

New Capabilities of EvoBot: A Modular, Open-Source Liquid-Handling Robot

Farzad Nejatimoharrami¹, Andres Faina¹, and Kasper Stoy¹

Abstract

We introduce a robot developed to perform feedback-based experiments, such as droplet experiments, a common type of experiments in artificial chemical life research. These experiments are particularly well suited for automation because they often stretch over long periods of time, possibly hours, and often require that the human takes action in response to observed events such as changes in droplet size, count, shape, or clustering or declustering of multiple droplets.

Our robot is designed to monitor long-term experiments and, based on the feedback from the experiment, interact with it. The combination of precise automation, accurately collected experiment data, and integrated analysis and modeling software makes real-time interaction with the experiment feasible, as opposed to traditional offline processing of experiments.

Last but not least, we believe the low cost of our platform can promote artificial life research. Furthermore, prevalently, findings from an experiment will inspire redesign for novel experiments. In addition, the robot's open-source software enables easy modification of experiments.

We will cover two case studies for application of our robot in feedback-based experiments and demonstrate how our robot can not only automate these experiments, collect data, and interact with the experiments intelligently but also enable chemists to perform formerly infeasible experiments.

Keywords

Robotics and instrumentation, automated biology, systems engineering, programming, informatics and software

Introduction

We introduce a liquid-handling robot, called EvoBot, which focuses on research requiring low-throughput but long-lasting liquid-handling experiments. In addition, EvoBot enables easy modification of experiments based on the results of previous experiments. The motivation for developing EvoBot is because most of the robots in the market have been designed to maximize accuracy and throughput for short-duration experiments.

In spite of the emergence of affordable, commercial, low-throughput liquid-handling platforms in recent years, they are not extendible by the user and are not capable of performing reactive experiments. Andrew¹ is one such successful example with a semi-affordable price. OpenTrones² is another successful example that makes the price affordable even for small labs, through open-source software. Although these platforms can be versatile (e.g., Andrew has dominos of different kinds accommodating various consumables that can be configured as needed), they are closed and not extendible with new modules (e.g., a pump module). Furthermore, there is little sensing functionality involved to provide feedback for reactive experiments.

The need to perform reactive low-throughput liquid-handling experiments has resulted in the development of

custom platforms for specific tasks. DropBot³ is an example of robots developed to perform artificial life experiments in a chemistry laboratory. However, these platforms are not open source, and because they are not modular, they are not easy to extend or versatile enough to be used for different experiments.

We have developed a liquid-handling robot that is modular, open source, and affordable. EvoBot's modularity makes it reconfigurable, and it is extendible for reuse in diverse experiments. EvoBot's open-source software enables easy modification of experiments as well as easy software development for potential future modules. The production cost of EvoBot (excluding assembly) does not exceed \$1500, making it affordable even for small-scale labs. EvoBot has been used by

¹Robots, Evolution, and Art Lab (REAL), IT University of Copenhagen, Copenhagen, Denmark

Corresponding Author:

Farzad Nejatimoharrami, IT University of Copenhagen, Rued Langgaards Vej 7, DK-2300 Copenhagen S, Denmark.
Email: fnej@itu.dk

partner universities in the EVOBLISS EU project to perform diverse liquid-handling experiments.

We have integrated computer vision in EvoBot's design to enable feedback-based experiments. A key application area of EvoBot is in artificial chemical life, where the behavior of motile droplets in a reaction vessel such as a petri dish is of interest. Computer vision enables EvoBot to detect relevant changes in either individual droplet behavior (e.g., change in droplet area, position, speed, direction, acceleration, color, shape, and number of droplets) or group droplet behavior (e.g., droplets clustering or declustering). Having detected the specified behavior change, precise automation enables EvoBot to interact with the experiment (e.g., dispense or aspirate a chemical at a specific point relative to a droplet center).

The user interface provided with EvoBot provides a real-time video from the experiment, therefore enabling users to interact with the experiment while it is happening. Therefore, if the users observe an interesting behavior, they don't need to wait until the end of the experiment. They can modify the remaining steps of the experiment on the fly.

