

Research



**Cite this article:** Adamatzky A, Schnauß J, Huber F. 2019 Actin droplet machine. *R. Soc. open sci.* **6**: 191135.  
<http://dx.doi.org/10.1098/rsos.191135>

Received: 6 July 2019

Accepted: 4 November 2019

**Subject Category:**

Computer science

**Subject Areas:**

molecular computing/theory of computing/hybrid computing

**Keywords:**

actin network, computing, waves, logical gates, finite-state machine, automata

**Author for correspondence:**

Andrew Adamatzky  
e-mail: [andrew.adamatzky@uwe.ac.uk](mailto:andrew.adamatzky@uwe.ac.uk)

# Actin droplet machine

Andrew Adamatzky<sup>1</sup>, Jörg Schnauß<sup>2</sup> and Florian Huber<sup>3</sup>

<sup>1</sup>Unconventional Computing Laboratory, Department of Computer Science, University of the West of England, Bristol, UK

<sup>2</sup>Soft Matter Physics Division, Peter Debye Institute for Soft Matter Physics, Faculty of Physics and Earth Sciences, Leipzig University, Germany & Fraunhofer Institute for Cell Therapy and Immunology (IZI), DNA Nanodevices Unit, Leipzig, Germany

<sup>3</sup>Netherlands eScience Center, Science Park 140, 1098 XG Amsterdam, The Netherlands

AA, 0000-0003-1073-2662; JS, 0000-0002-6408-8676

The actin droplet machine is a computer model of a three-dimensional network of actin bundles developed in a droplet of a physiological solution, which implements mappings of sets of binary strings. The actin bundle network is conductive to travelling excitations, i.e. impulses. The machine is interfaced with an arbitrary selected set of  $k$  electrodes through which stimuli, binary strings of length  $k$  represented by impulses generated on the electrodes, are applied and responses are recorded. The responses are recorded in a form of impulses and then converted to binary strings. The machine's state is a binary string of length  $k$ : if there is an impulse recorded on the  $i$ th electrode, there is a '1' in the  $i$ th position of the string, and '0' otherwise. We present a design of the machine and analyse its state transition graphs. We envisage that actin droplet machines could form an elementary processor of future massive parallel computers made from biopolymers.

## 1. Introduction

Actin is a protein presented in forms of monomeric, globular actin (G-actin) and filamentous actin (F-actin) [1–3]. G-actin polymerizes into filamentous actin forming a double helical structure [4–6]. The filaments can be further arranged into bundles by various different mechanisms such as crowding effects, cross-linking or counter-ion condensation [7–15]. The bundles are conductive to travelling localizations—defects, ionic waves, solitons [16–25]. By interpreting the presence or absence of a travelling localization at a given site of the network at a given time step, we can implement logical functions. This approach was comprehensively developed and successfully tested on chemical systems in the framework of collision-based computing [26–31]. As actin networks can implement logical functions, they can compute. So, in [32] we proposed a road map to experimental implementation of cytoskeleton-based computing devices. We proposed that collision-based cytoskeleton computers implement logical gates via interactions between travelling localization: voltage solitons on actin filaments or tubulin

microtubules bundles. An architecture of cytoskeleton computers can be developed via programmable polymerization of actin networks. Such cytoskeleton computers would take data via electrical and optical means, the signals (solitons, conformational defects) initiated by the input stimuli will be travelling along the network and the computational will be implemented via collisions of the signals at the structural gates of the network.

Our approach—computing with excitation waves propagating on overall ‘density’ of the conductive material—has previously been presented by us in [33]. As conductive material we looked at networks of actin bundles which were arranged by crowding effects without the need of additional accessory proteins [10,11]. We demonstrated how to discover logical gates on a two-dimensional slice of the actin bundle network by representing Boolean inputs and outputs as spikes of the network activity. In a previous paper [33], we demonstrated, using numerical integration of FitzHugh–Nagumo model, that a two-dimensional actin network realized  $k$ -ary Boolean functions  $G: \{0, 1\}^k \rightarrow \{0, 1\}$ , when  $k$  input electrodes and one output electrodes are employed.

In the present paper, we develop a novel concept and computer modelling implementation of the actin network machine, which implements a mapping  $F: \{0, 1\}^k \rightarrow \{0, 1\}^k$ , where  $k$  is a number of electrodes, and ‘1’ signifies the presence of an impulse on the electrode and ‘0’ the absence. At a higher level, the machine acts as a finite-state machine, at the lower level, a structure of the mapping  $F$  is determined by interactions of impulses propagating on the three-dimensional network of actin bundles.

We also offer an alternative to a numerical integration used in [33]: an automaton model of a three-dimensional actin network. There is a substantial body of evidence confirming that automaton models are sufficient and appropriate discrete tools for modelling dynamics of spatially extended nonlinear excitable media [34–36], propagation [37], action potential [38,39], electrical pulses in the heart [40–42]. A major advantage of automata is that they require less computational resources than typical numerical integration approaches.

Results presented in the paper give a rather ‘computer engineering’ view on a computation implementable with travelling localizations on acting bundle networks. We do not speculate about potential biological meanings of the phenomena described. That could be a scope of future studies.

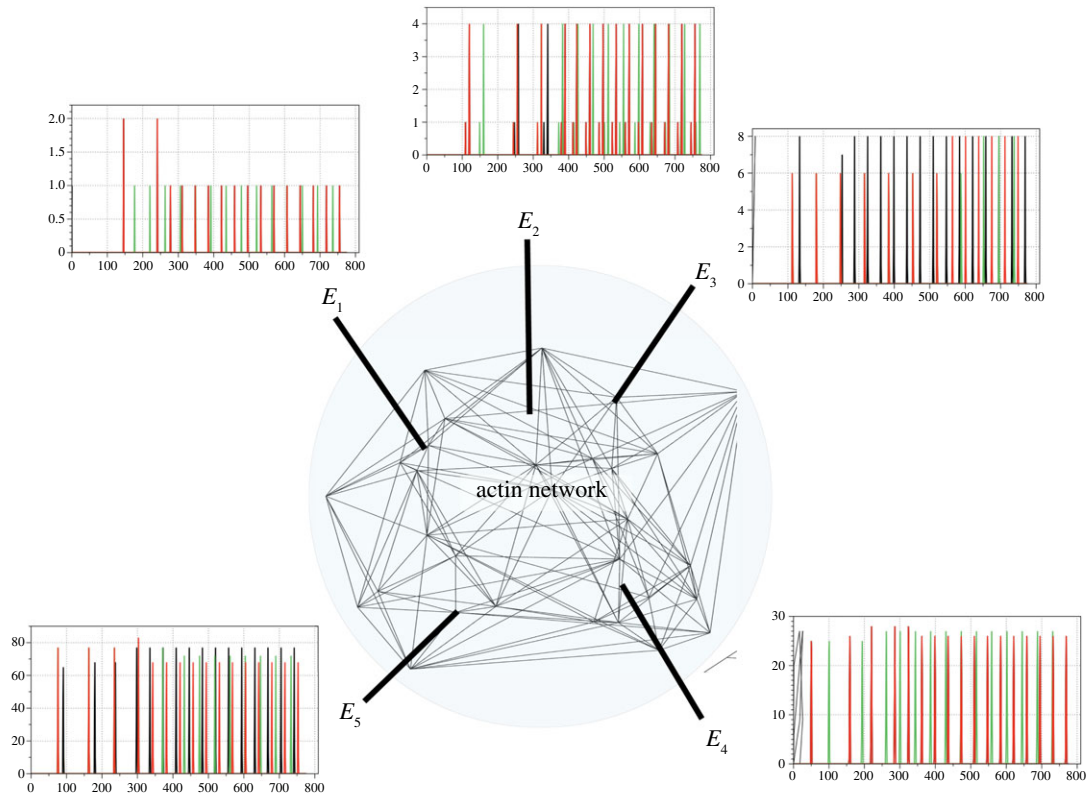
The paper is structured as follows. Our modelling approach is described in detail in §2. This includes a representation of a three-dimensional actin bundle network (§2.1), a structure of an automaton model to simulate propagation of impulses on the actin bundle network (§2.2), and an interface with the actin network (§2.3). In §2.4, we analyse dependencies of a number of Boolean gates implemented in the network on an excitation threshold and refractory period. Thus, we justify the selection of these parameters for the construction of the actin machine. The actin droplet machine is designed and analysed in §3. Section 4 discusses the results in a context of cytoskeleton computing and outlines directions for future research.

## 2. Methods

The overall approach is the following: we simulate the actin bundle network using three-dimensional arrays of finite-state machines, cellular automata. We select several domains of the network and assign them as inputs and outputs. We represent Boolean logic values with spikes of electrical activity, which are schematically represented as a virtual experiment in figure 1. We stimulate the network with all possible configurations of input strings and record spikes on the outputs. Based on the mapping of configurations of input spikes to output spikes, we reconstruct logical functions implemented by the network. In our design of the actin droplet machine, we consider outputs recorded on all electrodes at a given time step as a binary string and then represent the actin droplet machine as a finite-state machine whose states are binary strings of a given length.

### 2.1. Three-dimensional actin network

As a template for our actin droplet machine, we used an actual three-dimensional actin bundle network produced in laboratory experiments with purified proteins (figure 2). The underlying experimental method was shown to reliably produce regularly spaced bundle networks from homogeneous filament solutions inside small isolated droplets in the absence of molecular motor-driven processes or other accessory proteins [15]. These structures effectively form very stable and long-living three-dimensional networks, which can be readily imaged with confocal microscopy resulting in stacks of optical two-dimensional slices (figure 2). Dimensions of the network are the following: size along  $x$



**Figure 1.** A scheme of a virtual experiment. The actin bundle network is shown as a three-dimensional Delaunay triangulation. Electrodes are shown by thick lines and labelled  $E_1$  to  $E_5$ . Exemplary trains of spikes are shown near the electrodes.

coordinate is  $225 \mu\text{m}$  (width), along  $y$  coordinate is  $222 \mu\text{m}$  (height), along  $z$  coordinate is  $112 \mu\text{m}$  (depth), voxel width is  $0.22 \mu\text{m}$ , height  $0.22 \mu\text{m}$  and depth  $4 \mu\text{m}$ .

Original image:  $A_z = (a_{ijz})_{1 \leq i, j \leq n, 1 \leq z \leq m}$ ,  $a_{ijz} \in \{r_{ijz}, g_{ijz}, b_{ijz}\}$ , where  $n = 1024$ ,  $m = 30$ ,  $r_{ijz}, g_{ijz}, b_{ijz}$  are RGB values of the element at  $ijz$ ,  $1 \leq r_{ijz}, g_{ijz}, b_{ijz} \leq 255$  was converted to a conductive matrix  $C = (c_{ijz})_{1 \leq i, j \leq n, 1 \leq z \leq m}$  as follows:  $c_{ijz} = 1$  if  $r_{ijz} > 40$ ,  $g_{ijz} > 19$  and  $b_{ijz} > 19$ . The conductive matrices are shown in figure 3. The three-dimensional conductive matrix is compressed along the  $z$ -axis to reduce consumption of computational resources, scenario of the non-compressed matrix will be considered in future papers.

