SEVENTH FRAMEWORK PROGRAMME

Project no. 248992

Project acronym: NEUNEU

Project title: Artificial Wet Neuronal Networks from Compartmentalised Excitable Chemical Media

Small or medium-scale focused research project (STREP)

Deliverable 4.6 - Designs of sequential logical circuits evolved. Report or publication on algorithms

The deliverable consists of three parts: Firstly, an extended abstract submitted to the ECAL 2013 conference, summing up the architectural concepts of the NEUNEU project. Secondly, a report titled "Using Re-Entrant Networks of Repeated Units to Facilitate the Automated Design of Logic Gates in Unconventional Media: A New Method" on the novel technique for the automatic design of droplet- and other unconventional computers, combining evolution and self-organisation. And a third document, demonstrating the applicability of this approach for droplet networks. The meterial is intended for publication in a scientific journal.

Start date of project: 1.2.2010	Duration: 36 months

Project coordinator name:	Dr. Peter Dittrich
Project coordinator organisation name:	Friedrich Schiller University Jena

Synthetic signalling protocell networks as models of neural computation

Matthew D. Egbert[1,2,*], Gerd Grünert[1], Gabi Escuela[1] and Peter Dittrich[1]

[1] Biosystems Analysis Group, Friedrich Schiller University, Jena, Germany
[2] Center for Computational Neuroscience and Robotics, University of Sussex, Brighton, U.K.
[*] mde@matthewegbert.com

Introduction

Modern conventional computers are programmable, predictable and relatively easy to understand and engineered —at least compared to most complex non-linear systems. These properties are the result of various dynamical constraints that are universal to conventional computers, such as the clock mechanism that synchronises the update of logic gates and other components; the ubiquitous discretization steps (where continuous values are discretized into binary 1s and 0s); and the almost complete isolation of internal processes of computers from the environment of the computer. We are investigating an alternative computational medium composed of signalling synthetic protocells to explore the implications of relaxing some of these dynamical constraints that are typical of conventional computers. Is it possible to build useful and/or programmable computers out of unconventional media such as protocells that do not have a synchronizing clock? Or that do not employ a conventional representation of 0s and 1s? Or that are less decoupled from their environment?

The protocells that we are investigating are aqueous droplets suspended in oil. Each droplet contains the reagents for the Belousov-Zhabotinsky (BZ) oscillating chemical reaction (Zhabotinsky, 2007), resulting in self-exciting dynamical units that, when in contact with each other, are capable of propagating signals similar in some respects to signal transduction in biological neurons. Networks of these signalling protocells are therefore a kind of *wet* artificial neural network, sharing more in common with biological nervous tissue than conventional computer electronics.

It is envisaged that in the future more advanced protocells will be employed to make self-organising computers, or computers that can operate within the human body. But first it is necessary to develop a better understanding of how complex non-linear systems can be harnessed to accomplish useful or "minimally-cognitive" tasks (Beer, 2003) such as categorical perception, boolean logic, and dynamical control.

Moreover, by learning how to construct or assemble networks of complex non-linear units like the BZ-protocells we also gain insight into how other complex and non-linear "computational" media (such as nervous tissue) can conduct, modify and modulate signals and information, and how it can play an important role in the sensorimotor loops of a situated and embodied agent (Stewart et al., 2011). This bottom-up approach to the construction of alternative computational media is an important complement to the more widespread top-down neuroscience where biological neural networks are slowly being reverse engineered.

With these long and medium-term goals in mind, we have set out to (i) design functional collections of signalling protocells (comparable to the logic gates of conventional computing) that could be combined to produce more complex networks, (ii) identify effective signal encoding(s) that facilitate the transmission and manipulation of the signal by protocell networks, and (iii) identify design techniques and methodologies for creating functional signalling protocell networks out of complex non-linear media. To accomplish these goals, we are taking a three pronged approach involving *in vitro* experimentation, simulation and modelling to investigate the dynamical properties of the protocells and networks thereof; and experimental computer-aided design and machine-learning techniques to partially automate the development of functional protocell networks. We now briefly summarize our published results, before describing our current efforts.

Summary of published research

To elucidate the experimental foundations of working with wet chemical computers on microfluidic chips (King et al., 2012), the NeuNeu project consortium (www.neu-n.eu) has conducted various research projects involving simulation, modelling and experimentation. One branch of this research involves the investigation of droplet networks, where the droplets are assumed to be small enough that internal spatial dynamics can be ignored. In this vein, the computing potential of two-droplet systems has been demonstrated in experiment and simulation (Szymanski et al., 2011) and differential equation models have been identified that allow us to accurately describe droplet dynamics and interactions (Szymanski et al., 2011). More abstract simulation models have also been developed to make possible faster and larger-scale simulations (Gruenert et al., 2013), allowing us to analyse higher-level design principles and questions pertaining to system architecture, such as possible benefits of moving beyond naive or simple signal encodings (e. g. high firing-rate = 1, and low firing-rate = 0) to explore various alternatives (Gruenert et al., 2012).

In a parallel branch of simulation and experimental work, our collaborators have been investigating more spatial forms of computing, involving larger reservoirs containing sub-excitable BZ medium. In these conditions, isolated spatial propagating waves can form, combine and interfere in spatial and geometrical ways to accomplish computation-like tasks, such as logic gates (Holley et al., 2011; Adamatzky et al., 2012).

Ongoing research

Information measures for analysing and guiding the artificial evolution of unconventional computational media.

Following information theory (Shannon and Weaver, 1948) and information dynamics measures (Lizier, 2013) in cellular automata and in neural networks (Vicente et al., 2011), which help to identify information propagation, storage and modification systems, we are developing analysis tools for understanding the information flows of experimental and simulated droplet systems. These tools are intended to aid in the tracking and understanding of the flow of information through unconventional computational media, in a way that is largely independent of the encoding of the information and to thereby facilitate the search for complex and potentially useful system behaviours in random or evolved droplet networks, which are inherently less modular and decomposable than conventional engineered computational systems. We are also exploring the use of information theoretical measurements to constrain the design of functional networks. By identifying necessary changes in the state of information at different stages of computation, we believe it may be possible to guide machine-learning algorithms to more effectively design functional networks.

Defining computational-unit fitness implicitly using tautological closed loops.

To facilitate the artificial evolution of of Basic Composable Units ("BCUs" – c.f. logic gates) for unconventional computational media, we are developing a novel technique in which optimal BCU behaviour is defined not explicitly ("given this input, the unit should produce that output"), but implicitly, through its influence on network properties in a closed network consisting of multiple instances of the unit. The network is designed in such a way that only if the units are performing the desired task (e. g. acting as a NAND gate), will certain network properties hold (e. g. dynamics at two points in the network should be similar to each other and different to a third point), and machine-learning techniques tune the BCU properties maximise these network properties. In this way, we implicitly describe the desired behaviour of the units without overly constraining their design, allowing the artifical evolution to concurrently design the BCUs and the encoding of the signal that they operate upon.

References

Adamatzky, A., Holley, J., Dittrich, P., Gorecki, J., Costello, B. D. L., Zauner, K.-P., and Bull, L. (2012). On architectures of circuits implemented in simulated belousov–zhabotinsky droplets. *Biosystems*, 109(1):72–77.

Beer, R. D. (2003). The dynamics of active categorical perception in an evolved model agent. *Adaptive Behavior*, 11(4):209–243.

Gruenert, G., Escuela, G., and Dittrich, P. (2012). Symbol representations in evolving droplet computers. In Durand-Lose, J. and Jonoska, N., editors, *Proc. of Unconventional Computation and Natural Computation - 11th Intl. Conf. Orléan, France, September 3-7, 2012.*, volume 7445 of *Lecture Notes in Computer Science*, pages 130–140. Springer.