EvoBot can record accurate experiment data online. The possibility of providing real-time experiment data, in contrast to the offline data processing at the end of the experiments; the ability to place new droplets a precise distance from any droplet center; and the ability to track fast-moving droplets open doors to new possibilities in laboratory research. EvoBot's data logger provides various kinds of data about the experiment with 0.1 mm position accuracy and 4% droplet area accuracy at millisecond time steps. This enables building precise models for chemical experiments, and accurate verification of hypotheses. It should be noted that data from other types of sensors such as voltage or pH sensors also can be used readily to provide feedback for a specific experiment if required.

We will focus on the new capabilities of EvoBot. We have already demonstrated that EvoBot can perform feedback-based experiments by nurturing microbial fuel cells,⁴ performing single moving droplet experiments,⁵ and improving the quality of artificial life experiments.⁶ This article focuses on new capabilities, such as interaction with moving droplets and clusters of droplets. Also, we demonstrate 3D placement of droplets.

Experimental Section

We commence this section with a brief physical description of the robot. Then, we will look into the software architecture of the robot, including firmware, robot control, and vision application programming interface (API).

Physical Robot Description

EvoBot consists of an actuation layer on top, an experimental layer in the middle, and a sensing layer at the bottom.

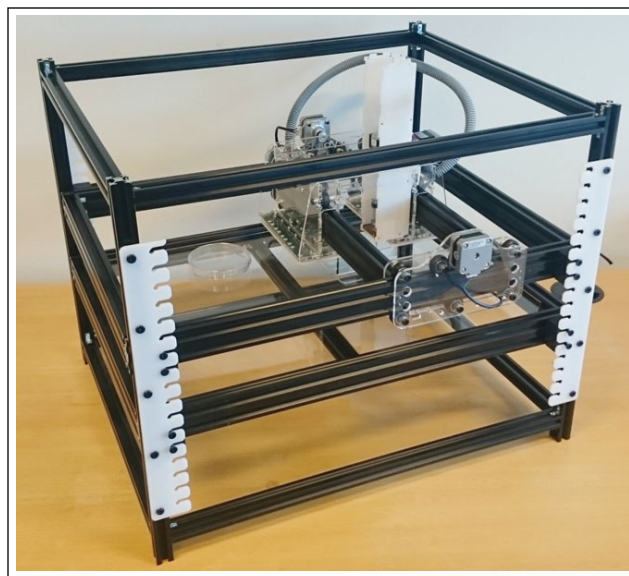


Figure 1. Overview of EvoBot.

EvoBot's design is based on open-source 3D printers, and we use an Arduino board and Ramps shield to control the electronics. **Figure 1** shows an overview of EvoBot, and **Figure 2** shows an overview of the actuation and experimental layers, as well as the head, syringe modules, and the camera.

The actuation layer comprises the robot head and modules mounted on it. EvoBot's modularity allows for support of modules of different kinds for various applications. The experiment-dependent modules could entail syringe modules for liquid dispensing or aspirating, grippers to move the containers over the experimental layer or dispose dirty containers, an OCT scanner module to perform OCT scans, an extruder module to 3D print, and other potential experiment-specific tools. Configuring EvoBot for different experiments is easy because different types of modules can be easily removed or plugged at the appropriate position. The head is responsible for moving the modules in the x - y plane, and the modules have motors to move vertically. Experiments performed with EvoBot determined the head precision to be ± 0.1 mm. The maximum speed of the robot satisfying this precision is 180 mm/s^3 .

The experimental layer consists of a transparent poly-methyl methacrylate (PMMA) sheet on which reaction vessels are positioned. Various types of reaction vessels can be placed on the experimental layer, such as Petri dishes, well plates, beakers, volumetric flasks, graduated cylinders, and Erlenmeyer flasks. The actuation layer interacts with the experimental layer by filling or emptying a specific volume to or from a syringe, washing a syringe, and disposing dirty containers while avoiding obstacles.

The sensing layer consists of a camera below the experimental layer to monitor the experiment, or required sensors

