A molecular computing approach to solving optimization problems via programmable microdroplet arrays

Si Yue Guo[a], Pascal Friederich[a,f], Yudong Cao[b,†], Tony Wu[a], Christopher J Forman[c], Douglas Mendoza[a,b], Matthias Degroote[a], Andrew Cavell[e], Veronica Krasecki[e], Riley J Hickman[a], Abhishek Sharma[d], Leroy Cronin*[d], Nathan Gianneschi[c], Randall H Goldsmith[e], Alán Aspuru-Guzik*[a,g,h,i]

[a] Department of Chemistry, University of Toronto, 80 St. George Street, Toronto, Ontario M5S 3H6 Canada

[b] Department of Chemistry and Chemical Biology, Harvard University, 12 Oxford Street, Cambridge, MA 02138, USA

[c] Department of Chemistry, Northwestern University, 2145 Sheridan Road, Evanston, IL 60208, USA

[d] School of Chemistry, University of Glasgow, Joseph Black Building, University Avenue, Glasgow, G12 8QQ, Scotland, UK

[e] Department of Chemistry, University of Wisconsin-Madison, 1101 University Avenue, Madison, WI 53706, USA

[f] Institute of Nanotechnology, Karlsruhe Institute of Technology, Hermann-von-Helmholtz-Platz 1, 76344 Eggenstein-Leopoldshafen, Germany

[g] Canadian Institute for Advanced Research (CIFAR) Senior Fellow, Toronto, Ontario, M5S 1M1, Canada

[h] CIFAR Artificial Intelligence Research Chair, Vector Institute, Toronto, Ontario, M5S 1M1, Canada

[i] Department of Computer Science, University of Toronto, 40 St. George Street, Toronto, Ontario, M5S 2E4, Canada

[†] Current Address: Zapata Computing Inc., Boston, MA 02110, USA

***Corresponding Authors.** Email: aspuru@utoronto.ca; Lee.Cronin@glasgow.ac.uk

Abstract

The search for novel forms of computing that show advantages as alternatives to the dominant von-Neuman model-based computing is important as it will enable different classes of problems to be solved. By using droplets and room-temperature processes, molecular computing is a promising research direction with potential biocompatibility and cost advantages. In this work, we present a new approach for computation using a network of chemical reactions taking place within an array of spatially localized droplets whose contents represent bits of information. Combinatorial optimization problems are mapped to an Ising Hamiltonian and encoded in the form of intra- and inter- droplet interactions. The problem is solved by initiating the chemical reactions within the droplets and allowing the system to reach a steady-state; in effect, we are annealing the effective spin system to its ground state. We propose two implementations of the idea, which we ordered in terms of increasing complexity. First, we introduce a hybrid classical-molecular computer where droplet properties are measured and fed into a classical computer. Based on the given optimization problem, the classical computer then directs further reactions via optical or electrochemical inputs. A simulated model of the hybrid classical-molecular computer is used to solve boolean satisfiability and a lattice protein model. Second, we propose architectures for purely molecular computers that rely on pre-programmed nearest-neighbour inter-droplet communication via energy or mass transfer.

Significance statement

Molecular computers—a promising alternative to semiconductor-based computers—not only compute but also re-organize matter, potentially admitting novel manufacturing approaches. The vast space of chemical reactions makes molecules a low-cost, energy-efficient, tunable and scalable computational vehicle. In molecular computers, memory and processing units combine into inherently parallelized single devices. We present a molecular computer for combinatorial optimization in a microdroplet array that employs an Ising Hamiltonian to map problems heuristically to droplet-droplet interactions. The droplets represent binary digits and problems are encoded in intra- and inter-droplet reactions. We propose two implementations: a hybrid classical-molecular computer that enforces inter-droplet constraints in a classical computer and a purely molecular computer where the problem is entirely pre-programmed in the droplet reactions.

1. Introduction

For science and technology to continue progressing at the current pace, we need our computational processing capabilities to keep growing. Conventional transistor-based computers follow the von Neumann architecture, where information is stored in memory units and processed in a central processing unit. Different logic gates, each acting on a single bit or multiple bits of information represented as a 0 or 1, are activated sequentially to perform an operation. However, as transistors are approaching tens of nanometers, i.e., the size of large molecules (1), high heat dissipation, and slow transfer rates between processors and memory are bringing about the breakdown of Moore's Law.

The search for alternatives to classical computers includes quantum computing, in which quantum mechanical phenomena allow information to exist as a superposition of states, not just individual binary bits (2, 3). For classical systems, computing with molecules is an attractive avenue given the vast chemical space and the relatively low energy dissipated in chemical reactions compared to transistors (4–6). Proposals for computers that exploit chemical processes have taken one or both of two broad approaches: 1) using chemistry and biochemistry to emulate circuit components or cellular automata, and 2) employing a large number of molecules to explore a combinatorial space in parallel. Examples for the first include reaction-diffusion systems (7), Belousov-Zhabotinsky oscillatory reaction (8), memristive polymers (9), and transcription regulation for cellular signaling (10, 11), and other chemical and biochemical analogues of logic gates (12, 13). In the second category of parallelized computing, we find microfluidic devices (14), nanofabricated networks (15–17), and adaptive amoebal networks (18). DNA computers have explored both approaches, from early work that involved sifting through combinations of nucleic acid sequences to recently using DNA self-assembly to carry out an algorithm process (19–21). However, each method presents challenges in programmability, scaling, and maintaining accuracy in large calculations.

Here we propose a new strategy for solving optimization problems using droplets spatially localized in a lattice and linked by a network of chemical reactions. Each droplet represents a binary variable that can communicate with its neighbouring droplets. We encode a problem in the intra- and inter-droplet interactions. As the droplets' contents evolve following the thermodynamics of the system, it reaches a steady-state where the ensemble of droplet states corresponds to a solution of the given problem. As we shall explain in **Section 2.1**, our proposed droplet-based computer relies on an alternative approach inspired by the pioneering work of Kirkpatrick et al.'s, which employs simulated annealing to solve combinatorial optimization problems (22). It is a heuristic method that intrinsically combines both memory and processing units in one device.

We ultimately envision a molecular computer that operates solely using chemical processes and without the aid of classical computers. This implementation, which we term the "purely molecular computer", uses pre-programmable chemical couplings, corresponding to a given problem, that determines neighbouring droplets' interactions. As a stepping stone to achieving complete autonomy, we have first conceived a "hybrid classical-molecular computer", where a classical

computer is used to enforce conditions that are then carried out by the droplet system. Both incarnations of the droplet-array computer would be inherently parallelized, easily scalable, efficient, specific purpose computing devices.

We shall present in **Section 2.2** the requirements for reactions to be suitable for our molecular computer. **Section 3** is devoted to a description and simulation of the hybrid classical-molecular computer and **Section 4**, the purely molecular computer. In **Section 5**, we will explore ways to implement our device physically. Lastly, in **Section 6**, we will discuss the advantages of this approach and examine the scalability of the system.

2. Concept

2.1 Optimization by simulated annealing

The basic principle of the droplet-array computer rests in the analogy between combinatorial optimization and statistical physics (22). Similarly to the ensemble behaviour of physical systems such as magnetic spins, an optimization problem is a complex system of interacting variables. Such interactions are captured in a cost function or Hamiltonian, where the constraints and requirements of the problem correspond to interaction energies between variables. At a given temperature, the equilibrium distribution of the configuration (ups and downs) of the spins follows a Boltzmann distribution, which favours the ground state of the Hamiltonian or the optimal (set of) solution(s) to the objective function. As the temperature lowers, the statistical weight (probability) of the ground state grows, which inspires the idea of simulated annealing to emulate the effect of such annealing process due to cooling for finding or approximating the ground state of the Hamiltonian.

A canonical model for magnetic spins in crystalline materials is the Ising or lattice spin Hamiltonian

$$H_{Ising} = -\mu \sum_{i=1}^{N} h_i s_i - J \sum_{i<j}^{N} s_i s_j, \quad s_i \in \{-1,1\}$$

whose terms describe a tradeoff between the cost of flipping a spin s_i in an external magnetic field h and the interaction energy between neighbouring spins, J, and where μ is the magnetic moment. Even though this model has only two parameters h and J, as we will discuss below in detail, it can already give rise to problems that are intractable in the worst cases. For encoding general combinatorial optimization problems, however, we allow for more tunability between pairs of coupled spins as well as the local fields on each spin, giving rise to the more general Ising Hamiltonian:

$$H_{general} = \sum_{i=1}^{N} \alpha_i s_i + \sum_{i<j}^{N} \beta_{ij} s_i s_j$$

in which the α and β coefficients are given by the problem.

In other words, the computational problem is encoded in the coefficients α_i, representing the local field for individual variables, and β_{ij}, representing the couplings between pairs of variables. For

the given values for the vector α and matrix β, the optimal configuration of s_i represents a solution that minimizes the Hamiltonian.

In order to implement the Ising model, we need a way to control the local field and couplings (one-body and two-body terms). In this work, we propose that chemical reactions can be used to achieve these features.