Gruenert, G., Szymanski, J., Holley, J., Escuela, G., Diem, A., Ibrahim, B., Adamatzky, A., Gorecki, J., and Dittrich, P. (2013). Multi-scale modelling of computers made from excitable chemical droplets. *International Journal of Unconventional Computing*, 9(3–4):237–266.

Holley, J., Jahan, I., Costello, B., Bull, L., and Adamatzky, A. (2011). Logical and arithmetic circuits in Belousov Zhabotinsky encapsulated discs. *Physical Review E*, 84(5):056110.

King, P. H., Corsi, J. C., Pan, B.-H., Morgan, H., de Planque, M. R., and Zauner, K.-P. (2012). Towards molecular computing: Co-development of microfluidic devices and chemical reaction media. *Biosystems*, 109(1):18–23.

Lizier, J. T. (2013). *The local information dynamics of distributed computation in complex systems*. Springer.

Shannon, C. E. and Weaver, W. (1948). A mathematical theory of communication. *The Bell Systems Technical Journal*, 27:379–423,623–656.

Stewart, J., Gapenne, O., and Di Paolo, E. A. (2011). *Enaction: Toward a New Paradigm for Cognitive Science*. MIT Press.

Szymanski, J., Gorecka, J. N., Igarashi, Y., Gizynski, K., Gorecki, J., Zauner, K.-P., and Planque, M. D. (2011). Droplets with information processing ability. *International Journal of Unconventional Computing*, 7(3):185–200.

Vicente, R., Wibral, M., Lindner, M., and Pipa, G. (2011). Transfer entropya model-free measure of effective connectivity for the neurosciences. *Journal of computational neuroscience*, 30(1):45–67.

Zhabotinsky, A. M. (2007). Belousov-zhabotinsky reaction. *Scholarpedia*, 2(9):1435.

Using Re-Entrant Networks of Repeated Units to Facilitate the Automated Design of Logic Gates in Unconventional Media: A New Method

Matthew Egbert, Gerd Gruenert, Peter Dittrich

Introduction

Unconventional Computing (UC) is a rapidly developing field, where materials that are not typical of modern computers, such as quantum systems (1), reaction diffusion systems (2,3), biological materials such as neurons, synthesized protocells, or DNA, are assembled or controlled to perform computational tasks. Some unconventional computers may be able to perform certain computational tasks orders of magnitude faster and with greater accuracy than conventional computers, and this is one primary motivation for the study of UC. A second motivation is the rapid technological progress in synthetic biology and nano-technology, where we are seeing the development of increasingly small and autonomous engineered "nanites" and synthetic protocells. The potential for these new forms of technology is radical, but before we can fully benefit from them, we must improve our ability to control these systems.

Current attempts to control these systems tend to work on a case-by-case basis, finding feedback loops or ways to couple environmental stimulus or "input" to system actions to produce a target behaviour. A more ambitious goal is to achieve *programmable control* of unconventional media, where motifs can be combined to assemble or configure (i.e., program) unconventional media to perform a wide variety of tasks. The effort to transition from *controllable* unconventional media to *programmable* unconventional media mirrors the transition in the 20th century from feedback-based machines of the cybernetics era, to modern programmable computers, and the benefits of programmability are demonstrated by the widespread use of computers in virtually every aspect of the modern world, from stock-trading to surgery to social interaction. It would be a tremendous technological advance to be able to program various unconventional media, such as protocells or chemical computers, but we do not yet know how to do so, nor do we have effective techniques for learning how to do so in a particular unconventional medium.

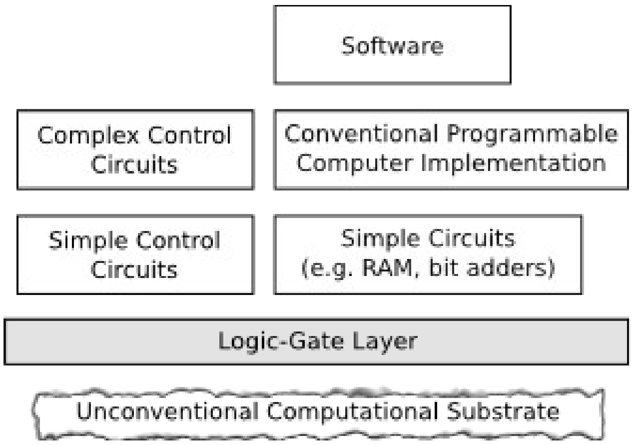

Illustration 1: A layered architecture, where some benefits of conventional computing are made available in unconventional media through the implementation of a logic-gate layer.

One possible approach to bringing the benefits of programmability to unconventional media, is to implement in the unconventional medium, a universal logic gate, such as NAND. All other basic logic operations can be implemented using only NAND or NOR (4), and, if implemented in a sufficiently robust and flexible manner, many NAND gates could be combined to provide the functionality of Turing-complete computer in an unconventional medium, and this could provide an abstraction layer (see Figure 1) upon which functional constructs and higher abstractions could be built. This paper presents a new technique for identifying configurations of unconventional media that operate as NAND-gates.

Unconventional engineering for unconventional media

For most unconventional media, where conventional engineering methods are difficult to apply, it is not clear how to build even simple signal-processing units such as NAND-gates. Fortunately, unconventional engineering techniques can be utilised, such as evolutionary algorithms (5), where instead of directly engineering a system, a “fitness function” is specified which defines how successful or “fit” a putative solution is at performing a target behaviour. A population of solutions is then randomly generated, and subjected to a set of evolution-inspired processes involving “mutations” (small random changes), selection (removal of less fit individuals from the population) and procreation/recombination (highly fit individuals are duplicated and combined with one another to produce a new solution that is added to the population). Part of the purported advantage of machine learning techniques such as evolutionary algorithms is that we can avoid biases brought by the experience of engineering conventional systems, and identify solutions that are different from those that would emerge through standard engineering practices (6). For instance, Thompson et al. (7) used an evolutionary algorithm to configure a field-programmable gate array to operate as a tone-discriminator circuit. Free from the constraints of conventional electronics engineering, such as “divide-and-conquer” or “building-block” based design, the evolutionary algorithm exploited unconventional features of the medium, such as “cross-talk” between neighbouring hardware, to accomplish the tone-discrimination task (8). The work by Thompson et al. resulted in an unconventional circuit design that takes advantage of unknown or poorly understood material properties, and the method presented in this paper has a similar goal – to use evolutionary algorithms to identify, in an unprejudiced way, configurations of an unconventional medium that operate NAND-gates.

A naïve evolutionary algorithm approach to designing NAND gates in an unconventional medium would involve the pre-specification of how bits are represented in the medium, and the use of these values as inputs for testing putative solutions to evaluate how they respond to inputs (0,0), (0,1), (1,0) and (1,1). If the NAND gates are to be assemble-able, the signals used for input must be the same as those produced by the output of the gate, so the output of these networks would be compared to the pre-specified representations, and the more similar the output to the target NAND output, the higher the fitness.

