

RESEARCH ARTICLE

Computational capabilities of a multicellular reservoir computing system

Vladimir Nikolić^{1,2*}, Moriah Echlin^{3,4,5}, Boris Aguilar³, Ilya Shmulevich^{3*}

1 Bioinformatics Graduate Program, The University of British Columbia, Vancouver, BC, Canada, **2** Canada's Michael Smith Genome Sciences Centre at BC Cancer, Vancouver, BC, Canada, **3** Institute for Systems Biology, Seattle, WA, United States of America, **4** Prostate Cancer Research Center, Faculty of Medicine and Health Technology, Tampere University, Tampere, Finland, **5** Tays Cancer Center, Tampere University Hospital, Tampere, Finland

* vnikolic@bcgsc.ca (VN); ilya.shmulevich@isbscience.org (IS)



OPEN ACCESS

Citation: Nikolić V, Echlin M, Aguilar B, Shmulevich I (2023) Computational capabilities of a multicellular reservoir computing system. PLoS ONE 18(4): e0282122. <https://doi.org/10.1371/journal.pone.0282122>

Editor: Claudio Zandron, University of Milano–Bicocca: Università degli Studi di Milano-Bicocca, ITALY

Received: November 30, 2022

Accepted: February 7, 2023

Published: April 6, 2023

Copyright: © 2023 Nikolić et al. This is an open access article distributed under the terms of the [Creative Commons Attribution License](https://creativecommons.org/licenses/by/4.0/), which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

Data Availability Statement: The simulation code and data produced can be found online at <https://github.com/vlad0x00/multicell-rc> and the archived version is available at Zenodo (<https://doi.org/10.5281/zenodo.7212658>). The archived version also has a container with all the software dependencies installed.

Funding: V.N. was supported in part by a scholarship from the UBC Bioinformatics Graduate Program (www.bioinformatics.ubc.ca) via the NSERC (www.nserc-crsng.gc.ca) CREATE

Abstract

The capacity of cells to process information is currently used to design cell-based tools for ecological, industrial, and biomedical applications such as detecting dangerous chemicals or for bioremediation. In most applications, individual cells are used as the information processing unit. However, single cell engineering is limited by the necessary molecular complexity and the accompanying metabolic burden of synthetic circuits. To overcome these limitations, synthetic biologists have begun engineering multicellular systems that combine cells with designed subfunctions. To further advance information processing in synthetic multicellular systems, we introduce the application of reservoir computing. Reservoir computers (RCs) approximate a temporal signal processing task via a fixed-rule dynamic network (the reservoir) with a regression-based readout. Importantly, RCs eliminate the need of network rewiring, as different tasks can be approximated with the same reservoir. Previous work has already demonstrated the capacity of single cells, as well as populations of neurons, to act as reservoirs. In this work, we extend reservoir computing in multicellular populations with the widespread mechanism of diffusion-based cell-to-cell signaling. As a proof-of-concept, we simulated a reservoir made of a 3D community of cells communicating via diffusible molecules and used it to approximate a range of binary signal processing tasks, focusing on two benchmark functions—computing median and parity functions from binary input signals. We demonstrate that a diffusion-based multicellular reservoir is a feasible synthetic framework for performing complex temporal computing tasks that provides a computational advantage over single cell reservoirs. We also identified a number of biological properties that can affect the computational performance of these processing systems.

Introduction

Information processing plays an essential role in cellular systems, enabling cells to adapt their behavior in response to changes in internal and external environmental conditions. By processing information through the heterogeneous, nonlinear interactions of their molecular

program in High-Dimensional Biology. The funds for the publication fee for this manuscript were provided by Institute for Systems Biology (isbscience.org). The funders had no role in study design, data collection and analysis, decision to publish, or preparation of the manuscript.

Competing interests: The authors have declared that no competing interests exist.

components, cells can respond to fluctuations in environmental signals [1, 2], search for food [3–5], navigate paths [6, 7], and make behavioral decisions during cellular regulation [8–10]. Similarly, when cells aggregate in multicellular communities, the interactions between cells give rise to community-level computation, such as collective decision-making [11, 12], distributed sensing [13, 14], structure formation [15], and immune surveillance [16, 17].

In addition to its importance in natural systems, information processing is playing an increasingly significant role in synthetic biology. In single cell systems, multi-input synthetic molecular circuits are being explored for their potential as sensors, diagnostic tools, and responsive agents for ecological, industrial, and biomedical applications [18]. Synthetic intracellular circuits have been implemented in both prokaryotic and eukaryotic cells through a variety of techniques using gene- [19, 20], DNA recombinase- [21, 22], and RNA-based mechanisms [23–25]. Even minimal cell-like architectures solely consisting of encapsulated synthetic biochemical circuits have been designed to process external molecular signals [26, 27]. However, single cell engineering has its limitations [28]. Because molecular components cannot be spatially isolated, separate circuits require unique molecular wiring. The number of independent circuits that can reasonably be engineered is therefore limited by the number of manageable molecular species [29, 30]. More practically, the complexity of engineered circuits is constrained by the additional metabolic burden they introduce to the cell [31–33]. Synthetic circuits are also often highly function-specific with limited repurposing [30, 34]. To overcome these limitations, synthetic biologists have begun engineering multicellular systems through the combination of simple circuits implemented in individual cells (or minimal cells) [35, 36]. In this way, the burden on individual cells is reduced, subfunctions are separated, and circuit design is modular. Moreover, synthetic biologists can take advantage of other inherent features of multicellular systems, such as the capacity for parallel and distributed computing, robustness to failure, modularity, and scalability [30, 34, 37]. Currently, multicellular systems have been designed that compute complex Boolean functions [28, 38–44], act as memory devices [45], behave as glucose sensors [46], generate specified spatial patterns [47], and recognize spatial patterns [48] among other functions [49].

