Research

CrossMark
click for updates

Towards heterotic computing with droplets in a fully automated droplet-maker platform

Alon Henson, Juan Manuel Parrilla Gutierrez, Trevor Hinkley, Soichiro Tsuda and Leroy Cronin

School of Chemistry, University of Glasgow, Glasgow G12 8QQ, UK

The control and prediction of complex chemical systems is a difficult problem due to the nature of the interactions, transformations and processes occurring. From self-assembly to catalysis and self-organization, complex chemical systems are often heterogeneous mixtures that at the most extreme exhibit system-level functions, such as those that could be observed in a living cell. In this paper, we outline an approach to understand and explore complex chemical systems using an automated droplet maker to control the composition, size and position of the droplets in a predefined chemical environment. By investigating the spatio-temporal dynamics of the droplets, the aim is to understand how to control system-level emergence of complex chemical behaviour and even view the system-level behaviour as a programmable entity capable of information processing. Herein, we explore how our automated droplet-maker platform could be viewed as a prototype chemical heterotic computer with some initial data and example problems that may be viewed as potential chemically embodied computations.

1. Introduction

The implementation of unconventional computing using a chemical system has a relatively long history in the field. Kuhnert [1] was one of the first people who demonstrated the computing capability of chemical reactions for image processing. In chemical unconventional
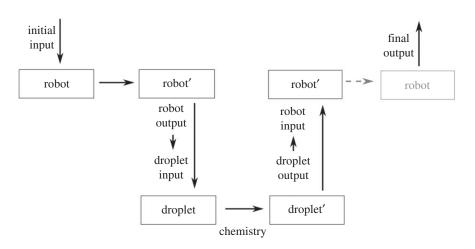
THE ROYAL SOCIETY
PUBLISHING

computing, excitation waves of the Belousov–Zhabotinsky (BZ) reaction are often employed to implement computing systems. For example, Fredkin & Toffoli [2] have shown that various types of logic gates can be implemented in a similar way to a collision-based, billiard-ball computer. The BZ reaction has also been known to be capable of planning optimal paths, such as maze solving [3], robot navigation [4] and computation of a Voronoi diagram [5]. These systems demonstrate the computational capabilities of chemical systems to carry out computations in a distributed fashion by exploiting the massively parallel potential of chemical systems. However, one grand aim is to exploit the computational potential of molecules from parallel assemblies, to interacting ensembles, down to the individual molecule without direct addressing. While most 'conventional' implementations are mostly done in a bulk solution, water-in-oil (W/O) droplets have been increasingly used to compartmentalize the BZ reaction in recent years [6–8]. Such compartmentalized chemistry has also been employed to implement chemical computers: Tompkins *et al.* [9] used W/O droplets encapsulating the BZ reaction to test Turing's theory of chemical morphogenesis. This is because BZ W/O droplets are shown to be capable of constructing Boolean logic gates as well as circuits by limiting the propagation of excitation waves to occur only at the interface between droplets [10,11]. Experimental implementations of chemical logic circuits using BZ droplets have also been partly done [12]. BZ reaction vesicles, which encapsulate the BZ reaction in vesicle membranes, have been theoretically conceived, but have yet to be implemented experimentally [13,14].

In relation to the heterotic computing framework, little work has been done in chemical unconventional computing. This is partly due to the difficulty in controlling the nonlinear behaviour of excitation waves, which propagate in all directions uniformly without control. However, it has been shown that the BZ reaction can be optically controlled by incorporating light-sensitive molecules, such as ruthenium complexes, in the reaction mixture and shining visible light onto it [15]. Exploiting the light sensitivity of the BZ reaction, Toth *et al.* [16,17] have demonstrated the coevolution of a cellular automata-based controller and the BZ reaction medium to construct collision-based logic gates, such as AND, NAND and XOR. The system could be viewed, in terms of the heterotic computing framework, as a coupled system of conventional computing (evolutionary algorithm) and unconventional computing (BZ reaction medium) components. However, the chemical medium was treated as a fixed component due to the difficulty of exchanging subcomponents (i.e. the chemicals involved in the BZ reaction). This results in a lack of programmability for an unconventional computing system. To overcome the issue, we have introduced a robotic liquid-handling platform to dynamically change the composition of chemical components. This would be expected to greatly increase the programmability and therefore improve the computing capability as well as the flexibility of chemical unconventional computers.