A fault with this approach lies in pre-specification of the representation of ones and zeros. In conventional computers, data such as images, programs, text, etc. are represented as sequences of “ones and zeros” which are represented by high and low voltages at different physical locations within the computer. In an unconventional medium, such as a protocell-based computer, it may or may not be possible to similarly represent bits using low or high concentrations of chemicals. Even if it is possible to do so, there also may be other, less conventional bit-representations, such as different oscillation rates, the relative concentration of chemicals, etc, and some of these bit-representations will be more amenable to NAND-based computation in the unconventional medium. That is to say, they will be more easily transmitted and manipulated by configurations of the unconventional medium to accomplish NAND operations, requiring less intricate or fragile configurations of the medium. In some cases, perhaps, the engineer will be clever or lucky, and the unconventional medium will be compatible with her pre-specified bit representations. However, in many cases, the unconventional medium would either be incapable of acting as a NAND gate for the particular pre-specified bit-representations, or the configuration for doing so would be difficult or impossible to find. How can we identify these representations in an unbiased way?

An improvement might be to utilise co-evolution (9): simultaneously evolving two populations, one that specifies the representation, and one that tries to optimise system configurations that manipulate those representations like NAND gates. This would avoid the pre-specification of the zero and one representations, but would have the disadvantage of increasing the total search space.

In the next section we propose a new, alternative approach that allows for the automated identification of NAND-gate functionality in unconventional media, without requiring a pre-specification of bit-representations and without increasing the size of the search-space.

Re-Entrant Repeated Unit Networks

Our new technique employs feedback-networks, in which putative solutions are coupled to copies of themselves, so that the output of a putative solution eventually returns as its input. By comparing the state of the system at different points within these Re-Entrant Repeated UNit (RERUN) networks, it is possible to evaluate how well the solutions are acting as certain kind of signal-processing units (e.g. as a NAND-gate) without pre-specifying the how the signal is represented.

NOT-Gate RERUN Network

We shall describe the technique by presenting the simplest possible example: the NOT-Gate RERUN network depicted on the right of Figure 2. This figure depicts an unconventional computing unit and its identical copy (gray pentagons), connected such that the output of one is the input of the other and vice versa. In this circuit, if the

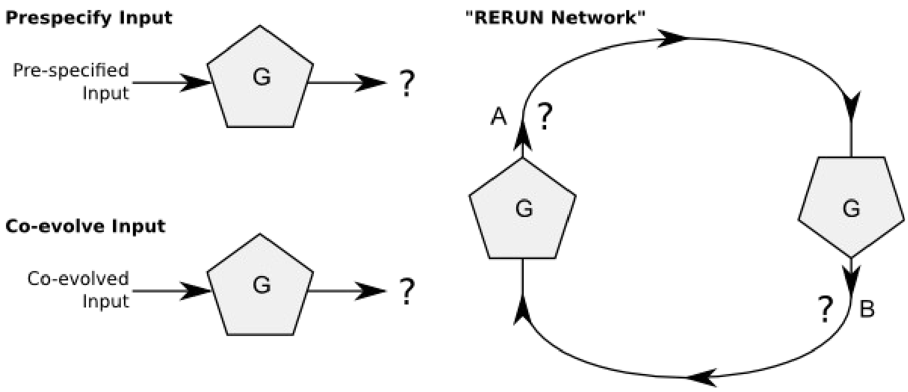

Figure 2: ***Three approaches to evolving logic gates in unconventional media.***

output of each of these identical gates (indicated by '
unchanging, then the units can be interpreted as actin
points A and B can be interpreted as representing '0' a
observation provides us with all we need to formulate
functionality in the repeated unconventional units wit
are represented in the system. Simply by comparing t
identical units differ, we can evaluate the quality of th
gate. If the output of one is often similar to the output
poorly as a NOT-gate, but if it is always measurably c
the unit is operating effectively as a NOT-gate, and th
by the output of the units in the circuit. This approach
solutions because it does not require the pre-specifica
measurement for detecting how the system can vary.

NAND-Gate RERUN Network

Figure 3 depicts a RERUN network that can be used to imply NAND-gate functionality. To create this network, we enumerate four identical putative NAND-gate units, associating each one with each of the four possible inputs, (0,0), (0,1), (1,0) and (1,1) and the target output (NAND) – see Table 1. We then connect the units in a way that satisfies the following constraints:

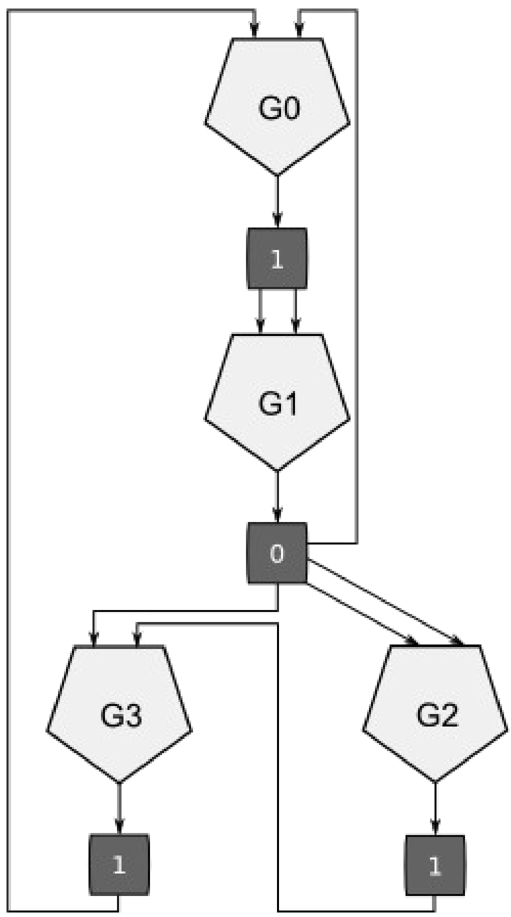

Figure 3: ***NAND-gate RERUN network.*** *Output of identical putative NAND gates (pentagons G0-G3) are labelled according to target output symbol (symbol in dark squares).*

- The inputs for each unit (as specified in Table 1) must come from outputs of the matching target value. So, for example, the inputs for G0 could not come from G2 and G3, because the input of G0 is (1,0) and both G2 and G3 have a target output of 1. The inputs for G2 must both come from G1, because it is the only unit with a target output of 0, etc.
- The input of every unit is the combination of the output of one or two *other* units.
- The output of every unit is used by at least one other unit.

UNIT	Input 0	Input 1	Target Output (NAND)
G0	1	0	1
G1	1	1	0
G2	0	0	1
G3	0	1	1

Table 1: ***Inputs and target outputs for NAND-implying closed-circuit.***

There are a variety of other possible NAND RERUN networks (e.g. other ways to write the units together to satisfy the conditions above, or a simpler, three-unit RERUN network that considers the inputs (0,1) and (1,0) to be equivalent). We focus on the RERUN network depicted in Figure 3 for the remainder of the paper.

The fitness of the repeated unit in terms of acting as a NAND-gate can be evaluated, similarly to the NOT-gate RERUN network, by comparing the network dynamics at different locations. Specifically, the output of G0,G2, and G3 should all be similar (as all of these gates should output 1), and each of these should be different from the output of G1 (which should output 0). If, that is the case, then the repeated units must either be acting effectively as a NAND gate (or, equivalently, as a NOR gate, if we reverse our interpretation of which state represents 0 and which represents 1).

In the next section, we use this NAND RERUN network, in conjunction with an evolutionary algorithm and a fitness function that compares the state of the RERUN network as described above, to identify configurations of an unconventional media that operate as NAND gate.

Application and Results

Continuous-time Recurrent Neural Network

In this section, we demonstrate use a NAND RERUN network in conjunction with an evolutionary algorithm to identify a configuration of continuous-time recurrent neural network (CTRNN) nodes (10) that operates effectively as a NAND-gate.

