Interactive Cloud Experimentation for Biology: An Online Education Case Study

Zahid Hossain [1,3], Xiaofan Jin [3], Engin W. Bumbacher [2], Alice M. Chung [3], Stephen Koo [1]
Jordan D. Shapiro [3], Cynthia Y. Truong [3], Sean Choi [1], Nathan D. Orloff [3]
Paulo Blikstein [2,1], Ingmar H. Riedel-Kruse [3]

[1]Department of Computer Science, [2]School of Education, [3]Department of Bioengineering
Stanford University
{zhossain, ingmar}@stanford.edu

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.

Author Keywords

Remote Experimentation; Cloud Computing; Biology; Automation; Education; Cloud Experimentation; Cloud Lab

ACM Classification Keywords

K.3.1 Computers and Education: Computer Uses in Education; J.3 Computer Application: Life and Medical Sciences; B.4.0 Hardware Input/Output and Data Communications: General

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 [30], or where online courses [19] do not include lab sections [15, 48]. On the other hand, citizen-science and crowd-sourcing projects have demonstrated how non-scientists can make relevant scientific contributions [18, 41], especially when users can design experiments that are centrally executed in batch by a technician [32]. 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. 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 [31, 37, 44, 49] 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 [25, 28, 34], 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, 22, 26, 27]. Few “cloud lab” companies have emerged recently that execute biology experiments in a centralized location [5, 1, 4]. 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 in-

CHI 2015, April 18 - 23 2015, Seoul, Republic of Korea
Copyright 2015 ACM 978-1-4503-3145-6/15/04?$15.00
http://dx.doi.org/10.1145/2702123.2702354

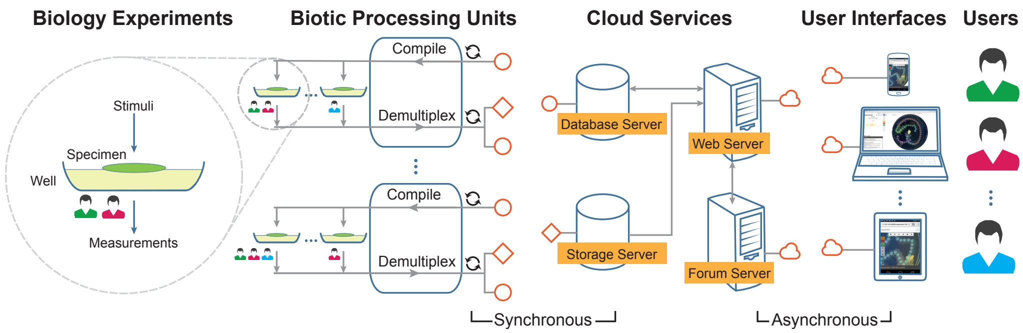

Figure 1. We developed a general architecture for a *cloud experimentation* system that allows multiple users to optimally share high-throughput equipment online to run many interactive biological experiments in parallel and to collaborate. Experiments are carried out in wells and can be shared by multiple 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.

teractive 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. 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.

System Architecture and Biotic Processing Unit (BPU)

Biological investigations are diverse (Fig. 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 [46] 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 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. 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 [28, 34], 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 [33]. 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. 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 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 [16, 21] and biology [17]. 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. This implementation also permitted instructors to perform learning analytics [10, 11, 14]. We selected the chemotactic response of the slime mold *Physarum polycephalum* (Fig. 2, SOM1) as our experimental paradigm. *P. polycephalum* is a single-celled, multi-nuclei, cytoplasmic organism that forms active and dynamic tube networks to search for food [6, 7, 35, 42, 43, 50]. These interesting macroscopic growth and foraging phenotypes represent both multicellular behavior and development [36], inspiring many questions for further investigation in areas ranging from basic biology and biophysics to abstract computation [6].