## 2.2. Automaton model

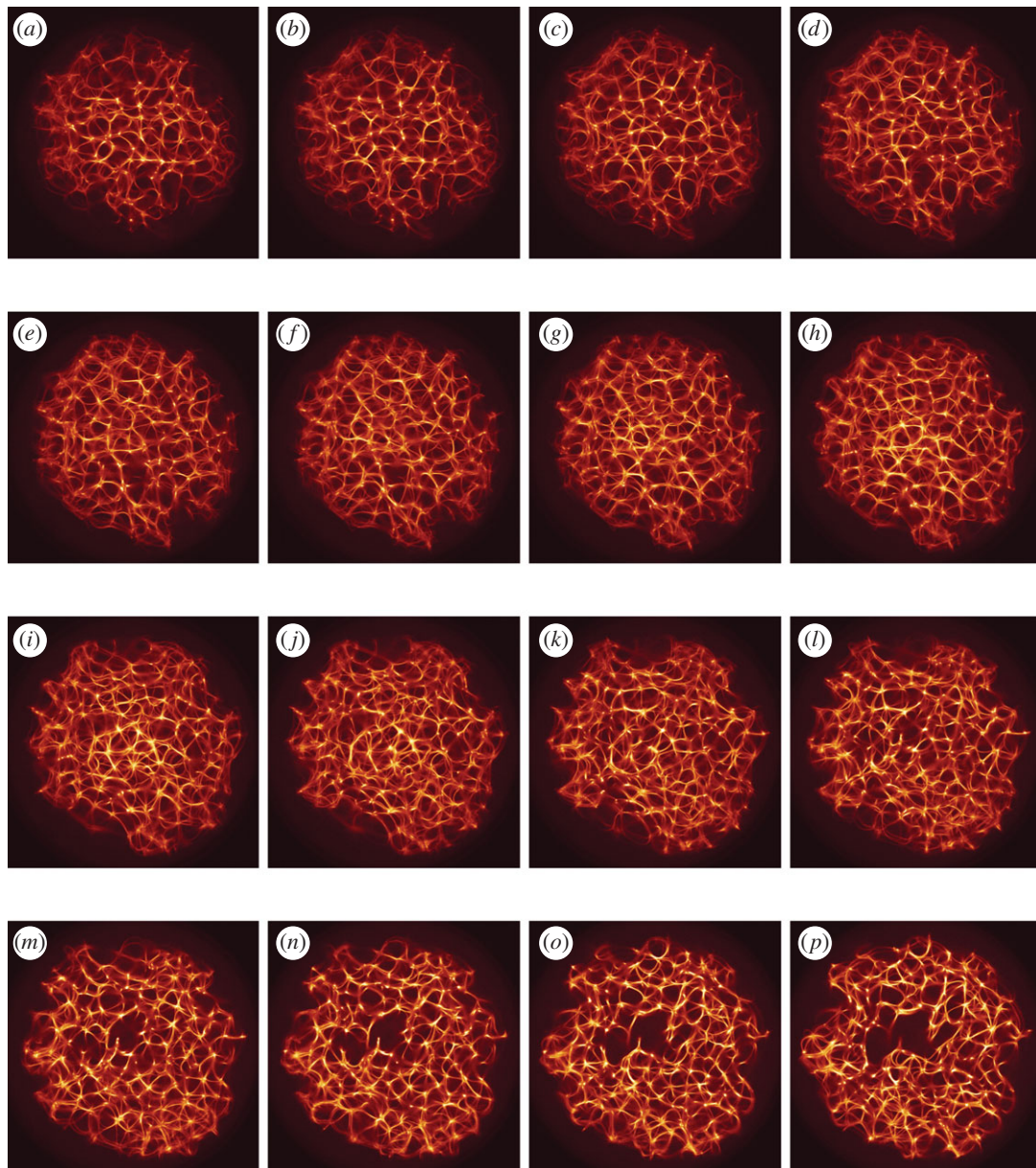
To model activity of an actin bundle network we represent it as an automaton  $\mathcal{A} = \langle C, Q, r, h, \theta, \delta \rangle$ .  $C \subset \mathbf{Z}^3$  is a set of voxels, or a conductive matrix  $C$  defined in §2.1. Each voxel  $p \in C$  takes states from the set  $Q = \{\star, \bullet, \circ\}$ , excited ( $\star$ ), refractory ( $\bullet$ ), resting ( $\circ$ ) and is complemented by a counter  $h_p$  to handle the temporal decay of the refractory state. Following discrete time steps, each voxel  $p$  updates its state depending on its current state and the states of its neighbourhood  $u(p) = \{q \in C : d(p, q) \leq r\}$ , where  $d(p, q)$  is an Euclidean distance between voxels  $p$  and  $q$ ;  $r \in \mathbf{N}$  is a neighbourhood radius.  $\theta \in \mathbf{N}$  is an excitation threshold and  $\delta \in \mathbf{N}$  is refractory delay. All voxels update their states in parallel and by the same rule:

$$p^{t+1} \begin{cases} \star, & \text{if } (p^t = \circ) \text{ and } (\sigma(p)^t > \theta) \\ \bullet, & \text{if } (p^t = \star) \text{ or } ((p^t = \bullet) \text{ and } (h_p^t > 0)) \\ \circ, & \text{otherwise} \end{cases}$$

and

$$h_p^{t+1} = \begin{cases} \delta, & \text{if } (p^{t+1} = \bullet) \text{ and } (p^t = \star) \\ h_p^t - 1, & \text{if } (p^{t+1} = \bullet) \text{ and } (h_p^t > 0) \\ 0, & \text{otherwise.} \end{cases}$$

Every resting ( $\circ$ ) voxel of  $C$  excites ( $\star$ ) at the moment  $t + 1$  if a number of its excited neighbours at the moment  $t$ ,  $\sigma(p)^t = |\{q \in u(p) : q^t = \star\}|$ , exceeds a threshold  $\theta$ . An excited voxel  $p^t = \star$  takes the refractory



**Figure 2.** Exemplary  $z$ -slices of a three-dimensional actin bundle network reconstructed as described in [15]. (a)  $z = 1$ , (b)  $z = 2$ , (c)  $z = 3$ , (d)  $z = 4$ , (e)  $z = 5$ , (f)  $z = 6$ , (g)  $z = 7$ , (h)  $z = 8$ , (i)  $z = 9$ , (j)  $z = 10$ , (k)  $z = 11$ , (l)  $z = 12$ , (m)  $z = 13$ , (n)  $z = 14$ , (o)  $z = 15$  and (p)  $z = 16$ .

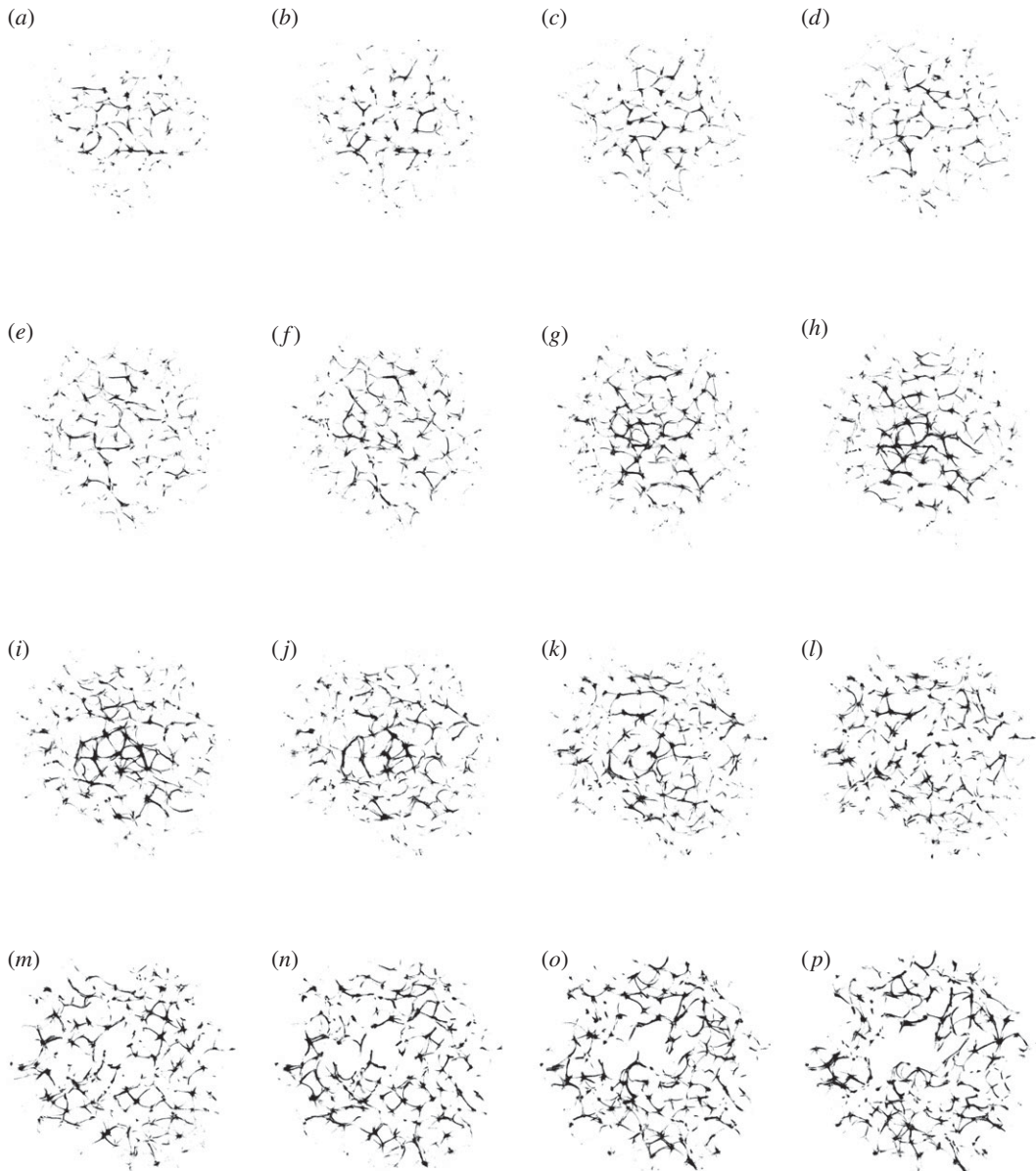
state  $\bullet$  at the next time step  $t + 1$  and at the same moment a counter of refractory state  $h_p$  is set to the refractory delay  $\delta$ . The counter is decremented,  $h_p^{t+1} = h_p^t - 1$  at each iteration until it becomes 0. When the counter  $h_p$  becomes zero the voxel  $p$  returns to the resting state  $\circ$ . For all results shown in this manuscript, the neighbourhood radius was set to  $r = 3$ . Choices of  $\theta$  and  $\delta$  are considered in §2.4.

### 2.3. Interfacing with the network

To stimulate the network and to record activity of the network, we assigned several domains of  $\mathbf{C}$  as electrodes. We calculated a potential  $p_x^t$  at an electrode location  $c \in \mathbf{C}$  as  $p_c = |\{z : d(c, z) < r_e \text{ and } z^t = \bullet\}|$ , where  $d(c, z)$  is an Euclidean distance between sites  $x$  and  $z$  in three-dimensional space. We have chosen an electrode radius of  $r_e = 4$  voxels and conducted two families of experiments with two configurations of electrodes.

