



## OpenAIR@RGU

### The Open Access Institutional Repository at Robert Gordon University

<http://openair.rgu.ac.uk>

This is an author produced version of a paper published in

Soft Computing (ISSN 1432-7643, eISSN 1433-7479)
--------------------------------------------------

This version may not include final proof corrections and does not include published layout or pagination.

#### Citation Details

##### Citation for the version of the work held in 'OpenAIR@RGU':

<b>GERRARD, C. E., MCCALL, J., MACLEOD, C. and COGHILL, G. M., 2015. Applications and design of cooperative multi-agent ARN-based systems. Available from <i>OpenAIR@RGU</i>. [online]. Available from: <a href="http://openair.rgu.ac.uk">http://openair.rgu.ac.uk</a></b>
-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

##### Citation for the publisher's version:

<b>GERRARD, C. E., MCCALL, J., MACLEOD, C. and COGHILL, G. M., 2015. Applications and design of cooperative multi-agent ARN-based systems. <i>Soft Computing</i>, 19 (6), pp. 1581-1594.</b>
----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

#### Copyright

Items in 'OpenAIR@RGU', Robert Gordon University Open Access Institutional Repository, are protected by copyright and intellectual property law. If you believe that any material held in 'OpenAIR@RGU' infringes copyright, please contact [openair-help@rgu.ac.uk](mailto:openair-help@rgu.ac.uk) with details. The item will be removed from the repository while the claim is investigated.

The final publication is available at Springer via <http://dx.doi.org/10.1007/s00500-014-1330-9>

# Applications and Design of Cooperative Multi-agent ARN based Systems

Claire E. Gerrard • John McCall • Christopher Macleod • George M. Coghill

Claire. E. Gerrard • John McCall

IDEAS Research Institute, Robert Gordon University, Aberdeen, AB10 7GJ, UK

email: c.e.gerrard@rgu.ac.uk

John McCall

email: j.mccall@rgu.ac.uk

Geroge M. Coghill

School of Computing Science, University of Aberdeen, Aberdeen, AB24 3FX, UK.

email: g.coghill@abdn.ac.uk

Christopher Macleod

School of Engineering, Robert Gordon University, Aberdeen, AB10 7GJ, UK.

email: chris.macleod@rgu.ac.uk

**Keywords** Artificial Biochemical Network (ABN), Artificial Chemistry, Artificial Neural Network (ANN), Swarm Robotics

**Abstract** The Artificial Reaction Network (ARN) is an Artificial Chemistry inspired by Cell Signalling Networks (CSNs). Its purpose is to represent chemical circuitry and to explore the computational properties responsible for generating emergent high-level behaviour. In previous work, the ARN was applied to the simulation of the chemotaxis pathway of *E. coli* and to the control of quadrupedal robotic gaits. In this paper, the design and application of ARN-based cell-like agents termed Cytobots are explored. Such agents provide a facility to explore the dynamics and emergent properties of multicellular systems. The Cytobot ARN is constructed by combining functional motifs found in real biochemical networks. By instantiating this ARN, multiple Cytobots are created, each of which is capable of recognizing environmental patterns, stigmergic communication with others and controlling its own trajectory. Applications in biological simulation and robotics are investigated by first applying the agents to model the life-cycle phases of the cellular slime mould *D. discoideum* and then to simulate an oil-spill clean-up operation. The results demonstrate that an ARN based approach provides a powerful tool for modelling multi-agent biological systems and also has application in swarm robotics.

## 1 Introduction

In recent years, researchers have become increasingly interested in the complex behaviours displayed by individual cells (Ford 2009; West et al. 2007). For example, the cellular slime mould *D. discoideum* (Dd), starts life as a collection of solitary amoebae which actively hunt bacterial prey. But on starvation these cells secrete a cAMP (cyclic adenosine monophosphate) signal resulting in a complex aggregation response and the formation of a travelling multicellular “slug”. Dd also has a symbiotic relationship with its bacterial prey using a primitive form of “farming” to ensure sufficient food availability within a new environment (Brock, 2011).

In order to generate this emergent high-level behaviour, a cell must be able to store and process information. This is accomplished by Cell Signalling Networks (CSNs) which function as the cell’s internal processing machinery. They do this by manipulating chemical data within elaborate networked hierarchical control structures which connect chemical species together in productive or inhibitory unions. In this way, cells are able to respond to changes within their environment, communicate with other cells, and perform internal self-maintenance operations (Bray 1995). Several researchers have highlighted the processing capabilities of these networks (Bray 1995; Arkin and Ross 1994; Bhalla 2003) and their similarities to Artificial Neural Networks (ANNs) (Bray 1995; Bhalla 2003). As discussed later, some have identified structural motifs common to many CSNs which form basic computational processing units.

The Artificial Reaction Network (ARN) is an Artificial Chemistry technique inspired by biological CSNs. The design, computational properties, mathematical formalism (Gerrard et al. 2013) and validation (Gerrard et al. 2011) of the ARN have already been discussed in detail. The ARN was previously used to simulate the chemotaxis pathway of *E. coli* (Gerrard et al, 2011), in pattern recognition and to generate complex temporal waveforms to control limbed robots (Gerrard et al. 2012a, b; 2013). Previous work focused on exploring the properties and mechanisms which lead to high-level behaviour in individual cells. The focus of this work is to explore those which result from groups of interacting cells using a new technique termed the “Cytobot”. Cytobots are cell-like agents which, under direction of their internal ARN, autonomously move within and respond to their environment. Like other Artificial Chemistry approaches (Joachimczak et al. 2013; Shen et al. 2004; Guo et al. 2009 ), a Cytobot system is composed of multiple cell-like components which communicate with each other and control their local actions via artificial chemicals. The specific objectives of the results presented here are as follows: Firstly, to explore the mechanisms and computational properties that lead to emergent high-level behaviour within and between groups of interacting cells. Secondly, to investigate applications of the ARN technique in biological simulation, and finally as a distributed robotic control system.

The following novel work is presented: 1) A complete overview of the design of Cytobots including their biological background and computational properties; and 2) a Cytobot based simulation of the life phases of *Dd*; and 3) a simulated oil-spill clean-up operation using a Cytobot swarm.