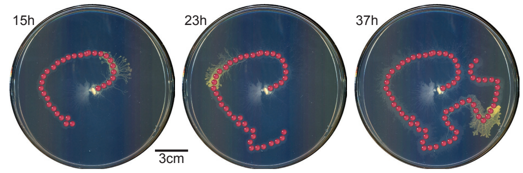

Figure 2. The spatiotemporal chemotactic response of the slime mold *P. polycephalum* (yellow) to an oatmeal solution trail (red) offers a scientifically interesting experimental paradigm with high-dimensional input/output spaces. Food trails of liquid oatmeal pipetted onto an agar surface lead to growth 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. 3.

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. 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. 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. 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. 4A) to connect these BPUs (Fig. 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,

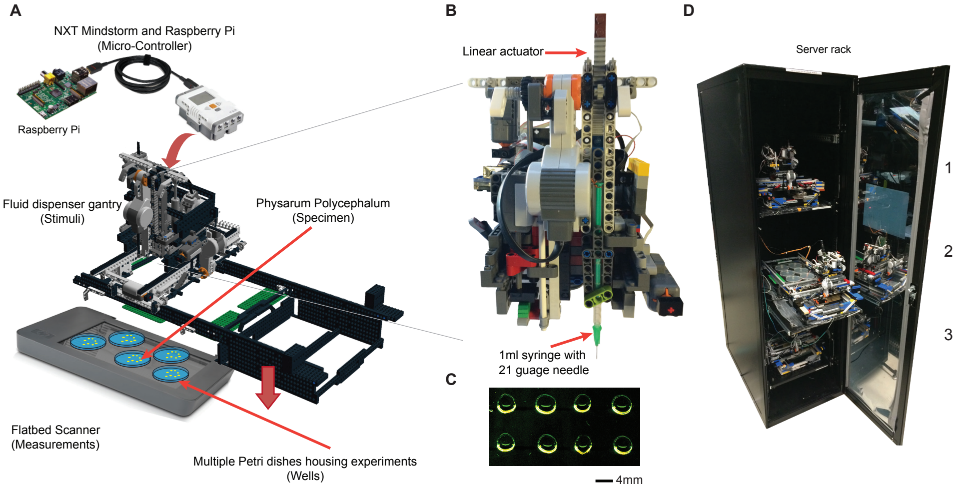

Figure 3. We developed an instance of a BPU for automated fluid handling and imaging to execute versatile biology experiments. (A) A Lego-based gantry for positioning a pipette (stimuli) in the x, y, and z planes is placed atop a flatbed scanner (measurements). Biological materials are housed in standard 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 $10\ \mu$L. (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).

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 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. 4, SOM2-4 for details). After selecting an experiment from the dashboard (Fig. 4A), the user is directed to the web interface (Fig. 4B), which is essentially a time-lapse movie player showing the selected Petri dish. The user can now select various UI elements (Fig. 4C) to play back and investigate the collected time-lapse data interactively (Fig. 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. 4B): a *temporal brush* leads to a trail that grows incrementally over time instead of dispensing all of the liquid at once. Being a web application, our UI runs on most systems, including mobile phones (Fig. 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.

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

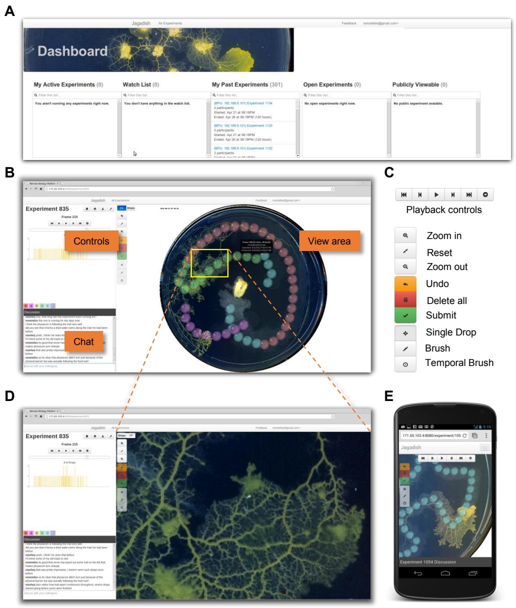

Figure 4. The frontend web-based UI allows the user to program instructions through visual programming, to analyze experimental data remotely, and to collaborate with other users. (A) The dashboard view displays five lists of experiments, enabling easy navigation through all past and currently running 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). (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.

