TH Chi. Active-constructive-interactive: A conceptual framework for differentiating learning activities. *Topics in Cognitive Science*, 1(1):73–105, 2009.

[31] Clark A Chinn and Betina A Malhotra. Epistemologically authentic inquiry in schools: A theoretical framework for evaluating inquiry tasks. *Science Education*, 86(2):175–218, 2002.

[32] Nate J. Cira, Alice M. Chung, Aleksandra K. Denisin, Stefano Rensi, Gabriel N. Sanchez, Stephen R. Quake, and Ingmar H. Riedel-Kruse. A Biotic Game Design Project for Integrated Life Science and Engineering Education. *PLoS Biol*, 13(3):e1002110, March 2015.

[33] Seth Cooper, Firas Khatib, Adrien Treuille, Janos Barbero, Jeehyung Lee, Michael Beenen, Andrew Leaver-Fay, David Baker, Zoran Popović, and Foldit Players. Predicting protein structures with a multiplayer online game. *Nature*, 466(7307):756–760, August 2010.

[34] Steve Cooper and Mehran Sahami. Reflections on stanford's moocs. *Communications of the ACM*, 56(2):28–30, 2013.

[35] Fernando J. CorbatÃş, Marjorie Merwin-Daggett, and Robert C. Daley. An Experimental Time-sharing System. In *Proceedings of the May 1-3, 1962, Spring Joint Computer Conference*, AIEE-IRE '62 (Spring), pages 335–344, New York, NY, USA, 1962. ACM.

[36] National Research Council. Guide to Implementing the Next Generation Science Standards. Technical report, Committee on Guidance on Implementing the Next Generation Science Standards, Board on Science Education, Division of Behavioral and Social Sciences and Education, Washington, DC: The National Academies Press, 2015.

[37] Thomas M Cover and Joy A Thomas. *Elements of information theory*. John Wiley & Sons, 2012.

[38] T de Jong, M C Linn, and Z C Zacharia. Physical and Virtual Laboratories in Science and Engineering Education. *Science*, 340(6130):305–308, April 2013.

[39] Gabriel Díaz, Félix García Loro, Manuel Castro, Mohamed Tawfik, Elio Sancristobal, and Santiago Monteso. Remote electronics lab within a mooc: Design and preliminary results. In *2013 2nd Experiment@ International Conference (exp. at'13)*, pages 89–93. IEEE, 2013.

[40] Bodo Diehn. Phototaxis and sensory transduction in euglena. *Science*, 181(4104):1009–1015, 1973.

[41] Daniel C Edelson. Design research: What we learn when we engage in design. *The Journal of the Learning sciences*, 11(1):105–121, 2002.

[42] D.T. Emerson. The work of jagadis chandra bose: 100 years of millimeter-wave research. *Microwave Theory and Techniques, IEEE Transactions on*, 45(12):2267–2273, Dec 1997.

[43] S K Esche, C Chassapis, J W Nazalewicz, and D J Hromin. 32nd Annual Frontiers in Education. In *Conference on Frontiers in Education*, pages T2E–1–T2E–6. IEEE, 2002.

[44] Yoav Etsion and Dan Tsafrir. A short survey of commercial cluster batch schedulers. *School of Computer Science and Engineering, The Hebrew University of Jerusalem*, 44221:2005–13, 2005.

[45] Ernesto Fabregas, Gonzalo Farias, Sebastián Dormido-Canto, Sebastián Dormido, and Francisco Esquembre. Developing a remote laboratory for engineering education. *Computers & Education*, 57(2):1686–1697, 2011.

[46] Adrian Fessel, Christina Oettmeier, Erik Bernitt, Nils C Gauthier, and Hans-Günther Döbereiner. Physarum polycephalum percolation as a paradigm for topological phase transitions in transportation networks. *Physical review letters*, 109(7):078103, 2012.

[47] ND Finkelstein, WK Adams, CJ Keller, PB Kohl, KK Perkins, NS Podolefsky, S Reid, and R LeMaster. When learning about the real world is better done virtually: A study of substituting computer simulations for laboratory equipment. *Physical Review Special Topics-Physics Education Research*, 1(1):010103, 2005.

[48] Armando Fox. Cloud computing-what's in it for me as a scientist? *Science*, 331(6016):406–407, 2011.

[49] G. J. Franx. A simple solution for the m/d/c waiting time distribution. *Operations Research Letters*, 29(5):221–229, Dec 2001.