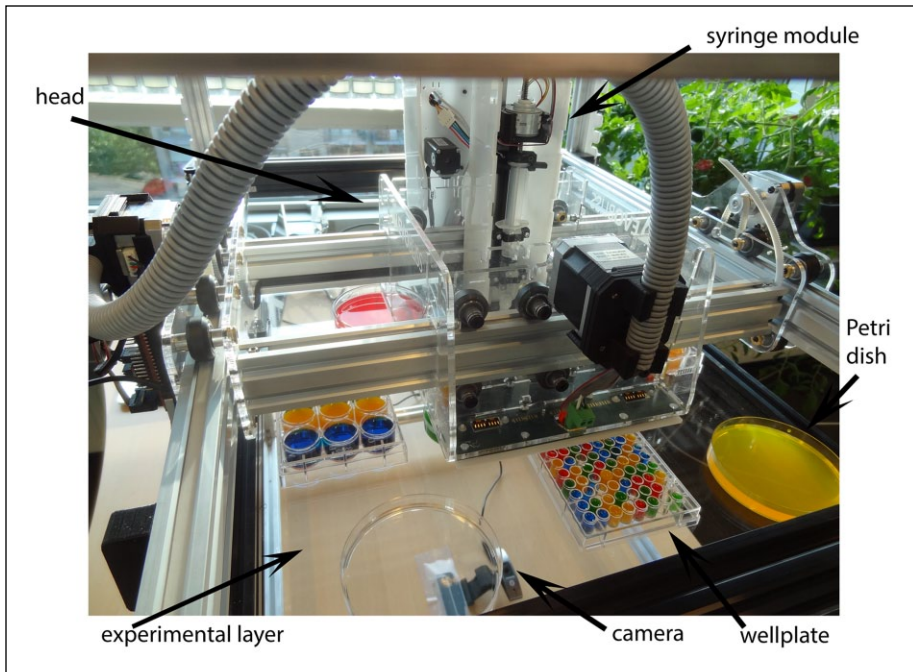


Figure 2. Overview of the actuation, experimental, and sensing layers. EvoBot’s actuation layer consists of a moving head on which various modules, such as syringe modules, can be mounted. The experimental layer accommodates different vessels, and the camera at the bottom acts as the sensing layer by collecting experiment data.

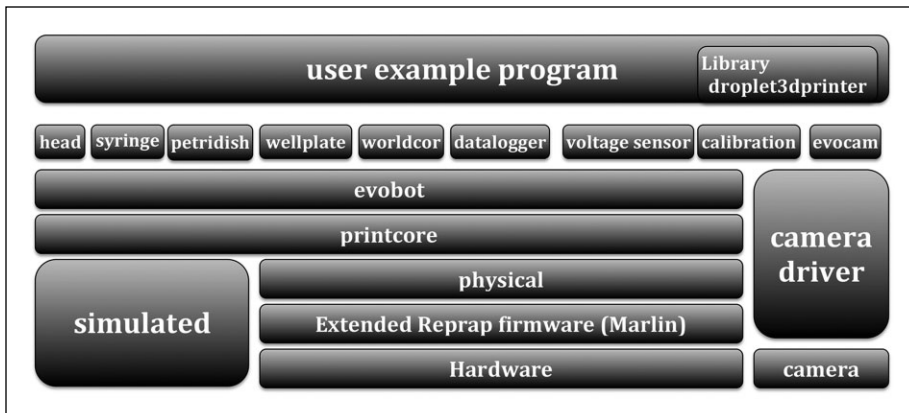


Figure 3. Software architecture of EvoBot. The API developed for EvoBot provides the basis for performing a large class of experiments.

depending on the experiment. The sensing layer collects data from the experiment and provides feedback for the robot to interact with the experiment.

Software Architecture

The software for EvoBot has been developed in Python and has been tested on OS X, Linux, and Windows. **Figure 3** shows the software architecture of EvoBot. At the top layer, the user programs and runs the experiment. Depending on the command, an API module like the syringe module will be responsible for sending the corresponding G-code (or G programming language used in computer-aided manufacturing to control automated machine tools) to the next layer down. The syringe module, Petri dish, well plate, and head

have methods for their own behavior or their interaction with other modules. The bottom layer, EvoBot, represents the robot as a whole. It is mainly used to initialize the robot, and turn it on or off. One layer down, the open-source library “printcore” is responsible for checking errors. The physical layer is responsible for configuring a serial connection on a specific port and baud rate, and for reading to and writing from the serial port. In contrast, the simulation module enables the user to verify his or her program in simulation mode prior to running it on the robot. The API developed for EvoBot provides the basis for performing a large class of experiments.

EvoBot uses an extended version of Marlin firmware to add support for controlling syringe modules. The robot’s firmware resides on an Arduino Mega, and it is the link between

software and hardware, interpreting commands from the G-code file and controlling the motion accordingly.