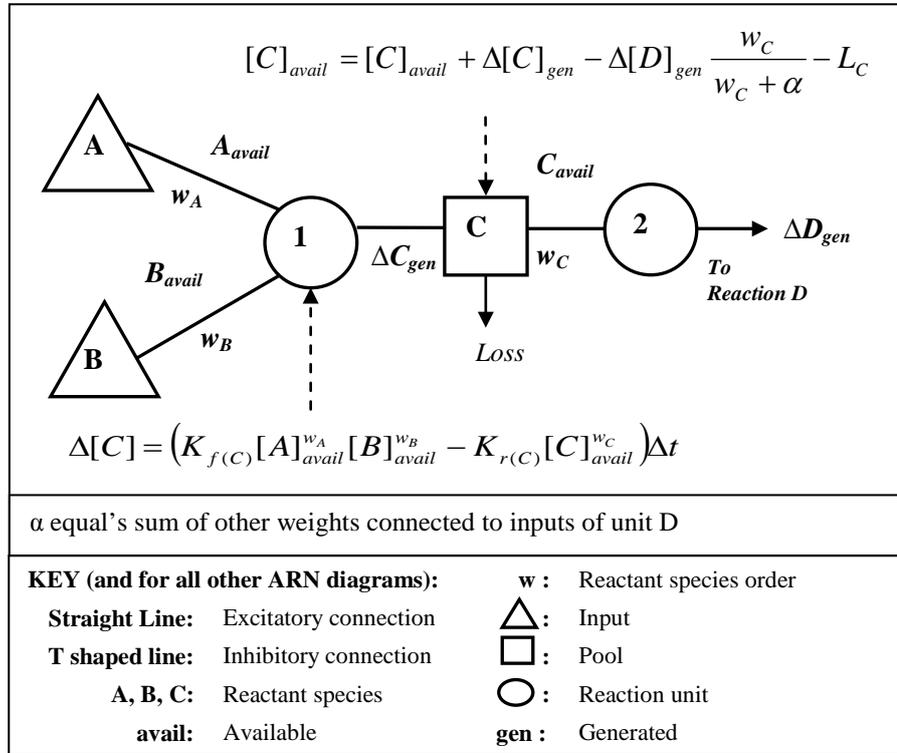
The paper is structured as follows: Section 2 briefly summarises the ARN representation. Section 3 discusses the biological background and behaviour of the Cytobots. Section 4 presents a complete overview of the Cytobot ARN design and discusses the biological functional motifs of which the network is composed. The experiments in section 5 explore the applications and properties of Cytobot systems. The first experiment (section 5.1) applies the Cytobots to the simulation of the Foraging and Aggregation phases of *Dd*. The phase times and emergent behaviours are compared with the literature. The results show that Cytobots are able to accurately model the behaviour of individual unicellular organisms, and that arising from interactions among such groups. They also demonstrate a high-level of flexibility where, for example, the pathway within an individual cell may be modified and its effects on high-level behaviour of the entire cell group viewed over time. In a further experiment (section 5.2), robotic swarm applications are investigated and a Cytobot swarm is applied to a simulated oil-spill clean-up operation. The results compare well with other related methods and show that Cytobots may have practical applications within the real-world as a physical robotic swarm.

## **2 The Artificial Reaction Network**

A brief summary of the ARN is provided here; a full account can be found in our previous paper published in this journal (Gerrard et. al, 2013).

The ARN focuses on the inherent networked properties of CSNs and is specifically designed to represent “biological circuitry”; it consists of linked processing units connected together via weighted connections and for this reason may be described as “connectionist”. It is a networked representation similar to other ACC models (Zeigler and Banzaf 2000; Eikelder et al. 2009). As shown in Fig. 1, the ARN comprises a set of networked reaction nodes (circles), pools (squares), and inputs (triangles) and is depicted as a directed weighted graph. Each pool stores the current available amount of a particular chemical species (*avail*); thus, the complete set of pool concentrations at time  $t$ , corresponds to the current state of the system. While many ACs assume a well-stirred reactor, the use of pools approximates a chemical compartment, allowing a representation of the spatial compartmentalisation which occurs within cells. This also provides a means to represent flow structures such as membrane channels and transport processes. Inputs are a special type of pool which are of fixed value and thus can be used to represent the continuous flow of environmental inputs or enzymes. Data is processed by reaction units which transform incoming pool values to connected outgoing pool values.

Connections symbolise the flow of chemical into and out of reaction units and their weight ( $w$ ) corresponds to reaction order. Connections provide a means to create complex control structures by combining inhibitory or excitatory unions.



**Fig. 1** Schematic diagram of a simplified Artificial Reaction Network (ARN). Reactant chemicals A and B react at unit 1. The rate of the reaction at unit 1 at time  $t$  is given by Eq. (1). The current concentration in pool C is updated using Eq. (2).

Figure 1 shows the reaction between species A and B to produce species C. The result of applying Euler's method to the differential rate equation is given by Eq. (1) and this is used to calculate each reaction unit's temporal flux value over the time interval  $\Delta t$ .

$$\Delta[C] = (K_{f(C)}[A]_{avail}^{w_A}[B]_{avail}^{w_B} - K_{r(C)}[C]_{avail}^{w_C})\Delta t \quad (1)$$

This result is then used to update the current concentration of each reaction's connecting pools. As mentioned previously, conserved mass values are used throughout the experiments detailed in this work and thus each pool is updated at each time interval for example at pool C.

$$[C]_{avail} = [C]_{avail} + \Delta[C]_{gen} - \Delta[D]_{gen} \frac{w_C}{w_C + \alpha} - L_C \quad (2)$$

Non-conserved values may also be modelled and the mathematics for this is given in our previous work (Gerrard et. al 2013).

### 3 Cytobot Behaviour

A Cytobot has 2 behavioural modes which are based on the chemotaxis behaviour of *D. discoideum* amoebae. These modes and their biological basis are described below.

#### 3.1 Biological Basis of Cytobot Behaviour: Chemotaxis of *D. discoideum* Amoebae

The Dd life cycle has 4 stages: Vegetation, Aggregation, Culmination and Migration, a detailed description of the biology is given by Devreotes (1989). During its Vegetative stage the organism consists of a collection of amoebae which navigate toward food by moving up gradients of folic acid secreted by their bacterial prey. The trajectory of these cells is a pattern of motion similar to a random biased walk. Dd cells extend pseudopods in a random direction; those extended toward sources of chemoattractants, such as food, are maintained; while those extended toward less favourable conditions are retracted. The overall result is movement up the gradient of attractant (Andrew and Insall 2007).

When the food resource has been depleted, the amoebae begin to starve and enter the Aggregation phase. Starving cells secrete cAMP (cyclic adenosine monophosphate), which serves as a signal to attract surrounding amoebae towards each other, resulting in a densely populated aggregate (Devreotes 1989) commonly referred to as a “mound”. Aggregating cells are polarized, thus one side becomes the leading edge, which always faces in the direction of travel (McCann et al. 2010). Depending on parameters such as environmental conditions and the cell population density, migrating cells can form transient emergent patterns such as streams and spirals (McCann et al. 2010; Dallon and Othmer 1997). Streaming describes a pattern of motion where cells line up in close-order files, with the head of one following the rear of another (McCann et al. 2010) and the spiral pattern describes streams of concentric cells spiralling toward the centre of the aggregate.

#### 3.2 Cytobot Foraging Mode

In Foraging mode, a Cytobot performs a pattern of motion based on the previously described chemotaxis of Dd cells during their Vegetative phase. The Cytobots approximate this behaviour as a random biased walk by performing alternate periods of forward motion termed “runs” and random redirections called “tumbles”. The bias is provided by reducing the tumble frequency when moving toward more favourable conditions (for example up food gradient), thus increasing the length of the run. Like aggregating Dd cells, each Cytobot is polarised, and will always face toward the direction of travel. At each new position P, an agent redirects itself to face a new random angle between 0 and 360 degrees (a tumble). The agent then moves forward in a straight line for a number of time-steps, based on the level of detected food at P (a run). The Cytobot consumes all the food (if present) at each location it passes through.

### 3.3 Cytobot Starvation Mode

The Cytobot Starvation mode is based on the pattern of motion displayed by starving cells of *Dd*. The Cytobot enters Starvation mode if it has not consumed food within a fixed time period. During this phase, the Cytobots respond to detected levels of environmental cAMP. Depending on the particular experiment, this chemical may already be present within the environment or it may be released by the starving Cytobots. In this mode, both run and tumble behaviours differ from that in the Foraging phase. Rather than turning in a random direction, a new direction is calculated by weighting the turn toward the highest concentration of artificial cAMP within the surrounding area. The run period, instead of being variable, is a fixed length, which is set according to the particular experiment.