Though different molecular mechanisms are utilized in synthetic multicellular systems, they all rely on intracellular architecture that is designed for a specific function. A flexible computing framework in neural networks, reservoir computing (RC), does not require such specification. Reservoir computing is a signal processing framework in which a signal is input into a fixed-rule dynamic network (the reservoir) and the state of the network is used to continuously approximate a function defined over the input [50, 51]. A reservoir computer consists of a signal node(s), a dynamical network (reservoir), and an output node(s). The signal node changes in value over time, passing this value to the reservoir via a subset of reservoir nodes. The reservoir propagates the signal and projects it into a higher dimensional space, retaining a memory of past signal values owing to feedback within the reservoir. Finally, the output node performs a form of regression analysis using another subset of reservoir nodes in order to approximate a given function applied to the input signal over a window of time. Different functions can be approximated simply by retraining the parameters used for regression. Importantly, retraining the output node does not involve any adjustment to the structure or dynamics of the reservoir itself.

Previous work in reservoir computing has shown that a diverse collection of physical systems can be utilized as reservoirs for a range of signal processing tasks, reviewed in [52]. In theory, any dynamical system that demonstrates a fading memory property can act as a reservoir [53]. Both structurally and dynamically, cells are suitable reservoirs. Intracellular networks are fixed and naturally tuned to balance robustness with responsiveness (i.e. fading memory) [54]. Moreover, multiple possible mechanisms—utilizing physical, chemical, and bioelectric

mediums—exist for signal input and signal readout. Furthermore, a cellular reservoir computer would be a widely applicable synthetic computing system that would require minimal engineering as the training is performed in the readout layer and not within the reservoir itself. Initial work has already demonstrated the capacity of cells to act as reservoirs. Namely, single cell gene regulatory network reservoirs have been explored in living [55, 56] and theoretical systems [57, 58].

As with other synthetic cellular systems, reservoir computers that take advantage of multicellular populations have the potential for more complex and robust computation compared to single cell reservoirs. Previous work has shown evidence of multicellular reservoirs in the prefrontal cortex [59], cerebellum [60], and glial networks [61]. Similarly, in vitro neural systems have been shown to act as reservoirs [62, 63]. This work has demonstrated that cells can act together to form a reservoir when connected via neuroelectric signaling. Here, we extend reservoir computing in multicellular populations to the more wide-spread mechanism of diffusion-based signaling, which is used by both prokaryotic and eukaryotic cells and can serve as a means of short- and long-distance communication. Namely, we simulated a proof-of-concept multicellular reservoir computer, in which a 3D community of cells communicating via diffusible molecules serves as the reservoir. We show that multicellular reservoir computing via diffusion-based communication is a feasible synthetic framework for performing complex temporal computing tasks.

Model description

The goal of the multicellular Reservoir Computer (RC) is to approximate an objective function f , given an input sequence of T binary values: $V = (v_1, v_2, \dots, v_T)$, where $v_i \in \{0, 1\}$, $1 \leq i \leq T$. The function f is defined on a w sized window of sequential values from V and outputs a binary value: $f: X \rightarrow Y$, where $X = (v_{t-w+1}, v_{t-w+2}, \dots, v_t)$, $w \leq t \leq T$ and $Y \in \{0, 1\}$. In this work, we have tested the performance of the RC on two time-dependent functions, temporal median and temporal parity, over multiple window sizes. The temporal median and parity functions test the memory and processing power of the reservoir, respectively, and have been used previously for benchmarking reservoir performance [64–67]. The median function outputs 1 if there are more 1 bits in the window and 0 otherwise. The parity function outputs 1 if there is an odd number of 1 bits within the window, and 0 otherwise.

Community organization & communication

The RC is implemented as a community of cells of different strains arranged in a 3-dimensional regular grid of cube shape or in a number of stacked square layers. A visualization of a cell community model can be seen in Fig 1. The cells communicate with each other using Extracellular Signalling Molecules (ESMs). Cells are able to secrete ESMs which diffuse into their surroundings and are able to register the presence of an ESM concentration above a given threshold. The diffusion of ESMs is modeled by solving the standard diffusion equation:

$$\frac{\partial C_i^m}{\partial t} = \beta \Delta C_i^m - \alpha C_i^m + \frac{\eta_i^m}{V_c} \quad (1)$$

where C_i^m is the concentration of ESM molecule m at location i , β is the diffusion coefficient, α is the degradation rate, η_i^m is the secretion rate of molecule m by the cell at location i , and V_c is the volume occupied by the secreting cell. $\frac{\eta_i^m}{V_c}$ term is 0 if there is no cell at location i . α and β values are constant and same for all ESM molecules. Here we assume that diffusion is a much faster process than regulation of the gene network responsible for secretion and reception, and