EvoBot's vision API is responsible for processing the camera frames from the experimental layer to extract data about the experiment (e.g., droplet behaviors). We use OpenCV3, NumPy, Matplotlib, and Webcolors libraries for image data analysis. To recognize droplets in artificial chemical life experiments and analyze their behavior, we use a series of image-processing operations, including different filters. We smooth the image to remove noise, and we convert the color space of camera frames from BGR to HSV for better accuracy and robustness to lighting changes. We then threshold the HSV image to extract only the droplets. We apply morphological operators, namely closing (i.e., dilation of image frames followed by erosion) and then opening (i.e., erosion of the resulting frame followed by a dilation), to remove noise and recognize droplets more accurately. The next step is to find the contours in the binary image to detect droplets. Then, the moments are calculated to determine the center of droplets. The center coordinate is used to calculate other droplet information (e.g., speed, direction, or acceleration). We use Hough transform to detect Petri dishes.

To extract relevant experiment data, EvoBot's vision API needs to calibrate the coordinate system of the robot and the camera because they are in millimeters and pixels, respectively. To this end, we use affine transformation, which represents a relation between two images and can be used to express rotations, translations, and scale operations. We need five coefficients to calculate the affine transformation matrix. This is done by asking the user to click on the needle tip three times to provide six equations for calculating the affine transform coefficients. This step is done only once, when the robot is set up.

The world coordinate module transforms coordinate systems of different modules to each other, enabling moving different modules to the same location. Syringe modules can be placed at 17 different positions on EvoBot's head, and up to 11 syringes can be mounted on EvoBot simultaneously. An object, such as a Petri dish or a well plate, can be defined relative to any of the 17 coordinate systems, and its coordinates will be calculated automatically relative to other syringes, enabling precise access to the same object by any desired syringe module. Even if using multiple syringes, only one of the syringes needs to be calibrated.

We have equipped EvoBot with a data logger that is able to log different types of experiment data, including head x and y position; head speed and acceleration; syringe x, y, and z position; and plunger vertical position throughout time. The data logger can also log events, such as aspirating or dispensing with a syringe, washing a syringe, and various experiment steps. The user can save the log as either a ".dat" or ".csv" extension. The Excel-compatible .csv format enables users to analyze these data easily, and it provides compatible data for advanced data-mining applications with

software such as Weka or python-based Orange. Also, sophisticated plots of real-time experiment results can be depicted using the Matplotlib library. In addition, EvoBot's vision API enables users to record the video of experiments when their desired events occur.

EvoBot comes with a default configuration file, and each user has a local configuration on top of that, enabling users to use the same software independent of the specific parameters of the robot they use. We use this configuration structure because we have built seven copies of EvoBot with different dimensions and modules for different users, according to their needs. The software detects the type, serial port, and slot number for all modules mounted on the head. In addition, the users use different versions, sizes, and numbers of syringe modules based on the types of experiments they perform. Furthermore, EvoBot may be connected to different serial ports on different computers. The camera is also assigned different IDs when connected to different ports or computers.

Furthermore, the EvoBot software provides an interactive graphical user interface, which enables chemists to interact with the experiment as shown in **Figure 4**. The users are provided with a live video of the experiment, showing real-time data about the experiment including droplet data, such as size or speed, and time elapsed since dispense. Based on these data, the chemist can interact with the experiment. The user will select one of the syringes, specify the volume to either aspirate or dispense liquid, and click on the point where he or she wants to perform the interaction. Data about the interaction (e.g., time elapsed from interaction) also will be displayed.

Results and Discussion

In this section, we demonstrate the general nonreactive liquid-handling functionality of the robot by performing routine liquid-handling experiments with the robot and providing examples of precise droplet placement. Thereafter, the reactive functionality of the robot is investigated. EvoBot has been used for performing numerous reactive liquid-handling experiments. We will include two use cases, because droplet behaviors in these experiments and the required type of robot interaction are comprehensive enough to be generalized to various reactive experiments. The use cases are examples of what is possible with EvoBot, and they are functionalities requested by our partners in the EVOBLISS EU project to enable them not to perform a specific experiment but a class of experiments. We compare the way nonautomated experiments were performed without the robot to demonstrate the ease, accuracy, and precision of results obtained by the robot. Furthermore, easy configurability of the robot for "needed to be modified" experiments is examined. We will discuss how EvoBot has realized experiments that formerly were difficult to perform.