In the first family of experiments  $\mathcal{E}_1$ , we studied frequencies of two-input-one-output Boolean functions implementable in the network. We used 10 electrodes, their coordinates are listed in table 1 and a configuration is shown in figure 4a. Electrodes  $E_0$  representing input  $x$  and  $E_9$  representing input  $y$  are





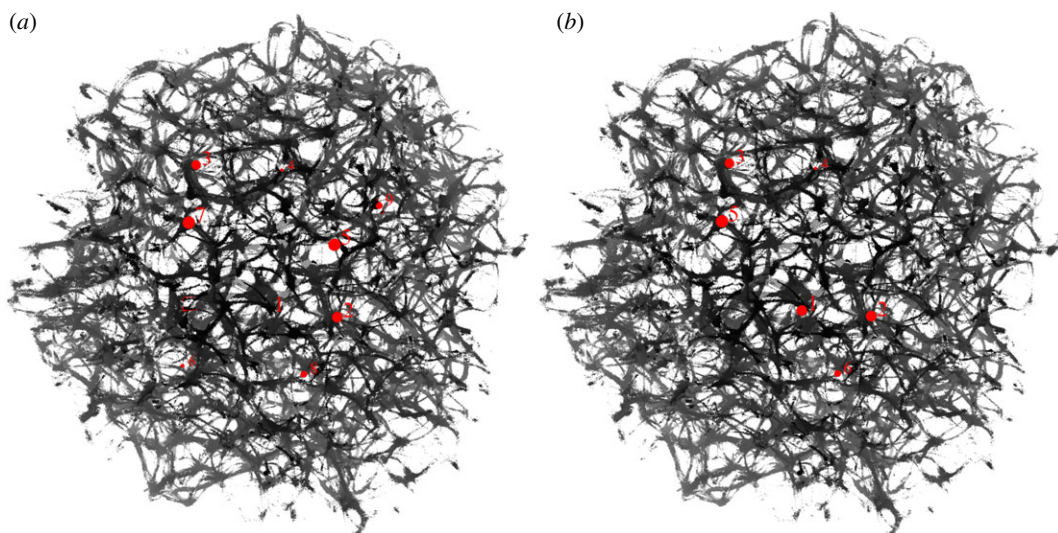
**Figure 3.** Exemplary  $z$ -slices of ‘conductive’ geometries  $C$  selected from the three-dimensional actin bundle network shown in figure 2, which were reconstructed as described in [15]. (a)  $z = 1$ , (b)  $z = 2$ , (c)  $z = 3$ , (d)  $z = 4$ , (e)  $z = 5$ , (f)  $z = 6$ , (g)  $z = 7$ , (h)  $z = 8$ , (i)  $z = 9$ , (j)  $z = 10$ , (k)  $z = 11$ , (l)  $z = 12$ , (m)  $z = 13$ , (n)  $z = 14$ , (o)  $z = 15$  and (p)  $z = 16$ .

the input electrodes, all others are output electrodes representing outputs  $z_1, \dots, z_8$ . Results are presented in §2.4. In the second family of experiments  $\mathcal{E}_2$ , we used six electrodes (table 2 and figure 4*b*). All electrodes were considered as inputs during stimulation and outputs during recording of the network activity.

Exemplary snapshots of excitation dynamics on the network are shown in figure 5. Domains corresponding to the two electrodes  $e_0$  and  $e_9$  (table 1 and figure 4*a*) have been excited (figure 5*a*). The excitation wave fronts propagate away from  $e_0$  and  $e_9$  (figure 5*b*). The fronts traverse the whole breadth of the network (figure 5*c*). Due to the presence of circular conductive paths in the network, the repetitive patterns of activity emerge (figure 5*d*). Recordings of potential and videos of experiments are available within the Zenodo repository [43].

## 2.4. Selecting excitation threshold and refractory delay to maximize a number of logical gates

To design an actin droplet machine with complex behaviour, we need to find values of refractory delay and excitation threshold for which the actin bundles network executes a maximum of Boolean gates. To map dynamics of the network onto sets of gates, we undertook the following trials of stimulation:



**Figure 4.** Configurations of electrodes in the three-dimensional network of actin bundles used in (a)  $\mathcal{E}_1$  and (b)  $\mathcal{E}_2$ . Depth of the network is shown by level of grey. Sizes of the electrodes are shown in perspective.

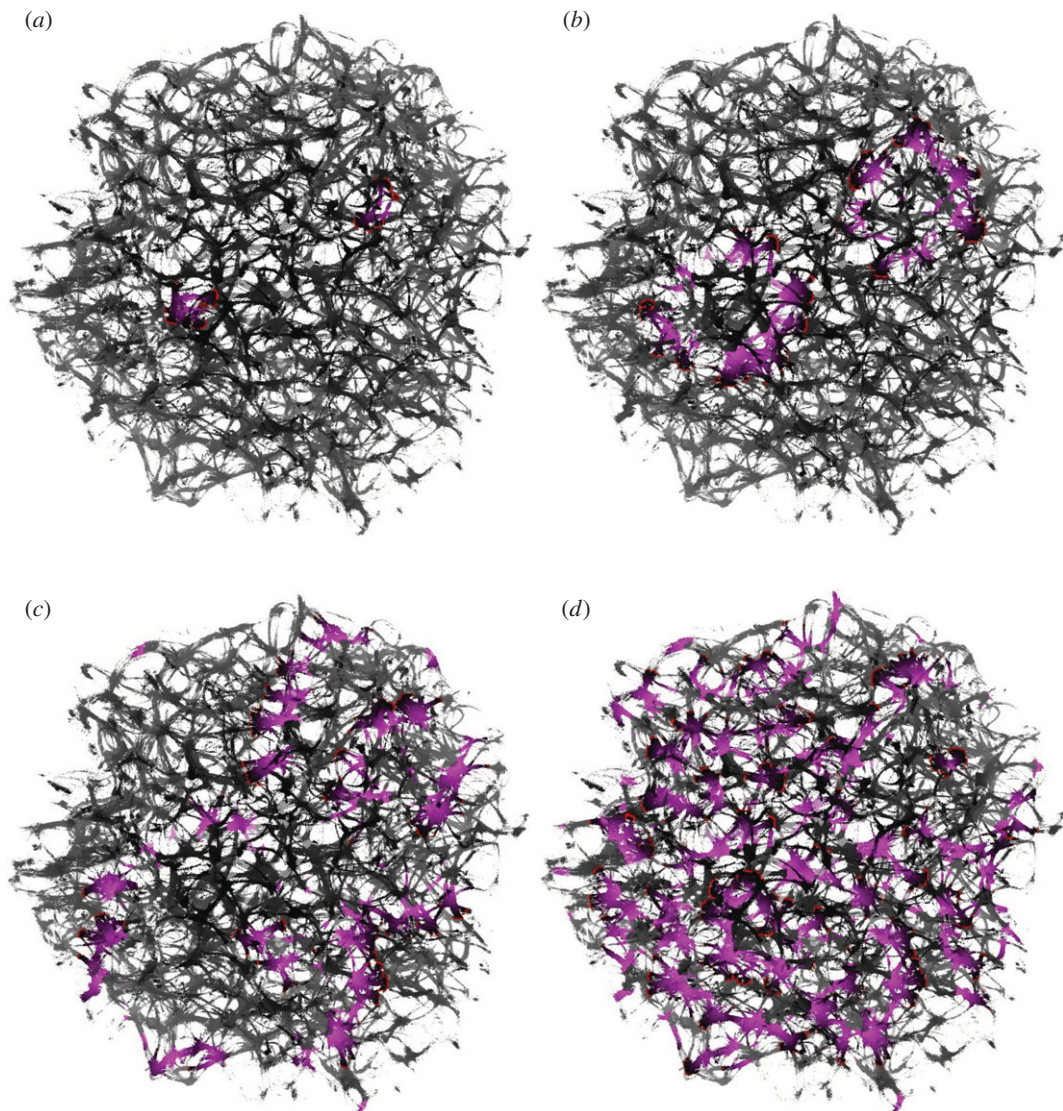
**Table 1.** Coordinates of electrodes in experiments family  $\mathcal{E}_1$ .

$e$	$i$	$j$	$z$
1	369	567	6
2	509	580	10
3	631	590	10
4	382	322	12
5	533	331	23
6	626	463	7
7	358	676	22
8	369	424	7
9	572	691	17
10	705	394	17

1. fixed refractory delay  $\delta = 20$  and excitation threshold  $\theta = 4, 5, \dots, 12$ ,
2. fixed excitation threshold  $\theta = 7$ , and refractory delay  $\delta = 10, 15, 17, \dots, 24, 30$ .

An example of the network spiking activity as a response to stimulation is shown in figure 1. We stimulated the network with all possible configurations of inputs, recorded the network's electrical dynamics and then extracted logical gates as follows. For each possible combination  $(i, j, k)$ ,  $1 \leq i, j, k \leq 6$ ,  $i \neq j$ ,  $i \neq k$ ,  $j \neq k$ , we considered electrodes  $\mathcal{E}_i$  and  $\mathcal{E}_j$  to be inputs, representing Boolean variables  $x$  and  $y$ , respectively, and electrode  $\mathcal{E}_k$  as output electrode, representing the result of a Boolean function. To input  $x = \text{TRUE}$ , we applied a current to electrode  $\mathcal{E}_i$ , to input  $y = \text{TRUE}$  to electrode  $\mathcal{E}_j$ . Then we recorded the potential at electrode  $\mathcal{E}_k$ . Two-input-one-output logical functions were extracted from the spiking events as follows. Assume each spike represents logical TRUE and that spikes being less than six iterations closer to each other happen at the same moment. Then a representation of gates by spikes and their combination will be as shown in figure 6.

For each combination  $(\rho, \theta)$ , we counted the numbers of gates OR  $(x + y)$ , AND  $(xy)$ , XOR  $(x \oplus y)$ , NOT-AND  $(\bar{x}y)$ , AND-NOT  $(x\bar{y})$  and SELECT  $(x, y)$ . We found that overall the total number of gates  $\nu(\theta)$  realized by the network decreases with increase of  $\theta$  (figure 7a). The function  $\nu(\theta)$  is nonlinear and could be adequately described by a five degree polynomial. The function reaches its maximal value at  $\theta = 7$  (figure 7a). OR gates are most commonly realized at  $\theta = 11$ , AND gates at  $\theta = 6$  and XOR gates at  $\theta = 5$  as well as  $\theta = 7$



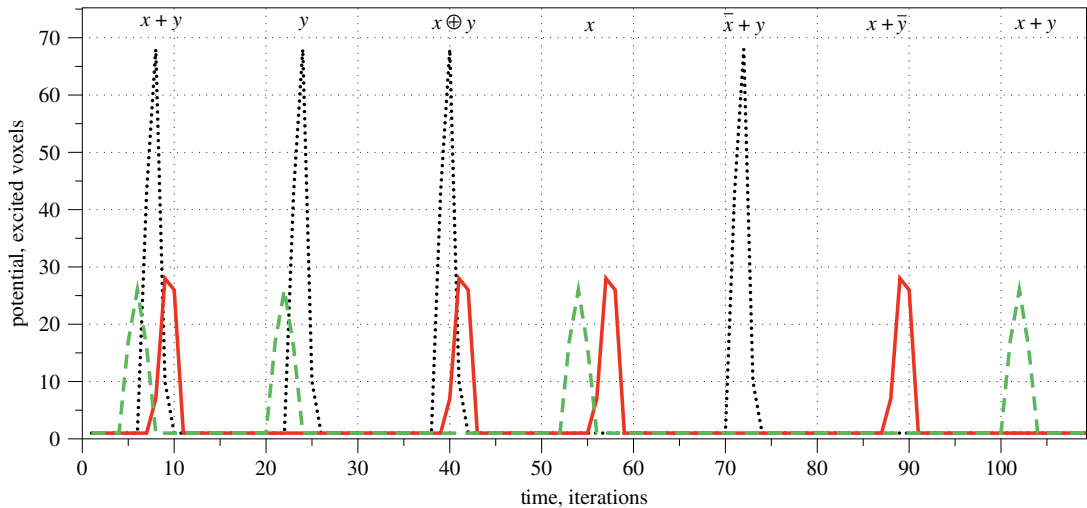
**Figure 5.** Snapshots of excitation dynamics on the network. The excitation wavefront is red and the refractory tail is magenta. The excitation threshold is  $\theta = 7$  and the refractory delay is  $\delta = 20$ , (a)  $t = 13$ , (b)  $t = 50$ , (c)  $t = 200$  and (d)  $t = 500$ .