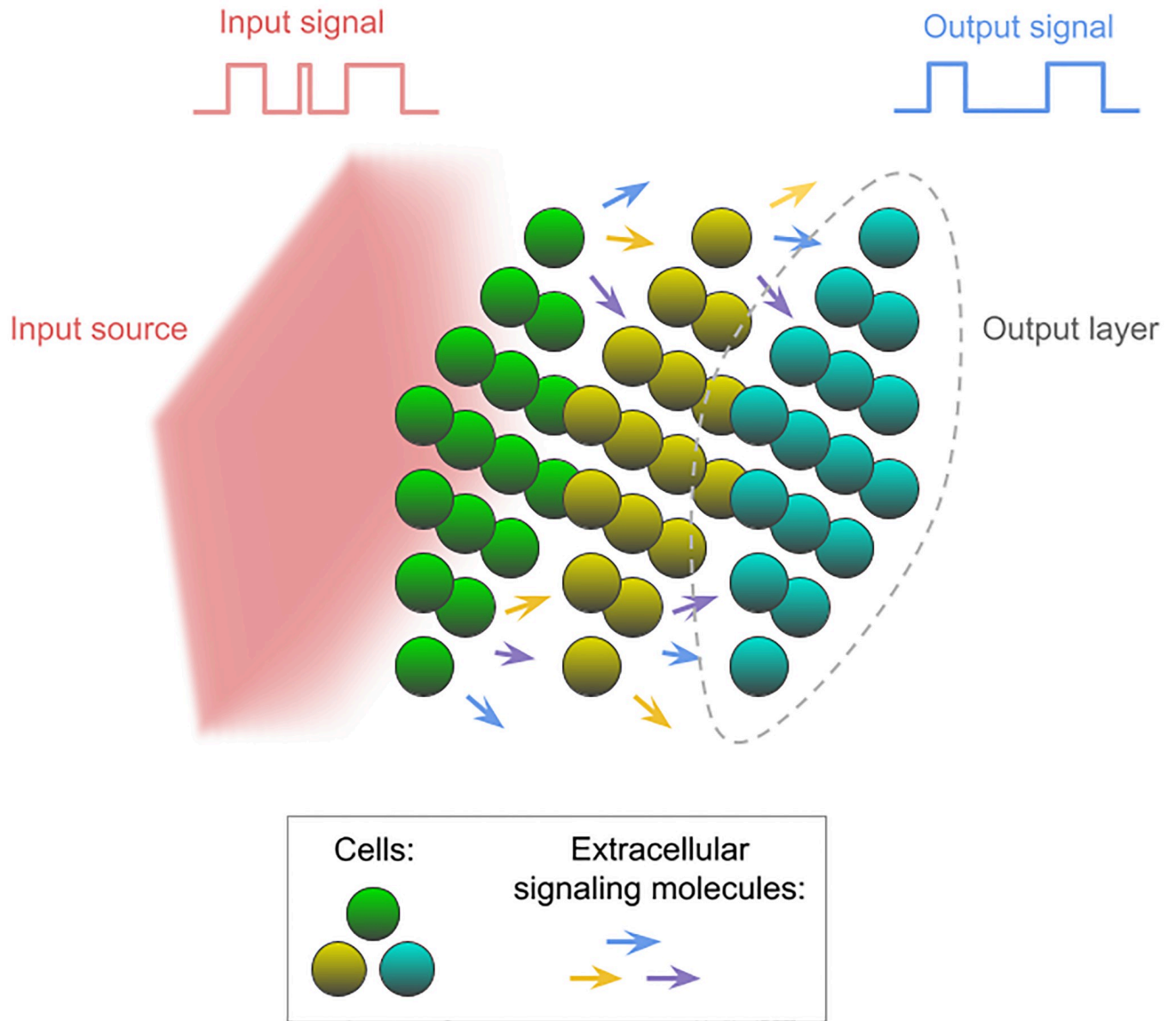


Fig 1. A visual example of a cell community. Cells are arranged in 3D space in a number of square shaped layers along the input signal source axis. The input signal molecules are produced from a simulation domain wall if the value of the input signal in the current simulation time step is 1. The input signal molecules dissipate as they travel through the simulation domain, activating a specified number of cell layers. The cells belonging to these layers register the input and communicate with further layers using ESMs. A specified number of cell community layers at the end opposite of the input source are used for output readout. Their gene values are read at every simulation time step and a regression analysis is performed in order to approximate the objective function.

<https://doi.org/10.1371/journal.pone.0282122.g001>

so we solve the steady state of the equation:

$$0 = \beta \Delta C_i^m - \alpha C_i^m + \frac{n_i^m}{V_c} \tag{2}$$

This mechanism facilitates information propagation throughout the community. At the ends of the simulation domain, Neumann boundary condition is enforced with a derivative of 0. The Neumann boundary condition specifies the value of the derivative of a variable in a partial differential equation (PDE) at a given end of the domain (boundary). By fixing the value of the

derivative at 0, we ensure the ESM levels at simulation domain boundaries stay unchanged, as opposed to, for example, dissipating. If the boundaries imposed a different value on ESM levels, it would affect the communication between the cells close to the boundary which we would like to avoid.

Each cell has ESM receptor and secreter genes that control communication and can be either ON or OFF. ESM receptors transition into the ON state if the concentration of the corresponding ESM is above the threshold θ in the cell's surroundings and into the OFF state otherwise. The value of θ is constant and the same for all ESM molecules. When the secretion genes are in the OFF state, they produce ESMs with a basal secretion rate and diffuse it into the cell's surroundings. When they are in the ON state, they produce ESMs with an active secretion rate that is higher than basal. For the receptor genes to register ESMs presence, the threshold θ is defined as a factor of the basal secretion rate of a single cell. In other words, the ESM concentration should be θ times the basal in order to be registered. Incorporating basal secretion in this way allows for a more general model of signaling dynamics better able to capture the leaky nature of real systems [68, 69]. Another key parameter of the ESM communication mechanism is the effective interaction distance λ , which is derived from the molecule concentration decay rate and the diffusion coefficient: $\lambda = \sqrt{\beta/\alpha}$. Increasing λ allows ESMs to travel further from their source, enabling more distant cells to communicate with each other.