[50] Jon Froehlich, Leah Findlater, Marilyn Ostergren, Solai Ramanathan, Josh Peterson, Inness Wragg, Eric Larson, Fabia Fu, Mazhengmin Bai, Shwetak Patel, et al. The design and evaluation of prototype eco-feedback displays for fixture-level water usage data. In *Proceedings of the SIGCHI conference on human factors in computing systems*, pages 2367–2376. ACM, 2012.

[51] JST Fujita, RF Cassaniga, and FJR Fernandez. Remote laboratory. In *Industrial Electronics, 2003. ISIE'03. 2003 IEEE International Symposium on*, volume 2, pages 1104–1106. IEEE, 2003.

[52] Janice D Gobert, Michael Sao Pedro, Juelaila Raziuddin, and Ryan S Baker. From log files to assessment metrics: Measuring students' science inquiry skills using educational data mining. *Journal of the Learning Sciences*, 22(4):521–563, 2013.

[53] Raymond E Goldstein. Green algae as model organisms for biological fluid dynamics. *Annual review of fluid mechanics*, 47:343, 2015.

[54] L Gomes and S Bogosyan. Current Trends in Remote Laboratories. *IEEE Transactions on Industrial Electronics*, 56(12):4744–4756.

[55] Christophe Gravier, Jacques Fayolle, Bernard Bayard, Mikael Ates, and Jeremy Lardon. State of the art about remote laboratories paradigms-foundations of ongoing mutations. *International Journal of Online Engineering*, 4(1), 2008.

[56] John D. Hansen and Justin Reich. Democratizing education? Examining access and usage patterns in massive open online courses. *Science*, 350(6265):1245–1248, December 2015.

[57] Hayden Harvey, Molly Havard, David Magnus, Mildred K Cho, and Ingmar H Riedel-Kruse. Innocent fun or "microslavery"? *Hastings Center Report*, 44(6):38–46, 2014.

[58] V. J. Harward, J. A. del Alamo, S. R. Lerman, P. H. Bailey, J. Carpenter, K. DeLong, C. Felknor, J. Hardison, B. Harrison, I. Jabbour, P. D. Long, T. Mao, L. Naamani, J. Northridge, M. Schulz, D. Talavera, C. Varadharajan, S. Wang, K. Yehia, R. Zbib, and D. Zych. The iLab Shared Architecture: A Web Services Infrastructure to Build Communities of Internet Accessible Laboratories. *Proceedings of the IEEE*, 96(6):931–950, June 2008.

[59] Erika Check Hayden. The Automated Lab. *Nature*, 516(7529), December 2014.

[60] Ruben Heradio, Luis de la Torre, Daniel Galan, Francisco Javier Cabrerizo, Enrique Herrera-Viedma, and Sebastian Dormido. Virtual and Remote Labs in Education: a Bibliometric Analysis. *Computers & Education*, 98:14 – 38, 2016.

[61] Lambertus Hesselink, Dharmarus Rizal, and Eric Bjornson. Cyberlab: Remote access to laboratories through the world-wide-web. *Retrieved September*, 26(2005):27–39, 2000.

[62] Zahid Hossain, Engin W. Bumbacher, Alison Brauneis, Monica Diaz, Andy Saltarelli, Paulo Blikstein, and Ingmar H. Riedel-Kruse. Biology Remote Lab Enables Authentic Inquiry-Based Learning at MOOC Scale. In *[Submitted]*, CHI '16. ACM, 2016.

[63] Zahid Hossain, Engin W. Bumbacher, Alice M. Chung, Honesty Kim, Casey Litton, Ashley D. Walter, Walter, Kemi Jona, Paulo Blikstein, and Ingmar H. Riedel-Kruse. Interactive and Scalable Biology Cloud Experimentation for Scientific Inquiry and Education. *Nature Biotechnology*, 2016. [In Press].

[64] Zahid Hossain, Alice M Chung, and Ingmar H Riedel-Kruse. Real-time and turn-based biology online experimentation. In *Remote Engineering and Virtual Instrumentation (REV), 2015 12th International Conference on*, pages 86–89. IEEE, 2015.

[65] Zahid Hossain, Xiaofan Jin, Engin W. Bumbacher, Alice M. Chung, Stephen Koo, Jordan D. Shapiro, Cynthia Y. Truong, Sean Choi, Nathan D. Orloff, Paulo Blikstein, and Ingmar H. Riedel-Kruse. Interactive Cloud Experimentation for Biology: An Online Education Case Study. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*, CHI '15, pages 3681–3690, New York, NY, USA, 2015. ACM.