CTRNN are universal dynamical approximators (10) that have been widely used in conjunction with evolutionary algorithms to produce a broad variety of dynamical systems. A CTRNN consists of a set of interconnected nodes, where the state or “activation” of each node changes continuously in time as a function of its current state and the state of its neighbours, according to the following differential equation.

$$\frac{\mathrm{d}y_i}{\mathrm{d}t} = \frac{1}{\tau}(-y_i + \sum w_{ji}\sigma(y_j + \theta_j)) \qquad \text{(Equation 1)}$$

In this equation, y_i is the state of the ith node, τ_i is the ith node's time constant, w_{ji} is the weight from the jth to the ith node, $\sigma()$ is the sigmoidal function $\sigma(x) = \frac{1}{1+x}$, and β_j is a bias term associated with the jth node.

A network of these nodes was generated by replacing each unit in the NAND RERUN network (Figure 3) with a two CTRNN nodes in a linear chain, resulting in the network depicted in Figure 4, where nodes (ovals) and labelled target outputs (dark squares) are indicated. Each two-node unit is identical; therefore, there are only six tunable parameters for this network: the τ and β parameters for each node, and two weight parameters: one specifying the weights of the inputs to the cluster (e.g. from CTRNN node N1 to N2), and one specifying the weight of connection from the first node to the second node in each unit (e.g. from N7 to N0).

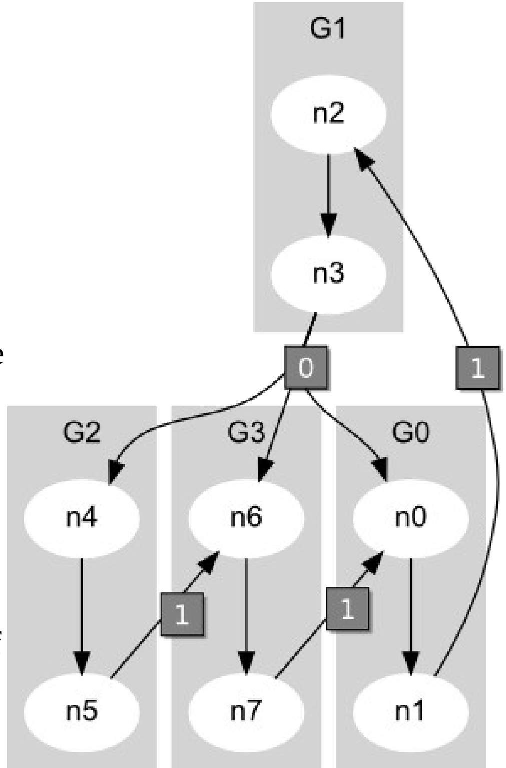

Illustration 4: ***CTRNN NAND RERUN network.*** *Repeated identical putative NAND gates (G0-G3), each composed of two CTRNN nodes (n0-n7) connected according to the NAND RERUN topology, with target outputs labelled (dark squares).*

These parameters were optimised using the Evolutionary Strategie (ES) provided by the *inspyred* Python package (11) to maximise the difference between network locations with different labels, and to minimise difference between locations marked with similar labels.

Each fitness evaluation consisted of four trials. At the start of each trial every node in the network is initialised with a state selected from a Gaussian distribution near zero (σ = 0.01, mean = 0) and the differential equations were integrated using the ode.integrate method of SciPy (12) for a trial duration of 10 time units. Note that because of the repetition of identical units in the network, the system must be initialised asymmetrically. If the system is initialised in a uniform manner, every unit (G0 – G3) would initially produce exactly the same output. They would then therefore all receive exactly the same input, and this would continue indefinitely with no way for the system to break symmetry. Without breaking symmetry there is no way for any of the gates to produce different outputs and there will be no success in identifying configurations of units that perform as desired.

In the CTRNN NAND RERUN network (Figure 4), there are three locations labelled '1' and one location (the output of N3) labelled '0'. Fitness is evaluated by making pairwise evaluations of the similarity between all of these locations. This is evaluated by the following function c(i,j) which takes the mean difference between the state nodes i and j during the last 2.5 time units of a trial, and normalises this values to be between 0 (similar) and 1 (dissimilar).

$$\mathrm{c}(i,j) = \frac{1}{20} \cdot \left| \sum_{t=7.5}^{10} \frac{n_t^i - n_t^j}{2.5} \right| \qquad \text{(Equation 2)}$$

For comparisons between locations with the same label, similarity is rewarded (first two terms in the fitness function below) and when the labels are different, dissimilarity is rewarded (third term).

$$\text{fitness} = \prod_i^{\text{zeros}} \prod_j^{\text{zeros}} (1 - \mathrm{c}(i,j)) \cdot \prod_i^{\text{ones}} \prod_j^{\text{ones}} (1 - \mathrm{c}(i,j)) \cdot \prod_i^{\text{zeros}} \prod_j^{\text{ones}} (\mathrm{c}(i,j)/k)$$

(Equation 3)

Results

We succeeded at evolving 2-Node CTRNN with NAND-gate functionality. The evolutionary strategie quickly identified parameters that resulted in a relatively high-fitness network. Figure 5 shows an example of the dynamics of each of the four units (CTRNN nodes N1,N3,N5, and N7) during a trial of the evolved network. N3 (indicated with the dashed line) is the only node that is associated with a '0' output, and it is apparent here that as desired, the artificial evolution has identified parameters that successfully cause the state of the RERUN network to be similar in areas with the same label, and to differ where the labels are different. Locations in the network labelled '1' all approach a value near 0, and the location marked '0' approaches 10 – these are the CTRNN states that have emerged via the RERUN-based evolution as suitable representations of ones and zeros in the unconventional medium.

Evolved Parameter	Value
τ_0	0.321
β_0	4.069
τ_1	0.487
β_1	-5.3
trans-unit w	-8.731
Intra-unit w	9.915

Table 2: ***Evolved parameter values.***

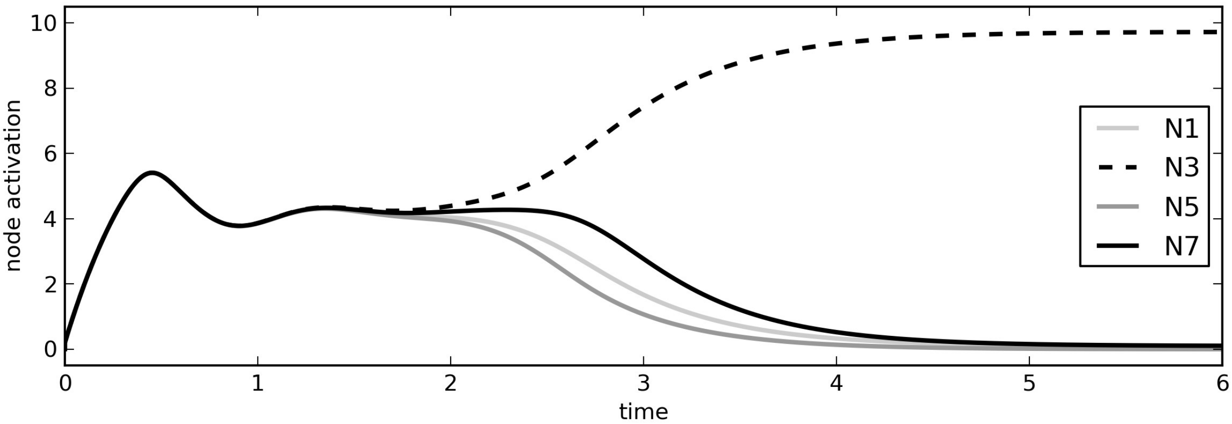

Figure 5: ***CTRNN node states during a trial of the evolved, high-fitness CTRNN RERUN network.*** *The output of nodes N1, N5, and N7, associated with bit-values of '1' (see Figure 4), approach a state near 0.0, while Node 3, associated with an output of '0' converges to a state near 10.*