The model can be generalized for higher-order terms (*k*-local).

$$H = h^{(0)} + \sum_{i}^{N} {h_i}^{(1)} s_i + \sum_{i<j}^{N} {h_{ij}}^{(2)} s_i s_j + \sum_{i<j<k}^{N} {h_{ijk}}^{(3)} s_i s_j s_k + \cdots$$

We shall proceed with this notation, where the $h^{(k)}$ is the tensor of rank k representing the coupling between k variables.

Quantum annealing devices were argued to be superior to classical annealing due to the availability of quantum effects such as superposition and tunneling (23, 24). This inspired several quantum and classical devices, including the D-Wave quantum annealer (25–27), the coherent Ising machine (28, 29), the recently-introduced Fujitsu-led application-specific CMOS-based digital annealer (30), and even more recently, the Toshiba simulated bifurcation algorithm (31). In **Table 1,** we give an overview of these optimization annealing machines and their characteristics.

Table 1*. A survey of optimization annealing machines and their capabilities*

	Algorithm	**Hardware**	**Number of bits**	**Connec-tivity**	**Typical time to solution**	**Date**	**Optimization Problem**	**Ref**
Classical								
Fujitsu Digital Annealer	Digital annealing	Application-specific CMOS	1024 variables	Full	1 - 10^2 s	April 2019	Spin glass	(30)
Toshiba Simulated Bifurcation	Simulated bifurcation	FPGAs and GPUs	2000 variables	Full	5×10^{-4} s	April 2019	MAX-CUT (NP-hard)	(31)
Quantum								
Coherent Ising Machine	Adiabatic quantum computation, quantum annealing	Optical parametric oscillators; laser pulses	2000 spins	Full	5×10^{-3} - 5×10^{-2} s	Oct 2016	MAX-CUT	(28, 29)
D-WAVE 2000Q Chimera	Quantum annealing	Superconducting qubits	2000 qubits	6	10^2 - 10^4 s	Jan 2017	Numerous applications	(26)
D-WAVE "next gen" Pegasus	Quantum annealing	Superconducting qubits	5000 qubits	15	n/a	April 2019	Numerous applications	(27)
Chemical								
Hybrid classical-molecular computer	Simulated annealing, stochastic gradient descent	Droplet array + classical computer	12 droplets → 10,000 droplets	Full	~6×10^2 s; to be reduced with smaller droplets	2019	k-SAT, lattice protein	This work
Purely molecular computer	Chemical annealing	Droplet array	10's droplets → 10,000 droplets	4-6	n/a		2-local k-SAT, 2-local lattice protein, TSP	
Long-term molecular computer	Chemical annealing	Droplet array	~1.5 million droplets (38)	4-6	n/a			
Inkjet printer	Chemical annealing	Microscopic nozzle (600 dpi)	~30 million droplets	4-6	n/a			

Adiabatic quantum computing and quantum annealing provide us with a method of solving for the Ising ground state, by evolving a system from the ground state of an accessible problem to that of a more difficult problem (23, 24, 32). Although a molecular computer remains a classical device, unable to avail itself of quantum tunneling across barriers in the potential energy surface relating to the problem Hamiltonian, the rationale can still be extended to simulated annealing.

In adiabatic quantum optimization, a system is initially in the ground state of a Hamiltonian H_0. To solve for the ground state of a problem Hamiltonian H_p, the Hamiltonian H is adiabatically changed over time following a function $a(t)$: $H(t) = [1 - a(t)]\, H_0 + a(t)\, H_p$

For simulated annealing, H_0 represents kinetic or thermal energy and H_p the potential energy, corresponding to the problem we would like to solve. Just like annealing metals, as the temperature of a system decreases, the spins or variables reach an optimal configuration.

For any classical optimization algorithm that can be mapped to the classical Ising system, the time to solution (TTS) to find the ground state asymptotically grows at most exponentially in n, i.e., $O(c^n)$, as it is needed to explore all possible outcomes, and finding an approximate solution grows as polynomially in n, i.e., $O(n^\gamma)$, where γ is a problem-dependent constant (32). By the Church-Turing thesis (4), the molecular computer should not have any exponential speed-up over any classical computing devices.

Simulated annealing was found to be well-suited to tackle NP and NP-hard combinatorial optimization problems (33). These are problems whose computational complexity increases exponentially with the number of variables, and for which no efficient (polynomial-time) algorithm is presently known. Nevertheless, since solving the Ising model was shown to be NP-hard, a good heuristic Ising model solver will be beneficial for solving real-world problems that are often NP-complete problems[1], such as the traveling salesperson problem, graph colouring, and boolean satisfiability (35). Barahona showed that solving the states of the Ising model (36) on a 2D lattice with arbitrary interactions is NP-complete. Since every instance of a problem in NP can be efficiently encoded as an instance of an NP-complete problem, Barahona's work implies that all problems in the NP complexity class can be mapped to instances of 2D Ising model (33), and thus can potentially be solved efficiently with a heuristic Ising solver. The work of Barahona and others (37), summarized in **Table 2**, also includes the complexity of solving different Ising models.

Table 2. *The complexity of the Ising ground state problem in one-, two-, and three-dimensions.*

Dimension and graph	Restrictions on coupling	Solving for the Ising ground state
1 D	none	P
2 D (planar graph)	$h_i = 0$	P
	$h_i \neq 0$	NP-hard
Non-planar graph	$J_{ij} \in \{-1,0,1\}$	NP-hard

[1] NP-complete problems belong to both NP and NP-hard classes. They are defined as decision problems where a polynomial-time reduction exists from every other problem in NP (34).

2.2 Description and Requirements

As described earlier, the proposed molecular computer consists of an array of droplets arranged in a lattice, which is shown in **Figure 1.** To apply the concept of adiabatic quantum optimization described above to the molecular computer, we begin with the random initialization of the droplet states. We can apply the problem Hamiltonian in the computer by programming the local field terms h_i and couplings J_{ij}. The droplet-array molecular computer enforces these parameters so that as the system anneals, it reaches the ground state of the problem, which can be read out and interpreted. Since it is not a quantum computer, the system will not tunnel between local minima in the potential energy surface that is associated with the problem Hamiltonian. Instead, we will require multiple initializations to attain the global minimum, thus solving a given problem.

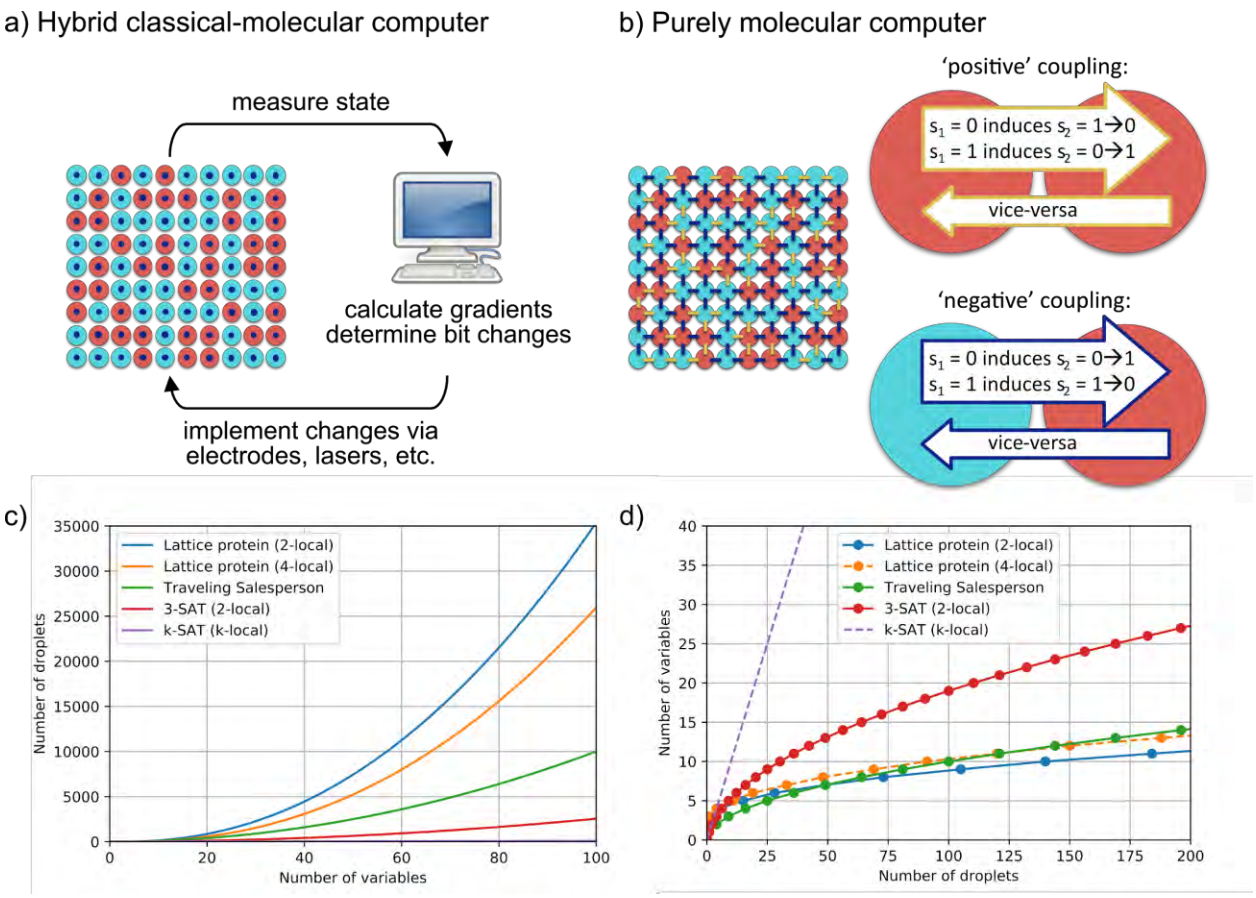

Figure 1. *Schematic of the droplet-array-based molecular computer. a) A hybrid classical-molecular computer in which measurements of droplet states are sent to a computer that calculates the gradient on each droplet and directs the changes to be implemented in situ. b) A purely molecular computer where the droplets interact via nearest neighbour couplings, either 'positive' or 'negative'. The couplings are programmed into the device in the initiation stage. c) The scaling of the number of droplets for different optimization problems: lattice protein folding (quadratic/2-local and quartic/4-local forms), traveling salesperson, boolean satisfiability (3-SAT reduced to 2-local and k-SAT k-local). d) The number of variables in these problems that a given droplet-array size can handle.*