Cell inner workings

Inside each cell, gene regulation is modeled with a Random Boolean Network (RBN) [70] implemented as a graph $G = \{G_V, G_E\}$, where G_V is a set of N nodes/genes: $G_V = \{g_1, g_2, \dots, g_N\}$, and G_E is a set of $2 \cdot (N - 1 - E)$ edges, where E is the number of ESMs. Each gene can either be OFF or ON: $g_i \in \{0, 1\}$, $1 \leq i \leq N$. All cells have the same number of genes defined as a simulation parameter (N). There are four types of genes in each cell: input signal gene, ESM receptors, ESM secreters, and reservoir genes. There is always a single input signal gene (1), a receptor and a secreter gene for each ESM ($2 \cdot E$), and the rest are reservoir genes (N_r), for a total of N genes. While the input and ESM genes are responsible for external communication, the reservoir genes represent the internal state of a cell.

A simulation is divided into discrete time steps. At every time step t , the input signal is diffused into the environment (as described in the following subsection) and the ESM secreted based on secreter gene states. Then Eq (2) is solved for every location in the simulation to determine molecule concentrations. Following this, the input signal gene g_1 and the ESM receptor genes determine their states based on the presence of the input signal and ESM molecules respectively in the cell's surroundings. For a corresponding gene g_i , if the molecule concentration is above a threshold, then $g_i(t) = 1$ and $g_i(t) = 0$ otherwise. The input signal gene g_1 has a different threshold from ESM receptors, as described in the following subsection. Finally, ESM secretion genes and reservoir genes determine their values based on the state of the other genes. The value of each of those genes, $g_i(t)$, is updated using a function f_i . f_i has k_i inputs chosen randomly and uniformly from G excluding ESM secreter genes, $g_i(t+1) = f_i(g_{j_1}(t), g_{j_2}(t), \dots, g_{k_i}(t))$, $g_j \in G \setminus G_{ESMsecreters}$. The fraction of genes whose f_i inputs include the input signal gene g_1 is controlled with the L simulation parameter. The truth table for f_i , its output for each possible combination of input values, is randomly generated. The average number of inputs (in-degree) for all f_i is 2 and the probability of f_i outputting a 1 is 0.5. These function parameters have been shown to maximize computing capacity [71, 72].

In each simulation, not all cells are described by the same set of functions. Each cell is randomly assigned a strain. Each strain has a distinct set of randomly generated edges in G ,

defining its RBN topology and a distinct truth table is generated for each g_i . The number of strains is a simulation parameter (S).

Community input/output

A random input signal sequence V is generated for each simulation. For every time step t , there is a corresponding input value v_t . The input signal is represented as molecules diffusing away from one of the simulation walls, as shown on the left part of Fig 1. The dynamics of the input signal molecules are modeled using the same diffusion equation as the ESMs. The difference is that the secretion term is always 0 and instead the production is modeled with Dirichlet boundary condition on the signal producing wall. The Dirichlet boundary condition specifies the value of a variable in a PDE at a given boundary. By fixing the concentration of molecules at a boundary to a value >0 , we simulate a source, and by fixing it to 0, a sink. All other walls still follow Neumann boundary condition with a derivative of 0. If the input value v_i is 1 in a simulation time step, the solution to the Dirichlet boundary condition is >0 (100,000), diffusing the molecules from the simulation wall into the environment, affecting nearby cells. The input molecules move towards the opposite wall of the simulation domain, dissipating along the way and penetrating I layers of the community, specified as a parameter. Each simulation calculates what threshold cells should have to the input signal in order for it to penetrate I layers. If v_i is 0 in a simulation time step, the Dirichlet boundary condition solution is 0 and no input signal molecule is produced.

As information is processed by the community, the goal is to obtain an approximation of the given function f (median or parity) on the input signal by reading the cell states of the cell community output layers. A number of cell community layers, specified as a parameter (O), on the side opposite of the input signal producing wall are used for output. After the ground truth for f output is calculated, the gene states g_i of all cells within the output layers are used as covariates for regression analysis with the ground truth values as desired output. Further description of the reservoir training and testing are provided in Materials and Methods.

Results

Here we have tested the performance of a cube-structured community as a reservoir with comparison to a single cell (Proof of Concept); analyzed the effect of parameter values on the cube-structured reservoir's performance (Sensitivity Analysis); and tested the performance of a 3-layered rather than cube-structured community of cells as a reservoir (Layered Community). In all analyses, accuracy has been used as a metric of performance. We define accuracy as the ratio of correctly approximated outputs using the regression model over the total number of outputs. Since the tested functions have binary output, at worst, the performance is 50%, which is equal to a random guess, and 100% at best, where all outputs are correctly guessed.

Proof of Concept

In this analysis, we simulated a cube-shaped community in which all cells are exposed to the input signal and all cells are used as outputs. In this way, we focused on the multicellular and communication aspects of the population rather than the effects of spatial heterogeneity in the input signal (input heterogeneity). The cube-shaped community was tested for approximating median and parity functions with window sizes 3, 5, 7, and 9. We also simulated a single cell to compare multicellular to unicellular performance. The fixed simulation parameters can be found in S1 Table in S1 File, with the only difference between the multicellular and the unicellular system being the number of cells, 12^3 and 1 respectively. The performance comparison can be seen in Fig 2.