2. Results and discussion

We suggest a heterotic computer that has two distinct physical systems. The first physical computing system is a robot capable of chemical manipulations which we have described in previous work [18]. The scheme for the operation of the computation system is shown in figure 1. After the initial input to the robot, it changes states and the output of the altered state of the robot allows for the chemistry to be set in motion in the second physical system, the droplet system. The different variables that the robot controls in the chemistry of the system are the inputs to it. These inputs along with the physical and chemical rules determine the outcome of the droplets. Although the chemistry is stochastic, there are behaviours that are exhibited by the droplets which are repeatable and reliable [19]. The behaviours are the outputs of the droplet system which can be analysed and extracted to serve as inputs for the robot, which will use a model to give new inputs to the chemical system. It is worth noting that this can be seen as having the inputs change the system's configuration, which then changes the inputs and so on, a basic principle in computing. This cycle can be repeated until the calculation has been done and the abstract layer decodes the output into the abstract solution.

Figure 1. Schematic of the two interacting computing systems. First, the robot is input a sequence to be executed that changes the robot into state robot'. The output of this state in encoded as the input for the droplet system. The droplets behave according to the chemical rules to form the state droplet'. The output is read by the robot, which uses that input to move to another state. This sequence repeats until the computation is done and the output is passed on.
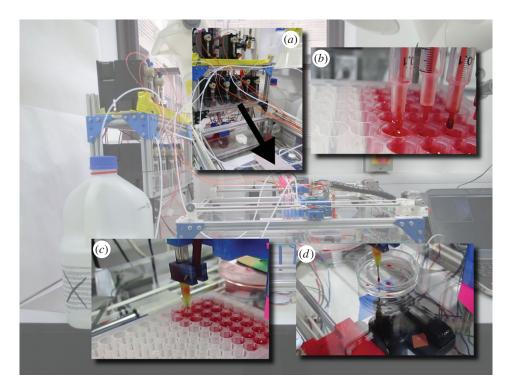
In order to construct this computational system, there is first a need to understand and create a model that can use the physical rules of the robot and droplet system in a reliable and deterministic fashion. We should be able to achieve this initial goal as the robot and the chemistry it creates are both controllable and measurable. We have already shown that we are able to use feedback control and genetic algorithms in combination with the high repeatability enabled by the robotic control of the chemical system to target specific behaviours. Being able to choose the outcome of a combined chemical and physical system means that, even if the underlying rules are not yet understood, the predictable manner by which they can now be investigated will allow for clarification of the dynamics of the system. The predictability as well as the ability to tune the behaviour of the system will also help elucidate the relation connecting the abstract layer to the computational layer, in this case a heterotic combination of a chemical manipulation robot and a chemical droplet system. In the following section, we will first overview our integrated robotic platform for forming oil-in-water (O/W) droplets, called dropletbot. When combined with evolutionary computing, the system allows us to 'program' the behaviour of O/W droplets. We then show that such programmed droplets can be potentially employed to implement computing systems, such as logic gates and cellular automata.

3. The robot

The platform used in this research as an automated medium for applying unconventional computer paradigms to droplet-based chemistry is based on a RepRap [20] three-dimensional printer, in which the thermoplastic extruder was replaced with a liquid-handling carriage and the X- and Y-axes were elongated in order to obtain a bigger working space. Some other minor changes required adding a camera to monitor the experiments, and moving most of the mechanical elements to the sides in order to avoid visual collisions with the running experiments. The robot can be seen in figure 2. Most of the mechanical and liquid-handling elements were designed and three-dimensionally printed using a RepRap three-dimensional printer (the thermoplastic material used was polylactic acid), which means that, if there is a need for modifications in order to introduce new technical capabilities, new elements can be easily three-dimensionally printed and attached to the robot. More information about the robot's construction and capabilities can be found elsewhere [18].