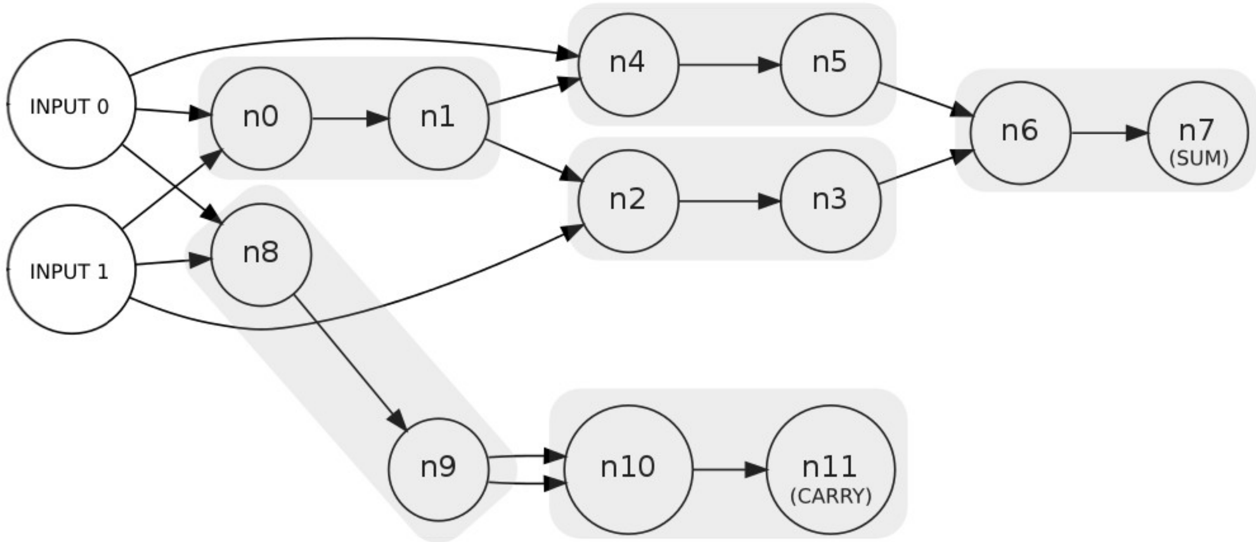

Illustration 6: ***CTRNN half-adder network.*** Each circle represents a CTRNN node. These are configured in pairs (highlighted by grey-rounded rectangles) that putatively act as NAND-gates, thanks to the RERUN-Network based evolution. The network should act as a half-adder, such that inputs 0 and 1 produce SUM and CARRY outputs as specified in Table 3.

To confirm the success of the evolution, we assembled a NAND-based half-adder network out of the evolved CTRNN units (Figure 6). In this network, nodes 0 – 7 operate as four of the evolved CTRNN-based NAND gates assembled into an XOR gate, used to calculate the SUM output of the half-adder. Gates 8 – 11 are two evolved CTRNN-based NAND gates assembled into an AND gate, used to calculate the CARRY output of the half-adder. The output of the network thus are the state of nodes 7 (SUM) and 11 (CARRY) and the input of the half-adder is determined by setting the state of the nodes “Input 0” and “Input 1” to the values determined during evolution to represent '0' and '1' (see Table 1). We used only approximations of these values, considering '0' to be represented by a node excitation of 10.0, and '1' to be represented by a node excitation of 0.0.

INPUT				**OUTPUT (SUM)**			**OUTPUT (CARRY)**		
Symbol Input 0	**Symbol Input 1**	**N13 State**	**N15 State**	**Correct Symbol**	**N7 Target State**	**N7 Final State**	**Correct Symbol**	**N11 Target State**	**N11 Final State**
'0'	'0'	10	10	'0'	10	9.732	'0'	10	9.733
'0'	'1'	10	0	'1'	0	0.098	'0'	10	9.732
'1'	'0'	0	10	'1'	0	0.098	'0'	10	9.732
'1'	'1'	0	0	'0'	10	9.732	'1'	0	0.000

Table 3: Half-adder input and output.

We performed two tests with the half-adder network. First, to confirm that the network can produce the correct output for a given input, we simulated the network with fixed, unchanging inputs for each of the four possible input sets. Table 3 shows the input, target output, and actual output for the four test cases after the network has been simulated for 10 time units. Comparing the target output and the final state of nodes 7 and 11, it is clear that the network has successfully produced output expected from a half-adder.

We then tested if the network was capable of dynamically changing in response to changes in the network input. To make this test, we simulated the half-adder network for 130 time units, changing the input every 10 time units, to test each of the 12 possible input transitions. Figure 7 shows the input and output nodes of this simulation, along with the error (distance from correct target value).

For every possible transition, this error term rapidly decreases to near 0. We conclude that the network is capable of correctly transitioning from any input state to any other input state, in each case, rapidly converging on the correct output values.

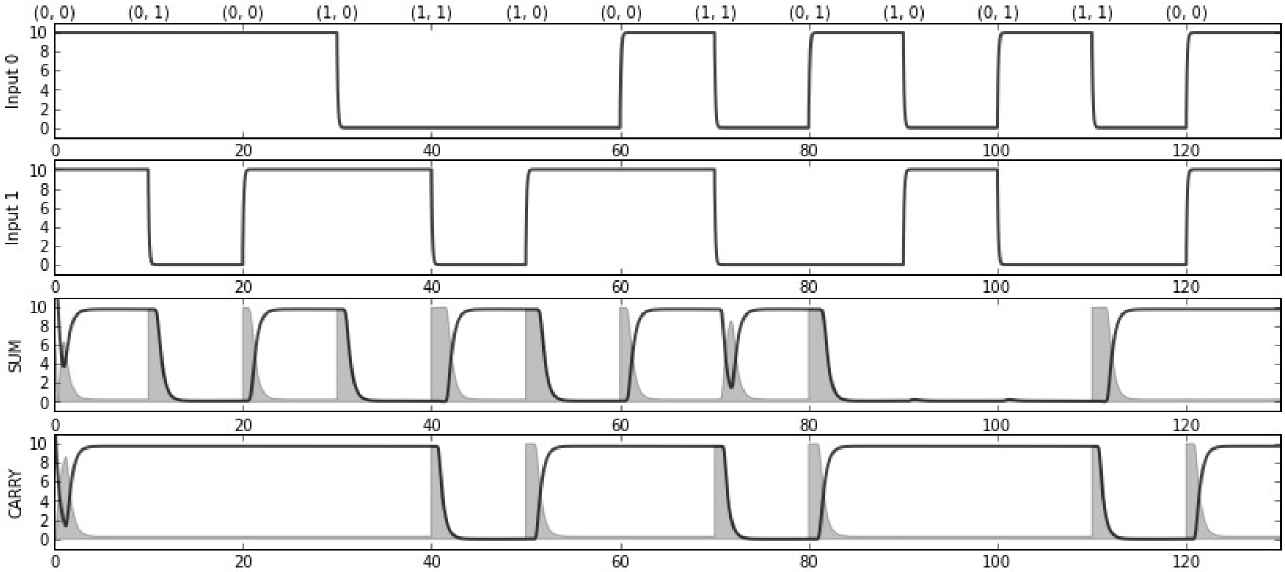

Figure 7: ***CTRNN half-adder transition test.*** *Inputs (top two rows) are varied to simulate all of the possible transitions between inputs and the SUM and CARRY outputs (bottom two rows) of the network respond correctly. In the bottom two rows, the state of outputs nodes are plotted as curves and the error (distance from target output) is shaded in gray. For each transition, the network rapidly approaches the correct target values.*

Discussion

We have proposed a new techniques for identifying configurations of unconventional media that act as NAND-gates. The method involves the construction of feedback-circuits, where a potential configuration of the unconventional medium is connected to copies of itself. By comparing states at different locations in these feedback-circuits, it is possible to evaluate how well the configuration is operating as a NAND-gate. This evaluation can be accomplished without pre-specifying how the bit-values are represented in the network, requiring only way(s) of measuring how system dynamics can be different or similar at different locations within the network. By avoiding the pre-specification of the representation of bit-values in the system, we let artificial evolution or other search algorithms/techniques identify bit-value representations that are more natural to the unconventional medium in the sense that they are easier for the medium to transmit and manipulate in NAND-operations than arbitrarily pre-specified representations.