## 4 Cytobots: Design and Implementation

In previous work it was shown that the ARN can be used to model the reactions of specific proteins involved in signalling pathways (Gerrard et al. 2011). To enable the Cytobot ARN to produce the behaviour of chemotaxing *Dd* amoeba, rather than simulating specific protein interactions, a more abstract method was employed. There are a number of reasons for this approach. For instance, there are significant gaps in our current knowledge of the chemical interactions involved within this pathway (Manahan et al. 2004), and thus it is not possible to create an accurate representation. When modelling such a network researchers often adopt a modular approach- where related signalling events are grouped into functional units (Manahan et al. 2004). In this way, the Cytobot ARN was designed by dividing functions into modular units. The functional modules are constructed by combining “structural motifs” from real biochemical networks. Such motifs, each of which perform distinct computational functions, have been identified by a number of researchers (Tyson and Novak 2010; Bray 1995; kholodenko 2006). There are a number of advantages in this approach. Firstly, these motifs are universally found in other pathways, and thus show that the ARN is capable of potentially modelling any pathway. Furthermore, by creating entire systems composed of these motifs illustrates how biologically plausible motifs can be combined and cooperate together to produce functionally distinct pathways and how such pathways cooperate (a feature of crosstalk) to produce overall cellular behaviour. These functional motifs and the manner in which they are combined within the Cytobot ARN control system is discussed below.

A summary of common structural motifs, their computational function, structure (in the previously defined ARN format), and biological examples of each is provided in Table 1. Note that these motifs are shown for simplicity as 2 or 3 component forms but there are larger versions with the same function; for example, an additional component may be added to motif 9

to create a 4 component oscillator. One important biological example is the universal signalling motif of a phosphorylation cycle. Here a signalling protein is interconverted by opposing enzymes (a kinase and a phosphatase) between its phosphorylated ( $Y_p$ ) and non-phosphorylated forms ( $Y_s$ ). In a multisite phosphorylation cycle, feedback from either form can cause oscillations between stable states or render the cycle into a bistable switch, where the low and high  $Y_p$  concentrations correspond to “on” and “off” states (kholodenko 2006). A cascade of such bistable cycles can produce multiple stable states, allowing the complex interdependent control of many cellular functions. For example, the cell’s transition into mitosis is governed by the sequential activation or inactivation of such kinases (CDK1/Cdc2) (kholodenko 2006). Space prevents a detailed discussion of each motif but an in-depth account including the biological mechanism, structure and examples is provided by Tyson and Novak (2010).

## 4.1 Functional Motifs in Biochemical Networks

**Table 1** Functional Motifs in Biochemical Networks

Motif No., Name and Description	Structure (in ARN format)	Biological Example
<b>1. Excitatory (E)</b> The presence of X activates Y		Elementary motif common throughout most pathways. E.g. Ras is a membrane associated protein that is normally activated in response to the binding of extracellular signals such as growth factors (Tyson and Novak 2010).
<b>2. Inhibitory (Y)</b> The presence of X inhibits Y. Acts as a NOT gate.		Elementary motif common throughout most pathways. E.g. E-cadherin (a calcium-dependent cell-cell adhesion molecule) suppresses cellular transformation by inhibiting β-catenin (Tyson and Novak 2010).
<b>3. Positive Feedback Loop (PFL)</b> The presence of X activates Y and in turn the presence of Y activates X		The pathway of caspase activation is essential for apoptosis induction. A PFL exists between caspase-3 and caspase-9 (Tyson and Novak 2010).
<b>4. Negative Feedback Loop (NFL)</b> The presence of X activates Y and in turn the presence of Y inhibits X		The proteins Mdm2 and p53 (p53 is a tumour suppressor protein) are involved in a NFL which functions to keep the level of p53 low in the absence of p53-stabilizing signals (Tyson and Novak 2010).
<b>5. Double-negative Feedback (DNF)</b> The presence of X inhibits Y and the presence of Y inhibits X		BAX is protein which promotes apoptosis by competing with BCL. A DNF is formed between the proteins BAX and BCL (Tyson and Novak 2010).
<b>6. Branch (B)</b> The presence of X activates Y and Z		The transcription factors such as E2F or P53 frequently modulate the expression of more than one gene. Enzymes often modify more than one substrate e.g. CycB-dependant kinase (Tyson and Novak, 2010).
<b>7. Logic Gate (LG1)</b> <b>AND gate:</b> 2 excitatory connections from X and Y when both X and Y are present they activate Z. <b>NOR gate:</b> two inhibitory connections from X and Y. Both X and Y must be absent for Z to be activated. <b>SWITCH:</b> Excitatory connection from X and inhibitory connection from Y. The presence of X but not Y activates Z		<b>AND:</b> The protein gCam 2 kinase becomes active when both calcium ions (Ca <sup>2+</sup> ) and Calmodulin (CaM) are present (Bray, 1995). <b>NOR:</b> The activity of transcription factor E2F is a NOR function of RB and CycB where E2F is active when both RB and CycB are inactive (Tyson and Novak, 2010). <b>SWITCH:</b> The enzyme aspartate transcarbamylase has multiple catalytic sites. It is activated by binding of its substrates (aspartate and carbamoyl phosphate) and inactivated by cytidine triphosphate causing its substrates to dissociate (Bray 1995).
<b>8. Logic Gate (LG2)</b> <b>OR Gate:</b> : 2 excitatory connections from X and Y when either X or Y are present they activate Z		For instance, Ras is a membrane associated protein that is activated by a number of different signals. E.g. in response to the binding of extracellular signals such as a number of growth factors (Tyson and Novak 2010).
<b>9. Oscillator (OSC)</b> The presence of X activates Y. In turn the presence of Y activates Z but inhibits X. The presence of Z inhibits Y and activates X.		The cyanobacteria clock protein KaiC has a closed cycle of phosphorylation and dephosphorylation states (composed of KaiA, KaiB and KaiC). In the structure shown left, all 3 chemicals oscillate and each inhibits the reaction clockwise left. Oscillators may have less inhibitory connections, the number of which is dependent on the mobility of the reaction species. However, the presence of all inhibitors increases stability in the presence of fluctuating environmental parameters e.g. temperature.
Key:	 Either inhibitory or excitatory. Excitatory connection Inhibitory connection	X/Y/Z: Chemical species  Reaction Chemical substrate
* Motifs may combine arbitrary numbers of components.		



MC activates run and MB and MD activates stop. Thus, if pool MC contains a chemical for 10 time-steps, the agent will move forward for 10 time-steps. The other subnetworks inhibit (Table 1, motif 2) or excite (Table 1, motif 1) the reaction units of the MOnet, to allow or prevent chemical flow.

Note that this oscillator motif allows the Cytobot ARN to function easily as the control system for the motor actuators of a wheeled robot. Here, MC would switch on all wheel motors, while MA would switch on left-wheel motors only, thus turning the robot. The remaining pools would act as off switches.

#### 4.2.2 The Food and Run Length Network

The Food Network (FNet) interfaces with the environment at pool FA, using an excitatory connection (Table 1, motif 1) and inhibits the Run Length network (RLnet) in accordance with the level of detected food. The forward rate of reaction at node F0 is 1, thus the content of FA is transferred to pool FB in a single time-step. The presence of chemical FB inhibits (Table 1, motif 2) R0 for a number of time-steps, according to the level of food (by setting forward rate of unit F1 to 1 and weight to 0, this can be an exact correlation). The RLnet is a 3 component oscillator (Table 1, motif 9). While reaction R0 is inhibited, it prevents pool RC from emptying. RC inhibits reaction M2 (Table 1, motif 2) of the MOnet thus preventing pool MC from emptying for the same number of time-steps. As discussed previously, the number of time-steps which pool MC contains the token unit represents the number of time-steps to move forward.