The platform itself consists of four different domains. The first is a series of liquid pumps (figure 2a) which connect the chemical reactants to the robot's liquid-handling carriage. The

4

rsta.royalsocietypublishing.org Phil. Trans. R. Soc. A **373**: 20140221

Figure 2. The fully automated platform can be seen in the background, with the liquid pumps and reactant bottles to the left, the working space with the liquid-handling carriage in the centre, and the electronics and computer to the right. The liquid pumps (*a*) and the moving carriage are used to pump the reactants to the 96-array well plate (*b*) after positioning the outlet above each desired well. Each well contains a magnetic stirrer to mix the formulation. The carriage (*c*) is also equipped with the automated syringe, which takes up the mixture from the designated well, and then generates the set of droplets in the Petri dish (*d*). Once this process is finished, the camera will record everything that happens during the experiment.

second is the droplet maker, which consists of a 96-array well plate (figure 2*b*), to which the liquid is added by the pumps; each well is equipped with a magnetic stirrer and an automated syringe, which takes up the liquid from a well plate (figure 2*c*) and generates droplets using mechanical pulses from a servo motor. The third is the experimental arena, which consists of a glass Petri dish (figure 2*d*), where the droplets are generated, and a video camera, which monitors and records the experiments. Finally, the last domain is the electronic hardware, which in our case consists of two Arduino Mega boards to control the pumps; the stepper motors which move the robot in the X–Y plane; the automated syringes; and a conventional computer, which generates 'g-code' instructions for X–Y carriage movement, based on an algorithm, and then sends them to be executed.

The platform works by starting with the computer designing one experiment. In our case, this means deciding the reactants which are going to be part of the experiment, and the quantities of the reactants, as well as the number of droplets, their sizes and their positions. In most cases, the size of the droplets was fixed at $5\,\mu l$, the population size at four droplets and the positions at four different points distributed in a square shape within the Petri dish. The droplet composition was based on four reactants (1-octanol, diethyl phthalate (DEP), 1-pentanol and either dodecane or 1-octanoic) mixed in different ratios, and the aqueous phase was a fixed solution (tetradecyltrimethylammonium bromide (TTAB), pH 13) with a fixed liquid quantity per experiment. Once the computer defines all the parameters to perform the experiment, this information is sent to the Arduino boards, which act as robotic controllers. The first action is to fill the Petri dish with the aqueous phase. The robot moves its carriage over the Petri dish, and the relevant pump fills the Petri dish. The next step is to generate the compositions of the droplets.

Table 1. This shows the physical inputs and measured outputs for the droplet robot.

droplet inputs/ environment inputs	robot output	droplet expression	droplet behaviours	droplet measures
four chemical volumes; 12 bit space; pH of aqueous phase; additives in the aqueous phase	population of droplets; total volumes fixed	droplets interact with each other and environment	motion; division; rotation	pixels moved; number; degrees

The robot moves its liquid-handling carriage over a defined position in the 96-array well plate, and the pumps are actuated to pump the liquid from the reactant bottles to the chosen well. Once the mixture is prepared, a magnetic stirrer mixes everything for a few seconds, and then the automated syringe extracts some of the mixture. The syringe is then moved over to the Petri dish, where it is lowered so that the tip of the needle is just over the air–aqueous interface, and four droplets are formed by generating pulses with a servo motor. After the four droplets are placed over the aqueous phase, a white background covers the Petri dish, and the experiment starts. The camera records everything that happens inside the Petri dish for 60 s. Once the experiment is finished, the Petri dish is cleaned. To do this, the Petri dish is automatically washed three times with acetone, followed by three washes with the aqueous phase. While the Petri dish is cleaned with acetone, the syringe is also cleaned. Once this experiment is finished, information from the experiment is sent back to the computer, and the computer defines a new experiment.
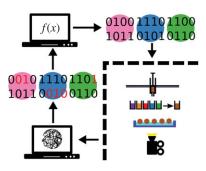
4. The droplets

Our droplet system is based on O/W emulsions. Both the aqueous phase and the oil phase contain surfactant molecules. In the aqueous phase, there is a fixed quantity of TTAB, pH 13. In the oil phase, there is a mix of different oils: 1-octanol, DEP, 1-pentanol, dodecane or 1-octanoic. Our droplets have a size of $5\,\mu l$, and the Petri dish where they are placed contains 2 ml of aqueous phase. The dynamics of the system are mostly based on Marangoni effects [21–23], where the different oils that assemble the droplet dissolve into the aqueous phase in different ratios, creating localized instabilities within the oil–water interface; these act as precursors to the pattern of behaviours we obtain in our droplet experiments.