To achieve molecular computing by annealing as described above, the chemistry taking place in the droplet array must fulfill several requirements. First, the system should exhibit physical properties that can be manipulated, read, and mapped as two states, so that each droplet can correspond to a binary variable. Second, changes of the states must be reversible and controllable by the state of their neighbouring droplets. Third, in order to implement interactions between the states that correspond to coupling terms of the Ising model, there must be connections between the droplets that allow for programmable positive and negative coupling interactions.

We propose two approaches to implementing such a system: first, a hybrid classical-molecular computer in which an array of droplets is connected to a classical computer; and second, a purely molecular computer. In **Table 1** we compare the two approaches and give the roadmap to scale to millions of droplets, as is currently achievable with inkjet printers at 600 dpi resolution (38).

In the hybrid classical-molecular computer, measurements of each droplet's state will be passed to a classical computer software that computes the interaction based on a given problem Hamiltonian and outputs instructions for the droplet-array to implement the calculated coupling, using, *e.g.* electrodes or optical excitations (see **Section 5**). In the case of the purely molecular computer, the coupling between nearest-neighbour droplets is pre-programmed in physical or chemical interaction between the droplets that take the form of mass or energy exchange (see **Section 4**).

The reversibility of the chemical reaction is essential to maintaining the symmetry of the problem. For instance, two droplets that are positively coupled to each other have a coefficient $J > 0$. The contribution to the Hamiltonian $-Js_1s_2$ is minimized when the two droplets are in the same state, either both 0 or both 1, which means both reactions 0 → 1 and 1→0 must be achievable. (This symmetry is broken as interactions with other droplets are taken into account, leading to possible frustration as they all compete with each other.)

We must emphasize that the molecular computer does not have *a priori* knowledge of the solution to the problem. It merely imposes the problem Hamiltonian on the droplets via a local field and droplet couplings. As the chemical system evolves under these conditions, it will attempt to minimize its thermodynamic free energy, which is the physical impetus to explore distinct solution states associated with the problem Hamiltonian, imposed by the couplings. Depending on the given problem, it may have degenerate ground states corresponding to multiple optimal configurations.

We expect that the kinetics of the problem, as well as imperfections in the settings of the molecular computer, will result in trapping in local minima for particular experiments, requiring the repetition of the computation to sample from the low-energy states of the problem. The alternative approaches mentioned earlier, namely simulated annealing, quantum annealing, and the Toshiba simulated bifurcation algorithm, share this challenge (30, 31).

3. Modeling and application of the hybrid classical-molecular computer

Tuning the intra- and inter-droplet chemical reaction to satisfy the requirements described above is not a trivial task. To test chemical systems for the droplet-based computer, we propose first developing a hybrid classical-molecular computer. A hybrid device also has the benefits of non-local interactions, full connectivity between all droplets, and higher-order couplings that involve *k* droplets (*k*-local interaction).

Our proposed implementation of a hybrid classical-molecular computer uses an array of microdroplets containing various chemical compounds. Manipulation of droplets can be achieved with electrostatic fields and electrochemical transformations induced by electrodes or optical stimuli induced by light-sources. The change in the states of droplets should be detectable by optical means, which allows for a convenient readout of the states. For the sake of simplicity in this concept paper, we use an abstract nomenclature of states 0 and 1. We shall assume that the droplets fluctuate within this range of states as a result of external stimuli controlled by a classical computer. We will discuss possible experimental schemes to build such a system in **Section 4**. One can think of the value associated with each droplet as the droplet's progress along an arbitrary reaction coordinate.

To explore the capabilities and potential application areas of the hybrid classical-molecular computer, we implemented an *in-silico* model of it. The model is composed of two parts, a set of readable and writable states that represents the droplet-array in the molecular computer and a computer algorithm that, given a model Hamiltonian, calculates and applies stepwise changes to these states. The model will be described below in **Section 3.1**.

We applied the *in-silico* model of the hybrid classical-molecular computer to two applications that we discuss in **Sections 3.2** and **3.3**. The first application is Boolean satisfiability (in particular 3-SAT), which was proven to be NP-complete (39). The second application is a simplified protein model comprising of six types of amino acids on a 2D square lattice. To solve these problems using the molecular computer, we map them to Ising Hamiltonians, initialize the droplets in random states and encode the coupling elements of the Ising Hamiltonian in the couplings between the droplets. Annealing of the system leads to a local optimum, from which a solution can be read out and checked on validity. Repetition of this procedure allows us to determine if a Boolean satisfiability problem has a solution that fulfills all conditions or that identifies low-energy conformations of the protein. A more exhaustive list of mappings from NP-complete problems to the Ising Hamiltonian is presented in Ref. (36).

3.1 Simulation by stochastic gradient descent

We modeled the annealing process of the hybrid classical-molecular computer using a stochastic gradient descent simulation. As shown in the methods section, it is possible to convert Boolean satisfiability problems, as well as the lattice protein folding model, to an Ising model that can be represented in the form of tensors $h^{(0)}, h^{(1)}, \ldots, h^{(n)}$. The first term, $h^{(0)}$, is an offset to the global

energy that does not influence the states of the system, $h^{(1)}$ is a vector of bias terms to the states of each droplet, $h^{(2)}$ is a matrix of two-droplet coupling terms, $h^{(3)}$ is a tensor containing three-droplet coupling terms, and so on. The 3-SAT problem can be converted to an Ising model with up to three-body interaction terms, while the lattice-protein model contains up to four-body interactions. As discussed earlier, the hybrid classical-molecular computer can handle interactions of any order (requiring *n* droplets for *n*-body terms), while the purely molecular computer will be limited to nearest-neighbour two-droplet interactions due to the limited connectivity between the droplets. It is, however, possible to convert higher-order interactions such as three- and four-droplet to two-body interactions by introducing ancillary states in the system (40). While our model of the hybrid classical-molecular computer can process any order of many-body interactions, we used the two-body representation of the lattice-protein model that includes ancillary states.

To simulate the annealing process of the droplet-array, we use an iterative procedure. In each step, the state of the system is read, all gradients acting on the states in terms of bias potentials ($h^{(1)}$) and many-body interactions ($h^{(2)}$to $h^{(n)}$) are calculated and changes to states are applied with a finite step size. We run the procedure in two modes, ensemble annealing mode and trajectory mode. In the ensemble annealing mode, we run many short annealing simulations from random initial states and analyse the statistics of the final states or search for the global optimum. In contrast, in trajectory mode, we run one instance of the molecular computer over a large number of steps but add a certain level of noise on the gradients to overcome barriers between local optima. Potentially, to overcome kinetic trapping, the hybrid classical-molecular computer could benefit from accelerated sampling methods developed for droplet dynamics simulations, *e.g.,* basin-hopping, metadynamics and umbrella sampling.

3.2 Application 1: Boolean satisfiability

In order to solve Boolean satisfiability problems such as the 3-SAT problem with our computational model of the hybrid classical-molecular computer, it is necessary to convert it from its conjunctive normal form (CNF) to an Ising model (see Appendix). In the case of 3-SAT problems, the Ising model will contain up to three-body coupling terms, which requires connectivity that goes beyond that of a purely molecular computer. We tested the satisfiability of random 3-SAT problems with N variables and M unique clauses of $k = 3$ literals and compared the results to solutions obtained using the Davis-Putnam-Logemann-Loveland (DPLL) algorithm (41) (see **Figure 2a**). The results are shown in **Figure 2b**, where we observe a transition from a high probability of a problem being satisfiable when clause density, *i.e.,* the ratio of M/N, is small to a low probability at large M/N ratios. The results obtained with the simulation are in good agreement with the results obtained using the (exact) DPLL algorithm.