**Table 2.** Coordinates of electrodes in experiments family  $\mathcal{E}_2$ .

$e$	$i$	$j$	$z$
1	369	567	6
2	509	580	10
3	631	590	10
4	382	322	12
5	533	331	23
6	369	424	7
7	572	691	17
8	705	394	17

(figure 7*b*). A number of AND-NOT gates implemented by the network reaches its highest value at  $\theta = 6$  then drops sharply after  $\theta_8$  (figure 7*c*). NOT-AND gates are more common at  $\theta = 5, 7, 9, 11$ , while SELECT( $x$ ) has its peak at  $\theta = 7$  and SELECT( $y$ ) at  $\theta = 8, 9$  (figure 7*c*). A total number of gates realized in the network with the excitability threshold fixed to  $\theta = 7$  decreases with the increase of  $\delta$ . Oscillations





**Figure 6.** Representation of two-inputs-one-output Boolean gates by combinations of spikes. Black dotted line shows the potential at an output electrode when the network was stimulated by input pair  $(x, y) = (\text{FALSE}, \text{TRUE})$ , red solid by  $(\text{TRUE}, \text{FALSE})$  and green dashed by  $(x, y) = (\text{TRUE}, \text{TRUE})$ .

of  $\nu(\delta)$  are visible at  $15 \leq \delta \leq 25$  (figure 7d). The three highest values of  $\nu(\delta)$  are achieved at  $\delta = 10, 17$  and  $20$ . Let us look now at the dependence of the numbers of OR, AND and XOR gates of the refractory delay  $\delta$  in figure 7e. The number of OR gates increases with  $\delta$  increasing from 10 to 15, but then drops substantially at  $\delta = 18$  to reach its maximum at  $\delta = 19$ . Numbers of gates AND and XOR behave similarly to each other. They both have a pronounced peak at  $\delta = 20$  (figure 7e).

The gate frequency analysis presented in this section allows us to choose  $\theta = 7$  and  $\delta = 20$  for an actin droplet machine constructed in the next section.

### 3. Actin droplet machine

An actin droplet machine is defined as a tuple  $\mathcal{M} = \langle \mathcal{A}, k, \mathbf{E}, \mathbf{S}, F \rangle$ , where  $\mathcal{A}$  is an actin network automaton, defined in §2.2,  $k$  is a number of electrodes,  $\mathbf{E}$  is a configuration of electrodes,  $\mathbf{S} = \{0, 1\}^k$ ,  $F$  is a state-transition function  $F: \mathbf{S} \rightarrow \mathbf{S}$  that implements a mapping between sets of all possible configurations of binary strings of length  $k$ . In the experiments reported here  $k = 6$ .

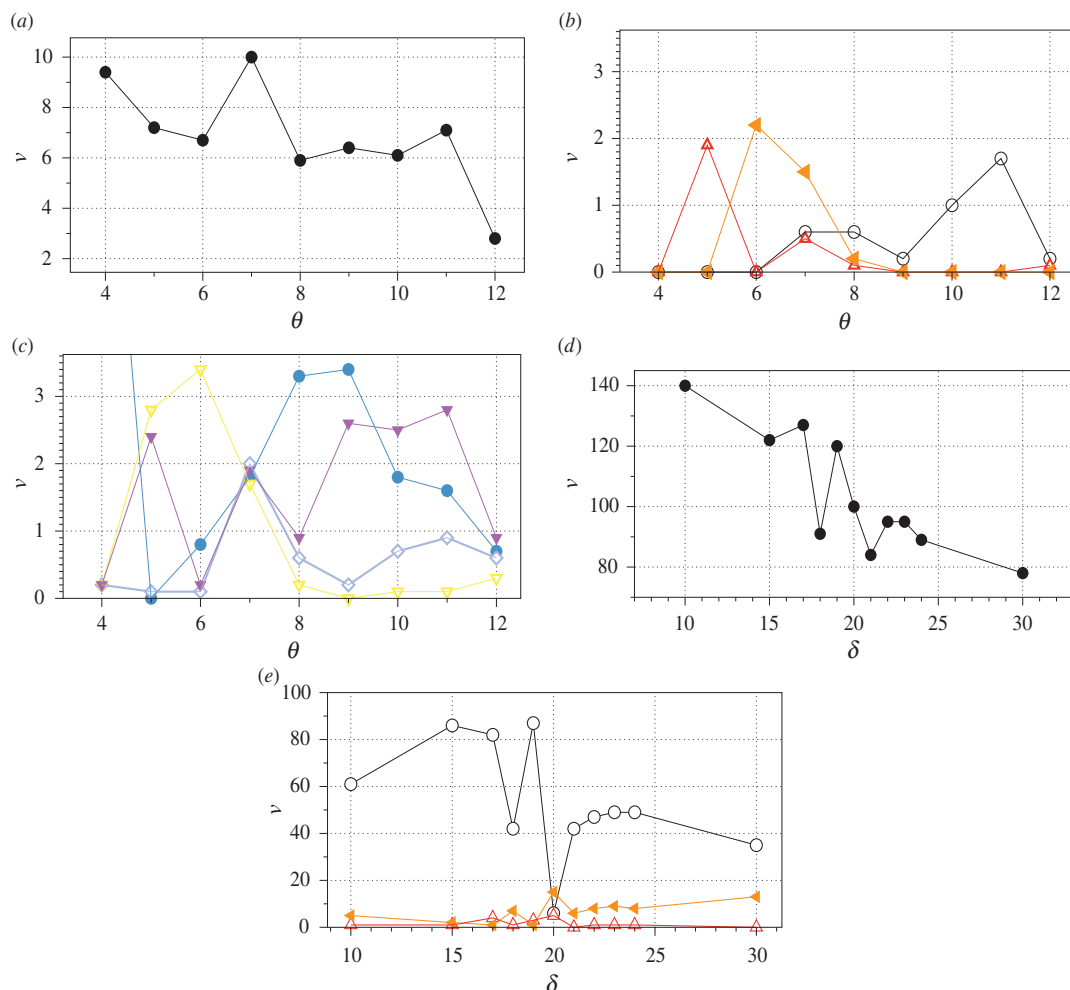
In our experiments, we have chosen six electrodes, their locations are shown in figure 4b and exact coordinates in table 2. Thus,  $F: \{0, 1\}^6 \rightarrow \{0, 1\}^6$  and the machine  $\mathcal{M}$  has 64 states. We represent the inputs and the machine states in decimal encoding. Spikes detected in response to every input from  $\{0, 1\}^6$  are shown in figure 8.

Global transition graphs of  $\mathcal{M}$  for selected inputs are shown in figure 9. Nodes of the graphs are states of  $\mathcal{M}$ , edges show transitions between the states. These directed graphs are defined as follows. There is an edge from node  $a$  to node  $b$  if there is such  $1 \leq t \leq 1000$  that  $\mathcal{M}^t = a$  and  $\mathcal{M}^{t+1} = b$ .

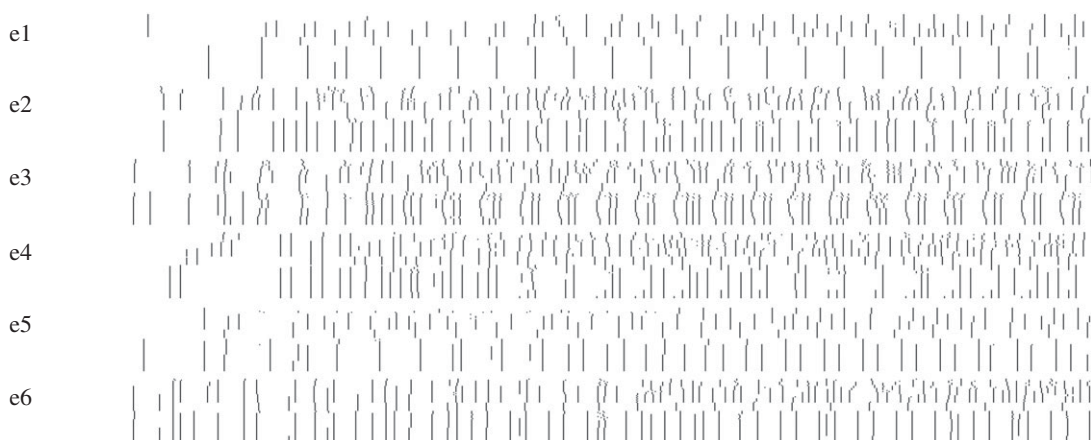
Let us now define a weighted global transition graph  $\mathcal{G} = \langle \mathbf{Q}, \mathbf{E}, w \rangle$ , where  $\mathbf{Q}$  is a set of nodes (isomorphic to the  $\{0, 1\}^6$ ), and  $\mathbf{E}$  is a set of edges, and weighting function  $w: \mathbf{E} \rightarrow [0, 1]$  assigning a number of a unit interval to each edge. Let  $a, b \in \mathbf{Q}$  and  $e(a, b) \in \mathbf{E}$  then a normalized weight is calculated as  $w(e(a, b)) = \left( \sum_{i \in \mathbf{Q}, t \in \mathbf{T}} \chi(s^t = a \text{ and } s^{t+1} = b) \right) / \left( \sum_{d \in \mathbf{Q}, t \in \mathbf{T}} \sum_{\mathbf{Q}, t \in \mathbf{T}} \chi(s^t = a \text{ and } s^{t+1} = d) \right)$ , with  $\chi$  takes value '1' when the conditions are true and '0' otherwise. In words,  $w(e(a, b))$  is a number of transitions from  $a$  to  $b$  observed in the evolution of  $\mathcal{M}$  for all possible inputs from  $\mathbf{Q}$  during time interval  $\mathbf{T}$  normalized by a total number of transition from  $a$  to all other nodes. The graph  $\mathcal{G}$  is visualized in figure 10a. Nodes which have predecessors are 1–6, 8–10, 12, 16–21, 24, 25, 28, 32–34, 36–38, 40, 41, 44, 48–50, 52, 53, 56. Nodes without predecessors are 7, 11, 13–15, 22, 23, 26, 27, 29–31, 35, 39, 42, 43, 45–47, 51, 54, 55, 57–63.