5. Using 'in silico' algorithms to program droplets

As explained above in the description of the robot, the robot itself is connected to a computer which defines the experiments by setting the input variables for the droplet system. In our platform, the input variables are: the ratio and composition of reactants, which includes the oil and water phases; several basic chemical properties, such as the pH of the aqueous phase; the order in which the reactants are added to the mixture, as well as the time between addition steps; the wait period after all the reactants are added, which defines the length of time that the solution is mixed using the magnetic stirrer; the number and size of the placed droplets; the duration of the experiment; and any online modifications to the experiment such as changing the pH or adding any materials that will promote chemotaxis. From a single droplet experiment, we obtain the position of the droplet in every frame, and some analytical measurements such as changes of pH or spectral analysis. Using the droplets' positions over time, basic system dynamics can be easily calculated, such as their speed, turning angle, divisions or fusions (table 1).

This information sent from the computer to the robot, and then back to the computer, can be used to program the droplet system as described. In our case, by 'programming a droplet' we mean to define a droplet composition which will manifest a given behaviour. By behaviour, we
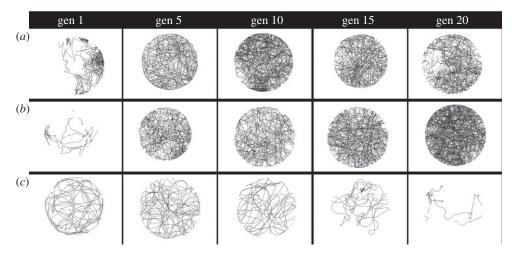
Figure 3. A schematic diagram of the robotic platform coupled to an optimization algorithm. A computer (top left) generates a number of input 'formulations' (top right), and each formulation is tested as droplets in a Petri dish (bottom right). The behaviour of the droplets is monitored by a camera mounted under the dish and trajectories of the generated droplets are analysed in a computer to generate formulations for the following round of droplet chemistry.

refer to any kind of physically dynamic manifestation, such as moving in a certain pattern, with a certain speed, or to divide, split, bounce around the Petri dish walls, collide with each other, and so on. It is important to note that the chemistry chosen will define the droplets' reaction space, and this in turn will determine how the droplets are to be programmed. Not every set of chemical variables will manifest the same set of behaviours. This is important as the state-space defined by chemical reactions, as well as material behaviour, will be vast with trillions upon trillions of possible states. Therefore, exploration of the state-space using an algorithm can assist the robot to navigate through this reaction space; see figure 3, in which the computer defines the desired function. Based on the previous data generated in relation to this function, a new experiment is defined, the reactants are chosen and a set of droplets is placed on a Petri dish. The experiment is monitored, and, when it is completed, the results of the physical computation are sent back to the computer; the computer then calculates the difference between the desired function and the actual function as defined by the droplets in the Petri dish. Using this error, the computer introduces variations on the experiment formulation, which is repeated, and its results are tested again against the desired function. This iterative process continues until the functions converge to some defined value. At this point, the computer has succeeded in programming the droplets to have some defined functionality.
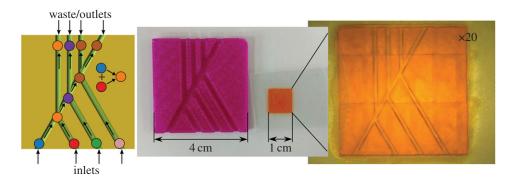
An example of this process applied to real experiments can be seen in [18], where a genetic algorithm was used to navigate a defined reaction space and optimize droplet formulations to produce a set of fitness functions. Figure 4 shows the droplet trajectories for 1 min experiments for three different fitness functions: division, movement and vibration. For each fitness function, snapshots of generation 1 and generation 20 are shown. In a genetic algorithm, generation 1 is usually completely random, while generation 20, as in this case, uses all the information generated during previous generations to navigate the reaction space to the point where the set fitness function localizes a global or local maximum. Once all the data are collected and the fitness functions are evolved to their best value, the robot knows how best to program droplets to achieve each possible fitness function. In our previous research, a genetic algorithm was used, but any other kind of artificial intelligence or optimization method would work, to enable programming of a droplet-based system.