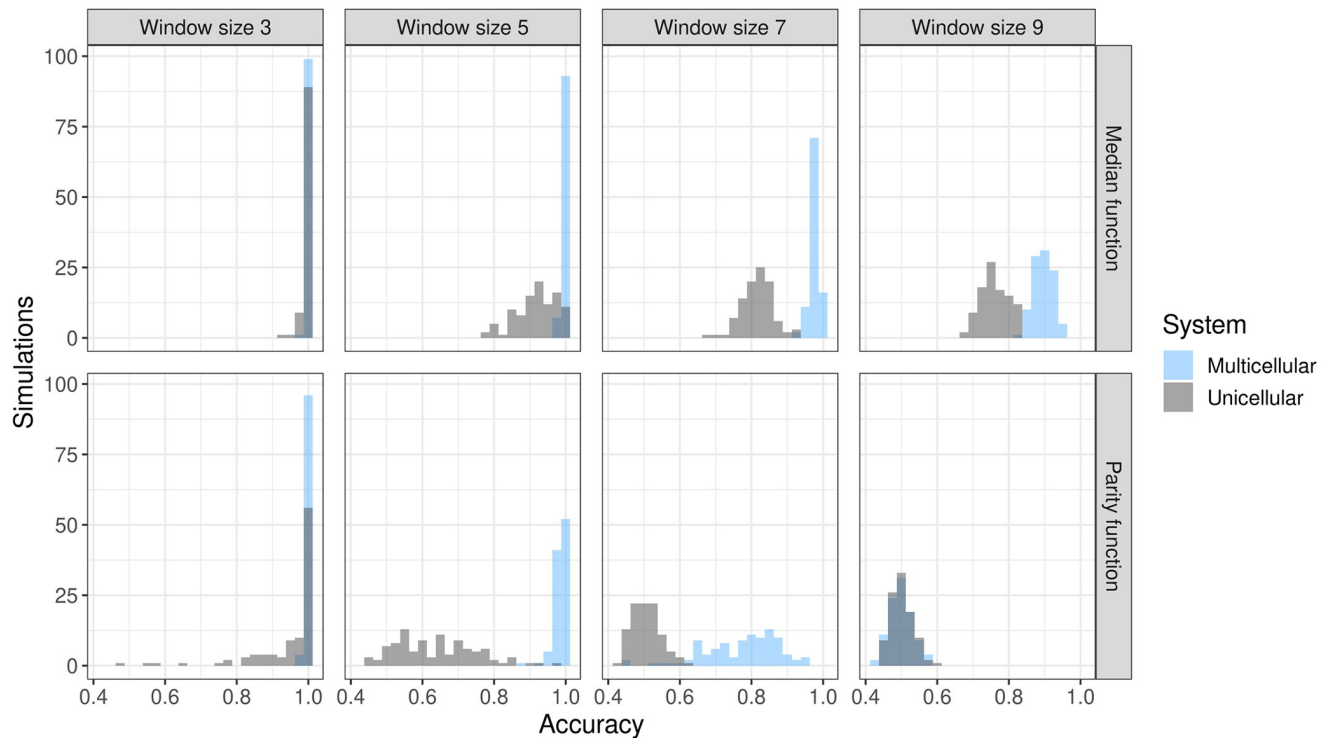


Fig 2. Simulation results from the Proof of Concept analysis. The distribution of reservoir accuracy is shown for median and parity functions with different function window sizes. To generate the distributions, each combination of window size and function was simulated 100 times. As parity is a harder function to approximate, the distributions are skewed towards the left compared to median function. For window size 3, both functions are well approximated, with almost all simulations performing with 100% accuracy. As window size increases, distributions move towards 50% (random chance) accuracy, with parity function reaching around 50% for window size 9 for both unicellular and multicellular systems. These results demonstrate that a community of cells can act as an RC and can outperform a single cell in that capacity.

<https://doi.org/10.1371/journal.pone.0282122.g002>

The results demonstrate that a community of cells has the capacity to approximate binary functions even with larger window sizes. For estimating the median function, the mean accuracy remains near 1 for all window sizes tested, but does decrease as the window size increases. For estimating the parity function, a computationally harder task, the community performs nearly perfectly for window sizes 3 and 5, but mean accuracy rapidly decreases for larger window sizes. At window size 9, mean accuracy is around 50%, or as good as a random guess, demonstrating the computational capacity limit of this setup. In all cases except parity window size 9 where both systems are at their limit, multicellular communities achieve a higher average accuracy when compared to the performance of a comparable unicellular reservoir simulated here as well as to previous reports of single cell Boolean Network (BN) reservoirs [57].

To check how these results generalize to other signal processing tasks, we tested all possible Boolean functions (256) for a 3 bit window, and 256 randomly sampled Boolean functions for a 5 bit window (S1 Fig in S1 File). Additionally, a recursive version of functions was also tested, in which the least recent input window bit is replaced with the output of the function from the previous timestep, i.e.: $f_t = f(f_{t-1}, v_{t-w+2}, \dots, v_t)$, $w \leq t \leq T$, where w is the window size, t is the current timestep, T is the total number of timesteps, and v is the input signal array. The number of reservoir genes (N_r) per cell is varied between 10, 20, 30, and 100. In all cases, the multicellular system performs better on average, as we have seen with the median and parity functions.