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 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.

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. 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. 5B) [29] . 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. 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. 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. 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 [12, 15,

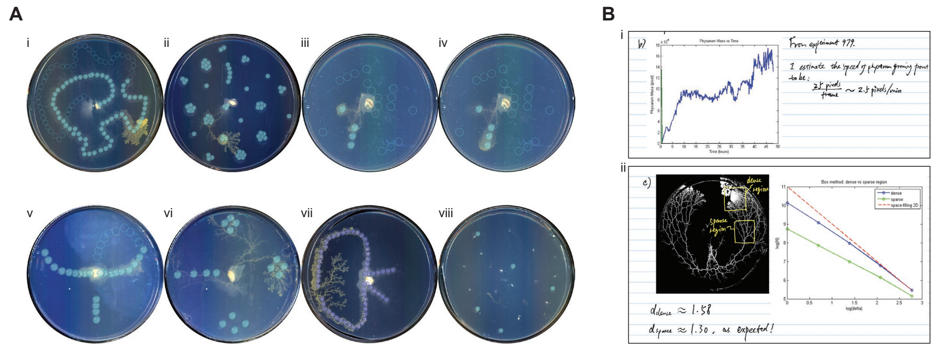

Figure 5. Integrating our *cloud experimentation* system into a lecture-based course on the biophysics of multicellular systems empowered biology experimentation by abstracting away hands-on skills and logistics. (A) Examples of experiments carried out by students throughout the course. The students 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.

24, 30, 38, 45, 52]. 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. 6A), which would not be possible in a virtual environment (e.g., PhET [38]). Ultimately, each instructional medium (conventional lab, virtual lab, online lab) has its own benefits [45], whereas combined or hybrid approaches normally achieve better learning gains [15, 20]. Hence, an exciting future research area is how to achieve optimal synergy between *cloud experimentation* and existing educational media [15].

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. 5Av-viii). During this process, he observed that *P. polycephalum* often does not stop growing even when all food stimuli are depleted (Fig. 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. 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 branching dynamics and morphology of the organism changes over time and with the organism's size (mass) (Fig. 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. 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. 6Ci), which seemingly went against the notion of "self-similarity across scales" in fractals [29] that had been discussed earlier in the course. The students' models iterated from symmetric (Fig. 6Cii) to probabilistic (Fig. 6Ciii) branching models, and eventually developed a dynamic model with several phenomenological rules (Fig. 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. 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. 6D, SOM6) between the model and the experimental data, and obtained quantitative agreement [15].

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 [7, 23], 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 biol-

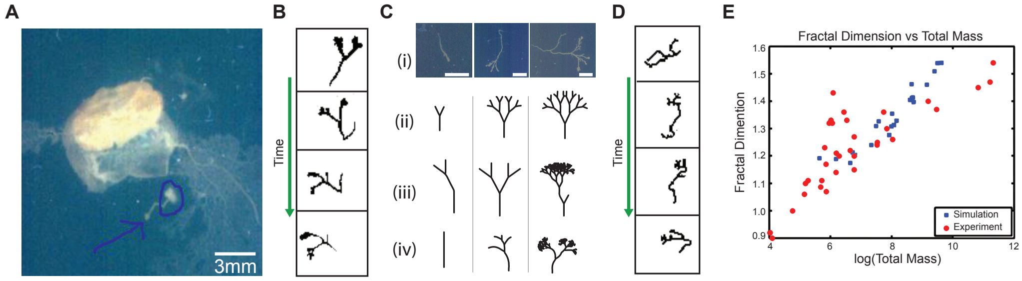

Figure 6. Chance observation by one student and subsequent iterative biophysical model development and experimental testing represent genuine discovery and the scientific method. (A) Email attachment by student sent to the system administrator illustrating the observation that small pieces of *P. polycephalum* 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.

ogy 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 [13].

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.