6. The computations

Although at present we have been using the droplet robot, as presented here, in an observational mode, and also optimizing behaviours using an algorithm, we now outline some possible 'model' computations that we believe this platform sets the ground work for. Firstly, we propose that the dropletbot can be used as a platform to physically implement automata. In this scenario, we

7

rsta.royalsocietypublishing.org *Phil. Trans. R. Soc. A* **373**: 20140221

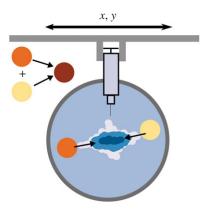|	gen 1	gen 5	gen 10	gen 15	gen 20
(*a*)					
(*b*)					
(*c*)					

Figure 4. Trajectories of droplet movements in a Petri dish every five generations from the first generation (left) to the twentieth generation. (*a*) Dividing droplets, (*b*) moving droplets and (*c*) vibrating droplets.

Figure 5. A possible scheme for digital calculations with droplets of differing chemistries. Allowing for four inputs of droplets and monitoring the four outputs, it would be possible to measure the resulting chemical reactions taking place when the droplets combine at the different junctions. The two photographs on the right are three-dimensionally printed devices at two different scales implementing the schematic design.

consider the system of robot and inputs as state 0, then we place a droplet which makes the system go to state $+1, +2, \ldots n$, based on the rules of physics and chemistry. We will then control state 0 to make droplets generate different states as we have recently shown [18]. In chemical droplets it is quite easy to increase the system complexity by having built-in chemical reactions in the droplets. For instance, a droplet with chemical component a is fused with another droplet containing component b, which results in the chemical reaction a + b (or no reaction at all if so designed). By selecting a series of chemistries that could change the physical properties (e.g. colour, motility, division and rotation), the complexity of the droplet system can be increased exponentially. In this set-up, the robot behaves like a stack calculator, where you always have a 'parent droplet' and program fusion of other droplets into an existing droplet in order to perform operations. This could be done either simply in a Petri dish or in a more controlled way, such as in microfluidics or a three-dimensional printed device. For example, it is very well known [24] that droplet fusion and division can be achieved by restricting the geometry of the droplet environment in a microfluidic device. In our macroscopic droplet environment, this can be possible using three-dimensional printed droplet trials, as shown in figure 5. In this case, droplets

Figure 6. The use of the dropletbot to conduct experiments with droplets with different chemistries in a Petri dish where the syringe is adding chemicals to affect the behaviour of the droplets by creating the gradient conditions needed for chemotaxis to occur.

that have already been programmed are introduced from inlets on the channels. The droplets are sequentially collided one by one and computation can be carried out by mixing the internal contents of the droplets and an output is read at the outlet. Such a droplet-based collision system enables us to implement logic gates (collision-based computing [2]).

Alternatively, chemical computation can be carried out in a Petri dish with freely moving droplets as in figure 6. As shown above, the robot can introduce droplets to the dish. Although droplets would move autonomously and stochastically, multiple numbers of droplets could be introduced and their statistical behaviour interpreted as computations. This could be done with pre-programmed droplets equipped with defined chemistry: the droplets would be programmed for a given behaviour using an optimization algorithm as outlined above. In this scenario, the robot would trace the path of the droplet and the droplets would go through that path and collide and merge with each other. By adding more droplets and more paths, more collisions would be engineered and the robot could even be used to intercept and remove a proportion of the droplet volume to ensure that the droplet entities conform to a standard size. Finally, by changing the droplet population, for example a computational environment with many droplets and droplet types, the formulation of the droplet would allow rules to be applied to both predicting and understanding which droplets would be able to merge upon collision (figure 6). In this way, the robot would be truly programming the chemistry as the droplets would control the sequence of reactions autonomously, as well as changing other parameters such as pH, adding chemical gradients for chemotaxis, or by removing and adding other members to and from the population.