[66] Clara M Ionescu, Ernesto Fabregas, Stefana M Cristescu, Sebastin Dormido, and Robin De Keyser. A remote laboratory as an innovative educational tool for practicing control engineering concepts. *IEEE Transactions on Education*, 56(4):436–442, 2013.

[67] Mineo Iseki, Shigeru Matsunaga, Akio Murakami, Kaoru Ohno, Kiyoshi Shiga, Kazuichi Yoshida, Michizo Sugai, Tetsuo Takahashi, Terumitsu Hori, and Masakatsu Watanabe. A blue-light-activated adenylyl cyclase mediates photoavoidance in Euglena gracilis. *Nature*, 415(6875):1047–1051, February 2002.

[68] Firas Khatib, Seth Cooper, Michael D Tyka, Kefan Xu, Ilya Makedon, Zoran PopoviÄĞ, David Baker, and Foldit Players. Algorithm discovery by protein folding game players. *Proceedings of the National Academy of Sciences*, 108(47):18949–18953, 2011.

[69] Honesty Kim, Lukas Cyrill Gerber, Daniel Chiu, Seung Ah Lee, Nate J Cira, Sherwin Yuyang Xia, and Ingmar H Riedel-Kruse. Ludusscope: Accessible interactive smartphone microscopy for life-science education. *PloS one*, 11(10):e0162602, 2016.

[70] Sang-Hoon Kim. Fractal dimensions of a green broccoli and a white cauliflower. *arXiv preprint cond-mat/0411597*, 2004.

[71] David Klahr, Lara M Triona, and Cameron Williams. Hands on what? the relative effectiveness of physical versus virtual materials in an engineering design project by middle school children. *Journal of Research in Science Teaching*, 44(1):183–203, 2007.

[72] Fanwei Kong, Liang Yuan, Yuan F Zheng, and Weidong Chen. Automatic liquid handling for life science a critical review of the current state of the art. *Journal of laboratory automation*, 17(3):169–185, 2012.

[73] Ger Koole and Avishai Mandelbaum. Queueing models of call centers: An introduction. *Annals of Operations Research*, 113(1-4):41–59, 2002.

[74] Juraj Krajčovič, Matej Vesteg, and Steven D Schwartzbach. Euglenoid flagellates: a multifaceted biotechnology platform. *Journal of biotechnology*, 202:135–145, 2015.

[75] Michael Lebert, Peter Richter, and Donat-Peter Häder. Signal perception and transduction of gravitaxis in the flagellate euglena gracilis. *Journal of plant physiology*, 150(6):685–690, 1997.

[76] Jeehyung Lee, Wipapat Kladwang, Minjae Lee, Daniel Cantu, Martin Azizyan, Hanjoo Kim, Alex Limpaecher, Sungroh Yoon, Adrien Treuille, and Rhiju Das. Rna design rules from a massive open laboratory. *Proceedings of the National Academy of Sciences*, page 201313039, 2014.

[77] Seung Ah Lee, Alice M. Chung, Nate Cira, and Ingmar H. Riedel-Kruse. Tangible Interactive Microbiology for Informal Science Education. In *Proceedings of the Ninth International Conference on Tangible, Embedded, and Embodied Interaction*, TEI '15, pages 273–280, New York, NY, USA, 2015. ACM.

[78] Sharona T Levy and Uri Wilensky. Mining students' inquiry actions for understanding of complex systems. *Computers & Education*, 56(3):556–573, 2011.

[79] Byung-Ro Lim. Challenges and issues in designing inquiry on the web. *British Journal of Educational Technology*, 35(5):627–643, 2004.

[80] Robert A Littleford. Culture of Protozoa in the Classroom. *The American Biology Teacher*, 22(9):551–559, 1960.

[81] John W Lloyd. Practical Advantages of Declarative Programming. In *Joint Conference on Declarative Programming, GULP-PRODE*, volume 94, page 94, 1994.

[82] Manuel E Macías and Israel Méndez. elab-remote electronics lab in real time. In *Frontiers In Education Conference-Global Engineering: Knowledge Without Borders, Opportunities Without Passports, 2007. FIE'07. 37th Annual*, pages S3G–12. IEEE, 2007.

[83] Kouichi Morimoto, Atsuhito Takemura, and others. Developing Teaching Materials using LED for a Phototaxis in Euglena. *Journal of Research in Science Education*, 45(3), 2005.

[84] Toshiyuki Nakagaki, Hiroyasu Yamada, and Masahiko Hara. Smart network solutions in an amoeboid organism. *Biophysical chemistry*, 107(1):1–5, 2004.