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 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 [47], 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 [40] come for free with *cloud experimentation* as we were able to track every student's activ-

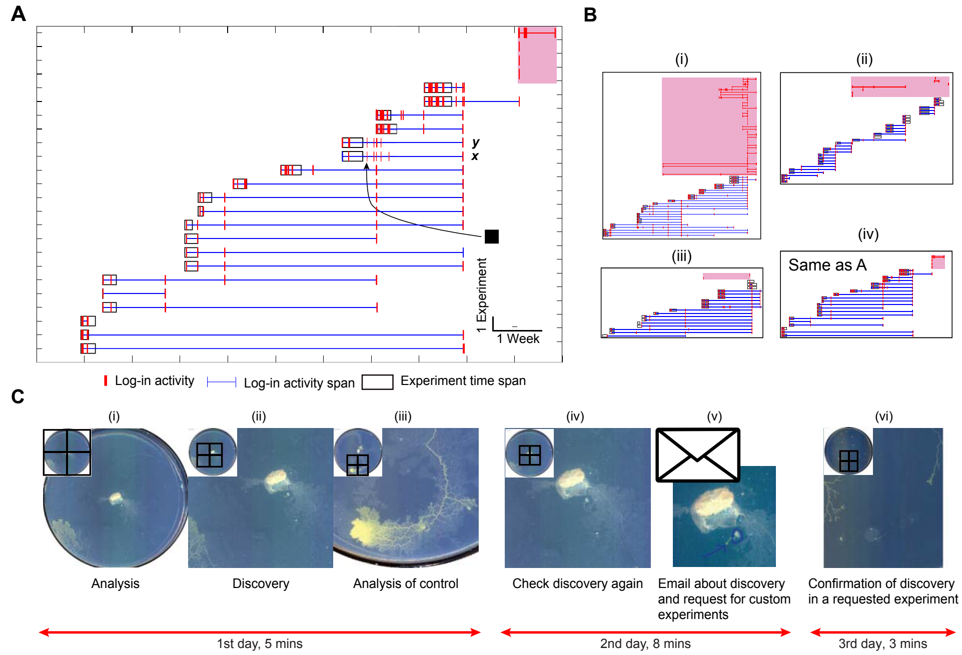

Figure 7. Rich user interaction data can be mined with visualization for student assessment and learning research. (A) The coarsest level of visualization in which the interaction history (experimentation and data analysis) of a single student is shown over the entire course in chronological order. Red 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. 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.

ities in full detail as demonstrated in Figure 7. An instructor could potentially use these activity logs to visualize emerging patterns, for example Figure 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 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 [31] and motorized microscopes [37]), or equipment control is closed during a run (such as real-time PCR machines [51], 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 [39]. 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, 22, 25, 26, 27, 28] 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.

ACKNOWLEDGEMENTS
This work has been supported in part by Stanford BioX IIP, Stanford VPOL grants, Lemann Center for Educational Entrepreneurship and Innovation in Brazil, NSF Cyberlearning (NSF 1324753), NSF Bifocal Modeling (NSF 1055130), SIGF, BioX graduate student fellowship and REU. We like to thank J. Grayson, G.I. Lee and P. Floyd.

Author Contribution
Idea: IRK; Hardware: ZH, IRK, JS; Software: ZH, SK, SC; Study design and evaluation: ZH, IRK, EB, PB; Study execution: ZH, XJ, AC, IRK; Supporting: JS , CT, NO; Writing: ZH, IRK.