7. Conclusion

One of the key elements of heterotic computing is to take advantage of the unconventional computing substrate rather than forcing the substrate to compute, often with great programming, technical and power constraints. However, we are so familiar with conventional computing architectures, driven by digital logic on a silicon substrate, that finding the correct process to convert potential natural computing architectures that take advantage of the laws of physics, chemistry and biology will require a new abstract 'translation' framework for constructing the computation [25]. In order to completely understand how to construct a new theory for computation, and then explore how existing physical systems may be better for one type of problem as opposed to others, we need to rigorously define the practical considerations for carrying out a computation. In this contribution, we have outlined a hybrid robo-chemical platform. In previous work, we started to explore this process by considering whether

nano-cellular electron storage systems could be used as cellular automata [26]. In that case, we hypothesized that, rather than having a universal set of laws, the physical constraints would define the local rules for the crystal-CA. Our droplet robot promises to be much more flexible insofar as the droplet compositions can be defined automatically, the environment modified, as can any image recognition and algorithmic optimization process. By being able to program many different chemical reactions, physical behaviours and link the two, the dropletbot has great potential to be used as a heterotic chemical computer. In further work, we are going to explore a range of embodied optimization problems that take advantage of the massively parallel nature of the molecular state-space, the stochastic properties and also the possibility for engineering several physical processes. The challenge will be to recognize where the system is truly computing, and then whether the computing process is faster, less expensive, more adaptable, with fewer layers and easier to program than current multi-layered digital computer systems.

References

1. Kuhnert L. 1986 A new optical photochemical memory device in a light-sensitive chemical active medium. *Nature* **319**, 393–394. (doi:10.1038/319393a0)
2. Fredkin E, Toffoli T. 1982 Conservative logic. *Int. J. Theor. Phys.* **21**, 219–253. (doi:10.1007/BF01857727)
3. Steinbock O, Tóth Á, Showalter K. 1995 Navigating complex labyrinths: optimal paths from chemical waves. *Science* **267**, 868–871. (doi:10.1126/science.267.5199.868)
4. Adamatzky A, Costello BDEL, Melhuish C, Ratcliffe N. 2003 Experimental reaction–diffusion chemical processors for robot path planning. *J. Intell. Robot. Syst.* **37**, 233–249. (doi:10.1023/A:1025414424756)
5. Adamatzky A, Lacy BD. 2003 On some limitations of reaction–diffusion chemical computers in relation to Voronoi diagram and its inversion. *Phys. Lett. A* **309**, 397–406. (doi:10.1016/S0375-9601(03)00206-8)
6. Szymanski J, Gorecki J, Hauser MJB. 2013 Chemo-mechanical coupling in reactive droplets. *J. Phys. Chem. C* **117**, 13 080–13 086. (doi:10.1021/jp402308t)
7. Delgado J, Li N, Leda M, González-Ochoa HO, Fraden S, Epstein IR. 2011 Coupled oscillations in a 1D emulsion of Belousov–Zhabotinsky droplets. *Soft Matter* **7**, 3155. (doi:10.1039/c0sm01240h)
8. Toiya M, Vanag VK, Epstein IR. 2008 Diffusively coupled chemical oscillators in a microfluidic assembly. *Angew. Chem.* **120**, 7867–7869. (doi:10.1002/ange.200802339)
9. Tompkins N, Li N, Girabawe C, Heymann M, Ermentrout GB, Epstein IR, Fraden S. 2014 Testing Turing's theory of morphogenesis in chemical cells. *Proc. Natl Acad. Sci. USA* **111**, 4397–4402. (doi:10.1073/pnas.1322005111)
10. Bull L, Holley J, Costello BDL, Adamatzky A. 2013 Toward Turing's A-type unorganised machines in an unconventional substrate: a dynamic representation in compartmentalised excitable chemical media. In *Computing nature* (eds G Dodig-Crnkovic, R Giovagnoli), pp. 185–199. Studies in Applied Philosophy, Epistemology and Rational Ethics, vol. 7. Berlin, Germany: Springer.
11. Adamatzky A, Holley J, Dittrich P, Gorecki J, De Lacy Costello B, Zauner KP, Bull L. 2012 On architectures of circuits implemented in simulated Belousov–Zhabotinsky droplets. *BioSystems* **109**, 72–77. (doi:10.1016/j.biosystems.2011.12.007)
12. Szymanski J, Gorecka JN, Igarashi Y, Gizynski K, Gorecki J, Zauner K, de Planque MRR. 2011 Droplets with information processing ability. *Int. J. Unconv. Comput.* **7**, 185–200.
13. Adamatzky A, De Lacy Costello B, Holley J, Gorecki J, Bull L. 2011 Vesicle computers: approximating a Voronoi diagram using Voronoi automata. *Chaos Solitons Fractals* **44**, 480–489. (doi:10.1016/j.chaos.2011.01.016)