Let us convert  $\mathcal{G}$  to an acyclic non-weighted graph of more likely transitions  $\mathcal{G}^* \langle \mathbf{Q}, \mathbf{E}^* \rangle$ , where  $e(a, b) \in \mathbf{E}^*$  if  $w(e(a, b)) = \max\{w(e(a, c)) \mid e(a, c) \in \mathbf{E}\}$ . That is for each node we select an outgoing edge with maximum weight. The graph is a tree, see figure 10b. Most states apart from 1, 2, 4, 8, 16, 20, 32 are Garden-of-Eden configurations, which have no predecessors. Indegrees  $\nu()$  of not-Garden-of-Eden nodes are  $\nu(20) = 1$ ,

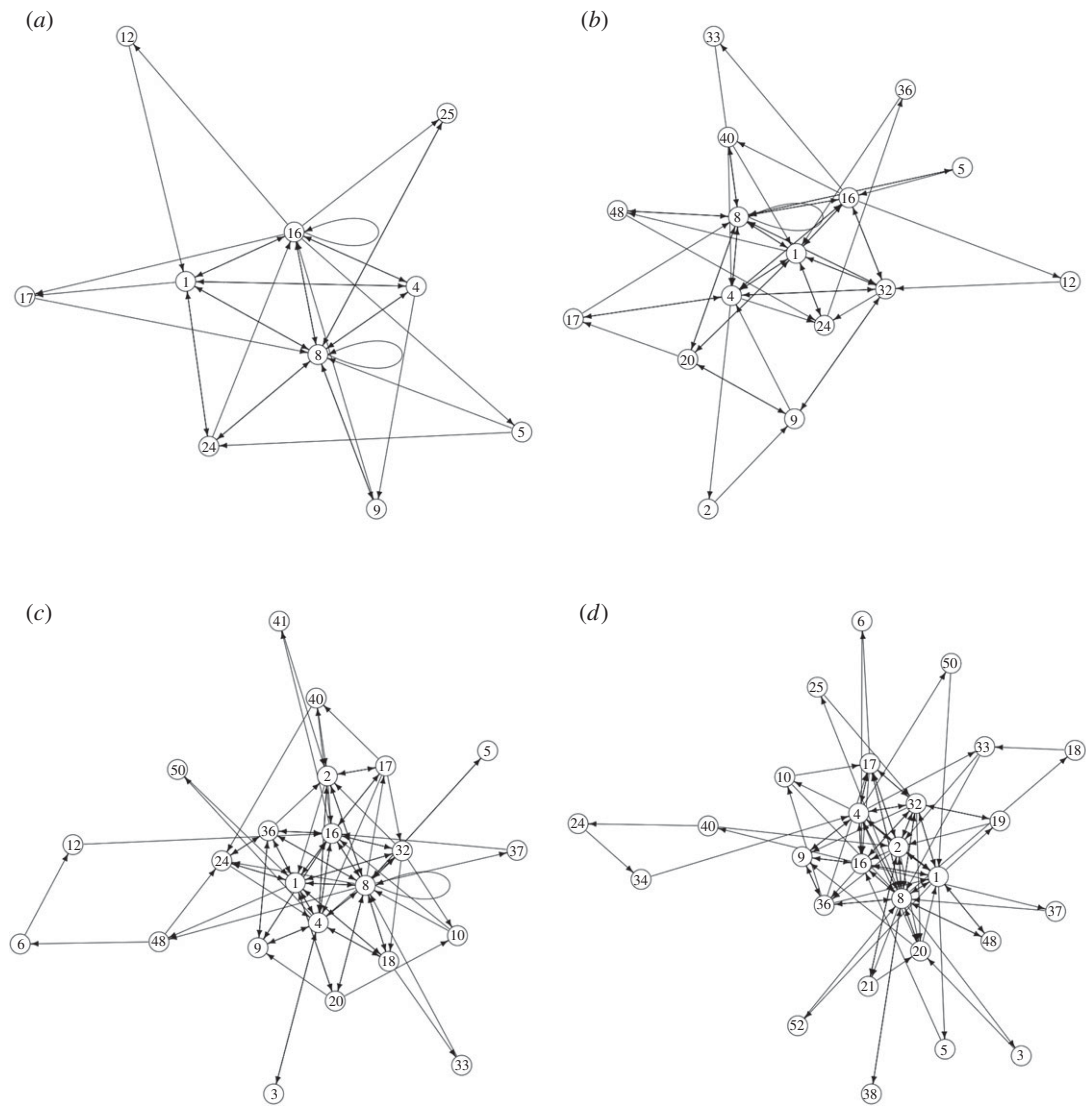




**Figure 7.** An average number  $\nu$  of gates realizable on each of the electrodes  $e_1, \dots, e_8$  depends on threshold  $\theta$  of excitation when the refractory delay  $\delta$  is fixed to 20 (abc) and on refractory delays  $\delta$  when the threshold  $\theta$  is fixed to 7 (def). (a) Number of gates  $\nu$  versus threshold  $\theta$ ,  $\delta = 20$ . (b) Number of OR (black circle), AND (orange solid triangle) and XOR (red blank triangle) gates,  $\delta = 20$ . (c) Number of NOT-AND (yellow blank triangle), AND-NOT (magenta solid triangle), SELECT(X) (cyan blank rhombus), SELECT(Y) (light blue disc),  $\delta = 20$ . (d) Number of gates  $\nu$  versus delay  $\delta$ ,  $\theta = 7$ . (e) Number of OR (black circle), AND (orange solid triangle) and XOR (red blank triangle) gates,  $\theta = 7$ .



**Figure 8.** All spikes recorded at each electrode for input binary strings from 1 to 63. The representation is implemented as follows. We stimulate the  $\mathcal{M}$  with strings from  $\{0, 1\}^6$  and represent a spike detected at time  $t$  by a black pixel at position  $t$  along horizontal axis. A plot of each electrode  $e_i$  represents a binary matrix  $\mathbf{S} = (s_{zt})$ , where  $1 \leq z \leq 63$  and  $1 \leq t \leq 1000$ :  $s_{zt} = 1$  if the input configuration was  $z$  and a spike was detected at moment  $t$ , and  $s_{zt} = 0$  otherwise.

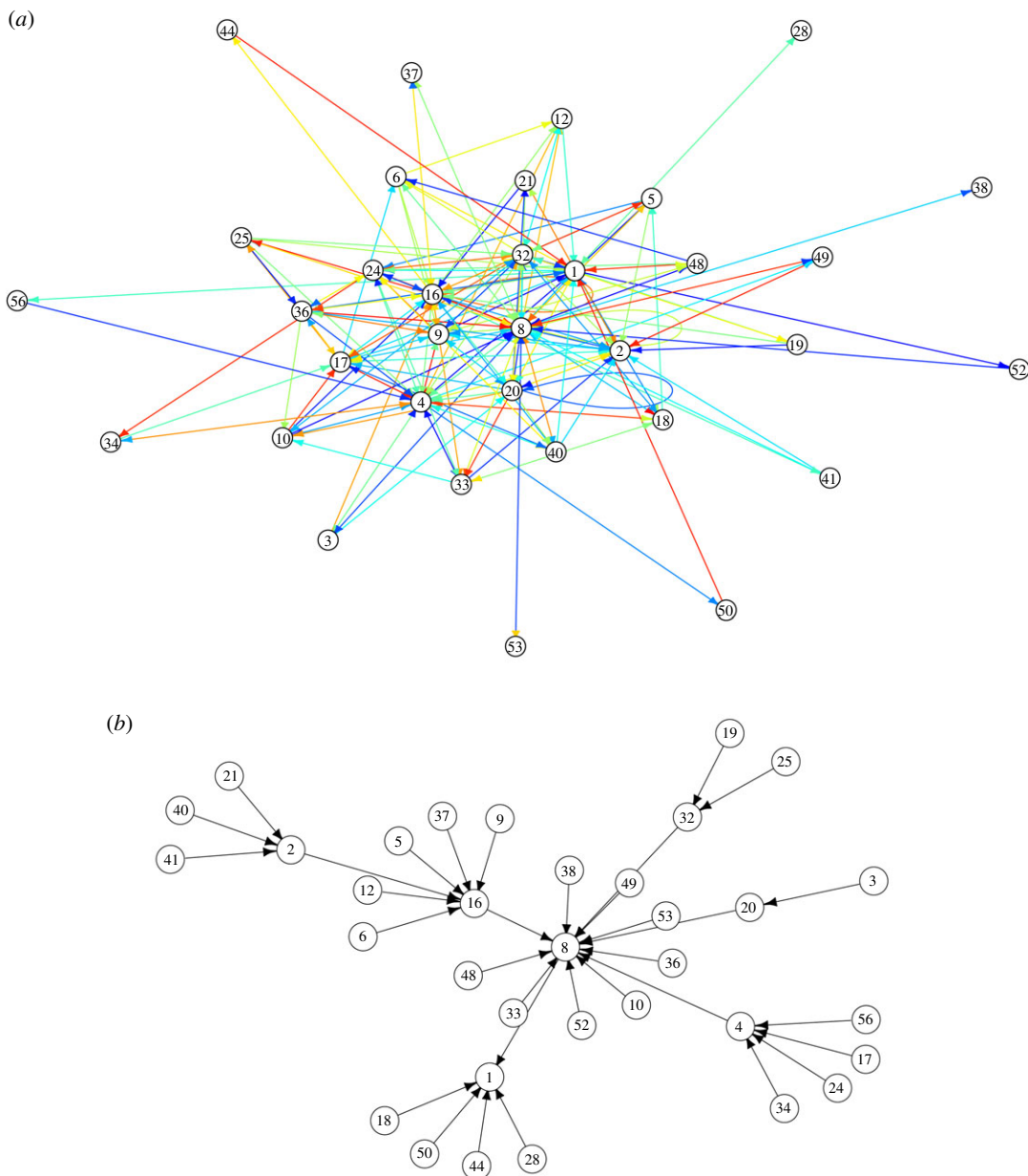


**Figure 9.** State transitions of machine  $\mathcal{M}$  for selected inputs  $I$ . A node is a decimal encoding of the  $\mathcal{M}$  state ( $e_0^t \dots e_5^t$ ). (a)  $I = 5$ , (b)  $I = 15$ , (c)  $I = 31$  and (d)  $I = 63$ .

$v(32) = 2, v(2) = 3, v(4) = 4, v(1) = 5, v(16) = 6, v(8) = 12$ . There is one fixed point, the state 1, corresponding to the situation when a spike is recorded only on electrode  $e_5$ ; it has no successors.

By analysing  $\mathcal{G}$  we can characterize a richness of  $\mathcal{M}$ 's responses to input stimuli. We define a richness as a number of different states over all inputs, as shown in table 3, and distribution in figure 11a. A number of states produced increases from under five for beginning of  $\mathcal{M}$  evolution and then reaches circa seven states on average. Oscillations around this value are seen in (figure 11a). Figure 11b shows a number of different nodes, generated in evolution of  $\mathcal{M}$ , stimulated by a given input. There are below 15 different states found in the evolution in responses to inputs 1 to 21 (21 corresponds to binary input string 010101); then a number of different nodes stay around 25. The diagram figure 11c shows how many inputs might lead to a given state/node of  $\mathcal{M}$ . Some of the states/nodes are seen to be Garden-of-Eden configurations E (nodes without predecessors) and thus could not be generated by stimulating  $\mathcal{M}$  by sequences from  $\mathbf{Q} - \mathbf{E}$ .

Assume  $\mathbf{T}$  is a set of temporal moments when the machine responded at least to one input string with a non-zero state. Configurations at each transition  $t$  can be considered as outputs representing the function  $g : \{0,1\}^6 \rightarrow \{0,1\}^6$ . As we can see in table 3, transitions at  $t = 41$  and  $t = 53$  correspond to the highest number of different binary strings ( $e_1, \dots, e_6$ ). The graph corresponding to  $g(41)$  at  $t = 41$  is shown in figure 12 and is not connected. The small component consists of fixed point 40 (string '101000') with two leafs 39 ('100111') and 38 ('100110'). The largest component has a tree structure at



**Figure 10.** (a) Global graph of  $\mathcal{M}$  state transitions. Edge weights are visualized by colours: from lowest weight in orange to highest weight in blue. (b) Pruned global graph of  $\mathcal{M}$ : only transitions with maximum weight for any given predecessors are shown, each node/state has at most one outgoing edge.

large, with cycle 2 ('000010')–1 ('000001') as a root. Other nodes with most predecessors are 8 ('001000'), 16 ('010000'), and 18 ('010010').