REFERENCES
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. Synthego. http://www.synthego.com (last accessed: 19th Dec 2014).
5. Transcriptic. https://www.transcriptic.com (last accessed: 19th Dec 2014).
6. Adamatzky, A. *Physarum machines: computers from slime mould*, vol. 74. World Scientific, 2010.
7. Alim, K., Amselem, G., Peaudecerf, F., Brenner, M. P., and Pringle, A. Random network peristalsis in physarum polycephalum organizes fluid flows across an individual. *Proceedings of the National Academy of Sciences 110*, 33 (2013), 13306–13311.
8. Alparslan, E., and Fuatince, M. Image enhancement by local histogram stretching. *IEEE Transactions on Systems Man and Cybernetics 11* (1981), 376–385.
9. Alves, G. R., Ferreira, J. M., Muler, D., Erbe, H., Hine, N., Alves, J. B. M., Pereira, C. E., Chiang, L., Herrera, O., and Sucar, E. 2005 IEEE Conference on Emerging Technologies and Factory Automation. In *2005 IEEE Conference on Emerging Technologies and Factory Automation*, IEEE (2005), 1023–1030.
10. Baker, R. S., and Yacef, K. The state of educational data mining in 2009: A review and future visions. *JEDM-Journal of Educational Data Mining 1*, 1 (2009), 3–17.
11. Berland, Matthew and Baker, Rachel and Blikstein, Paulo. (in press) Learning Analytics in Constructivist, Inquiry-based learning environments. *Technology, Knowledge, and Learning*.
12. Blikstein, P. Connecting the science classroom and tangible interfaces: the bifocal modeling framework. *Proceedings of the 9th International Conference of the Learning Sciences, Chicago, IL.* (2010).
13. Blikstein, P. 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, ACM (New York, NY, USA, 2013), 173–182.
14. Blikstein, P. Multimodal learning analytics. In *Proceedings of the Third International Conference on Learning Analytics and Knowledge*, ACM (2013), 102–106.
15. Blikstein, Paulo. (in press) Bifocal Modeling: Comparing Physical and Computational Models Linked in Real Time. *Playful Learning Interfaces*, 317–352.
16. Bose, J. C. On the determination of the wave-length of electric radiation by diffraction grating. *Proceedings of the Royal Society of London 60*, 359-367 (1896), 167–178.
17. Bose, J. C. *Comparative electro-physiology: a physico-physiological study*. Longmans, Green, and Co., 1907.
18. Cooper, S., Khatib, F., Treuille, A., Barbero, J., Lee, J., Beenen, M., Leaver-Fay, A., Baker, D., Popović, Z., and Players, F. Predicting protein structures with a multiplayer online game. *Nature 466*, 7307 (Aug. 2010), 756–760.
19. Cooper, S., and Sahami, M. Reflections on stanford's moocs. *Communications of the ACM 56*, 2 (2013), 28–30.
20. de Jong, T., Linn, M. C., and Zacharia, Z. C. Physical and Virtual Laboratories in Science and Engineering Education. *Science 340*, 6130 (Apr. 2013), 305–308.
21. Emerson, D. The work of jagadis chandra bose: 100 years of millimeter-wave research. *Microwave Theory and Techniques, IEEE Transactions on 45*, 12 (Dec 1997), 2267–2273.
22. Esche, S. K., Chassapis, C., Nazalewicz, J. W., and Hromin, D. J. 32nd Annual Frontiers in Education. In *Conference on Frontiers in Education*, IEEE (2002), T2E–1–T2E–6.
23. Fessel, A., Oettmeier, C., Bernitt, E., Gauthier, N. C., and Döbereiner, H.-G. Physarum polycephalum percolation as a paradigm for topological phase transitions in transportation networks. *Physical review letters 109*, 7 (2012), 078103.