We successfully demonstrated the potential of the technique, using an evolutionary algorithm to optimise the parameters of a 2-Node CTRNN network to operate as a NAND-gate. The technique identified bit-representations associated with low and high state values of the CTRNN nodes and parameters that allowed the CTRNN networks to perform successfully as NAND-gates. Tests were conducted to show that these CTRNN-based NAND-gates could be coupled to produce a half-adder system that produces the correct output for all possible inputs and for all possible transitions between inputs.

During our investigations we noticed two interesting details that are worthy of further comment. When simulated in the feedback-circuit, the evolved CTRNN network, did not always succeed at producing high-fitness outputs. Depending upon the initial conditions, the network would sometimes fail, for instance, to produce different output for the output of units G1 and G3, (CTRNN nodes N7 and N3) (see Figure Fehler: Referenz nicht gefunden). In fact, in 100 trials of the evolved

CTRNN feedback-circuit, only 52 simulations succeeded in producing high-fitness scores. But, when simulated as part of a half-adder, we never once noticed an incorrect output. This has prompted us to consider the possibility that the tight feedback in the feedback-circuits can produce dynamics or attractors that are unstable in the absence of feedback. In the case of the CTRNN network, this had no deleterious effects upon the evolved NAND gate, which worked flawlessly in the half-adder tests. It is however, conceivable, that this feedback could cause problems in other systems, producing high-fitness solutions that do not effectively function in non-re-entrant (or less tightly re-entrant) logic circuits.

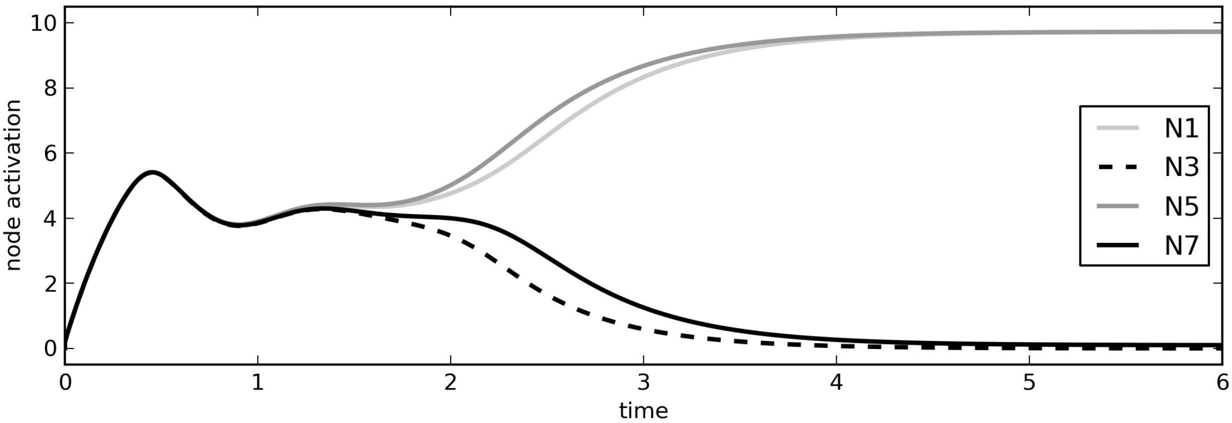

Illustration 8: Low-fitness initial conditions for the evolved CTRNN based NAND-gate.

In this report, we have described and demonstrated the potential utility of a new method that may be broadly applicable for creating new computational devices out of unconventional media. Although we have demonstrated that the technique can be used successfully for some systems, it remains future research to determine how generally applicable the idea is, and for what systems it can work and cannot. It is also will be interesting to investigate how this technique could be used on its own, i.e. *without* a genetic algorithm. If a sufficiently complex system is wired up in a RERUN network such as those presented above, and the output of the repeated units is measured in a variety of ways, it may be possible to identify an existing dynamic in the system that naturally operates as a NAND gate.

References

1. Nielsen MA, Chuang IL. Quantum computation and quantum information. Cambridge university press; 2010.

2. Adamatzky A, Costello BDL, Asai T. Reaction-Diffusion Computers. Elsevier; 2005.

3. De Lacy Costello B, Toth R, Stone C, Adamatzky A, Bull L. Implementation of glider guns in the light-sensitive Belousov-Zhabotinsky medium. Phys Rev E. 2009 Feb 27; 79(2):026114. Available from: http://link.aps.org/doi/10.1103/PhysRevE.79.026114

4. Sheffer HM. A Set of Five Independent Postulates for Boolean Algebras, with Application to Logical Constants. Trans Am Math Soc. 1913 Oct 1 ;14(4):481–8. Available from: http://www.jstor.org/stable/1988701

5. Holland JH. Adaptation in natural and artificial systems: an introductory analysis with applications to biology, control, and artificial intelligence. University of Michigan Press; 1975.

6. Harvey I, Paolo ED, Wood R, Quinn M, Tuci E. Evolutionary robotics: A new scientific tool for

studying cognition. Artif Life. 2005;11(1-2):79–98.

7. Thompson A, Layzell P, Zebulum R. Explorations in design space: unconventional electronics design through artificial evolution. IEEE Trans Evol Comput. 1999;3(3):167–96.

8. Thompson A. Evolving electronic robot controllers that exploit hardware resources [Internet]. Springer; 1995. Available from: http://link.springer.com/chapter/10.1007/3-540-59496-5_332

9. Hillis WD. Co-evolving parasites improve simulated evolution as an optimization procedure. Phys Nonlinear Phenom. 1990 Jun; 42(1–3):228–34. Available from: http://www.sciencedirect.com/science/article/pii/0167278990900762

10. Beer RD. On the Dynamics of Small Continuous-Time Recurrent Neural Networks. Adapt Behav. 1995 Mar 1 ;3(4):469–509. Available from: http://adb.sagepub.com/cgi/content/abstract/3/4/469

11. Inspyred: Bio-inspired Algorithms in Python. . Available from: http://inspyred.github.io/

12. Jones E, Oliphant T, Peterson P, et al. SciPy: Open source scientific tools for Python. 2001. Available from: http://www.scipy.org/

13. Zhabotinsky AM, Zaikin AN. Autowave processes in a distributed chemical system. J Theor Biol. 1973;40(1):45–61.

14. Adamatzky A, Holley J, Dittrich P, Gorecki J, De Lacy Costello B, Zauner K-P, et al. On architectures of circuits implemented in simulated Belousov-Zhabotinsky droplets. Biosystems. 2012;109(1):72–7.