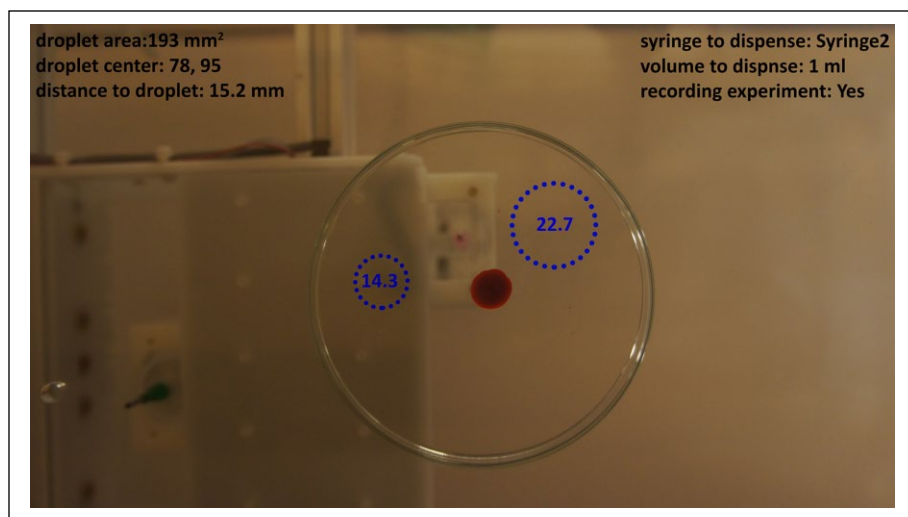


Figure 4. EvoBot's interactive graphical user interface. The user interface displays experiment information live and enables users to interact with the experiment in real time. The dotted circles show the estimated propagation of injected salt droplets. The elapsed time since dispensing is displayed in the center of the circle.

General Liquid-Handling Experiments

EvoBot can perform general liquid-handling experiments because it can perform 1-N, N-1, and N-N experiments; mix the contents of a reaction vessel; and wash reaction vessels. 1-N refers to experiments like aspirating liquid from one chemical vessel and dispensing in different chemical vessels (e.g., filling all wells in a well plate with a specific liquid from a Petri dish). N-1 refers to experiments like emptying wells of a well plate in a garbage collector. N-N refers to experiments like transferring liquid from specific wells in a well plate to the corresponding wells in another well plate. The robot is intelligent enough to fill a syringe with the same liquid if that syringe runs out of liquid. Due to these functionalities, EvoBot is able to perform serial dilutions, enzyme-linked immunosorbent assay (ELISA), and PCR.

Droplet Placement

EvoBot can be used to place droplets precisely in two or three dimensions. The droplets can be placed in specific patterns such as circles or rectangles, or in sophisticated geometric shapes. Droplet pattern printing in 3D is a novel opportunity because it is difficult to achieve the necessary precision in the vertical direction if performed manually. **Figure 5** shows droplets placed in 3D patterns. This was obtained by injecting salt solution into silicone. The salt density was calculated in a way so the injected droplets would float in the silicone. Different droplet volumes at different positions and heights were injected to form a 3D pattern. Then, the silicone was cured in an oven for 10 min.

Use Case I: Sensor Input Feedback for Aspirating Droplet

In this experiment, a moving droplet is aspirated by the syringe module when the droplet speed goes lower than a specified threshold. To do this, the vision API of EvoBot tracks the droplets and analyzes droplet speed. This experiment is performed by first adding a 40 μL droplet of a 1 M NaCl solution (i.e., 10 μmol of NaCl) at 10 mm distance from the edge of a 90 mm diameter Petri dish containing 3 mL of 10 mM sodium decanoate (pH 11), and then dispensing a 20 μL decanol droplet over the sodium decanoate, as can be seen in **Supplementary Video 1**. The droplet moves toward the salt gradient. When the speed goes lower than a threshold, the syringe tip aspirates the droplet. We repeated the experiment 10 times with the same droplet. **Figure 6** shows the droplet speed throughout time obtained from EvoBot's vision API.