From the transitions  $g(41)$ , we can reconstruct Boolean functions realized at each of six electrodes (the functions are minimized and represented in a disjunctive normal form):

$$e_0 : f_0(x_0, \dots, x_5) = x_0 \cdot \bar{x}_1 \cdot x_2 \cdot x_3 + \bar{x}_0 \cdot x_1 \cdot \bar{x}_3 \cdot \bar{x}_4 \cdot \bar{x}_5 + x_0 \cdot \bar{x}_1 \cdot x_2 \cdot \bar{x}_3 \cdot x_4 + \bar{x}_0 \cdot \bar{x}_2 \cdot x_3 \cdot x_4 \cdot x_5 + \bar{x}_0 \cdot x_1 \cdot \bar{x}_2 \cdot x_3 \cdot \bar{x}_4 + \bar{x}_1 \cdot x_2 \cdot \bar{x}_3 \cdot \bar{x}_4 \cdot x_5 + \bar{x}_0 \cdot \bar{x}_1 \cdot \bar{x}_2 \cdot \bar{x}_3 \cdot \bar{x}_4 \cdot x_5 + \bar{x}_0 \cdot x_1 \cdot \bar{x}_2 \cdot x_3 \cdot x_4 \cdot \bar{x}_5$$

$$e_1 : f_1(x_0, \dots, x_5) = x_0 \cdot \bar{x}_1 \cdot x_2 \cdot x_3 + \bar{x}_0 \cdot x_1 \cdot \bar{x}_3 \cdot \bar{x}_4 \cdot \bar{x}_5 + x_0 \cdot \bar{x}_1 \cdot x_2 \cdot \bar{x}_3 \cdot x_4 + \bar{x}_0 \cdot \bar{x}_2 \cdot x_3 \cdot x_4 \cdot x_5 + \bar{x}_0 \cdot x_1 \cdot \bar{x}_2 \cdot x_3 \cdot \bar{x}_4 + \bar{x}_1 \cdot x_2 \cdot \bar{x}_3 \cdot \bar{x}_4 \cdot x_5 + \bar{x}_0 \cdot \bar{x}_1 \cdot \bar{x}_2 \cdot \bar{x}_3 \cdot \bar{x}_4 \cdot x_5 + \bar{x}_0 \cdot x_1 \cdot \bar{x}_2 \cdot x_3 \cdot x_4 \cdot \bar{x}_5$$

$$e_2 : f_2(x_0, \dots, x_5) = x_0 \cdot \bar{x}_1 \cdot x_2 \cdot x_3 + \bar{x}_0 \cdot x_1 \cdot \bar{x}_3 \cdot \bar{x}_4 \cdot \bar{x}_5 + x_0 \cdot \bar{x}_1 \cdot x_2 \cdot \bar{x}_3 \cdot x_4 + \bar{x}_0 \cdot \bar{x}_2 \cdot x_3 \cdot x_4 \cdot x_5 + \bar{x}_0 \cdot x_1 \cdot \bar{x}_2 \cdot x_3 \cdot \bar{x}_4 + \bar{x}_1 \cdot x_2 \cdot \bar{x}_3 \cdot \bar{x}_4 \cdot x_5 + \bar{x}_0 \cdot \bar{x}_1 \cdot \bar{x}_2 \cdot \bar{x}_3 \cdot \bar{x}_4 \cdot x_5 + \bar{x}_0 \cdot x_1 \cdot \bar{x}_2 \cdot x_3 \cdot x_4 \cdot \bar{x}_5$$

$$e_3 : f_3(x_0, \dots, x_5) = x_0 \cdot \bar{x}_1 \cdot x_2 \cdot x_3 + \bar{x}_0 \cdot x_1 \cdot \bar{x}_3 \cdot \bar{x}_4 \cdot \bar{x}_5 + x_0 \cdot \bar{x}_1 \cdot x_2 \cdot \bar{x}_3 \cdot x_4 + \bar{x}_0 \cdot \bar{x}_2 \cdot x_3 \cdot x_4 \cdot x_5 + \bar{x}_0 \cdot x_1 \cdot \bar{x}_2 \cdot x_3 \cdot \bar{x}_4 + \bar{x}_1 \cdot x_2 \cdot \bar{x}_3 \cdot \bar{x}_4 \cdot x_5 + \bar{x}_0 \cdot \bar{x}_1 \cdot \bar{x}_2 \cdot \bar{x}_3 \cdot \bar{x}_4 \cdot x_5 + \bar{x}_0 \cdot x_1 \cdot \bar{x}_2 \cdot x_3 \cdot x_4 \cdot \bar{x}_5$$



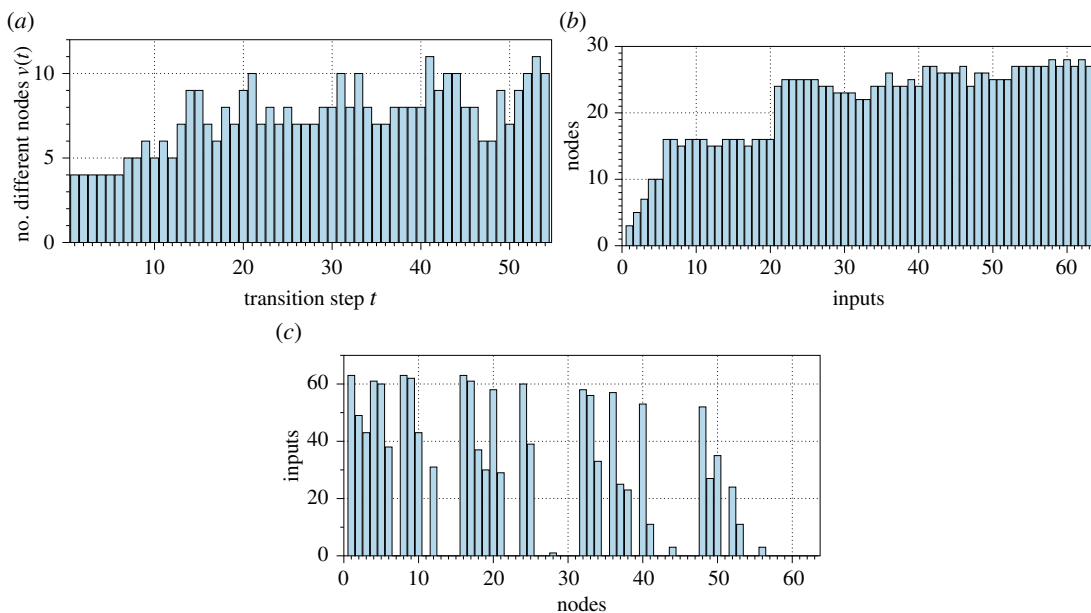
**Table 3.** Fifty-four state transitions of  $\mathcal{M}$  over all possible inputs:  $t$  is a transition step,  $\mu(t)$  is a number of different states appeared over all possible inputs,  $\mathbf{P}(t)$  is a set of nodes appeared at  $t$ .

$t$	$\mu(t)$	$\mathbf{P}(t)$
1	3	8, 9, 1,
2	3	16, 32, 8,
3	3	1, 16, 32,
4	3	8, 1, 16,
5	3	1, 8, 16,
6	3	16, 8, 1,
7	4	8, 1, 16, 4,
8	4	1, 16, 8, 5,
9	5	16, 1, 8, 4, 5,
10	4	16, 1, 8, 4,
11	5	8, 1, 16, 20, 4,
12	4	1, 16, 8, 20,
13	6	16, 8, 1, 17, 4, 20,
14	8	8, 16, 17, 4, 20, 1, 32, 2,
15	8	1, 16, 8, 4, 2, 10, 20, 32,
16	6	16, 4, 8, 1, 10, 32,
17	5	16, 1, 4, 8, 9,
18	7	8, 16, 4, 1, 17, 10, 9,
19	6	1, 8, 16, 17, 4, 10,
20	8	16, 1, 8, 17, 4, 24, 10, 2,
21	9	8, 16, 1, 17, 32, 24, 9, 4, 10,
22	6	16, 1, 8, 32, 9, 4,
23	7	8, 1, 16, 4, 32, 9, 17,
24	6	1, 16, 17, 4, 32, 8,
25	7	16, 1, 8, 4, 17, 32, 9,
26	6	8, 16, 4, 12, 1, 17,
27	6	1, 8, 16, 4, 17, 32,
28	6	16, 8, 4, 1, 24, 32,
29	7	8, 1, 4, 16, 12, 24, 32,
30	7	16, 1, 8, 4, 17, 2, 32,
31	9	8, 1, 24, 16, 12, 4, 2, 17, 32,
32	7	1, 16, 8, 24, 17, 2, 40,
33	9	16, 8, 1, 4, 40, 17, 24, 32, 2,
34	7	8, 1, 16, 24, 40, 4, 32,
35	6	1, 16, 8, 4, 24, 2,
36	6	16, 8, 1, 17, 4, 32,
37	7	8, 16, 17, 4, 1, 40, 2,
38	7	1, 8, 16, 17, 4, 24, 2,
39	7	16, 1, 8, 17, 9, 4, 2,
40	7	8, 16, 4, 1, 24, 40, 2,
41	10	1, 8, 16, 9, 17, 4, 18, 24, 40, 2,

(Continued.)

Table 3. (Continued.)

$t$	$\mu(t)$	$P(t)$
42	8	16, 1, 8, 4, 18, 33, 40, 24,
43	9	8, 1, 16, 4, 24, 33, 18, 32, 34,
44	9	1, 16, 8, 4, 17, 33, 24, 32, 40,
45	7	16, 8, 4, 1, 12, 24, 34,
46	7	8, 1, 16, 4, 24, 18, 34,
47	5	1, 16, 8, 4, 33,
48	5	16, 8, 1, 4, 17,
49	8	8, 1, 16, 4, 20, 32, 24, 19,
50	6	1, 16, 8, 4, 17, 32,
51	8	16, 8, 1, 4, 17, 32, 41, 19,
52	9	8, 16, 4, 1, 32, 33, 41, 2, 19,
53	10	1, 8, 16, 4, 20, 10, 2, 41, 32, 19,
54	9	16, 1, 8, 5, 17, 4, 2, 32, 19,