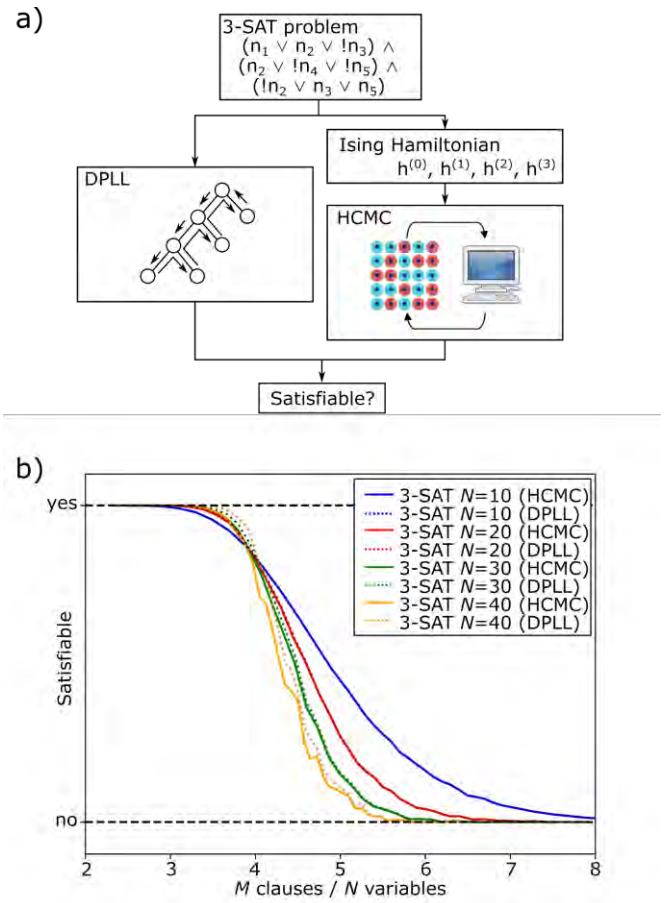

Figure 2. *a) Illustration of two methods to solve the 3-SAT problem. b) Probability of satisfiability as a function of clause density (ratio of clauses and variables) for a random 3-SAT problem, showing agreement between the simulated hybrid classical-molecular computer (HCMC, solid lines) and the solutions of the Davis-Putnam-Logemann-Loveland (DPLL) algorithm. Each point is the average of up to 10000 3-SAT problems.*

3.3 Application 2: Lattice protein model

As a second application, we tested the simulated hybrid classical-molecular computer on the two-dimensional lattice protein model presented in Refs (42, 43). The proteins in this model consist of a sequence of amino acids that can fold onto a two-dimensional square lattice. The protein conformation is represented as a sequence of turns, where each turn is encoded by two bits (00: down, 01: right, 10: left, 11: up, see **Figure 3a** and **3b** for a description and an example of a folding).

The energy of each conformation contains two terms: clashes of two amino acids are penalized with positive terms of size λ whereas attracting interactions between specific amino acids (i and j, $|i-j|>1$) are rewarded with negative energy terms ϵ_{ij} in case amino acids i and j are direct, non-diagonal neighbours (native contact). We can construct an Ising Hamiltonian with up to four-body interactions that encodes both types of interactions. This requires ancillary bits that activate and deactivate depending on the state of the bits that encode the physical conformation of the

protein. The overlap parameter λ can be chosen in a way that the energy spectrum of the protein model and thus of the Ising Hamiltonian has negative or zero energies for all conformations without clash and positive energies for all other conformers. The global ground state of the Hamiltonian yields the protein conformation in which the highest possible number of native bonds are formed. Non-clashing conformers with fewer or less strong native bonds are local optima with energies smaller than zero, whereas non-clashing conformers without any native bond formed have zero energy ("unfolded").

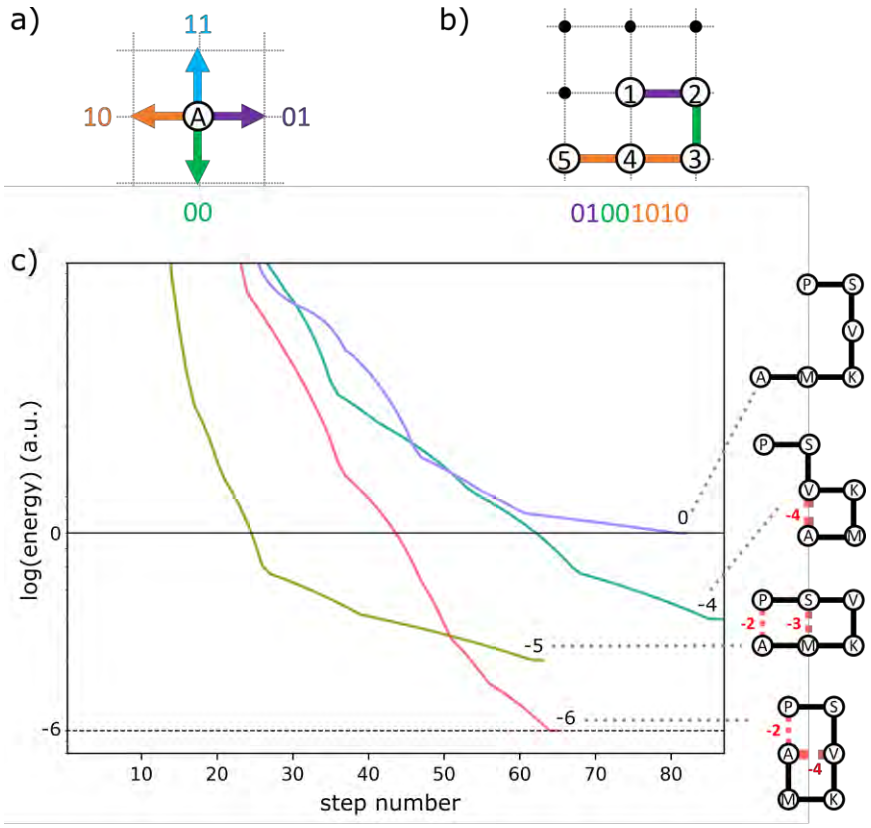

Figure 3. *Lattice protein model simulated on the molecular computer. a) encoding of each link in the amino acid chain as two bits, b) an example 5-link chain with the corresponding sequence of bits, c) results of a simulation for a 6-link chain, where each curve corresponds to an instance of random initialization followed by annealing to a local minimum, with the corresponding folded configuration. The ground state configuration, with an energy of -6, is shown in the lower right corner.*

As described in **Section 3.1**, the simulated hybrid classical-molecular computer is capable of solving Ising models with four-body interactions, whereas any purely molecular computer will only have local connections that can encode up to two-body interactions. There, the four-body Ising model of the protein was converted to a two-body Ising model, which requires the introduction of further bits ("reduction bits"). This procedure is described in Ref. (42). The results of annealing simulations of the lattice protein model are shown in **Figure 3c**. Selected trajectories that ended at low energy conformations show the energy as a function of the step number. Formation of native contacts, as well as annealing of the ancillary bits, reduce the total energy of the system.

The final states shown in **Figure 3c** are local optima of the Hamiltonian. The barriers between these local optima are considerably higher than the energy differences between the local optima (further details are provided in the SI), which makes it impossible for a (stochastic) gradient descent algorithm to overcome the barrier between local and global optima. To find the global optimum and, thus, the ground state of the protein, it is necessary to run many instances of the molecular computer. No speed-up can be expected compared to an algorithm that searches through all 2^K states of the protein, with K being the number of bits that encode the physical conformation of the protein ($K = 7$ in our example).

4. Future: purely molecular computer

Ultimately, we envisage a purely molecular version of the droplet-array computer that anneals to a ground state configuration solely by the physical and chemical interactions between droplets. In the absence of an external classical computer to enforce droplet-droplet couplings, the problem must be pre-programmed into the contents of the droplets and the interactions of neighbouring droplets. These programmable couplings can take place via mass exchange (diffusion, biological membranes) or energy exchange (excitons, Förster resonance energy transfer (FRET) (44)).

Such a system benefits from the complete parallelization of the problem. For a QUBO problem, a hybrid computer still requires that, at each step of the optimization, a classical computer calculates n^2 terms representing the pairwise energies of the n variables, and even more for problems with higher-order terms. A purely molecular computer would implement these couplings n times faster since all pairwise interactions occur simultaneously, in $O(n)$ time.

Rather than using a classical computer to perform stochastic gradient descent, as we did in the hybrid version of the device, the system is driven, kinetically and thermodynamically, to its ground state. The key is to select the appropriate reactions and chemicals. A few proposed mechanisms are detailed in **Section 5**.

While the hybrid computer does not make use of the spatial arrangement of the droplets, in the purely molecular version, each droplet is connected only to those droplets with which it can physically interact. A problem must therefore be mapped to this graph using minor-embedding techniques to address the sparse connectivity (45, 46).

Additionally, this device can only implement two-body terms, which means auxiliary ancilla droplets are needed for reducing *k*-local problems (PUBO) to a quadratic expression (QUBO). Such algorithms are commonly used to map problems to various quantum devices such as the D-WAVE quantum annealer and the Rigetti quantum computer (40, 47).

$$H = c + \sum_{i}^{N} h_i s_i + \sum_{i<j}^{N} J_{ij} s_i s_j$$

In the following section, we discuss practical considerations for building a physical device.