24. Finkelstein, N., Adams, W., Keller, C., Kohl, P., Perkins, K., Podolefsky, N., Reid, S., and LeMaster, R. 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 (2005), 010103.
25. Fujita, J., Cassaniga, R., and Fernandez, F. Remote laboratory. In *Industrial Electronics, 2003. ISIE'03. 2003 IEEE International Symposium on*, vol. 2, IEEE (2003), 1104–1106.
26. Gomes, L., and Bogosyan, S. Current Trends in Remote Laboratories. *IEEE Transactions on Industrial Electronics 56*, 12, 4744–4756.
27. Gravier, C., Fayolle, J., Bayard, B., Ates, M., and Lardon, J. State of the art about remote laboratories paradigms-foundations of ongoing mutations. *International Journal of Online Engineering 4*, 1 (2008).
28. Hesselink, L., Rizal, D., and Bjornson, E. Cyberlab: Remote access to laboratories through the world-wide-web. *Retrieved September 26*, 2005 (2000), 27–39.
29. Kim, S.-H. Fractal dimensions of a green broccoli and a white cauliflower. *arXiv preprint cond-mat/0411597* (2004).
30. Klahr, D., Triona, L. M., and Williams, C. 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 (2007), 183–203.
31. Kong, F., Yuan, L., Zheng, Y. F., and Chen, W. Automatic liquid handling for life science a critical review of the current state of the art. *Journal of laboratory automation 17*, 3 (2012), 169–185.
32. Lee, J., Kladwang, W., Lee, M., Cantu, D., Azizyan, M., Kim, H., Limpaecher, A., Yoon, S., Treuille, A., and Das, R. Rna design rules from a massive open laboratory. *Proceedings of the National Academy of Sciences* (2014), 201313039.
33. Lloyd, J. W. Practical Advantages of Declarative Programming. In *Joint Conference on Declarative Programming, GULP-PRODE*, vol. 94 (1994), 94.
34. Macías, M. E., and Méndez, I. 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*, IEEE (2007), S3G–12.
35. Nakagaki, T., Yamada, H., and Hara, M. Smart network solutions in an amoeboid organism. *Biophysical chemistry 107*, 1 (2004), 1–5.
36. Niizato, T., Shirakawa, T., and Gunji, Y.-P. A Model of Network Formation by *Physarum* Plasmodium: Interplay Between Cell Mobility and Morphogenesis. *Biosystems 100*, 2 (2010), 108–112.
37. Pepperkok, R., and Ellenberg, J. High-throughput fluorescence microscopy for systems biology. *Nature Reviews Molecular Cell Biology 7*, 9 (2006), 690–696.
38. Perkins, K., Adams, W., Dubson, M., Finkelstein, N., Reid, S., Wieman, C., and LeMaster, R. Phet: Interactive simulations for teaching and learning physics. *The Physics Teacher 44*, 1 (2006), 18–23.
39. Riedel-Kruse, I. H., Chung, A. M., Dura, B., Hamilton, A. L., and Lee, B. C. Design, engineering and utility of biotic games. *Lab on a Chip 11*, 1 (2011), 14–22.
40. Romero, C., and Ventura, S. 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 (2010), 601–618.
41. Seung, S. EyeWire. https://eyewire.org, last accessed: 18th September, 2014.
42. Tero, A., Takagi, S., Saigusa, T., Ito, K., Bebber, D. P., Fricker, M. D., Yumiki, K., Kobayashi, R., and Nakagaki, T. Rules for biologically inspired adaptive network design. *Science 327*, 5964 (2010), 439–442.
43. Tero, A., Yumiki, K., Kobayashi, R., Saigusa, T., and Nakagaki, T. Flow-network adaptation in physarum amoebae. *Theory in Biosciences 127*, 2 (2008), 89–94.
44. Trietsch, S., Hankemeier, T., and Van der Linden, H. Lab-on-a-chip technologies for massive parallel data generation in the life sciences: A review. *Chemometrics and Intelligent Laboratory Systems 108*, 1 (2011), 64–75.
45. Triona, L. M., and Klahr, D. 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 (2003), 149–173.
46. Van Deursen, A., Klint, P., and Visser, J. Domain-specific languages: An annotated bibliography. *Sigplan Notices 35*, 6 (2000), 26–36.
47. Vasilescu, B., Filkov, V., and Serebrenik, A. Stackoverflow and github: associations between software development and crowdsourced knowledge. In *Social Computing (SocialCom), 2013 International Conference on*, IEEE (2013), 188–195.
48. Waldrop, M. M. 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 (2013), 268–270.
49. Whitesides, G. M. The origins and the future of microfluidics. *Nature 442*, 7101 (July 2006), 368–373.
50. Wohlfarth-Bottermann, K. Oscillatory contraction activity in Physarum. *Journal of Experimental Biology* (1979).
51. Wong, M. L., and Medrano, J. F. Real-time pcr for mrna quantitation. *Biotechniques 39*, 1 (2005), 75.
52. Zacharia, Z. C., Olympiou, G., and Papaevripidou, M. Effects of experimenting with physical and virtual manipulatives on students' conceptual understanding in heat and temperature. *Journal of Research in Science Teaching 45*, 9 (2008), 1021–1035.