Sensitivity analysis

To examine how community parameters, especially those related to intercellular communication, affect reservoir performance, sensitivity analysis was performed using Partial Rank Correlation Coefficient (PRCC) [73]. Briefly, sensitivity analysis generates random sets of input parameters and assesses the accuracy of the model for every parameter set in order to determine the rank correlation between each parameter x_i , $1 \leq i \leq k$, where k is the number of parameters, and the output y (Accuracy). Correlation is calculated after rank-transforming values, and so non-linear effects are ignored. When calculating correlation for each parameter, the effect of other parameters is cancelled out (further explained in Methods and Materials).

The model was tested to approximate parity function with a window size of 7. The community is cube-shaped and all cells are used for output. The results of Sensitivity Analysis can be found in Table 1, including the range within which parameters have been randomly sampled. Details about the sampling approach can be found in Methods and Materials. The rest of the parameters are kept fixed and can be found in S2 Table in S1 File.

As demonstrated in previous RBN reservoir studies [57, 64], the fraction of genes connected to the input gene (L) and the number of reservoir genes (N_r) strongly positively correlate with accuracy ($\rho_{R(L),R(y)} = 0.77$, $\rho_{R(N_r),R(y)} = 0.55$). Here, we demonstrate that the number of cells (C) and the number of strains (S) also positively correlate with accuracy ($\rho_{R(C),R(y)} = 0.42$, $\rho_{R(S),R(y)} = 0.33$). Additionally, the number of cell layers that receive the input signal (I) moderately positively correlates with accuracy ($\rho_{R(I),R(y)} = 0.08$). Despite being responsible for intercellular communication, the number of ESMs (E), λ , and θ only have a weak correlation ($\rho_{R(E),R(y)} = -0.06$, $\rho_{R(\lambda),R(y)} = 0.04$, $\rho_{R(\theta),R(y)} = -0.06$).

Layered Community

This analysis involves a cell community of three square shaped layers along the input signal axis, as depicted in Fig 1. The model was tested to approximate parity function with a window size of 3. This function and window size were chosen because the difficulty of the task balances the accuracy well between 50% and 100%. In this setup, the threshold for the input signal is such that the signal is undetectable past the first layer. Additionally, only the last cell layer is used for output. The ESM λ parameter value is such that the signal is only detectable to immediately neighbouring cells in the same or adjacent layer. Because of this, cells only receive a strong enough molecular signal from their nearest neighbors. Consequently, the first layer cannot directly communicate with the third output layer. The information has to propagate through the middle layer, and so it takes two simulation time steps to reach the output layer.

Table 1. Sensitivity analysis results, including parameter range, rank correlation (ρ_R), and p -value. As shown in earlier works, L and N_r have strong correlation with reservoir accuracy. These results add C and S to the set of parameters that are strongly correlated with accuracy. The number of input layers, I , the number of ESMs, λ , and θ all have noticeably lower correlation.

Parameter	Range	ρ_R	p -value
Gene input fraction (L)	[0.1, 1.0]	0.77	$< 1.00e - 10$
Reservoir genes (N_r)	[10, 500]	0.55	$< 1.00e - 10$
Cells (C)	[1, 15^3]	0.42	$< 1.00e - 10$
Strains (S)	[1, 20]	0.33	$< 1.00e - 10$
Input layers (I)	[1, 15]	0.08	$< 1.00e - 10$
ESM λ	[2.5, 30]	0.04	$1.63e - 05$
ESMs (E)	[0, 20]	-0.06	$4.54e - 09$
ESM θ	[10.0, 100.0]	-0.06	$1.08e - 09$

<https://doi.org/10.1371/journal.pone.0282122.t001>

The intention here is to model the input heterogeneity present in real cellular communities, which can be the result of receptor heterogeneity, limited signal penetration, or engineered spatial segregation. Moreover, with this setup, we more closely tested the role of communication in the reservoir, since the output layer relies solely on intercellular communication for information about the input signal. An analysis of the average performance of the Layered Community for median and parity functions shows that reservoir performance is generally improved over the performance of single cells (S2 Fig in S1 File).

As with the Proof of Concept scenario, we tested the general accuracy of the Layered Community scenario for a wider set of functions. The tested functions include all possible Boolean functions (256) for a 3 bit window, and 256 randomly sampled Boolean functions for a 5 bit window (S3 Fig in S1 File). Recursive versions of functions, as previously described, are also tested. The number of reservoir genes (N_r) per cell is varied between 10, 20, 30, and 100. Notably, the Layered Community has a much wider distribution of accuracies compared to the Proof of Concept.

For the median and parity functions, we varied four parameters in order to determine their effect: number of reservoir genes (N_r), cells (C), strains (S), ESMs (E). N_r is relevant to the relationship between individual computational power and community computational power; C and S are relevant to the size and diversity of the reservoir; and E is relevant to the informational bandwidth of communication. The numbers for cells were chosen such that they can form three square layers without excess cells. Fig 3 shows the relationship between each parameter and accuracy, while keeping all other parameter values fixed (S3 Table in S1 File). Each datapoint is the average accuracy of 100 simulations.