5. Implementation

Given the requirements above for the hybrid and purely molecular computers, numerous chemical systems can be considered as viable candidates. A physical system capable of implementing either the hybrid classical-molecular computer or the purely molecular computer should have properties that can be precisely defined and measured. Since each droplet will be subjected to changing constraints from neighbouring droplets (two-body terms, $h^{(2)}$) as the computation progresses, changes in these properties must be reversible.

Measurable properties of a droplet can be mapped onto a binary variable, for example, by using a threshold value. **Figure 4** gives two examples of physical properties that can be used to represent a droplet as a binary variable, namely polymer molecular weight distribution and pH. Other candidate properties include the concentration of a reagent, oxidation states, and colour. The state of each droplet can be read out in a non-interfering manner via optical or electrochemical means. For example, pH-sensitive dyes or ion indicators can be used to read out a fluorescence signal proportional to the pH or ion concentration in each droplet in the array (48). Functionalized electrode arrays can be used to address individual droplets and determine their pH. Fluorescent dyes sensitive to gelation or viscosity changes can be used to indicate the presence of polymer (49).

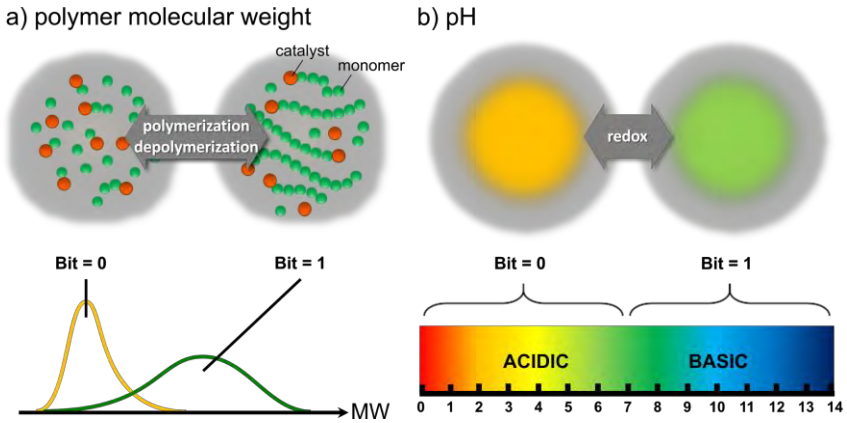

Figure 4. *Two approaches to representing a droplet as a binary variable: in a), polymerization and depolymerization giving different molecular weight distribution, and in b), redox reactions altering the pH of the droplet.*

One of the most versatile ways to form the basis of computing bits for the hybrid classical-molecular computer is using the pH of a droplet, by converting the logarithmic pH scale to a binary variable. The pH of a droplet can be reversibly adjusted using several chemical reactions, such as the addition of acid or base or by tuning electric potentials to drive redox reactions that generate or consume H^+.

In the hybrid classical-molecular computer, the inter-droplet couplings, whose nature and strength correspond to the problem being optimized, are enforced by an external control algorithm. In the purely molecular computer, these couplings should be pre-programmed into the inter- and intra-

droplet interactions. Information between neighbouring droplets can be shared via mass transfer or energy transfer. The former relies on the movement of molecules across the droplet membranes, which can take place through passive diffusion across the interface, by electrophoretic motion, or via pores installed in the membranes to enhance the rates of transport between droplets (50, 51). The latter energy transfer-based system could eliminate these considerations by allowing for inter-droplet communication without mass transfer.

To execute either version of the molecular computer, we must achieve precise control over droplet placement. Microfluidic-based robotic platforms, as well as inkjet printers, can be used to generate droplets with precise compositions and to place them in square or hexagonal packed arrays. In this way, the droplets can be situated on electrode arrays for external control of droplet couplings or the readout of droplet states. The states of the droplets can be determined via optical or electrochemical readout using the means mentioned above. For the hybrid classical-molecular computer, this readout can be sent to a classical computer that can implement the necessary couplings. The classical computer can manipulate the droplets through external stimuli (optical and electrochemical control) based on the set couplings. This control loop would continue until the problem reaches a minimum energy state. For the purely molecular computer, the couplings are pre-programmed into the droplet interactions, removing the need for a classical computer to impose these couplings.

6. Discussion

6.1 Problem scaling analysis of molecular computer

An essential factor in the choice of problems to solve on the microdroplet-array computer is how the number of droplets needed scales with the number of variables in the problem. Compared to quantum annealers, which are constructed using Josephson junctions cooled to near absolute zero temperature, the cost of adding droplets to the molecular computer is negligible compared to the overall cost of the setup.

Since the molecular computer takes Ising parameters as input, any problem that can be efficiently encoded in a spin Ising system can also be efficiently implemented on a microdroplet array computer. As far as we know, the parallelization of chemical reactions across all droplets could give the molecular computer an advantage over numerical gradient evaluations in a classical computer or GPU.

In **Figure 1c** and **1d** we plotted the scaling of four types of problems that can be represented by an Ising Hamiltonian and solved on either the hybrid or the purely molecular computer (36). They include the two applications given in the main text, boolean satisfiability and lattice protein folding, as well as the traveling salesperson (e.g. searching for the shortest path between a set of cities.)

Recall that the hybrid version of the molecular computer is fully connected and allows for k-local terms (higher-order expressions involving k variables). The purely molecular computer, on the other hand, is limited to 2-local, nearest-neighbour connections. Therefore the scaling formulas account for ancilla bits used in embedding and locality reduction algorithms.

On the hybrid computer, boolean satisfiability for *k* variables per clause (k-SAT) scales as $O(n)$ since each variable corresponds to one droplet. For three or more clauses, the problem is NP-complete. To reduce a 3-SAT problem to an Ising Hamiltonian (2-local), we need $(n-1)^2/4$ ancilla droplets (40). The traveling salesperson is a significant problem for logistical optimization. Given n nodes ("cities"), we require n^2 droplets to account for the sequence as well as labels of the cities visited, so the number of droplets scales as $O(n^2)$ (36). The mapping involves only quadratic terms (2-local), so it can be directly implemented on the purely molecular computer. Lastly, the lattice protein problem is the most complex since the Hamiltonian must encode self-avoidance constraints as well as interactions between neighbouring amino acids (42, 43). It scales as $n^2 log(n)$. We note that the scaling of the described problems is at most polynomial in the length of the input.

It has been shown that embedding QUBO problems on a sparsely connected graph scales linearly with the number of variables in the problem (47).

6.2 Advantages of the molecular computer

With a fully-realized molecular computer, there will be many advantages compared to conventional silicon-based computers, as well as compared to many other recent proposed computing architectures. If some of the proposed mechanisms in previous sections could be realized, we believe that a molecular computer to be particularly well-suited to solving combinatorial optimization problems, including some NP-problems.

Comparison to classical computation: Since the rules of classical physics still govern the operation of a molecular computer, we do not believe it will be more efficient than classical conventional computers in terms of time or space scaling. There are nevertheless many advantages to using chemical reactions for solving specific types of problems such as discrete optimization. First, this approach removes the need for physically defined, atomic-sized circuits. The molecular computer benefits from the intrinsic parallelization of chemical reactions, effectively solving a system of differential equations of motion in parallel and remove the need for memory to store each step of the optimization process. The computation speed is, however, limited by the intrinsic speed of droplet motions and chemical reactions, nuclear rearrangements being orders of magnitude slower than electronic rearrangements in the Born-Oppenheimer approximation.

Comparison to biomolecular computation: The droplet-array molecular computer also has advantages over biomolecular methods, such as DNA computing and cellular signaling, with faster operation speed, fewer errors, and significantly less complexity than biological cells. There is no need for post-processing or offline analysis of the chemical content, allowing for immediate readout. Droplet-array initialization, computation through annealing, and optical readout are all fast and automatable processes. The phase space of chemical reactions, mechanisms, reagents, is extensive. The contents of each droplet can be cost-effective, widely available, and safe materials.

Comparison to quantum computation: In recent years, quantum computing has gained much attention due to the potential for solving problems that are intractable for classical Turing machines. These include encryption (prime factorization), quantum chemistry, as well as the optimization problems described above. However, with quantum entanglement and superposition come challenges in error correction, noise and decoherence, and qubit scaling, which must be overcome before quantum computers attain their full potential. The droplet-array, on the other hand, is easily scalable through inkjet printing. The cost of adding additional droplets, representing bits, is negligible compared to the cost of the overall experimental setup (38). As well, there is little experimental overhead to operate the molecular computer, since we can select for chemical reactions that take place at or near room temperature. Information storage would also be trivial since the output of a calculation can be printed onto a piece of paper for future readout.

7. Conclusion

In sum, we have proposed a new heuristic method for computation with programmable droplet-arrays to solve combinatorial optimization problems. The device consists of a two-dimensional array of microdroplets that represents a set of interacting binary variables that evolve under an Ising Hamiltonian. Each droplet corresponds to one variable, whose value is determined by measuring a specific property of the droplet. A specific problem is solved by programming the intra-droplet contents and inter-droplet interactions. As the system evolves collectively, the droplet states approach the optimal solution of the problem through a process akin to annealing in materials.