#### 4.2.3 The Signalling Network

The Signalling Network (Snet) functions as a switch between Starvation and Foraging mode. A low food level triggers the starvation response and allows the Weighted Direction Network (WDnet) to control each new angle. Sufficient food will switch off the WDnet and allows the Chaotic Network (Cnet) to control each new angle. It is a 3 component oscillator (Table 1, motif 9) with a token unit of chemical flowing around it. Pool CA acts as the switch between Foraging and Starvation modes. Here the presence of chemical in CA inhibits the WDnet (Table 1, motif 2), while its absence switches on the WDnet; this in turn inhibits the Cnet, as shown in Fig. 2. In this oscillatory network, all reaction units have a forward rate of 0.5. This produces a continuously oscillating waveform and ensures a minimum number of time-steps for each behaviour. A NOR gate (Table 1, motif 7) activates pool CB in the absence of food chemical in both pools FB and FC of the Fnet, thus allowing pool CB to empty. An AND gate (Table 1, motif 7) will lead pool CA to eventually refill by activating pool CC, only when food is present in input FA and pool FC of the Fnet.

#### 4.2.4 The Weighted Direction Network

The Weighted Direction Network (WDnet) senses cAMP within the agent's immediate environment and calculates a tumble angle which is weighted toward higher levels. This network interfaces with the environment via a number of receptor pools (AW, ANW, AN, ANE, AEA) which sense the level of food around the Cytobot. These pools represent receptors and are positioned at points around the front of its perimeter (as shown in Fig. 3), allowing the agent to travel in a similar way to that of an aggregating Dd cell. Each receptor input pool forms one input of an AND gate (Table 1, motif 7); the other input is a static pool containing a fixed level of chemical in correspondence to its direction. Directions start from AW (west) with a corresponding numeric value of 0 (A00) and progress in 45 degree steps through each direction to east (thus, the maximum value is 180). As the receptor positions around the agent are fixed, directions are always relative to that in which the agent is facing. All connections have a weight of 1 with the exception of the connection between pool AD and reaction A12 which has a weight of -1. This negative connection weight raises the sum of food detected in AD to -1, which multiplied by AB, allows an average angle to be calculated. Detected signals are classed as being in one of the following cardinal or ordinal directions: W, NW, N, NE, and E. Thus signals are detected from all directions above the horizontal plane. The calculated angle interfaces with the remaining subnetworks at pool AE. Pool AE is the output of an OR gate (Table 1, motif 8), and its inputs are activated by either the WDnet or the Cnet. AE also forms the inhibitory input of a SWITCH (Table 1, motif 7), where the presence of chemical in MA and absence in AE activates pool MB of the MOnet. In the actual organism, receptors are set around the cell perimeter and direct movement appropriately. In this simulation, for simplicity, a count of the number of time-steps " $n$ ", that MA contains the token unit is processed to gain the new heading " $h$ " relative to the agents' current heading " $c$ " using Eq. (3):

$$h \equiv (n - 90) + c \quad (3)$$

##### Statement 1

```
IF (h > 360) THEN h = h - 360  
IF (h < 0) THEN h = h + 360
```

Thus, if the number time-steps is 120 and the agent is facing north, then the current heading would equal 0 and the new heading would equal 30.

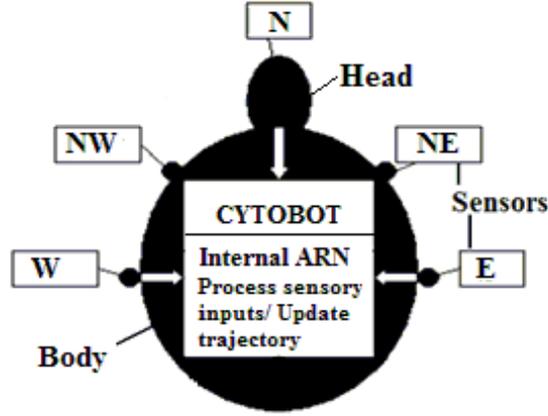


Fig. 3. Location of the Cytobot sensors around its perimeter.

#### 4.2.5 The Chaotic Network

The Chaotic Network (Cnet), shown in Fig. 2, is responsible for generating the pseudo-random angles which agents use to perform each Foraging mode tumble. It is a networked implementation of a Logistic Map, given by Eq. (4), where  $X_n$  is a state variable of value  $0 < X_n < 1$  at time-step  $n$  and  $\lambda$  is a system parameter of value  $1 \leq \lambda \leq 4$ :

$$X_{n+1} = \lambda X_n (1 - X_n) \quad (4)$$

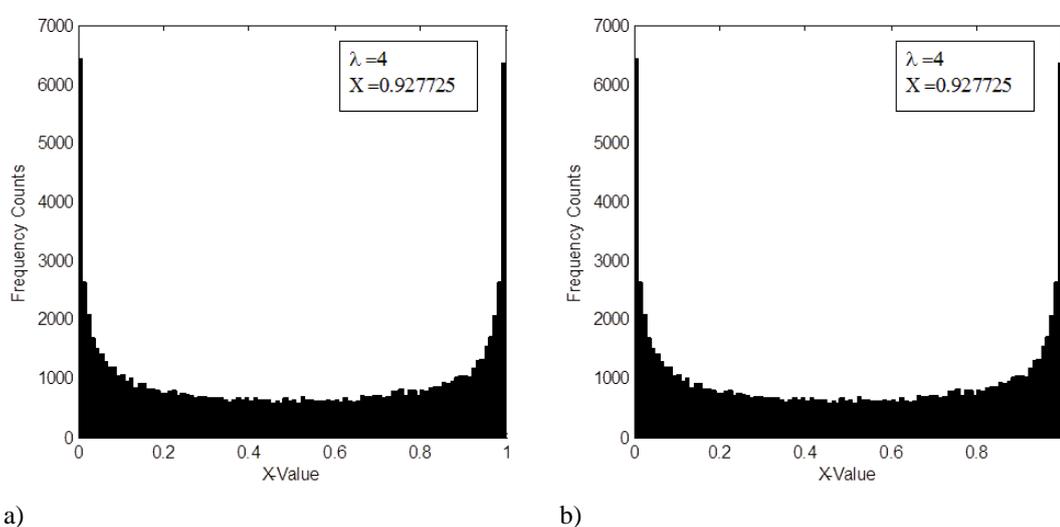
Without prior knowledge of the initial conditions, the output of the Logistic Map is not predictable; whereas, with prior knowledge it is deterministic. Therefore, the resulting series cannot be described as truly random, but as pseudo-random and its output has long been proposed as a pseudo-random number generator. Ulam and von Neumann (1947) were the first to examine this, and it has been successfully used in that capacity by several researchers (Patidar et al. 2009; Phatak and Rao 1995). The probability-density distribution of the Logistic Map, as given by Eq. (5) (where  $P(X)$  is the probability of  $X$  occurring at any time-step), is non-uniform (Patidar et al. 2009):

$$P(X) = \frac{1}{\pi \sqrt{X(1-X)}} \quad (5)$$

When  $\lambda=4$ , the distribution is “U” shaped with a higher probability of values closer to the minima and maxima of  $X$  and a symmetric distribution around the midpoint.

To implement the Logistic Map, a number of motifs are combined, including multiple branch motifs (Table 1, motif 6- KB activates KD and KE), PFLs (Table 1, motif 3- a multi component PFL exists where KA leads to activation of KE, which results in the activation of KA) and NFLs (Table 1, motif 4- KA activates KD which in turn inhibits KA). At the start of the simulation, pools KA and KB are initialized to the same random value (a unique number for

each Cytobot), between 0 and 1 (to 5 decimal places). This value represents the initial value of X of Eq. (4). All the other pools are initialized to 0, with the exception of the static pools KI and RK, whose initial values are 360 and 1 respectively. Reaction K2 is responsible for generating each new value of X and has a forward and reverse rate of 4 (the Logistic Map exhibits chaotic behaviour when  $\lambda$  is 4). The connection between KA and K2 has a weight of 1 and that between K2 and KB has a weight of 2. The remaining series of reactions function to copy the value of X 3 times; where 2 copies serve as the new initial values of KA and KB, and one participates in the final output of the network at KH. KI has a fixed value of 360 which allows the network to convert the pseudo-random number at KH to an angle value between 0 and 360 at reaction K0. However, reaction K0 cannot proceed until all 11 pools that inhibit it are empty.