Analysis of the speed of a droplet composed of different chemicals is of interest to chemists. EvoBot makes interacting with the experiment based on droplet speed possible. This is useful because the composition of an aspirated droplet with certain speed can be analyzed with an external instrument. A similar application is injecting a chemical in a moving droplet when the speed reduces.⁷ It is of interest for artificial life scientists to refuel a moving droplet, when it stops motion or when the speed is lower than a certain threshold, to make the droplet continue movement. Furthermore, collecting data such as droplet position, speed, acceleration, and size; analyzing these data; and potentially modeling droplet behavior are crucial for chemists.

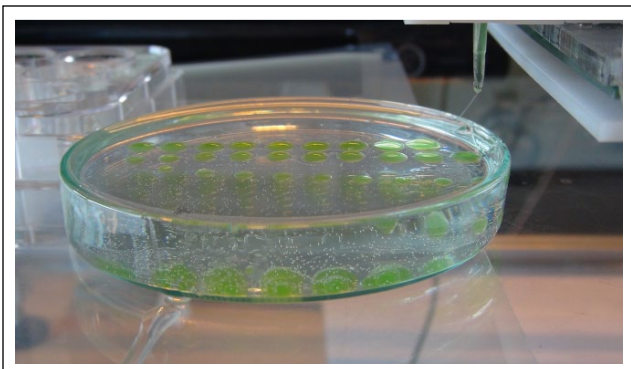


Figure 5. Droplet placement in 3D patterns. The 3D pattern was obtained by injecting salt solution into silicone, and then curing silicone in an oven.

Use Case 2: Sensor Input Feedback Affecting Group Droplet Behavior (Clustering Experiment)

In this experiment, multiple decanol droplets are added to a decanoate solution at different points. After a certain time, these droplets come together and form a cluster. Now, if a droplet of sodium chloride is added at a specific distance from the cluster, the experiment will be reversed, meaning the decanol droplets will decluster. However, this step takes a long time; depending on the set of parameters, it can be even as long as a day. Subject to the parameters of the experiment, such as the number of decanol droplets, how distant and in what pattern they are placed, the pH of decanoate solution, and room temperature, decanol droplets cluster at different times and patterns. In addition to these parameters, the location and time of salt injection affect the number of clusters during the course of the time and the total time needed to decluster completely.

The need to perform different long-running experiments with various parameters makes EvoBot ideal for these experiments. EvoBot's vision API enables it to detect clustering and declustering and count the number of clusters during the course of the experiment. Whereas humans need to waste time monitoring the experiment for declustering to happen, EvoBot can perform many of the experiments in parallel at the same time without the need for the presence of a human. In addition, humans are not as accurate as the robot at detecting the clustering, and not precise enough to inject salt at a specific point relative to the cluster at a certain time, and it is too tedious and inaccurate to count the clusters throughout the course of a long experiment.

Supplementary Video 2 demonstrates the feasibility of these types of experiments with the robot. First, the robot adds 0.5 mL of 10 mM sodium decanoate (pH 11) on one end of a 76×26 mm microscope slide, then adds another 0.5 mL of sodium decanoate in the center of the microscope slide. Therefore, the sodium decanoate is distributed all

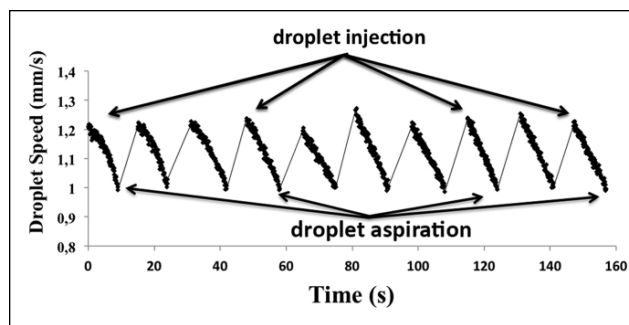


Figure 6. Droplet speed throughout time for use case 1. As can be seen in **supplementary video 1**, when droplet speed goes lower than a threshold, the robot aspirates the droplet. The experiment is repeated 10 times.