D-WAVE first adopted a quantum version of this approach; more recently, a classical digital annealer by Fujitsu. To our knowledge, this is the first proposal of a molecular computer operating in annealing mode. In its purely molecular version, the microdroplet-array computer benefits from all of the advantages of computing with molecules: concurrent information processing and storage, massive parallelization of chemical reactions, energy-efficient processes, vast phase space of molecules and reactions, cost-effectiveness, and scalability of the device.

As a stepping stone to a purely molecular computer, we first developed a hybrid model where a classical computer imposes the parameters of the optimization problem, and the information processing and storage is carried out by the individual droplets in the array. Throughout numerous iterations, the classical computer takes in the set of droplet states and performs a stochastic gradient descent algorithm to search for the optimal ground state using the droplets. A simulation of the hybrid classical-molecular computer demonstrated its ability to solve two NP-hard problems, reproducing the phase transition in boolean satisfiability (3-SAT) as a function of clause density, and identifying the ground state configuration in a lattice protein folding problem.

Our next step is to perform these calculations on a physical hybrid classical-molecular computer. The challenge of identifying suitable chemical reagents and reactions to program the microdroplet-array can be facilitated with machine learning and high throughput experimentation. We would also employ robotics and computer vision to operate the device. In due course, we shall tackle droplet miniaturization to scale up to thousands and eventually millions of droplets. In

the ultimate limit of this technology, we imagine a molecular computer operated by inkjet printing a problem onto a sheet of paper; by the time the ink dries, the problem is solved and imprinted onto the sheet. Applying the same concept to a 3D printed ink may open a novel route for manufacturing large scale assemblies of matter with nanoscopic precision.

Acknowledgments

We are grateful to the Defense Advanced Research Projects Agency (DARPA) for funding this project under award number W911NF-18-2-0036 from the Molecular Informatics program. P.F. acknowledges funding by the European Union's Horizon 2020 research and innovation program under the Marie Skłodowska-Curie grant agreement no. 795206 (MolDesign). L.C. thanks the ERC for an Advanced Grant (Smart-POM) and the EPSRC for funding grant EP/R009902/1. Computations were performed on the Niagara supercomputer at the SciNet HPC Consortium. SciNet is funded by: the Canada Foundation for Innovation; the Government of Ontario; Ontario Research Fund - Research Excellence; and the University of Toronto. A.A.-G. acknowledges the generous support from Anders G. Frøseth.

References

1. V. V. Zhirnov, R. K. Cavin, J. A. Hutchby, G. I. Bourianoff, Limits to binary logic switch scaling-a gedanken model. *Proc. IEEE* **9**, 1934–1939 (2003).

2. Y. Cao, *et al.*, Quantum Chemistry in the Age of Quantum Computing (2018) (March 27, 2019).

3. T. D. Ladd, *et al.*, Quantum computers. *Nature* **464**, 45–53 (2010).

4. M. Conrad, On design principles for a molecular computer. *Communications of the ACM* **28**, 464–480 (1985).

5. M. M. Waldrop, M. Mitchell Waldrop, The chips are down for Moore's law. *Nature* **530**, 144–147 (2016).

6. G. Rozenberg, T. Bäck, J. N. Kok, *Handbook of Natural Computing* (Springer, 2012).

7. L. Cronin, *et al.*, A programmable chemical computer with memory and pattern recognition (2019) https:/doi.org/10.26434/chemrxiv.7712564.v1 (October 28, 2019).

8. J. Holley, I. Jahan, B. De Lacy Costello, L. Bull, A. Adamatzky, Logical and arithmetic circuits in Belousov-Zhabotinsky encapsulated disks. *Physical Review E* **84** (2011).

9. B. Zhang, *et al.*, Redox gated polymer memristive processing memory unit. *Nature Communications* **10** (2019).

10. V. Galstyan, L. Funk, T. Einav, R. Phillips, Combinatorial Control through Allostery. *J. Phys. Chem. B* **123**, 2792–2800 (2019).

11. N. E. Buchler, U. Gerland, T. Hwa, On schemes of combinatorial transcription logic. *Proc. Natl. Acad. Sci. U. S. A.* **100**, 5136–5141 (2003).

12. E. Katz, Ed., *Molecular and Supramolecular Information Processing* (Wiley-VCH Verlag GmbH & Co. KGaA, 2012).

13. E. Katz, Ed., *Biomolecular Information Processing* (Wiley-VCH Verlag GmbH & Co. KGaA, 2012).

14. D. T. Chiu, E. Pezzoli, H. Wu, A. D. Stroock, G. M. Whitesides, Using three-dimensional microfluidic networks for solving computationally hard problems. *Proc. Natl. Acad. Sci. U. S. A.* **98**, 2961–2966 (2001).

15. D. V. Nicolau Jr, *et al.*, Parallel computation with molecular-motor-propelled agents in nanofabricated networks. *Proc. Natl. Acad. Sci. U. S. A.* **113**, 2591–2596 (2016).

16. J. Einarsson, New biological device not faster than regular computer. *Proc. Natl. Acad. Sci. U. S. A.* **113**, E3187 (2016).

17. D. V. Nicolau Jr, *et al.*, Reply to Einarsson: The computational power of parallel network exploration with many bioagents. *Proc. Natl. Acad. Sci. U. S. A.* **113**, E3188 (2016).

18. L. Zhu, S.-J. Kim, M. Hara, M. Aono, Remarkable problem-solving ability of unicellular amoeboid organism and its mechanism. *R Soc Open Sci* **5**, 180396 (2018).

19. D. Woods, *et al.*, Diverse and robust molecular algorithms using reprogrammable DNA self-assembly. *Nature* **567**, 366–372 (2019).

20. L. Adleman, Molecular computation of solutions to combinatorial problems. *Science* **266**, 1021–1024 (1994).

21. M. G. T. A. Rutten, Martin G T, F. W. Vaandrager, Johannes A A, R. J. M. Nolte, Encoding information into polymers. *Nature Reviews Chemistry* **2**, 365–381 (2018).

22. S. Kirkpatrick, C. D. Gelatt Jr, M. P. Vecchi, Optimization by simulated annealing. *Science* **220**, 671–680 (1983).

23. E. Farhi, *et al.*, A quantum adiabatic evolution algorithm applied to random instances of an NP-complete problem. *Science* **292**, 472–475 (2001).

24. G. E. Santoro, R. Martonák, E. Tosatti, R. Car, Theory of quantum annealing of an Ising spin glass. *Science* **295**, 2427–2430 (2002).

25. M. W. Johnson, *et al.*, Quantum annealing with manufactured spins. *Nature* **473**, 194–198 (2011).

26. S. Mandrà, H. G. Katzgraber, A deceptive step towards quantum speedup detection (2017) https:/doi.org/10.1088/2058-9565/aac8b2 (August 21, 2019).

27. Boothby K, Bunyk P, Raymond J, Roy A, “Next-generation topology of D-wave quantum processors” (D-wave Technical Report, 2019).

28. P. L. McMahon, *et al.*, A fully programmable 100-spin coherent Ising machine with all-to-all connections. *Science* **354**, 614–617 (2016).

29. T. Inagaki, *et al.*, A coherent Ising machine for 2000-node optimization problems. *Science* **354**, 603–606 (2016).

30. M. Aramon, *et al.*, Physics-Inspired Optimization for Quadratic Unconstrained Problems Using a Digital Annealer (2018) (March 28, 2019).

31. H. Goto, K. Tatsumura, A. R. Dixon, Combinatorial optimization by simulating adiabatic bifurcations in nonlinear Hamiltonian systems. *Sci. Adv.* **5**, eaav2372 (2019).

32. V. Bapst, L. Foini, F. Krzakala, G. Semerjian, F. Zamponi, The quantum adiabatic algorithm applied to random optimization problems: The quantum spin glass perspective. *Physics Reports* **523**, 127–205 (2013).

33. F. Barahona, On the computational complexity of Ising spin glass models. *J. Phys. A Math. Gen.* **15**, 3241–3253 (1982).

34. R. M. Karp, “Reducibility among Combinatorial Problems” in *Complexity of Computer Computations*, (1972), pp. 85–103.

35. B. Hayes, Can’t Get No Satisfaction. *Am. Sci.* **85**, 108–112 (1997).

36. A. Lucas, Ising formulations of many NP problems. *Frontiers in Physics* **2** (2014).

37. S. Istrail, Statistical mechanics, three-dimensionality and NP-completeness: I. Universality of intracatability for the partition function of the Ising model across non-planar surfaces (extended abstract) in *Proceedings of the Thirty-Second Annual ACM Symposium on Theory of Computing - STOC '00*, (ACM Press, 2000), pp. 87–96.

38. W. Feng, E. Ueda, P. A. Levkin, Droplet Microarrays: From Surface Patterning to High-Throughput Applications. *Advanced Materials* **30**, 1706111 (2018).

39. S. A. Cook, The complexity of theorem-proving procedures in *Proceedings of the Third Annual ACM Symposium on Theory of Computing - STOC '71*, (1971) https:/doi.org/10.1145/800157.805047.

40. R. Babbush, B. O'Gorman, A. Aspuru-Guzik, Resource efficient gadgets for compiling adiabatic quantum optimization problems. *Ann. Phys.* **525**, 877–888 (2013).