**Fig. 4** The Frequency distribution for each value of X when X is initialised to 0.927725 and  $\lambda=4$  resulting from: a) the chaotic network b) Recursive relation given by Eq. (2) run using Matlab.

These inhibitory connections (Table 1, motif 2) ensure that random angles are not output while the agent is in starvation mode, and that pool AE is empty before adding more chemical.

The ARN implementation of the Logistic Map was verified against the recursive relation shown in Eq. (4) using Matlab, where  $\lambda=4$ , initial X = 0.927725, and iterated for  $1 \times 10^5$  steps. The complete range of state-variables between 0 and 1 were divided into 100 equal subintervals and the frequency of occurrence of each subinterval interval was plotted. Similarly, the Cnet was run for  $1 \times 10^5$  cycles, using the same parameters of X (initial value) and  $\lambda$ . These results were processed in the same way and are shown in Fig. 4. The frequency distribution gained from the ARN is identical to that obtained using Matlab and by other researchers using the same parameters (Patidar et al. 2009). The same comparison was repeated 100 times at different values of X, and the ARN consistently produced the same values as Eq. (2).

## 5 Experiments and Results

In the following sections the methodology and results for the following experiments are presented: 1) A Cytobot based simulation of the Vegetative and Aggregative life-cycle phases of *Dd* including the transition; between them and 2) application of the same Cytobots in a simulated oil-spill clean-up operation.

### 5.1 *D. discoideum* Simulation

#### 5.1.1 *D. discoideum* Simulation Methodology

The experiments are grouped into two sets: Aggregation (AG1-10 of Table 2) which models the Aggregation phase only and Foraging to Aggregation (AGF3 and AGF8 of Table 2) where both the Vegetative and Aggregative phases are simulated, including the transition between them. Each experiment is performed at varying population densities of Cytobots ( $p$ ) and different distance ranges of detection of cAMP ( $r$ ). The experiments AGF3 and AGF8 are performed at the same  $p$  and  $r$  as experiments AG3 and AG8 respectively to compare the effect of the Foraging phase on the number of mounds formed and length of time to complete the Aggregation phase. The emergent patterns, numbers of mounds, and length of time to complete phases is examined and compared in both sets of experiments and with the literature. In each experiment the Cytobots move within a 2D simulated environment which represents an area of  $5.06 \text{ mm}^2$ - approximately half the maximum Aggregation territory reported in the literature (Dallon and Othmer 1997). A screen output shows the position of the Cytobots in real-time and is a grid of  $500 \times 500$  pixels where each is represented by a square of side  $4.5 \mu\text{m}$ . In nature, aggregating *Dd* cell densities are typically 250 to 1000 per  $\text{mm}^2$  (Dallon and Othmer 1997). Due to the computational resources required to manage a population of Cytobots within the upper range, a cell density at the lower biological range of 250 agents per  $\text{mm}^2$  (1250 Cytobots) and another at 150 per  $\text{mm}^2$  (750 Cytobots) were chosen.

**Table 2** *D. discoideum* Simulation Results

No.	Cytobots per mm <sup>2</sup> (p)	Range (r) in mm	Mean No. of mounds; ( $\sigma$ )	Aggregation Phase Mean time in Hours; ( $\sigma$ );
AG1	150	5	1 (0)	8.98 (0.09)
AG2	150	2.5	4 (0.31)	9.63 (0.17)
AGF3	150	1	5.9 (1.16)	9.47 (0.65)
AG3	150	1	5.2 (0.82)	9.92 (0.34)
AG4	150	0.5	8.4 (1.19)	10.23 (0.59)
AG5	150	0.1	14.2 (2.36)	10.6 (1.82)
AG6	250	5	1 (0)	8.95 (0.11)
AG7	250	2.5	1 (0)	9.6 (0.20)
AGF8	250	1	6.8 (1.81)	9.71 (0.87)
AG8	250	1	4.3 (0.37)	10.05 (0.58)
AG9	250	0.5	6.7 (1.62)	12.65 (1.94)
AG10	250	0.1	-	-

```

FOR each Cytobot
  Get current agents' facing direction CF
  Assign a value to direction CF using statement 1

  FOR each (index n) detected cAMP signal
    Get detected signal incoming direction CA
    Assign a value to direction CA using statement 1
    IF CA = CF THEN kn = 3
    ELSE IF CA = CF-1 OR CA = CF+1 THEN kn=2
    ELSE IF CA = CF-2 OR CA = CF +2 THEN kn=1
    ELSE kn=0
    END IF
    Calculate distance dn
    Store each CA with kn and dn
  END FOR

  Calculate WA for current agent using Eq. (7)
END FOR

Statement 1: East = 1; North East = 2; North = 3; North West =4;
West = 5

Where:
WA= total weight of direction A
N= total number of agents within range of detection
dn= distance of current agent from agent n
CA= direction of incoming signal detected by current agent
CF= the current agents facing direction
kn= value of cAMP signal from agent n

```

**Fig. 5** The strength of signal for each cardinal or ordinal direction above the horizontal plane of a Cytobot is calculated using this pseudocode. The result for each direction is used to set the corresponding direction input pool of the ARN WDnet.

In both sets of experiments the Cytobots are initialized at random positions in Foraging mode within the simulated environment. In the AGF experiments, the environment is initialized with a radial outwardly-decreasing gradient of food ( $z$ ), as described by Eq. (6), where  $x$  and  $y$  are Cartesian coordinates on the horizontal plane:

$$z = \sqrt{x^2 + y^2} \quad (6)$$

$$W_A = \sum_{n=1}^N \frac{k_n}{d_n} \quad (7)$$

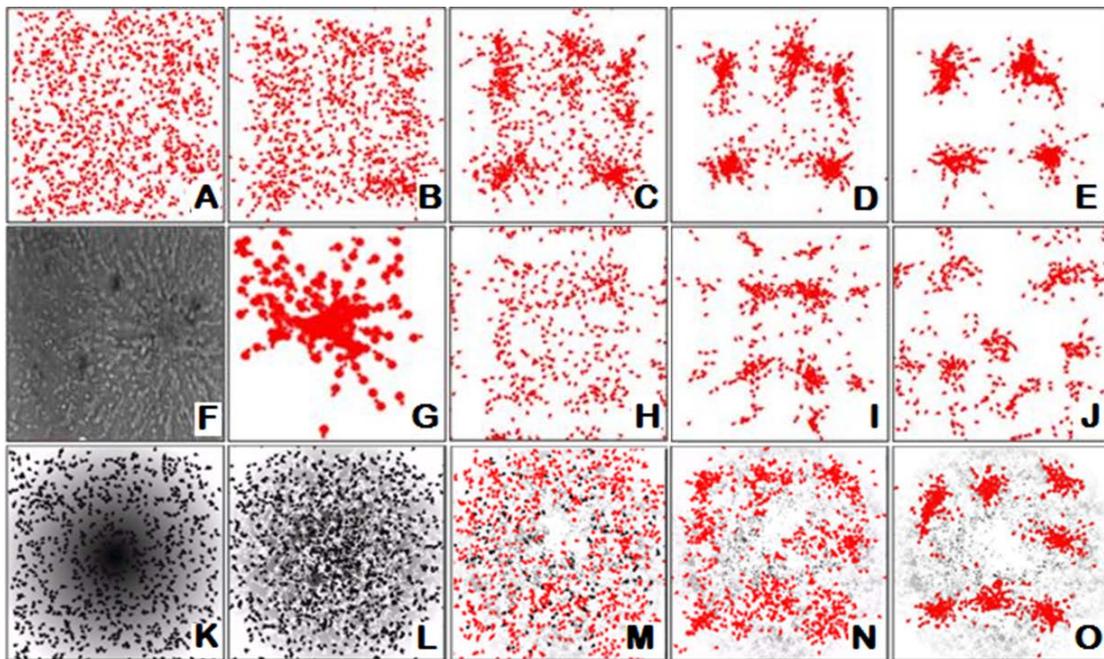
The Cytobots begin the experiment in the previously described Foraging mode and remain in this mode until the food resource is depleted and Starvation mode is triggered. In a real environment, food is non-uniformly distributed, may be regenerated and can move (in the case of bacterial prey). Thus, this setup is highly simplified, but is comparable to other simulations (Becker 2010).