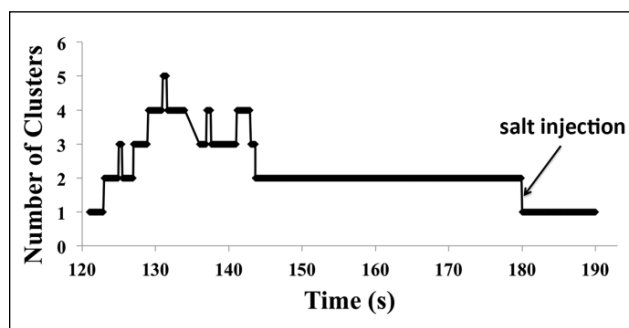


Figure 7. Number of clusters throughout time in a declustering experiment. The number of clusters is obtained from the vision API of EvoBot. As can be seen in the **Supplementary Video 2**, when droplets form a cluster, a salt solution is aspirated and dispensed at the specified distance to the cluster center.

over the slide. Then, the robot adds a row of five $5 \mu\text{L}$ decanol droplets and a row of four $5 \mu\text{L}$ decanol droplets, with a distance of 10 mm between droplets. **Figure 7** shows the number of clusters throughout time for this experiment obtained from EvoBot's vision API. As can be seen in the video, the robot detects the desired behavior as soon as a cluster forms, and it injects a $10 \mu\text{L}$ droplet of a 1 M NaCl solution (i.e., $10 \mu\text{mol}$ of NaCl) at 10 mm distance to the cluster center.

Conclusion

We introduced EvoBot, a robot that is affordable, open source, and modular. Owing to these characteristics, the robot is extendible and can also be used for feedback-based experiments. We demonstrated the application of EvoBot for new possibilities in non-feedback-based liquid-handling experiments such as 3D droplet placement. In addition, we investigated two use cases to manipulate experiments based on feedback from sensor input. Owing to the modularity of EvoBot, its application can be generalized for a wide range of feedback-based experiments.

Acknowledgments

The authors would like to thank EU Future and Emergent Technologies, who supported this work through EVOBLISS grant no. 611640, and the members of the EVOBLISS consortium. In particular, Jørn Lambertsen, who was involved in building EvoBot; and Jitka Cejkova, Martin Hanczyc, and Silvia Holler, who helped us perform artificial chemical life experiments, some of which are featured in the article. In addition, we would like to thank Laura Beloff and Jonas Jørgensen, who provided photos from the experiments they performed with the robot.

Declaration of Conflicting Interests

The authors declared no potential conflicts of interest with respect to the research, authorship, and/or publication of this article.

Funding

The authors disclosed receipt of the following financial support for the research, authorship, and/or publication of this article: The authors would like to thank EU Future and Emergent Technologies, who supported this work through EVOBLISS grant no. 611640, and the members of the EVOBLISS consortium.

References

1. Andrew Alliance. <http://www.andrewalliance.com> (accessed Aug 10, 2016).
2. Wohlsen, M. This Robot Could Make Creating New Life Forms as Easy as Coding an App. <http://www.wired.com/2014/11/opentrons-bio-robots/> (accessed Aug 10, 2016).
3. Gutierrez, J. M. P.; Hinkley, T.; Taylor, J. W.; et al. Evolution of Oil Droplets in a Chemorobotic Platform *Nat. Commun.* **2014**, *5*, 5571.
4. Faina, A.; Nejatimoharrami, F.; Stoy, K.; et al. Evobot: An Open-Source, Modular Liquid Handling Robot for Nurturing Microbial Fuel Cells. In *Proceedings of the 15th International Conference on the Synthesis and Simulation of Living Systems, Cancun, Mexico, July 4–8, 2016*; Gershenson, C., Froese, T., Siqueiros, J. M.; et al., Eds.; MIT Press: Cambridge, MA, 2016.
5. Nejatimoharrami, F.; Stoy, K.; Faina, A. An Open-Source, Low-Cost Robot for Performing Reactive Liquid Handling Experiments. Presented at the fifth annual SLAS (Society for Laboratory Automation and Screening) Conference and Exhibition, San Diego, CA, Jan 23–27, 2016; Paper ID 131765.
6. Nejatimoharrami, F.; Faina, A.; Cejkova, J.; et al. Robotic Automation to Augment Quality of Artificial Chemical Life Experiments. In *Proceedings of the 15th International Conference on the Synthesis and Simulation of Living Systems, Cancun, Mexico, July 4–8, 2016*; Gershenson, C., Froese, T., Siqueiros, J. M.; et al., Eds.; MIT Press: Cambridge, MA, 2016.
7. Hanczyc, M. M.; Gutierrez, J. M. P.; Nicholson, A.; et al. Creating and Maintaining Chemical Artificial Life by Robotic Symbiosis. *Artif. Life.* **2015**, *21*(1), 47–54.