15. Adamatzky A. Computing in nonlinear media and automata collectives. Bristol, UK: IOP Publishing Ltd.; 2001.

16. Szymanski J, Gorecka JN, Igarashi Y, Gizynski K, Gorecki J, Zauner K-P, et al. Droplets with information processing ability. Int J Unconv Comput. 2011;7:185–200.

17. Gruenert G, Szymanski J, Holley J, Escuela G, Diem A, Ibrahim B, et al. Multi-scale Modelling of Computers Made from Excitable Chemical Droplets. Int J Unconv Comput. 2013;9:237–66.

18. Escuela G, Gruenert G, Dittrich P. Symbol representations and signal dynamics in evolving droplet computers. Nat Comput. 2013;1–10. Available from: http://dx.doi.org/10.1007/s11047-013-9384-7

Supplementory material for “Using Re-Entrant Networks of Repeated Units to Facilitate the Automated Design of Logic Gates in Unconventional Media: A New Method”

Simulation of Computing Belousov-Zhabotinsky Droplet Networks

To demonstrate that the RERUN-method can be used to design NAND-gates in more complex non-linear systems, we used the method to identify configurations of simulated protocell-computers that can operate as NAND-gates.

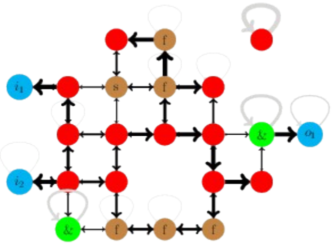

Abbildung 1: Evolved droplet network realizing the NOR function. Blue droplets are the inputs (left) and outputs (right), red droplets are normal droplets, brown droplets are faster (f) and slower (s) oscillating droplets, green droplets (&) are less excitable.

We investigated a chemical computing substrate for artificial chemical neurons: Droplets of the aqueous Belousov-Zhabotinsky (BZ) medium [Zaikin1970,Zhabotinsky1973], coated in a Lipid Monolayer and swimming in organic phase are considered for computation by propagation of pulses of chemical excitation [Adamatzky2001,Adamatzky2012]. The BZ medium in the droplet either self-excites after a period of inactivity or it can be triggered into a new excitation by adjacent droplets. This propagation of excitation from one droplet to the next is bounded by a refractory period after each excitation. There are complex and highly accurate ODE descriptions of the BZ medium [Szymanski2011], but they are computationally extensive. Instead we use an event-based simulation system 'DropSim', that abstracts complex chemical processes to only the three states 'excited', 'refractory' and 'responsive' and allows for very fast simulations of large droplet systems [Gruenert2013]. Nonetheless, evolution of particular boolean gates, especially the NAND gate is not trivial in this system [Escuela2013].

For the evolution of a NAND gate from simulated BZ droplets, we used planar designs in a 5x5 grid where each horizontally and vertically adjacent droplet is connected. The normal droplets in simulation stayed excited for one second, were refractory for five seconds and then self-excited after 10 seconds if not externally triggered. The signal propagation delay from one droplet to the next was one second. For each of these values, a normal distributed noise term of the standard deviation 0.05 s is added for each event, such that the simulation becomes non-deterministic.

The mutation function can exchange each of the 25 positions in the grid with 5 different droplet types, including no droplet. The remaining four droplet types were a normal droplet as described above, a slightly faster oscillating droplet with 0.8 times the original period, a slightly slower oscillating droplet with 1.25 times the original period and a less excitable droplet that requires two concurrent excitation at its neighbours to be triggered into an excitation. The probability for exchanging a droplet position in the grid was 0.05 per position.

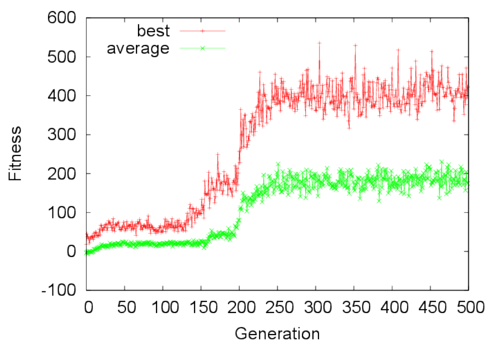

Abbildung 2: Best and average fitness per generation over the best evolution run.

Evolutionary studies were conducted in a 50 x multi-start approach, where over 500 generations, a population of 10 parents was selected by truncation selection from 37 children and 3 random immigrants. Single point cross over was used in recombination.

To evaluate the fitness of an individual, we connected the evolutionary designed droplet network in a RERUN network as proposed in Figure 3 of the main article. Additionally to the evolved part of the network, we added 'diode droplets' [Szymanski2011] that allow the signal propagation only in the direction that is symbolized by the arrows in Figure 3. We used the fitness formulation from Equation 3 (main paper) and averaged the three lowest of 24 simulation runs of 3000 seconds for each droplet network, where c(i,j) is the difference in spike numbers, counted at the output droplets.

Results

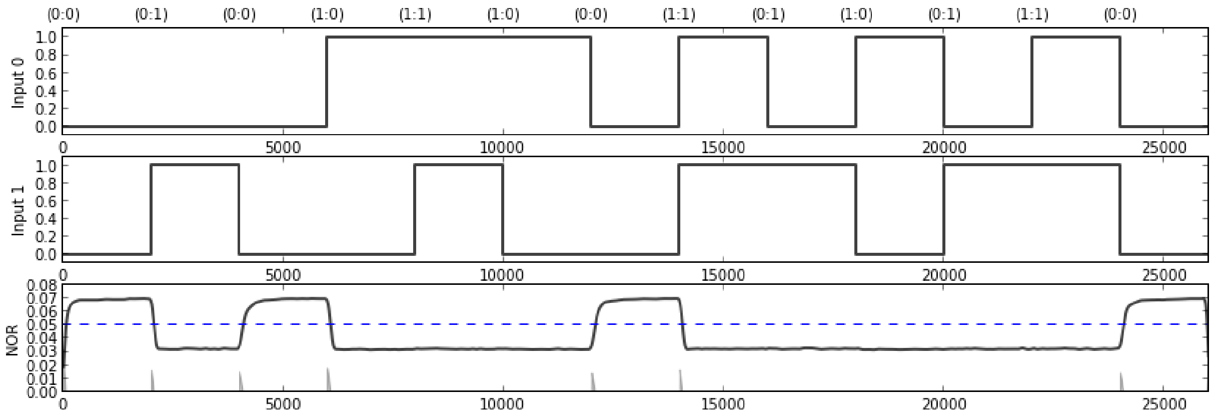

Abbildung 3: Ouput frequency of the evolved NOR network for all four input configurations, averaged over 100 runs. Each stimulation pattern is applied for 2000 seconds. We interpret a high oscillation frequency (more than 0.05 spikes/second) as a logical one and a low oscillation frequency (less than 0.05 spikes/second) as a logical zero. The blue, dashed line in the NOR plot indicates an externally applied threshold function that can be used to distinguish a high from a low frequency. The gray peaks in the lowest plot indicate the error of the NOR output: It indicates the distance of the output frequency from the blue threshold line, if the signal is on its wrong side.

The droplet network shown in Figure 1 is the final individual of the evolution run with the highest maximum fitness. The evolution dynamics of this run are shown in Figure 2, where it becomes obvious that even with the averaging, there is a strong deviation in the quality of each simulation run.