41. M. Davis, G. Logemann, D. Loveland, A Machine Program for Theorem-proving. *Commun. ACM* **5**, 394–397 (1962).

42. A. Perdomo-Ortiz, N. Dickson, M. Drew-Brook, G. Rose, A. Aspuru-Guzik, Finding low-energy conformations of lattice protein models by quantum annealing. *Sci. Rep.* **2** (2012).

43. R. Babbush, A. Perdomo-Ortiz, B. O'Gorman, W. Macready, A. Aspuru-Guzik, "Construction of Energy Functions for Lattice Heteropolymer Models: Efficient Encodings for Constraint Satisfaction Programming and Quantum Annealing" in *Advances in Chemical Physics*, (2014), pp. 201–244.

44. N. P. D. Sawaya, D. Rappoport, D. P. Tabor, A. Aspuru-Guzik, Excitonics: A Set of Gates for Molecular Exciton Processing and Signaling. *ACS Nano* **12**, 6410–6420 (2018).

45. V. Choi, Minor-embedding in adiabatic quantum computation: I. The parameter setting problem. *Quantum Information Processing* **7**, 193–209 (2008).

46. V. Choi, Minor-embedding in adiabatic quantum computation: II. Minor-universal graph design. *Quantum Information Processing* **10**, 343–353 (2011).

47. A. Perdomo-Ortiz, *et al.*, On the readiness of quantum optimization machines for industrial applications (2017) (March 13, 2019).

48. J. Han, K. Burgess, Fluorescent indicators for intracellular pH. *Chem. Rev.* **110**, 2709–2728 (2010).

49. P. Bosch, F. Catalina, T. Corrales, C. Peinado, Fluorescent probes for sensing processes in polymers. *Chemistry* **11**, 4314–4325 (2005).

50. Y. Elani, R. V. Law, O. Ces, Vesicle-based artificial cells as chemical microreactors with spatially segregated reaction pathways. *Nat. Commun.* **5**, 5305 (2014).

51. A. Huebner, *et al.*, Microdroplets: a sea of applications? *Lab Chip* **8**, 1244–1254 (2008).

Supplementary Information for

A molecular computing approach to solving optimization problems via programmable microdroplet arrays

Si Yue Guo, Pascal Friederich, Yudong Cao, Tony Wu, Christopher J Forman, Douglas Mendoza, Matthias Degroote, Andrew Cavell, Veronica Krasecki, Riley J Hickman, Abhishek Sharma, Leroy Cronin*, Nathan Gianneschi, Randall H Goldsmith, Alán Aspuru-Guzik*

Corresponding authors: Alán Aspuru-Guzik, Leroy Cronin
Emails: aspuru@utoronto.ca; Lee.Cronin@glasgow.ac.uk

This PDF file includes:

A. Boolean Satisfiability: Conjunctive Normal Formula to Ising Hamiltonian

A k-SAT Boolean formula of N variables consists of M clauses of k literals each. An example of a 3 variable 2-SAT in conjunctive normal form (CNF) is:

$$(\ x \text{ or } y\ ) \text{ and } (\ y \text{ and not } z\ )$$

To convert this expression into an Ising Hamiltonian, we represent each Boolean variable as a spin, $s_i \in \{-1, +1\}, i = 1, \dots, N$.

First, define a clause matrix:

$$\begin{cases} w_{ji} = 1\text{ , if clause } j \text{ includes } s_i \\ w_{ji} = -1\text{ , if clause } j \text{ includes not } s_i \end{cases}$$

Then each clause indicator:

$$v_j = \frac{1}{2^k}\prod_{i=1}^{N}(1 - w_{ji}s_i)$$

The overall Hamiltonian becomes:

$$H_{k-SAT} = \sum_{j=1}^{M} v_j = \frac{1}{2^k}\sum_{j=1}^{M}\prod_{i=1}^{N}(1 - w_{ji}s_i)$$

When the CNF is satisfied, $H_{k-SAT} = 0$. For $H_{k-SAT} > 0$, the value of H_{k-SAT} gives the number of violated clauses.

To use binary variables $x_i \in \{0,1\}$ instead of spin variables, we can use a simple transformation (1-2):

$$s_i = 2x_i - 1 \text{ or } x_i = \frac{1}{2}(s_i + 1)$$

B. Example 1: a four-droplet system

To illustrate how the molecular computer works, consider the following satisfiability problem of five clauses, each with two variables per clause (2-SAT). Although the 2-SAT problem is in P and not NP, it is the simplest and easiest to illustrate how the device operates.

In conjunctive normal form:

$$(s_1 \text{ or } s_2) \text{ and } (s_2 \text{ or not } s_3) \text{ and } (s_3 \text{ or } s_4) \text{ and } (\text{not } s_4 \text{ or } s_1) \text{ and } (\text{not } s_2 \text{ or not } s_4)$$

$$\text{or } (s_1 \vee s_2) \wedge (s_2 \vee \, !s_3) \wedge (s_3 \vee s_4) \wedge (!s_4 \vee s_1) \wedge (!s_2 \vee !s_4)$$

Figure S1 shows how this calculation would be implemented in a four-droplet system, using our simulation of a hybrid classical-molecular computer.

The problem is first mapped to an Ising Hamiltonian, according to the procedure described in **Section A** above.

$$\boldsymbol{H = h^{(0)} + \sum_i h_i^{(1)} s_i + \sum_{(i,j)} h_{ij}^{(2)} s_i s_j}$$

$$h^{(0)} = \frac{5}{4}$$

$$h^{(1)} = [-2 \quad -1 \quad 0 \quad 1]$$

$$h^{(2)} = \begin{bmatrix} 0 & 1 & 0 & -1 \\ 1 & 0 & -1 & 1 \\ 0 & -1 & 0 & 1 \\ -1 & 1 & 1 & 0 \end{bmatrix}$$

These arrays are entered into the control algorithm of the computer software. To run a calculation, we first randomly initialize each droplet state (Figure S1 panel b)). The control algorithm takes in the droplet states (panel d)), then computes the gradients on each droplet based on the Hamiltonian above (panel e)). These gradients are delivered to the droplets via changes in electric potential, for example. Over a number of iterations, the system reaches a minimum energy state and stops when the gradients reach zero. The final state is rounded to the nearest integer value, -1 or $+1$ and the result is read out, after a change of basis from ± 1 to $0{,}1$ (panel d)). In this case, a solution of $s_1 = 1,\ s_2 = 0,\ s_3 = 0,\ s_4 = 1$ satisfies all of the clauses in the formula given above.

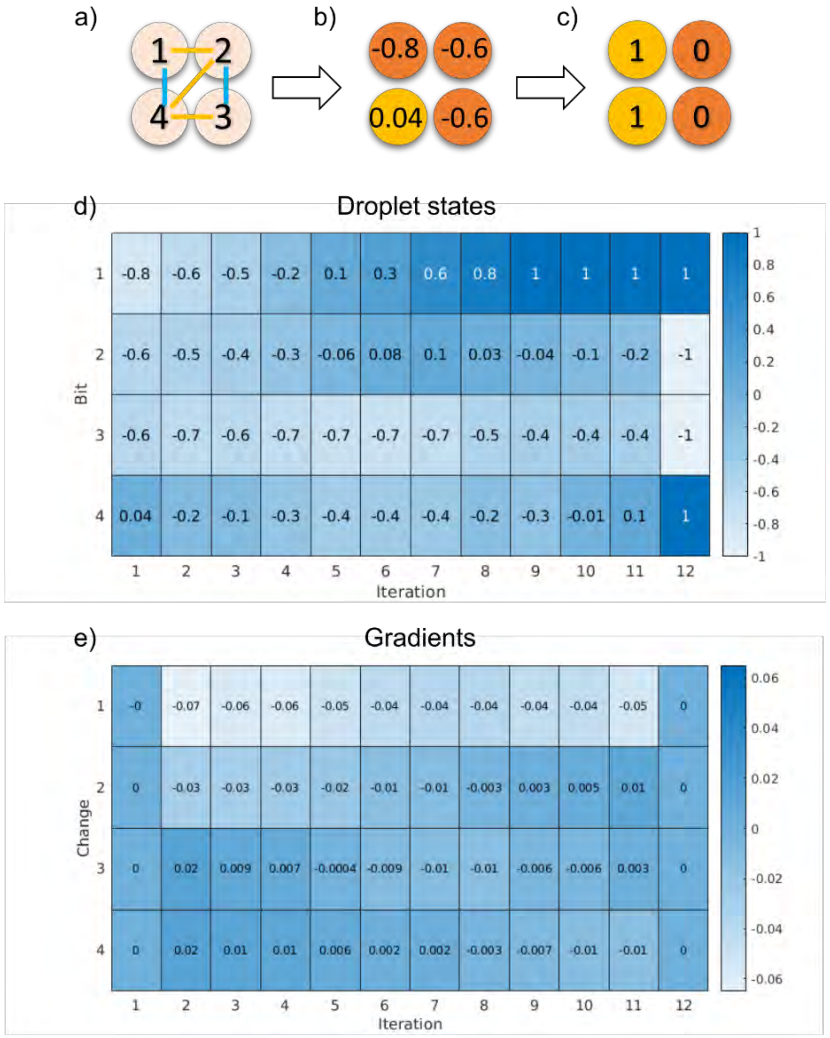

Figure S1 - Example of a 2-SAT problem of 4 variables with 5 clauses. In a) the clauses are indicated by the lines connecting the droplets (yellow for "positive" coupling, blue for "negative"). Panel b) gives the random initialization of the problem, giving each droplet an initial value between -1 and +1. The steps of the optimization process are simulated in the lower two panels: at each step, the droplet states are measured, and shown in d), and in e) the gradients calculated based on the Ising Hamiltonian. Once the system has reached a minimum, the final droplet states are mapped to a bit value of 0 or 1. ($s_i \in [-1,0) \rightarrow x_i = 0,\ s_i \in (0,1] \rightarrow x_i = 1$). One solution to the CNF is shown in panel c).