If a Cytobot does not detect food for a period of approximately 5 time steps (the exact number depends on the level of food detected in the recent past, because higher levels take longer to flow through the network) it will enter Starvation mode. Cytobots in Starvation mode emit a cAMP signal at equal strength in all directions around their circumference into the environment. Each Cytobot in Starvation mode detects the cAMP signal of all other starving agents within a radius  $r$ . The total value for each direction is calculated using the pseudocode given in Fig. 5 and these totals are used to set the weighted direction network input (receptor) pools. A range of  $r$  values were explored, including that of real Dd cells: 1, 0.5, and 0.1 mm (McCann et al. 2010). The cAMP signal degrades linearly with increasing distance from the emitting cell. Each cycle represents 1 minute of time. In this time an aggregating Cytobot moves  $9\mu\text{m}$ - a distance which corresponds to that of actual aggregating Dd cells (Rifkin and Goldberg 2006). Therefore, after 1 hour of motion a Cytobot travels a distance of  $540\mu\text{m}$ . In this simulation, just as in biology, there are always remaining cells that do not aggregate, and thus the simulation runs until 95% of agents are at a distance of less than 0.1mm from their nearest neighbour.

#### 5.1.2 *D. discoideum* Simulation Results and Discussion

The results for all 12 experiments are given in Table 2. Each experiment was performed 100 times. In the AG experiments an increase in  $p$  by 100 per  $\text{mm}^2$  resulted in a decrease in the number of mounds formed at each value of  $r$ , with the exception of experiment AG6. This is not surprising, as denser populations have more chance of interacting, and thus form fewer clusters, each having a higher number of agents. Similarly, decreasing  $r$  results in a general

increase in the number of mounds formed at both values of  $p$ . The likely reason for this is that as  $r$  decreases the Cytobots area of influence becomes increasingly smaller, and thus the number of isolated stable clusters with fewer agents increases. In the AGF experiments, agents generally focus on consuming food in each of the remaining areas of highest concentration (see Fig. 6K-L). Having consumed almost all the food, agents begin switching to starvation mode (Fig. 6M). In these experiments the number and location of resulting mounds differs from that of the AG experiments at the same values of  $r$  and  $d$ . For example experiment AG8 results in an average of 4.3 mounds within the test space (Fig. 6E) while AGF8 results in an average of 6.8 mounds and a general shift in mound formation further away from the centre of the environment (as shown in Fig. 6O). The likely explanation is that, at the time of switching to aggregation, the majority of cells were forced outward toward the next remaining highest concentration of food. Emergent behaviours and clustering patterns similar to the biological organism were also observed.



**Fig. 6** Screenshots of the Dd simulation. Dots represent the Cytobots (black- vegetative and red- aggregative cells), and greyscale colour represents the food distribution. A-E: Cytobot aggregation experiment AG8 at A- 1hr, B- 2hr, C-5hr, D- 8hr, E- 10hr; Image F- real Dd cells aggregating; G- Lower right hand corner of image C demonstrating streaming behaviour; H-J Shows pattern formation; K-O Cytobot experiment AGF8 at K-0hr vegetation, L-4hr vegetation, M-transition to aggregation 0hr aggregation, N-5hrs aggregation, O-10hr aggregation.

*Diagram F courtesy of T. Gregor, Laboratory for the Physics of Life, Princeton University, 2013 Used with permission.*

In experiments AG8-10 and AGF8 the value of  $r$  and  $p$  are within the ranges for real Dd cells. These experiments are used to compare the behaviors and aggregation time with the values for real Dd in the literature. In experiments AG8-9 and AFG8 mound formation completes within

the range reported for the actual organism of 9-13 hours (Cotter et al. 1992; Becker et al. 2010). These results are comparable with other work. For instance, Becker et al. (2010) report an aggregation time of 11.6 hours for a simulated population of Dd with a cell density of  $200\text{mm}^2$ . In experiment AG10, the population never satisfied the criteria for completion of mound formation where instead the agents appeared to move in a fashion reminiscent of Brownian motion. The likely explanation is firstly because the simulation does take into account glycoprotein's which allow aggregating cells to attach together on contact. Furthermore, because  $r$  is small, fewer agents are detected by each Cytobot. Thus momentarily larger clusters with higher attraction strength go undetected and quickly dissipate- an effect that would not occur if agents stayed together. As previously discussed, the Cytobots are polarized. Implementing the agents in this way allows us to observe whether or not the previously described streaming behavior occurs. A close-up of the right-hand corner of screenshot C is shown in Fig. 6G showing agents beginning to form a cluster. The protruding head of each agent can be seen clearly, and each lines up its head to the rear of another to form a stream. As can be seen in Fig. 6F, this is very similar to the streaming behavior in real cells of Dd. Other emergent patterns occurred during different experiments including spirals (Fig. 6J), symmetric patterns (Fig. 6I), and waves (Fig. 6H).

These results show that the Cytobots are able to simulate behaviour of individual unicellular organisms, and the emergent behaviours arising from their interaction. It highlights a potential use, as a means to simulate groups of interacting cells, for example a bacterial colony or tissue component within a multicellular organism. Applications include the modelling of the effects of disease (e.g. faulty gene expression) and pharmaceuticals on global behaviour. The results demonstrate the parallels between ARN agents and their biological counterpart; like amoebae, their internal network of spatially distributed dynamic chemical species allows them to autonomously coordinate and direct their movement, recognize and respond to patterns in the environment, and produce high-level behaviour.

## 5.2 Oil-spill Confrontation Simulation

### 5.2.1 Oil-spill Simulation Methodology

To illustrate a practical application of the Cytobot system within robotics they were used to tackle a simplified oil-spill clean-up simulation. The Cytobots move within a 2D environment containing an oil-spill on water. This oil is analogous to a distribution of food within a nutrient landscape. In the following 4 experiments the length of time it takes for a swarm of 3, 5, 8 and 15 Cytobots to clean up 95% of a simulated oil-spill is recorded. The agents move through the environment by switching between the two previously described behavioural modes- Foraging and Starvation. In these experiments, each Cytobot is controlled using the same ARN network

as used in the Dd simulation. To enable the Cytobots to behave differently, rather than modify the network, the interface between the Cytobots and the environment was altered. To achieve this, the concentration of oil surrounding the agents was used to represent both food and cAMP attractants. Thus, the amount of oil at each new position was fed into both the receptor pools of the WDnet and of the Fnet. At the start of each experiment, the Cytobots are distributed randomly within the environment, and the ARN network is initialized as previously described. The agents start the simulation in Foraging mode but during the simulation alternate between Foraging and Starvation modes. Starvation behaviour is triggered when the last positions (minimum of 2) contained zero food. In Starvation mode, instead of turning in a random direction, the new direction is weighted toward higher concentrations of food within its surrounding area. This behaviour forces exploration of unexplored search space because previously visited positions have a food level of 0. Consumption of environmental food therefore acts as a stigmergic signal, where agents are inclined to move up the nutrient gradient created by their foraging activities. On consuming a sufficient amount of food, the Cytobot switches back to Foraging mode, repeating this behaviour until 95% of the oil is consumed. Here, we model the spillage of 100 tonnes of Statfjord crude oil at 15°C under a wind speed of 5ms<sup>-1</sup>. The oil is distributed over a 2D sea surface of 300m by 200m, thus an area of area 60000m<sup>2</sup>, where 2 pixels corresponds to 1m, as shown in Fig. 7A. This particular oil type and parameter set were chosen in order to compare directly with work by Kakalis and Ventikos (2008) who present a robotic swarm concept for oil-spill confrontation. For this reason, we account for an initial response time of 14 hours. Based on the mathematical models found in Kakalis and Ventikos which account for the main factors of short term changes in oil characterization, the volume of oil after 14 hours is reduced to 150m<sup>3</sup>. Beyond this starting state, the volume is only influenced by the Cytobots. The speed of each agent is 0.5ms<sup>-1</sup> and is based on other robotic agents in oil cleaning scenarios (Kakalis and Ventikos 2008), thus the Cytobots move 1 pixel (0.5m) for every time step. The actual cleaning surface is 1m, thus the Cytobots clean a 2 pixel wide area in each time step.