[85] Zorica Nedic, Jan Machotka, and Andrew Nafalski. *Remote laboratories versus virtual and real laboratories*, volume 1. IEEE, 2003.

[86] Takayuki Niizato, Tomohiro Shirakawa, and Yukio-Pegio Gunji. A Model of Network Formation by *Physarum* Plasmodium: Interplay Between Cell Mobility and Morphogenesis. *Biosystems*, 100(2):108–112, 2010.

[87] Kazunari Ozasa, Jeesoo Lee, Simon Song, Masahiko Hara, and Mizuo Maeda. Gas/liquid sensing via chemotaxis of Euglena cells confined in an isolated micro-aquarium. *Lab on a Chip*, 13(20):4033–4039, 2013.

[88] Kazunari Ozasa, Jeesoo Lee, Simon Song, and Mizuo Maeda. Transient freezing behavior in photophobic responses of Euglena gracilis investigated in a microfluidic device. *Plant and Cell Physiology*, page pcu101, 2014.

[89] Aydogan Ozcan. Mobile phones democratize and cultivate next-generation imaging, diagnostics and measurement tools. *Lab on a Chip*, 14(17):3187–3194, 2014.

[90] Margus Pedaste, Mario Mäeots, Leo A Siiman, Ton De Jong, Siswa AN Van Riesen, Ellen T Kamp, Constantinos C Manoli, Zacharias C Zacharia, and Eleftheria Tsourlidaki. Phases of inquiry-based learning: Definitions and the inquiry cycle. *Educational research review*, 14:47–61, 2015.

[91] Rainer Pepperkok and Jan Ellenberg. High-throughput fluorescence microscopy for systems biology. *Nature Reviews Molecular Cell Biology*, 7(9):690–696, 2006.

[92] Katherine Perkins, Wendy Adams, Michael Dubson, Noah Finkelstein, Sam Reid, Carl Wieman, and Ron LeMaster. Phet: Interactive simulations for teaching and learning physics. *The Physics Teacher*, 44(1):18–23, 2006.

[93] Edward M Purcell. Life at low Reynolds number. *Am. J. Phys*, 45(1):3–11, 1977.

[94] Mitchel Resnick and Brian Silverman. Some Reflections on Designing Construction Kits for Kids. In *Proceedings of the 2005 Conference on Interaction Design and Children*, IDC '05, pages 117–122, New York, NY, USA, 2005. ACM.

[95] Ingmar H Riedel-Kruse, Alice M Chung, Burak Dura, Andrea L Hamilton, and Byung C Lee. Design, engineering and utility of biotic games. *Lab on a Chip*, 11(1):14–22, 2011.

[96] Pawel Romanczuk, Maksym Romensky, Dimitri Scholz, Vladimir Lobaskin, and Lutz Schimansky-Geier. Motion of Euglena gracilis: Active fluctuations and velocity distribution. *The European Physical Journal Special Topics*, 224(7):1215–1229, 2015.

[97] Maksym Romensky, Dimitri Scholz, and Vladimir Lobaskin. Hysteretic dynamics of active particles in a periodic orienting field. *Journal of The Royal Society Interface*, 12(108):20150015, 2015.

[98] Cristóbal Romero and Sebastián Ventura. Educational data mining: a review of the state of the art. *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on*, 40(6):601–618, 2010.

[99] Megan Sauter, David H Uttal, David N Rapp, Michael Downing, and Kemi Jona. Getting real: the authenticity of remote labs and simulations for science learning. *Distance Education*, 34(1):37–47, 2013.

[100] Eileen Scanlon, Stamatina Anastopoulou, Lucinda Kerawalla, and Paul Mulholland. How technology resources can be used to represent personal inquiry and support students' understanding of it across contexts. *Journal of Computer Assisted Learning*, 27(6):516–529, 2011.

[101] Bertrand Schneider, Jenelle Wallace, Paulo Blikstein, and Roy Pea. Preparing for future learning with a tangible user interface: The case of neuroscience. *IEEE Transactions on Learning Technologies*, 6(2):117–129, 2013.

[102] Heidi Schweingruber, Thomas Keller, Helen Quinn, and others. *A Framework for K-12 Science Education:: Practices, Crosscutting Concepts, and Core Ideas*. National Academies Press, 2012.

[103] Sebastian Seung. EyeWire. https://eyewire.org, last accessed: 18th September, 2014.

[104] Samuel K Sia and Matthew P Owens. Share and share alike. *Nat Biotech*, 33(12):1224–1228, December 2015.