Even though the RERUN network was designed to fulfil the NAND-gate function, a NOR-gate was actually evolved, when assuming that a high spike rate is interpreted as a logical one and a low spike rate as a logical zero. Obviously, when inverting this assignment, a NAND gate becomes a NOR gate for the same operation on high and low signals. Averaged over 100 simulation runs, we plotted the

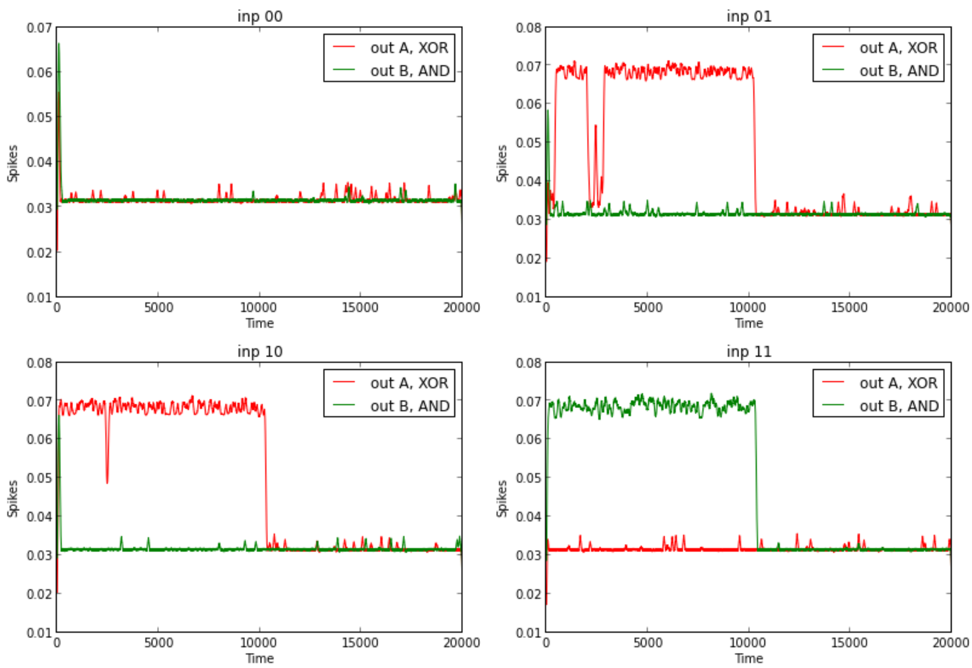

Abbildung 4: Half Adder functionality: both outputs of a typical run for 10000 seconds of stimulation and 10000 seconds of rest.

droplet network behaviour in Figure 3 for all switching processes between different input combinations.

When connecting the evolved NOR gates to a half-adder, it mostly implements the half-adder functionality, even though the influence of noise increases (cf. Figure 4).

This becomes more obvious when observing the different switching processes between input combinations in Figure 5. In the simulated 2000 s per input configuration, it took the half-adder network a lot longer to move the output spike frequency towards the desired region. This implies that it will not trivially be possible to build arbitrarily large logical systems from this NOR gate. Nonetheless, the gate was functional, even under noisy conditions.

Summing up, we used an evolutionary algorithm here to evolve an instance of a NAND gate that turned out to be rather a NOR gate by using an inverted encoding of high and low spike frequencies. In analogy to our experiments with the CTRNN networks, we also tested the NOR gates by coupling them to form a functional half-adder.

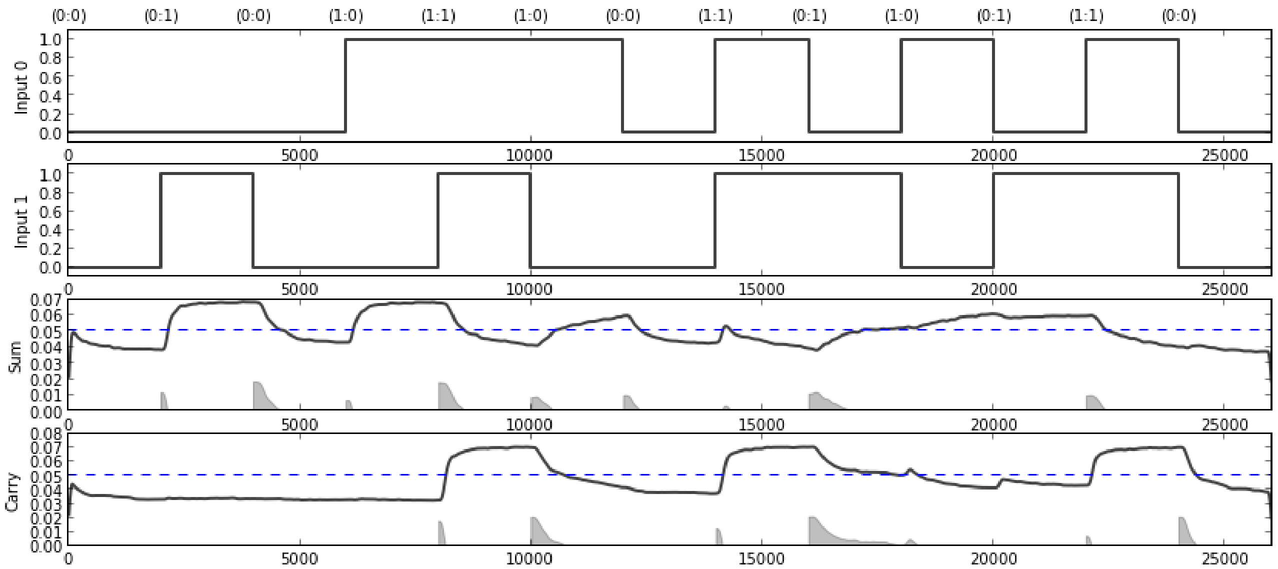

Abbildung 5: Ouput frequency of the half-adder network constructed from the evolved NOR gate for all four input configurations, averaged over 100 runs. Each stimulation pattern is applied for 2000 seconds. We interpret a high oscillation frequecy (more than 0.05 spikes/second) as a logical one and a low oscillation frequency (less than 0.05 spikes/second) as a logical zero. The gray peaks in the lower two plots indicate the error of the output: It indicates the distance of the output frequency from the blue threshold line, if the signal is on its wrong side.

References

1. Adamatzky, A.
   Computing in nonlinear media and automata collectives
   IOP Publishing Ltd., **2001**

2. Adamatzky, A.; Holley, J.; Dittrich, P.; Gorecki, J.; Costello, B. D. L.; Zauner, K.-P. & Bull, L. On architectures of circuits implemented in simulated Belousov–Zhabotinsky droplets
   Biosystems, 2012*, 109*, 72-77

3. Escuela, G.; Gruenert, G. & Dittrich, P.

Symbol representations and signal dynamics in evolving droplet computers
Natural Computing, Springer Netherlands, **2013**, 1-10

4. Gruenert, G.; Szymanski, J.; Holley, J.; Escuela, G.; Diem, A.; Ibrahim, B.; Adamatzky, A.; Gorecki, J. & Dittrich, P. Multi-scale Modelling of Computers Made from Excitable Chemical Droplets
International Journal of Unconventional Computing, **2013***, 9*, 237-266
5. Szymanski, J.; Gorecka, J. N.; Igarashi, Y.; Gizynski, K.; Gorecki, J.; Zauner, K.-P. & Planque, M. D. Droplets with information processing ability
International Journal of Unconventional Computing, **2011***, 7*, 185-200
6. Zaikin, A. N. & Zhabotinsky, A. M.
Concentration Wave Propagation in Two-dimensional Liquid-phase Self-oscillating System
Nature, **1970***, 225*, 535-537
7. Zhabotinsky, A. M. & Zaikin, A. N.
Autowave processes in a distributed chemical system.
Journal of Theoretical Biology, **1973***, 40*, 45-61