14. Holley J, Adamatzky A, Bull L, De Lacy Costello B, Jahan I. 2011 Computational modalities of Belousov–Zhabotinsky encapsulated vesicles. *Nano Commun. Netw.* **2**, 50–61. (doi:10.1016/j.nancom.2011.02.002)
15. Kádár S, Amemiya T, Showalter K. 1997 Reaction mechanism for light sensitivity of the Ru(bpy)$_3^{2+}$-catalyzed Belousov-Zhabotinsky reaction. *J. Phys. Chem. A* **101**, 8200–8206. (doi:10.1021/jp971937y)
16. Toth R, Stone C, Adamatzky A, de Lacy Costello B, Bull L. 2008 Dynamic control and information processing in the Belousov–Zhabotinsky reaction using a coevolutionary algorithm. *J. Chem. Phys.* **129**, 184708. (doi:10.1063/1.2932252)
17. Toth R, Stone C, Adamatzky A, de Lacy Costello B, Bull L. 2009 Experimental validation of binary collisions between wave fragments in the photosensitive Belousov–Zhabotinsky reaction. *Chaos Solitons Fractals* **41**, 1605–1615. (doi:10.1016/j.chaos.2008.07.001)
18. Gutierrez JMP, Hinkley T, Taylor JW, Yane K, Cronin L. 2014 Evolution of oil droplets in a chemorobotic platform. *Nat. Commun.* **5**, 5571. (doi:10.1038/ncomms6571)
19. Horibe N, Hanczyc MM, Ikegami T. 2011 Mode switching and collective behavior in chemical oil droplets. *Entropy* **13**, 709–719. (doi:10.3390/e13030709)
20. Jones R, Haufe P, Sells E, Iravani P, Olliver V, Palmer C, Bowyer A. 2011 RepRap—the replicating rapid prototyper. *Robotica* **29**, 177–191. (doi:10.1017/S026357471000069X)
21. Scriven LE, Sternling CV. 1960 The Marangoni effects. *Nature* **187**, 186–188. (doi:10.1038/187186a0)
22. Nagai K, Sumino Y, Kitahata H, Yoshikawa K. 2005 Mode selection in the spontaneous motion of an alcohol droplet. *Phys. Rev. E Stat. Nonlinear Soft Matter Phys.* **71**, 1–4. (doi:10.1103/PhysRevE.71.065301)
23. Sumino Y, Kitahata H, Seto H, Nakata S, Yoshikawa K. 2009 Spontaneous deformation of an oil droplet induced by the cooperative transport of cationic and anionic surfactants through the interface. *J. Phys. Chem. B* **113**, 15 709–15 714. (doi:10.1021/jp9037733)
24. Duncan PN, Nguyen TV, Hui EE. 2013 Pneumatic oscillator circuits for timing and control of integrated microfluidics. *Proc. Natl Acad. Sci. USA* **110**, 18 104–18 109. (doi:10.1073/pnas.1310254110)
25. Horsman C, Stepney S, Wagner RC, Kendon V. 2013 When does a physical system compute? *Proc. R. Soc. A* **470**, 20140182. (doi:10.1098/rspa.2014.0182)
26. Symes MD, Cronin L. 2011 The crystal computer—computing with inorganic cellular frameworks and nets. *Int. J. Nanotechnol. Mol. Comput.* **3**, 24–34. (doi:10.4018/jnmc.2011010103)