[105] Susan R Singer, Margaret L Hilton, Heidi A Schweingruber, et al. *America's lab report: Investigations in high school science*. National Academies Press, 2006.

[106] Ryan A Skilton, Richard A Bourne, Zacharias Amara, Raphael Horvath, Jing Jin, Michael J Scully, Emilia Streng, Samantha LY Tang, Peter A Summers, Jiawei Wang, and others. Remote-controlled experiments with cloud chemistry. *Nature chemistry*, 7(1):1–5, 2015.

[107] Atsushi Tero, Seiji Takagi, Tetsu Saigusa, Kentaro Ito, Dan P. Bebber, Mark D. Fricker, Kenji Yumiki, Ryo Kobayashi, and Toshiyuki Nakagaki. Rules for biologically inspired adaptive network design. *Science*, 327(5964):439–442, 2010.

[108] Atsushi Tero, Kenji Yumiki, Ryo Kobayashi, Tetsu Saigusa, and Toshiyuki Nakagaki. Flow-network adaptation in physarum amoebae. *Theory in Biosciences*, 127(2):89–94, 2008.

[109] The Concord Consortium. CODAP: Common Online Data Analysis Platform, September 2016.

[110] SJ Trietsch, T Hankemeier, and HJ Van der Linden. Lab-on-a-chip technologies for massive parallel data generation in the life sciences: A review. *Chemometrics and Intelligent Laboratory Systems*, 108(1):64–75, 2011.

[111] Lara M Triona and David Klahr. Point and click or grab and heft: Comparing the influence of physical and virtual instructional materials on elementary school students' ability to design experiments. *Cognition and Instruction*, 21(2):149–173, 2003.

[112] Arie Van Deursen, Paul Klint, and Joost Visser. Domain-specific languages: An annotated bibliography. *Sigplan Notices*, 35(6):26–36, 2000.

[113] Arie Van Deursen, Paul Klint, and Joost Visser. Domain-Specific Languages: An Annotated Bibliography. *Sigplan Notices*, 35(6):26–36, 2000.

[114] Wouter van Joolingen, Tom de Jong, and Angelique Dimitrakopoulout. Issues in computer supported inquiry learning in science. *Journal of Computer Assisted Learning*, 23(2):111–119, 2007.

[115] Wouter R Van Joolingen, Ton De Jong, and Angelique Dimitrakopoulou. Issues in computer supported inquiry learning in science. *Journal of Computer Assisted Learning*, 23(2):111–119, 2007.

[116] Bogdan Vasilescu, Vladimir Filkov, and Alexander Serebrenik. Stackoverflow and github: associations between software development and crowdsourced knowledge. In *Social Computing (SocialCom), 2013 International Conference on*, pages 188–195. IEEE, 2013.

[117] Mitchell M Waldrop. Education online: The virtual lab. confronted with the explosive popularity of online learning, researchers are seeking new ways to teach the practical skills of science. *Nature*, 499(7458):268–270, 2013.

[118] Jerry Wellington. America's lab report: Investigations in high school science. *Science Education*, 91(3):514–515, 2007.

[119] George M Whitesides. The origins and the future of microfluidics. *Nature*, 442(7101):368–373, July 2006.

[120] C E Wieman, W K Adams, and K K Perkins. PHYSICS: PhET: Simulations That Enhance Learning. *Science*, 322(5902):682–683, October 2008.

[121] Joseph Jay Williams. Improving learning in moocs with cognitive science. In *AIED 2013 Workshops Proceedings Volume*, page 49, 2013.

[122] KE Wohlfarth-Bottermann. Oscillatory contraction activity in Physarum. *Journal of Experimental Biology*, 1979.

[123] Marisa L Wong and Juan F Medrano. Real-time pcr for mrna quantitation. *Biotechniques*, 39(1):75, 2005.

[124] Ahmed Mohamed Fahmy Yousef, Mohamed Amine Chatti, Ulrik Schroeder, Marold Wosnitza, and Harald Jakobs. The state of MOOCs from 2008 to 2014: A critical analysis and future visions. In *International Conference on Computer Supported Education*, pages 305–327. Springer, 2014.

[125] Zacharias C Zacharia. Comparing and combining real and virtual experimentation: an effort to enhance students' conceptual understanding of electric circuits. *Journal of Computer Assisted Learning*, 23(2):120–132, 2007.

[126] Zacharias C Zacharia, Georgios Olympiou, and Marios Papaevripidou. Effects of experimenting with physical and virtual manipulatives on students' conceptual understanding in heat and temperature. *Journal of Research in Science Teaching*, 45(9):1021–1035, 2008.