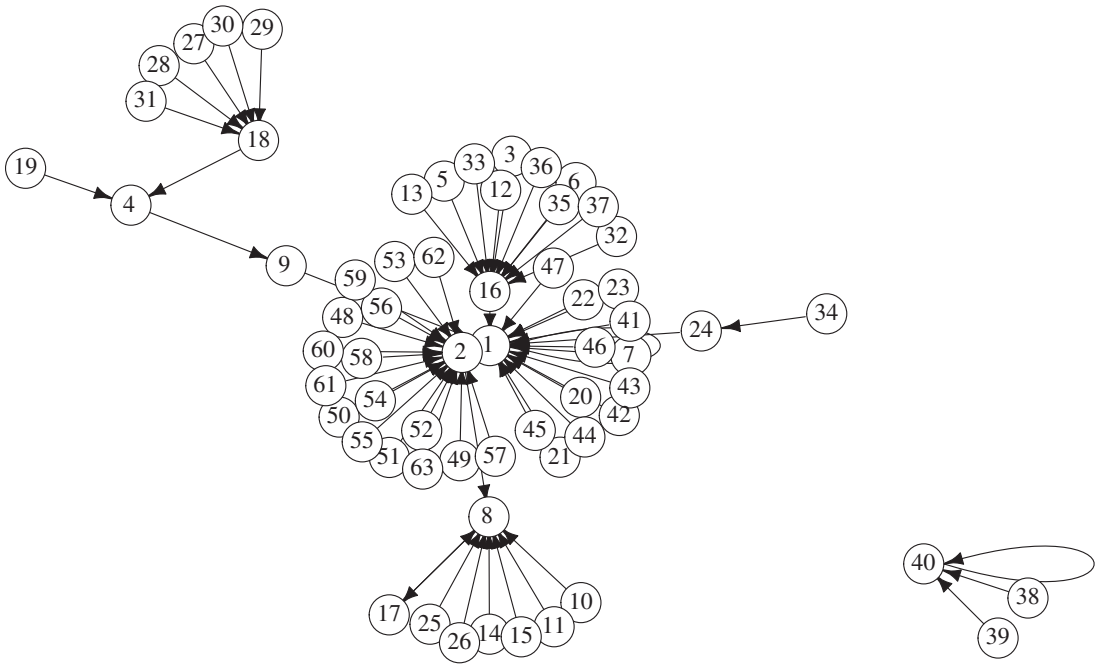
**Figure 11.** Distributions characterizing richness of  $\mathcal{M}$ 's responses. (a) Different states per transitions over all inputs. Horizontal axis shows steps of  $\mathcal{M}$  transitions. Vertical axis is a number of different states. (b) Nodes per input. Horizontal axis shows decimal values of input strings. Horizontal axis shows a number of different states/nodes generates in the evolution of  $\mathcal{M}$ . (c) Inputs per node.

$$e_4 : f_4(x_0, \dots, x_5) = x_0 \cdot \bar{x}_1 \cdot x_2 \cdot x_3 + \bar{x}_0 \cdot x_1 \cdot \bar{x}_3 \cdot \bar{x}_4 \cdot \bar{x}_5 + x_0 \cdot \bar{x}_1 \cdot x_2 \cdot \bar{x}_3 \cdot x_4 + \bar{x}_0 \cdot \bar{x}_2 \cdot x_3 \cdot x_4 \cdot x_5 + \\ \bar{x}_0 \cdot x_1 \cdot \bar{x}_2 \cdot x_3 \cdot \bar{x}_4 + \bar{x}_1 \cdot x_2 \cdot \bar{x}_3 \cdot \bar{x}_4 \cdot x_5 + \bar{x}_0 \cdot \bar{x}_1 \cdot \bar{x}_2 \cdot \bar{x}_3 \cdot \bar{x}_4 \cdot x_5 + \bar{x}_0 \cdot x_1 \cdot \bar{x}_2 \cdot x_3 \cdot x_4 \cdot \bar{x}_5$$

$$e_5 : f_5(x_0, \dots, x_5) = x_0 \cdot \bar{x}_1 \cdot x_2 \cdot x_3 + \bar{x}_0 \cdot x_1 \cdot \bar{x}_3 \cdot \bar{x}_4 \cdot \bar{x}_5 + x_0 \cdot \bar{x}_1 \cdot x_2 \cdot \bar{x}_3 \cdot x_4 + \bar{x}_0 \cdot \bar{x}_2 \cdot x_3 \cdot x_4 \cdot x_5 + \\ \bar{x}_0 \cdot x_1 \cdot \bar{x}_2 \cdot x_3 \cdot \bar{x}_4 + \bar{x}_1 \cdot x_2 \cdot \bar{x}_3 \cdot \bar{x}_4 \cdot x_5 + \bar{x}_0 \cdot \bar{x}_1 \cdot \bar{x}_2 \cdot \bar{x}_3 \cdot \bar{x}_4 \cdot x_5 + \bar{x}_0 \cdot x_1 \cdot \bar{x}_2 \cdot x_3 \cdot x_4 \cdot \bar{x}_5.$$

## 4. Discussion

Early concepts of sub-cellular computing on cytoskeleton networks as microtubule automata [44–46] and information processing in actin-tubulin networks [47] did not specify what type of ‘computation’ or ‘information processing’ the cytoskeleton networks could execute and how exactly they do this. We implemented several concrete implementations of logical gates and functions on a single actin filament [48] and on an intersection of several actin filaments [49] via collisions between solitons. We also used a reservoir-computing-like approach to discover functions on a single actin unit [50] and



**Figure 12.** Graph of  $g$  at  $t = 41$ .

filament [51]. Later, we realized that it might be unrealistic to expect someone to initiate and record travelling localizations (solitons, impulses) on a single actin filament. Therefore, we developed a numerical model of spikes propagating on a network of actin filament bundles and demonstrated that such a network can implement Boolean gates [33].

In the present paper, we reconsidered the whole idea of the information processing on actin networks and designed an actin droplet machine. The machine is a model of a three-dimensional network, based on an experimental network developed in a droplet, which executes mapping  $F$  of a space of binary strings of length  $k$  on itself. The machine acts as a finite state machine, which behaviour at a low level is governed by localizations travelling along the networks and interacting with each other. By focusing on a single element of a string, i.e. a single location of an electrode, we can reconstruct  $k$  functions with  $k$  arguments, as we have exemplified at the end of the §3. The exact structure of each  $k$ -ary function is determined by  $F$ , which, in turn, is determined by the exact architecture of a three-dimensional actin network and a configuration of electrodes.

Thus, potential future directions could be in detailed analysis of possible architectures of actin networks developed in laboratory experiments and evaluation on how far an exact configuration of electrodes affects the structure of mapping  $F$  and corresponding distribution of functions implementable by the actin droplet machine. The ultimate goal would be to implement actin droplet machines in laboratory experiments and to cascade several machines into a multi-processors computing architecture.

Conventional hardware is static. Actin networks reconfigure dynamically: some filaments disappear by depolymerization, new filaments appear by polymerization. This is not a disadvantage of the actin network computers because: (1) they operate with the speed several orders more than actin treadmilling rate, (2) actin networks can be stabilized, (3) we can employ dynamic reconfigurability in the computation.

A computation in actin bundle networks is implemented with travelling mechanical or electrical signals. Thus, we could estimate the speed of the signals propagation would be  $10^6 \mu\text{m s}^{-1}$ , for sound solitons, or  $10^5$ – $10^8 \mu\text{m s}^{-1}$ , for action potential speed [52]. Let us take the lowest estimate  $10^5 \mu\text{m s}^{-1}$ . Assuming maximum linear size of an actin droplet machine is *ca*  $250 \mu\text{m}$ , the machine can process about 400 parallel inputs per second, thus operating at 0.4 kHz frequency. Commonly, actin polymerization speed is estimated to be  $4 \times 10^{-1} \mu\text{m s}^{-1}$  [53]. An acting bundle has up to 500 actin filaments, which will not fail simultaneously. In fact, we have seen that the networks, once formed, could remain stable over hours without major rearrangements. In contrast with cells, no actin accessory proteins were used and no energy in the form of ATP was provided. The structures self-assembled solely driven by thermodynamic arguments into a stable, frozen state. If we would neglect these experimental findings and would assume an active network with high treadmilling rates, we



can consider the network being fixed for at least 10 s, which allows us to execute up to  $4 \times 10^3$  cycles of computation.

The life-time of the fixed network can be even substantially changed by using accessory proteins such as purely synthetic actin crosslinkers from DNA and peptides [54], increasing a ratio of integrin [55] and drebrin [56] peptides in the matrix solution, hardening the filaments with  $\alpha$ -actinin [57] and stabilizing the filament with synthetic mini-nebulin [58,59]. Using accessory proteins such as gelsolin, cofilin, formin and myosins would even allow us to speed up potential reconfiguration effects, enabling us to build up a dynamic computing system [6].

Dynamical reconfiguration of actin network computers can be used as an advantage for accelerating Boolean satisfiability solvers [60], reconfigurable data flow machine for implementing atomic functional programming languages [61], dynamical genetic programming on evolvable Boolean networks [62,63], and cryptographic applications [64].

Data accessibility. Supporting data are available at [43].

Authors' contributions. F.H. and J.S. provided experimental laboratory data. A.A. designed the model. All authors analysed the results and co-wrote the paper.

Competing interests. The authors declare that they have no competing interests.

Funding. A.A. was partially supported by EPSRC grant no. EP/P016677/1.