Mathematical modelling of an oil-spill is non-trivial and at best can offer a crude approximation of its actual trajectory. Most oil-spills quickly form a comet shape with most of the oil within the head, and a trail of sheen (Wang and Stout 2007). To represent a simplified version of the comet shaped spread, the area is divided into 100 3m x 200m segments. The first segment contains 0.015 tonnes of oil, and each subsequent segment increases by 0.03 tonnes from right to left.

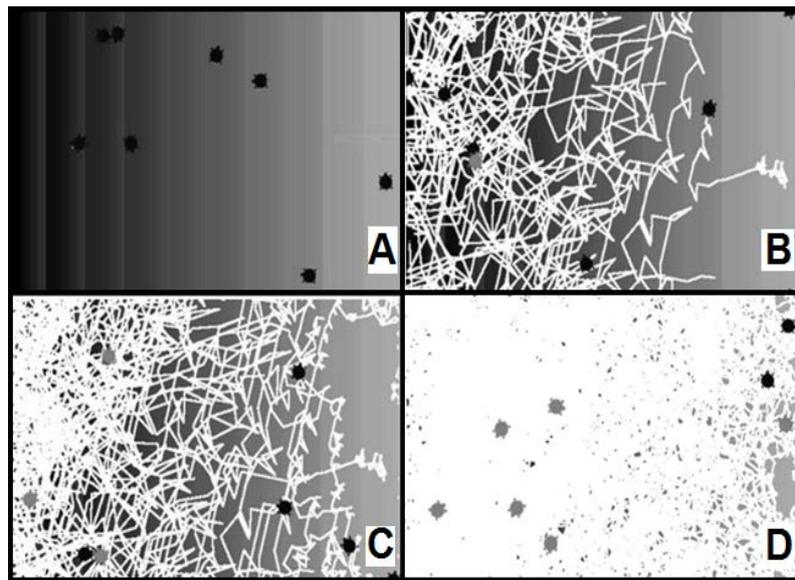
### 5.2.2 Oil-Spill Results and Discussion

In each experiment, a different number of Cytobots was deployed- 3, 5, 8 and 15 and the recovery rate achieved by each group were compared. The simulation time was measured from

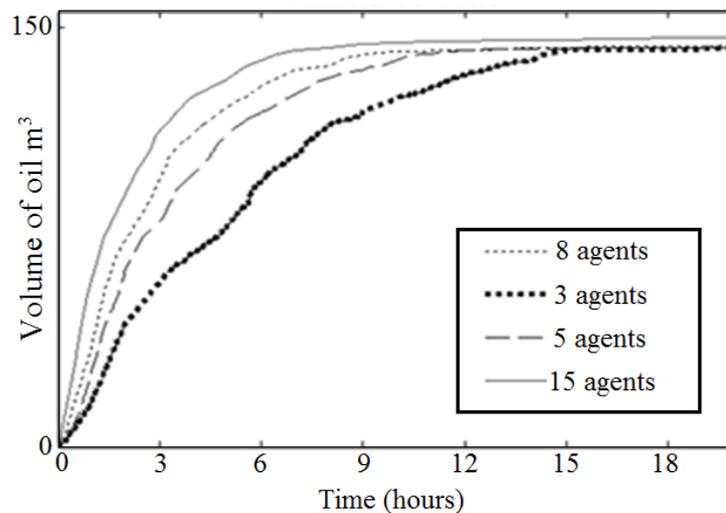
deployment of the Cytobots at 0 hours (14 hours after oil was spilled) and stopped when the Cytobots had collectively removed 95% of the 150m<sup>3</sup> of oil. Each experiment was run 100 times, and the average volume of oil consumed at 6 minute intervals was calculated. Figure 8 presents the average volume of oil consumed by the group of Cytobots against time. Figure 9 provides the average length of time taken to clean 95% of the oil (Avg. time) and standard deviation ( $\sigma$ ) for each experiment. By adding 2 additional agents to the group of 3 the length of time is reduced by 3.7 hours, thus 1.85 hour average difference per extra Cytobot. This difference decreases 1.12 hours per Cytobot for 8 agents, then to 0.76 per agent for 15. The variation can be accounted for by examining the agents' paths through the oil. Rates are much faster at the beginning of the experiments, where Cytobots move toward the oil-rich left side of the environment. This can be seen in the series of screenshots of a typical experiment shown in Fig. 7, where A shows the starting position at time 0, and B shows that after 2 hours the Cytobots have moved toward the left-hand side, focusing mainly on highly concentrated areas (consumed oil is shown in white). Initially, the rate of oil removal is high because Cytobots focus on the highest concentration areas and cannot go over their path, thus each new location results in consumption of oil. However, as time progresses, large patches become cleaned and a higher probability exists for the Cytobots to revisit previously cleaned areas. The consumption of oil in Fig. 7C-D at 4 and 9.6 hours respectively shows more clearly that Cytobots focus cleaning efforts on the area of highest concentration first, and are gradually forced to move toward the next highest concentration by the gradient created by their foraging activities.

Figure 7D shows the state of the oil at the end of the simulation, where only small patches remain mainly in areas of low oil concentration. These results can be compared to the simulation by Kakalis and Ventikos. Here, varying numbers of simulated EU-MOP robots are deployed to tackle 150m<sup>3</sup> of Strajford oil over 60000m<sup>3</sup> (as before). In this case, the robots have a slightly faster speed of 0.54m/s but have the same 1m skimming face. Each EU-MOP robot has a storage capacity of 2m<sup>3</sup> and a transit speed of 2.1ms<sup>-1</sup>. The times taken for 3, 5, 8, and 15 EU-MOPS are 54, 32, 20 and 10 hours respectively. For comparison, the results of our simulation can be adjusted to include unloading of the oil at a servicing vessel. Using the same storage capacity and transit speed and assuming the distance to the ship and back is 2 times 300m and that each Cytobot fills the same amount simultaneously, then the new times are 17.2, 12.7, 10.3 and 6.5 for 3, 5, 8 and 15 Cytobots respectively. The Kakalis and Ventikos simulation has several differences to the one reported here, particularly in the distribution of the oil. Also, some key parameters are missing from their paper, for example, the distance to the boat. Despite these differences, our results are very similar. For example, the reported simulation time for 15 EU-MOPS is 10 hours and in our simulation 5 and 8 Cytobots took 12.7 and 10.3 hours respectively. Given the differences in the simulation and differences in operation of the robots, the resulting clean-up times are comparable, showing that the Cytobots

have potential application as distributed robotic agents in real-world environments. This application demands an internal control system which can function without reference to other agents within the environment which are operating in parallel. By modifying the environment, (which in this case was consumption of food), the agents can stigmergically communicate and facilitate emergent behaviour. The Cytobots offer a unique range of abilities. Like cells, their internal network of spatially distributed dynamic chemical species allows them to autonomously coordinate and direct their movement, recognize and respond to patterns in the environment, and produce high-level behaviour.