Clause	$(s_1 \mathrm{V}\ s_2)$	$(s_2 \mathrm{V}\ !\, s_3)$	$(s_3 \mathrm{V}\ s_4)$	$(!\, s_4 \mathrm{V}\ s_1)$	$(!\, s_2 \mathrm{V}\ !\, s_4)$
Check solution	1 or 0	0 or ! 0	0 or 1	! 1 or 1	! 0 or ! 1
	True	True	True	True	True

C. Barriers in the lattice protein model

In order to investigate whether or not a (stochastic) gradient descent model will be able to escape local minima, we analysed the energy landscape between local minima using constrained optimizations at states between the optima. In particular, we interpolated between the ground state conformation with an energy of -6 (lower left) and a non-clashing, unfolded conformation with an energy of 0 (upper left). The energy barrier on a linear interpolation between the two states that only differ by one bit (Bit #7) is higher than 90, which is considerably more than the differences between all valid conformations (E = -6 to E = 0). Interpolations of other bitflips show the same results (3).

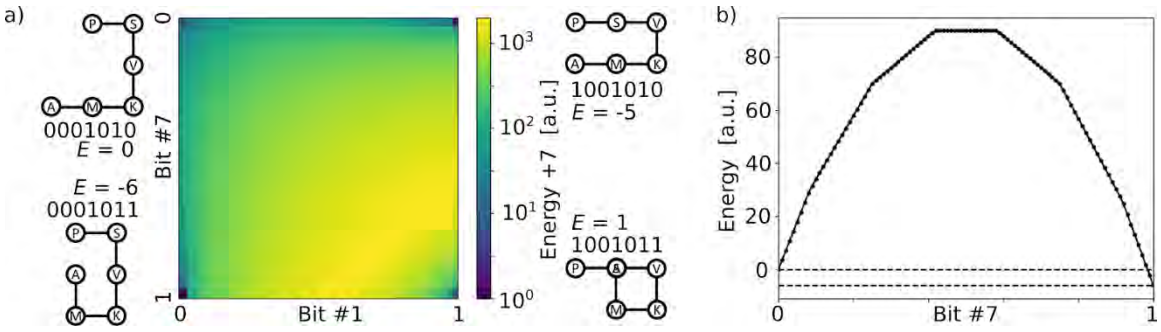

Figure S2 - a) Energy landscape between four configurations of the lattice protein. The energy differences at the edges (shown next to the heatmap) are considerably lower than the barrier height between them. b) The energy of the system on a line between the unfolded state 0001010 and the folded state 0001011 (Bit #1 is 0).

D. Example 2: Mapping bits to a hybrid and a purely-molecular computer for 9 variables

Here we give an example of how a Boolean satisfiability problem can be mapped to the hybrid computer and the purely-molecular computer implemented on two lattices. For the purposes of easily illustrating the relation between the variables, we will use a 2-SAT problem (9 variables, 17 clauses). The CNF is:

$$(1 \vee 4) \wedge (!2 \vee 5) \wedge (!3 \vee !9) \wedge (!2 \vee 4) \wedge (!3 \vee 5) \wedge (6 \vee !9) \wedge (!3 \vee 7) \wedge (!4 \vee 8) \wedge (2 \vee 9) \wedge (7 \vee !8) \wedge (!2 \vee !6) \wedge (1 \vee 6) \wedge (!4 \vee 5) \wedge (2 \vee 5) \wedge (!4 \vee 9) \wedge (3 \vee 4) \wedge (!2 \vee 3)$$

which can be represented as an Ising Hamiltonian with the bias, one-body, and two-body term coefficients:

$$H = h^{(0)} + \sum_i h_i^{(1)} s_i + \sum_{(i,j)} h_{ij}^{(2)} s_i s_j$$

$$h^{(0)} = \frac{9}{2}$$

$$h^{(1)} = \frac{1}{4}[-2 \quad 3 \quad 1 \quad 0 \quad -4 \quad 0 \quad -2 \quad 0 \quad 0]$$

$$h^{(2)} = \frac{1}{8}\begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & -1 & -1 & 0 & 1 & 0 & 0 & 1 \\ 0 & -1 & 0 & 1 & -1 & 0 & -1 & 0 & 1 \\ 1 & -1 & 1 & 0 & -1 & 0 & 0 & -1 & -1 \\ 0 & 0 & -1 & -1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & -1 \\ 0 & 0 & -1 & 0 & 0 & 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & -1 & 0 & 0 & -1 & 0 & 0 \\ 0 & 1 & 1 & -1 & 0 & -1 & 0 & 0 & 0 \end{bmatrix}$$

In Figure S3 we first represent this 2-SAT problem first as a graph and then show how to physically place this problem on a molecular computer (hybrid and purely-chemical versions). Since the hybrid classical-molecular computer is fully connected and relies on a classical computer to enforce the inter-droplet couplings, the droplets can be placed anywhere, in any order. For the purely-molecular computer, depending on the lattice, square or hexagonal, each droplet can be coupled to four or six nearest-neighbour

droplets. To overcome the spatial and connectivity constraints, we require minor embedding techniques and ancilla bits. For example, in Figure S3 c) the square lattice has an extra droplet for bits 3, 4, and 6. Although this invariably increases the size of the system, it ensures that all droplets can be pairwise coupled (4).

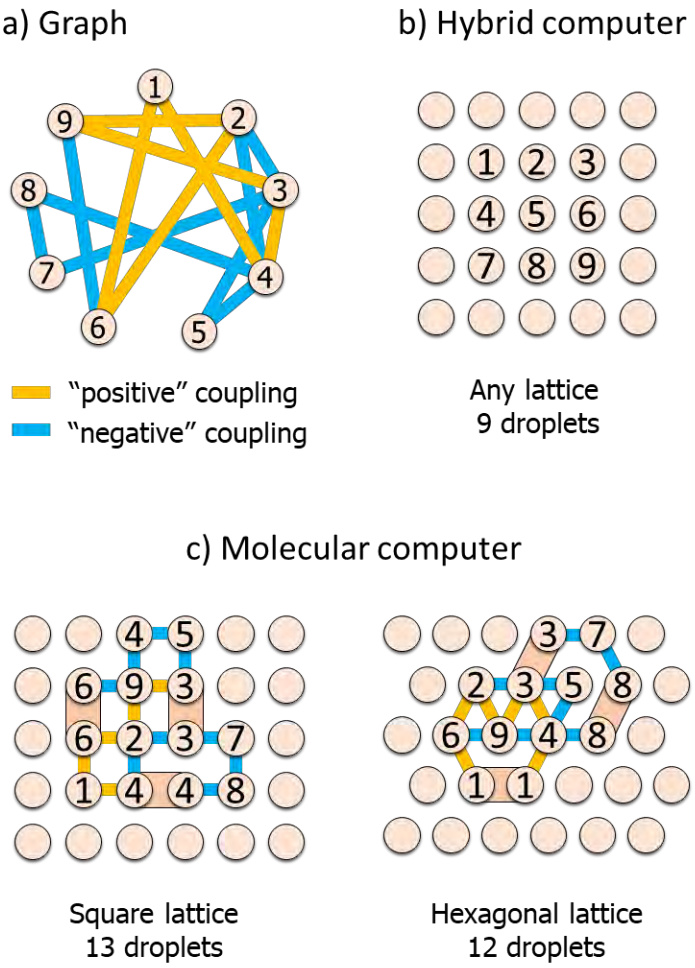

Figure S3 - Mapping a 9-variable 2-SAT problem onto a hybrid and purely-molecular computer. a) graph representing the problem, with "positive" couplings in yellow and "negative" couplings in blue, and its mapping to droplets on b) a hybrid computer, c) a purely-molecular computer on a square and hexagonal lattice.

References

1. A. Lucas, “Ising formulations of many NP problems”. Frontiers in Physics 2 (2014).
2. Z. Bian, F. Chudak, W.G. Macready, G. Rose “The Ising model: teaching an old problem new tricks” available at https://www.dwavesys.com/sites/default/files/weightedmaxsat_v2.pdf
3. R. Babbush, A. Perdomo-Ortiz, B. O’Gorman, W. Macready, A. Aspuru-Guzik, “Construction of Energy Functions for Lattice Heteropolymer Models: Efficient Encodings for Constraint Satisfaction Programming and Quantum Annealing” in Advances in Chemical Physics, (2014), pp. 201–244.
4. V. Choi, “Minor-embedding in adiabatic quantum computation: I. The parameter setting problem.” Quantum Information Processing 7, 193–209 (2008)