With the exception of the number of strains, all parameters have a monotonic positive relationship with accuracy. For the number of genes and cells, this aligns with the conclusion from Sensitivity Analysis, i.e., that they positively correlate with accuracy. However, gene contribution to accuracy plateaus after around 300 genes. Strains have also been found in Sensitivity Analysis to correlate positively, but in this setup plateau quickly and do not have a clearly monotonic relationship as they start to negatively correlate with accuracy after around 8 strains. Another contrast from Sensitivity Analysis is the clear contribution of the number of ESMs to accuracy. Unlike in Sensitivity Analysis, here the output cells did not have direct access to the input signal and relied on ESM communication through layers.

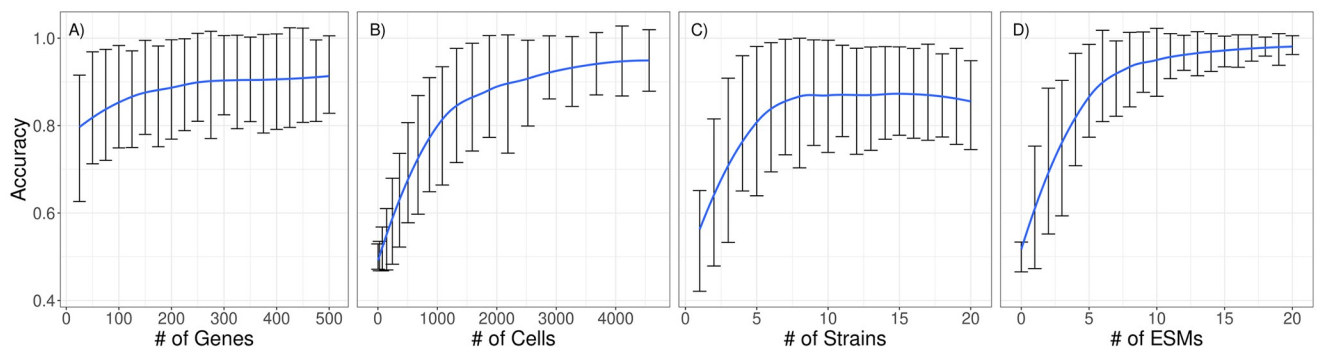


Fig 3. Simulation results from the Layered Community analysis. Four parameters were examined independently while keeping the rest fixed: the number of genes (A), cells (B), strains (C), and ESMs (D). The blue line shows the trend in average accuracy. Each data point along the trend line is the average accuracy of 100 simulations and the bars show \pm one standard deviation. This data shows that the number of cells and ESMs contribute the most to improving accuracy. These are also the parameters most closely responsible for carrying the signal across layers. Genes give a modest improvement to accuracy and plateau around 300. Finally, strains initially improve accuracy noticeably, but peak around 8, after which accuracy starts decreasing.

<https://doi.org/10.1371/journal.pone.0282122.g003>

Discussion

With this work, we aimed to demonstrate that cellular communities utilizing diffusion-based communication are capable of acting as reservoir computers. In natural systems, diffusible signaling molecules are utilized for a range of functions [11, 15, 74], such as the coordination of bacterial host-response with quorum signals [75] and the generation of an immune response with cytokine networks [76]. Diffusible signaling molecules have also been used to form synthetic multicellular networks for ecological functions [37] (e.g. glucose-sensitive insulin production [46]) as well as biocomputing ([35], e.g. as digital circuits [39, 40]). In reservoir computing, previous research has successfully applied neuronal communities with neuroelectric communication schemes as reservoirs. However, communities that rely on diffusion, a more widespread form of communication, had yet to be tested in this role. Here we have provided theoretical evidence demonstrating the potential of such reservoirs. We further demonstrated that these reservoirs are successful under conditions of both input homogeneity, when all cells have access to the input signal and can act as individual reservoirs, and input heterogeneity, when only some cells have access to the input signal and communication is necessary to function as a reservoir. The latter case is likely to be more applicable to engineered systems in which input heterogeneity can be introduced through receptor expression heterogeneity, reduced signal penetration, or externally imposed spatial segregation.

In addition to demonstrating that multicellular communities can act as reservoirs, we have shown that they can outperform their single cell counterparts. Not only was average accuracy higher, but cellular communities were able to perform tasks that single cells could not (Fig 2, parity function window size 5 and 7). However, the factors responsible for the increased computational power remain unclear and are the target for future work. Two potential factors are the increased number of individually capable reservoir computers (cells), from which a prediction can be crowd-sourced; and the connection of the total cellular network through communication, which has more components to contribute to the computing capacity. Multicellular communities use such collective computation, for example, in collective migration along chemotactic gradients [77, 78] and population-wide behavioral switches in response to environmental changes [79, 80]. We found that removing communication when it is not necessary has little effect on performance (S4 Fig in S1 File). Thus, crowd-sourcing from many cells and strains of cells likely plays a significant role in the success of multicellular reservoirs and collective computation may not be a straightforward contributor to reservoir success. Our brief analysis also suggests that communication can negatively impact accuracy when little memory is required for the signal processing task and positively impact accuracy when more memory is required (S5 Fig in S1 File, median function). Therefore, communication may be beneficially increasing the memory of the system while also detrimentally reducing the value of crowd-sourcing by reducing the independence of each cell.