**Fig. 7** Oil simulation using 8 Cytobots at A- 0 hours, B- 2 hours, C- 4 hours and D- 9.6 hours



**Fig. 8** Average volume of oil cleaned against time for each group of Cytobots

No. of Cytobots	Avg. finish times	Standard deviation ( $\sigma$ )
3	15.2	3.4
5	11.5	2.7
8	9.6	2.8
15	6.1	3.1

**Fig. 9** Average length of time taken to clean 95% of the oil-spill for each group of Cytobots.

## 6 Conclusions

The experiments outlined in this paper show the advantages of considering cell-signalling networks as a connectionist paradigm. This approach allows their structure to be easily visualised, manipulated and organised into hierarchical modular structures. These in turn facilitate the exploration of the limits of their processing capabilities and, from an artificial intelligence perspective, allows them to be compared directly with other forms of simulated and biological intelligence. It may also prove useful tool in biomedical research as it allows, for example, the effect of mutated proteins to be examined simply in isolation or in interconnected groups - this is of particular importance in cancer research.

The results presented above illustrate how common simple motifs, present in all CSNs, can be integrated together to form structured networks with sophisticated processing capabilities. This indicates that these may form universal building blocks from which higher-level functions can be built. The ARN based agents, constructed from these, behave in a very similar way to the real organisms; displaying two of their most interesting behaviours (foraging and aggregation), and so we may conclude that the ability to evolve this level of behaviour is probably fairly universal among such single-celled microbes.

In the next stage of work, it is hoped to use an ARN to explore how learning might arise in protists. In particular the extent to which learning and memory is genetically programmed into invariant CSNs, and how much of it is extrinsic to the organism – as a stigmergic system. Or whether it has a variable intrinsic aspect – for example, the use of modulating elements within the CSNs, which might act as primitive “memories”. In the experiments outlined in this paper, the stigmergic aspect of such memory was illustrated. For example, like the cAMP trail-following behaviour described above, the environment acts as a shared information depository in which to both facilitate collective manipulation of data and communicate the current global system state.

There may also be a role for the ARN in other areas which have yet to be explored, particularly those which would benefit from its connectionist approach described in the paragraphs above. For example, it would appear ideal for modelling the complex but interconnected pathways present in environmental science and soil chemistries.

*Supplementary information and code can be found at the following link:*

<https://drive.google.com/folderview?id=0B-xGVfJFH9UmZIJTV1pROFFRb00&usp=sharing>

## References

- Andrew N, Insall R (2007) Chemotaxis in shallow gradients is mediated independently of PtdIns 3-kinase by biased choices between random protrusions. *Nat. cell boil.* 9(2):193-200
- Arkin A, Ross J (1994) Computational functions in biochemical reaction networks. *Biophys. J* 67:560-578
- Becker M (2010) Simulation model for the whole life cycle of slime mould Dictyostelium Discoideum. In: *Proceedings of the European conference on modeling and simulation*, pp 247-253
- Bhalla U (2003) Understanding complex signaling networks through models and metaphors. *Prog. Biophys. Mol. Bio.* 81:41-65
- Bray D (1995) Protein molecules as computational elements in living cells. *Nature.* 376(6538):307-12
- Brock D, Douglas T, Queller D, Strassmann J (2011) Primitive agriculture in a social amoeba. *Nature* 469 (7330): 393-396
- Cotter D, Sands T, Viridy K, North M, Klein G, Satre M (1992) Patterning of development in Dictyostelium discoideum: factors regulating growth, differentiation, spore dormancy and germination. *Biochem. Cell Biol.* 70(10-11):892-919
- Dallon J, Othmer H (1997) A discrete cell model with adaptive signaling for aggregation of Dictyostelium discoideum. *Philos. T. Roy. Soc. B* 352(1351): 391–417
- Devreotes P (1989) Dictyostelium discoideum: a model system for cell-cell interactions in development. *Science* 245(4922): 1054-1058
- Ford B (2009) On intelligence in cells: The case for whole cell biology. *Interdisipl Sci Rev* 34(4):350-365
- Gerrard C, McCall J, Coghill G, Macleod C (2011) Artificial Reaction Networks. In: *Proceedings of the 11th UK Workshop on Computational Intelligence, Manchester: UK*, pp 20-26
- Gerrard C, McCall J, Coghill G, Macleod C (2012a) Temporal patterns in Artificial Reaction Networks. In: *Proceedings of The 22nd International Conference on Artificial Neural Networks Lausanne*, part 1, vol. 7552, pp 1-8
- Gerrard C, McCall J, Coghill G, Macleod C (2012b) Adaptive Dynamic Control of Quadrupedal Robotic gaits with Artificial Reaction Networks. In: *Proceedings of the 19th International Conference on Neural Information Processing Doha*, vol. 7663, part 1, pp 280-287
- Gerrard C, McCall J, Coghill G, Macleod C (2013) Exploring aspects of cell intelligence with artificial reaction networks. *Soft Computing* (in press)
- Guo H, Meng Y, Jin, Y (2009). A cellular mechanism for multi-robot construction via evolutionary multi-objective optimization of a gene regulatory network. *BioSystems* 98(3):193-203
- Joachimczak M, Kowaliw T, Doursat R, Wrobel B (2013) Controlling development and chemotaxis of soft-bodied multicellular animats with the same gene regulatory network In: *Advances in Artificial Life, ECAL*, vol. 12, pp. 454-461
- Kakalis N, Ventikos, Y (2008) Robotic swarm concept for efficient oil spill confrontation. *J. Hazard. Mater.* 154(1-3):880-7.
- Kholodenko B (2006) Cell Signaling dynamics in Time and Space. *Nat Rev Mol Cell Biol* 7(3):165-176
- Manahan C, Iglesias P, Long Y, Devreotes P (2004) Chemoattractant signaling in Dictyostelium discoideum. *Annu. Rev. Cell Dev. Biol.* 20, 223–253
- McCann C, Kriebel P, Parent C, Losert W (2010) Cell speed, persistence and information transmission during signal relay and collective migration. *J. Cell Sci.* 123:1724-1731
- Patidar V, Sud K, Pareek N (2009) A pseudo random generator based on chaotic logistic map and its statistical testing. *Informatica* 33:441-452
- Phatak S, Rao S (1995) Logistic map: A possible random-number generator. *Phys. Rev.* 51(4): 3670.
- Rifkin J, Goldberg R (2006) Effects of chemoattractant pteridines upon speed of D. discoideum vegetative amoeba. *Cell Motil. Cytoskeleton*, 63(1): 1-5
- Shen W, Will P, Galstyan A, Chuong C (2004) Hormone-inspired self-organization and distributed control of robotic swarms. *Auton. Robot* 17(1): 93-105
- ten Eikelder T, Crigins S, Steijaert M, Liekens A, Hilbers P (2009) Computing with feedforward networks of artificial biochemical neurons. In: *Proceedings of the 2<sup>nd</sup> International Workshop on Natural Computation, Japan: Springer*: vol 1 pp 38-47

- Tyson J, Novák B (2010) Functional motifs in biochemical reaction networks. *Annu. Rev. Phys. Chem* 61:219-240
- West S, Diggle S, Buckling A, Gardner A, Griffin A (2007) The social lives of microbes *Annu. Rev. Ecol. Evol. Syst.*38: 53-77
- Ziegler J, Banzhaf W (2000). Evolving a "nose" for a robot. In: *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO)*