## References

1. Straub F. 1943 Actin, II. *Stud. Inst. Med. Chem. Univ. Szeged* **3**, 23–37.
2. Korn ED. 1982 Actin polymerization and its regulation by proteins from nonmuscle cells. *Physiol. Rev.* **62**, 672–737. (doi:10.1152/physrev.1982.62.2.672)
3. Szent-Györgyi AG. 2004 The early history of the biochemistry of muscle contraction. *J. Gen. Physiol.* **123**, 631–641. (doi:10.1085/jgp.200409091)
4. Straub F. 1950 Muscle. *Annu. Rev. Biochem.* **19**, 371–388. (doi:10.1146/annurev.bi.19.070150.002103)
5. Korn ED, Carlier MF, Pantaloni D. 1987 Actin polymerization and ATP hydrolysis. *Science* **238**, 638–644. (doi:10.1126/science.3672117)
6. Huber F, Schnauss J, Röncke S, Rauch P, Müller K, Fütterer C, Käs JA. 2013 Emergent complexity of the cytoskeleton: from single filaments to tissue. *Adv. Phys.* **62**, 1–112. (doi:10.1080/00018732.2013.771509)
7. Wang YL. 1984 Reorganization of actin filament bundles in living fibroblasts. *J. Cell Biol.* **99**, 1478–1485. (doi:10.1083/jcb.99.4.1478)
8. Verkhovskiy AB, Svitkina TM, Borisy GG. 1995 Myosin II filament assemblies in the active lamella of fibroblasts: their morphogenesis and role in the formation of actin filament bundles. *J. Cell Biol.* **131**, 989–1002. (doi:10.1083/jcb.131.4.989)
9. Bartles JR. 2000 Parallel actin bundles and their multiple actin-bundling proteins. *Curr. Opin Cell Biol.* **12**, 72–78. (doi:10.1016/S0955-0674(99)00059-9)
10. Schnauss J, Golde T, Schuldt C, Schmidt BUS, Glaser M, Strehle D, Händler T, Heussinger C, Käs JA. 2016 Transition from a linear to a harmonic potential in collective dynamics of a multifilament actin bundle. *Phys. Rev. Lett.* **116**, 108102. (doi:10.1103/PhysRevLett.116.108102)
11. Schnauss J, Händler T, Käs JA. 2016 Semiflexible biopolymers in bundled arrangements. *Polymers* **8**, 274. (doi:10.3390/polym8080274)
12. Strehle D, Schnauss J, Heussinger C, Alvarado J, Bathe M, Käs JA, Gentry B. 2011 Transiently crosslinked F-actin bundles. *Eur. Biophys. J.* **40**, 93–101. (doi:10.1007/s00249-010-0621-z)
13. Strehle D, Mollenkopf P, Glaser M, Golde T, Schuldt C, Käs JA, Schnauss J. 2017 Single actin bundle rheology. *Molecules* **22**, 1804. (doi:10.3390/molecules22101804)
14. Huber F, Strehle D, Käs J. 2012 Counterion-induced formation of regular actin bundle networks. *Soft Matter* **8**, 931–936. (doi:10.1039/C1SM06019H)
15. Huber F, Strehle D, Schnauss J, Käs J. 2015 Formation of regularly spaced networks as a general feature of actin bundle condensation by entropic forces. *New J. Phys.* **17**, 043029. (doi:10.1088/1367-2630/17/4/043029)
16. Tuszyński J, Hameroff S, Satačić M, Trpisova B, Nip M. 1995 Ferroelectric behavior in microtubule dipole lattices: implications for information processing, signaling and assembly/disassembly. *J. Theor. Biol.* **174**, 371–380. (doi:10.1006/jtbi.1995.0105)
17. Tuszyński J, Luchko T, Carpenter E, Crawford E. 2004 Results of molecular dynamics computations of the structural and electrostatic properties of tubulin and their consequences for microtubules. *J. Comput. Theor. Nanosci.* **1**, 392–397. (doi:10.1166/jctn.2004.042)
18. Tuszyński J, Portet S, Dixon J, Luxford C, Cantiello H. 2004 Ionic wave propagation along actin filaments. *Biophys. J.* **86**, 1890–1903. (doi:10.1016/S0006-3495(04)74255-1)
19. Tuszyński J, Brown J, Crawford E, Carpenter E, Nip M, Dixon J, Satačić MV. 2005 Molecular dynamics simulations of tubulin structure and calculations of electrostatic properties of microtubules. *Math. Comput. Modell.* **41**, 1055–1070. (doi:10.1016/j.mcm.2005.05.002)
20. Priel A, Tuszyński JA, Cantiello HF. 2006 Ionic waves propagation along the dendritic cytoskeleton as a signaling mechanism. *Adv. Mol. Cell Biol.* **37**, 163–180. (doi:10.1016/s1569-2558(06)37008-7)
21. Tuszyński J, Portet S, Dixon J. 2005 Nonlinear assembly kinetics and mechanical properties of biopolymers. *Nonlin. Anal.: Theory Methods Appl.* **63**, 915–925. (doi:10.1016/j.na.2005.01.089)
22. Satačić M, Ilić D, Ralević N, Tuszyński JA. 2009 A nonlinear model of ionic wave propagation along microtubules. *Eur. Biophys. J.* **38**, 637–647. (doi:10.1007/s00249-009-0421-5)
23. Satačić M, Sekulić D, Živanov M. 2010 Solitonic ionic currents along microtubules. *J. Comput. Theor. Nanosci.* **7**, 2281–2290. (doi:10.1166/jctn.2010.1609)
24. Satačić M, Satačić B. 2011 Ionic pulses along cytoskeletal protofilaments. In *Journal of Physics: Conference Series*, vol. 329, p. 012009. IOP Publishing.
25. Kavitha L, Parasuraman E, Muniyappan A, Gopi D, Zdravković S. 2017 Localized discrete breather modes in neuronal microtubules. *Nonlinear Dyn.* **88**, 2013–2033. (doi:10.1007/s11071-017-3359-7)
26. Adamatzky A. 2000 Collision-based computing in biopolymers and their automata models. *Int. J. Mod. Phys. C* **11**, 1321–1346. (doi:10.1142/S0129183100001176)
27. Jakubowski MH, Steiglitz K, Squier R. 2002 Computing with solitons: a review and prospectus. In *Collision-based computing* (ed. A Adamatzky), pp. 277–297. Berlin, Germany: Springer.
28. Blair S, Wagner K. 2002 Gated logic with optical solitons. In *Collision-based computing*, pp. 355–380. Berlin, Germany: Springer.
29. Toth R, Stone C, de Lacy Costello B, Adamatzky A, Bull L. 2009 Simple collision-based chemical logic gates with adaptive computing. *Int. J. Nanotechnol. Mol. Comput. (IJNMC)* **1**, 1–16. (doi:10.4018/jnmc.2009070101)
30. Zhang L, Adamatzky A. 2009 Collision-based implementation of a two-bit adder in excitable cellular automaton. *Chaos* **41**, 1191–1200. (doi:10.1016/j.chaos.2008.04.050)

31. Adamatzky A ed. 2002 *Collision-based computing*. Berlin, Germany: Springer.
32. Adamatzky A, Tuszynski J, Pieper J, Nicolau DV, Rinaldi R, Sirakoulis G, Erokhin V, Schnauss J, Smith DM. 2019 Towards cytoskeleton computers. A proposal. In *From parallel to emergent computing* (eds A Adamatzky, S AKI, G Sirakoulis). CRC Group/Taylor & Francis.
33. Adamatzky A, Huber F, Schnauss J. 2019 Computing on actin filament bundles. *Sci. Rep.* **9**, 15887. (doi:10.1038/s41598-019-51354-y)
34. Markus M, Hess B. 1990 Isotropic cellular automaton for modelling excitable media. *Nature* **347**, 56–58. (doi:10.1038/347056a0)
35. Gerhardt M, Schuster H, Tyson JJ. 1990 A cellular automaton model of excitable media including curvature and dispersion. *Science* **247**, 1563–1566. (doi:10.1126/science.2321017)
36. Weimar JR, Tyson JJ, Watson LT. 1992 Diffusion and wave propagation in cellular automaton models of excitable media. *Physica D* **55**, 309–327. (doi:10.1016/0167-2789(92)90062-R)
37. Lechleiter J, Girard S, Peralta E, Clapham D. 1991 Spiral calcium wave propagation and annihilation in *Xenopus laevis* oocytes. *Science* **252**, 123–126. (doi:10.1126/science.2011747)
38. Ye P, Entcheva E, Grosu R, Smolka SA. 2005 Efficient modeling of excitable cells using hybrid automata. In *Proc. of CMSB*. vol. 5, pp. 216–227.
39. Atienza FA, Carrión JR, Alberola AG, Álvarez JLR, Muñoz JJS, Sánchez JM, Chávarri MV. 2005 A probabilistic model of cardiac electrical activity based on a cellular automata system. *Revista Española de Cardiología (English Edition)* **58**, 41–47. (doi:10.1016/S1885-5857(06)60233-8)
40. Saxberg BE, Cohen RJ. 1991 Cellular automata models of cardiac conduction. In *Theory of heart*, pp. 437–476. Berlin, Germany: Springer.
41. Dowe M, Martin Mantel R, Barkley D. 1997 Fast simulations of waves in three-dimensional excitable media. *Int. J. Bifurcation Chaos* **7**, 2529–2545. (doi:10.1142/S0218127497001692)
42. Siregar P, Sinteff J, Julien N, Le Beux P. 1998 An interactive 3D anisotropic cellular automata model of the heart. *Comput. Biomed. Res.* **31**, 323–347. (doi:10.1006/cbmr.1998.1485)
43. Adamatzky A, Huber F, Schnauss J. 2019 Supplementary materials for the paper ‘Actin Droplet Machine’. See <http://doi.org/10.5281/zenodo.2649293>.
44. Hameroff SR, Rasmussen S. 1989 Information processing in microtubules: biomolecular automata and nanocomputers. In *Molecular Electronics*, pp. 243–257. Berlin, Germany: Springer.
45. Rasmussen S, Karampurwala H, Vaidyanath R, Jensen KS, Hameroff S. 1990 Computational connectionism within neurons: a model of cytoskeletal automata subserving neural networks. *Physica D* **42**, 428–449. (doi:10.1016/0167-2789(90)90093-5)
46. Hameroff S, Rasmussen S. 1990 *Microtubule automata: sub-neural information processing in biological neural networks*. Hackensack, NJ: World Scientific.
47. Priel A, Tuszynski JA, Cantiello HF. 2006 The dendritic cytoskeleton as a computational device: an hypothesis. In *The emerging physics of consciousness*, pp. 293–325. Berlin, Germany: Springer.
48. Siccardi S, Tuszynski JA, Adamatzky A. 2016 Boolean gates on actin filaments. *Phys. Lett. A* **380**, 88–97. (doi:10.1016/j.physleta.2015.09.024)
49. Siccardi S, Adamatzky A. 2016 Logical gates implemented by solitons at the junctions between one-dimensional lattices. *Int. J. Bifurcation Chaos* **26**, 1650107. (doi:10.1142/S0218127416501078)
50. Adamatzky A. 2017 Logical gates in actin monomer. *Sci. Rep.* **7**, 11755. (doi:10.1038/s41598-017-11333-7)
51. Adamatzky A. 2018 On discovering functions in actin filament automata. (<http://arxiv.org/abs/quant-ph/180706352>).
52. Schmidt-Nielsen K. 1997 *Animal physiology: adaptation and environment*. Cambridge, UK: Cambridge University Press.
53. Pollard TD, Mooseker MS. 1981 Direct measurement of actin polymerization rate constants by electron microscopy of actin filaments nucleated by isolated microvillus cores. *J. Cell Biol.* **88**, 654–659. (doi:10.1083/jcb.88.3.654)
54. Lorenz JS, Schnauss J, Glaser M, Sajfutdinow M, Schuldt C, Käs JA, Smith DM. 2018 Synthetic transient crosslinks program the mechanics of soft, biopolymer-based materials. *Adv. Mater.* **30**, 1706092. (doi:10.1002/adma.201706092)
55. McGeachie AB, Cingolani LA, Goda Y. 2011 A stabilising influence: integrins in regulation of synaptic plasticity. *Neurosci. Res.* **70**, 24–29. (doi:10.1016/j.neures.2011.02.006)
56. Mikati MA, Grintsevich EE, Reisler E. 2013 Drebrin-induced stabilization of actin filaments. *J. Biol. Chem.* **288**, 19 926–19 938. (doi:10.1074/jbc.M113.472647)
57. Xu J, Tseng Y, Wirtz D. 2000 Strain Hardening of actin filament networks regulation by the dynamic cross-linking protein  $\alpha$ -actinin. *J. Biol. Chem.* **275**, 35886–35892. (doi:10.1074/jbc.M002377200)
58. Pappas CT, Krieg PA, Gregorio CC. 2010 Nebulin regulates actin filament lengths by a stabilization mechanism. *J. Cell Biol.* **189**, 859–870. (doi:10.1083/jcb.201001043)
59. Chu M, Gregorio CC, Pappas CT. 2016 Nebulin, a multi-functional giant. *J. Exp. Biol.* **219**, 146–152. (doi:10.1242/jeb.126383)
60. Zhong P, Martonosi M, Ashar P, Malik S. 1998 Solving Boolean satisfiability with dynamic hardware configurations. In *International Workshop on Field Programmable Logic and Applications*, pp. 326–335. Berlin, Germany: Springer.
61. Giraud-Carrier C. 1994 A reconfigurable data flow machine for implementing functional programming languages. *Sigplan Not.* **29**, 22–28. (doi:10.1145/185009.185014)
62. Bull L. 2009 On dynamical genetic programming: simple boolean networks in learning classifier systems. *Int. J. Parallel Emergent Distrib. Syst.* **24**, 421–442. (doi:10.1080/17445760802660387)
63. Bull L. 2016 On the evolution of Boolean networks for computation: a guide RNA mechanism. *Int. J. Parallel Emergent Distrib. Syst.* **31**, 101–113. (doi:10.1080/17445760.2015.1057590)
64. Rodríguez-Henríquez F, Saqib NA, Pérez AD, Koc CK. 2007 *Cryptographic algorithms on reconfigurable hardware*. New York, NY: Springer Science & Business Media.