For reservoirs that rely on communication to propagate the input signal, we found that performance can also exceed that of single cells (S2 and S5 Figs in S1 File); though we did observe a decrease in reservoir performance compared to reservoirs that do not need communication. Estimation of the parity function is also notably more difficult when communication is required. The inherent delay as the signal is propagated from input to output cells may be causing the decrease in performance. Previous reservoir computing research has shown that introducing a delay between receiving an input and estimating the function negatively affects performance [64]. Another possibility is that signal propagation between cells introduces noise and uncertainty to the signal, which would contribute to lower accuracy. Unexpectedly, the spatial organization of input and output cells has a very strong effect on reservoir performance. Populations with one half of cells receiving input and the other half used as output, randomly

intermixed in the community, are at a disadvantage (S5 Fig in [S1 File](#)). Intermixed populations actually increase in accuracy as window size increases. More targeted experimentation is needed to further explore the role of communication in reservoir behavior.

A multicellular reservoir's computing capacity also varies with properties inherent to both the individual cell and the cellular community. Here we focused on parameters that have the potential to be incorporated into the design of a multicellular reservoir computer. As previous work has shown for single cell RBN reservoirs, the number of genes within each cell and the number of those genes with input from the signal both positively correlate with the performance of multicellular reservoirs [57, 64]. Intuitively, more genes means more computing power and more genes wired to the input signal means less of an information bottleneck. Notably, there is a nontrivial amount of heterogeneity across reservoir performance with respect to the number of genes and the benefit of adding more genes to each cell quickly diminishes (Fig 3). For engineered systems this suggests that the burden to an individual cell can be kept low without losing out on much computational capacity. Since the work shown here uses random reservoirs, many existing cellular signaling systems have the potential to be repurposed for a signal of interest with little change to the existing architecture.

Considering properties unique to cellular communities, we found that the size, diversity, and number of signaling channels of the community positively correlate with reservoir performance. If crowd-sourcing is, in fact, a contributing factor to computing, then increasing the number and diversity of cells would be advantageous. The effect of ESMs is less straightforward. Accuracy in the communities that do not require communication (Proof of Concept) shows little correlation with ESM-related parameters, suggesting communication provides a minor improvement to these reservoirs. Although, it is also possible that the method used for sensitivity analysis is simply unable to detect a relationship since the interaction distance and threshold form a nonlinear communication parameter space. For the three-layered community, in which communication is necessary, increasing the number of signaling molecules has a large effect on accuracy, though the reason is not obvious. In metazoans, an expansion of the signaling network was observed as organisms evolved more complex cellular architecture [74]. A similar increase in complexity could be related to the increase in accuracy we observe. While more signaling molecules may benefit the efficacy and complexity of signal transmission, it would also reduce the independence of individual cells, which may negatively impact the reservoir which we observe when input/output cells are intermixed instead of layered (S5 Fig in [S1 File](#)). Overall, it appears that a minimum number of cells, strains, and ESMs is necessary for reservoirs to perform well, but that deficiency in one parameter can likely be compensated for by increasing another. This provides flexibility in the prospective engineering of synthetic communities.

Though we have tried to consider real-world conditions in this work, further work is required to identify all limitations that could be encountered in the bio-engineering process. For example, the number of genes that can directly interact with signal receptors may be severely limited in practice. Additional analysis to account for such limitations would be straightforward to test in a similar computational framework. Future work could also include a closer investigation of the effects of communication on computational power versus system memory as well as the effect of different communication regions described by the communication parameters (S6 Fig in [S1 File](#)).

What we have shown with this work is that multicellular communities can act as reservoirs. The accuracy of those reservoirs can be increased with parameters contributing to computational power. We have also identified that the communication scheme and community structure both play important roles in the type of computation performed. And lastly we have shown that, in addition to the benefits typically associated with multicellular systems,

cellular communities can easily outperform single cells with more flexibility in how they can be optimized.

Methods and materials

The simulations were implemented using the Biocellion v1.2 multicellular simulation framework [81]. Fixed simulation parameters can be found in S1 to S3 Tables in [S1 File](#) and their meaning is further described here.

Simulation details

All simulations run for $T = 1000$ time steps. Before each simulation, a random binary input signal array V is generated, equal in size to the number of simulation time steps. In each time step t , the input signal molecule is produced if v_t is 1 (Dirichlet boundary condition is 100,000), otherwise it is not (Dirichlet boundary condition is 0). A numerical solution is implemented that determines the threshold each cell should have to the input signal when it is being produced so that number of cell layers that detect the presence of the input signal molecule is equal to the value of the input layers I parameter. All cells that belong to the input layers register the input signal presence and set the state of the input gene g_1 to 1 for the duration of that simulation time step. The state of g_1 is 0 otherwise. For all simulations, v_0 is guaranteed to be 0 and all molecule concentrations are initialized to 0. The input signal gene g_1 and ESM receptor genes are initialized to 0 for all cells. The initial values of the rest of the genes are randomly generated.

For Proof of Concept and Sensitivity Analysis, cells are arranged in a cube shaped grid, whereas for the Layered Community setup, the cells are arranged in a grid of three square-shaped layers. Each cell occupies one simulation domain voxel and is centered in it.

Random Boolean networks

Every cell has N genes: one input signal gene (1), one receptor and one secreter gene for each ESM ($2 \cdot E$) and the rest are reservoir genes (N_r). When assigning edges in the RBN, the constraints are:

- Input gene and ESM receptor genes cannot have incoming edges.
- ESM secreter genes cannot have outgoing edges.

In order to stay within these constraints and have an average in-degree of two for the reservoir genes, there is a total of $\sum_{i=E+2}^N |e_i| = 2 \cdot (N - (1 + E))$ randomly and uniformly assigned edges in the RBN, where $|e_i|$ is the in-degree of gene i . The incoming edges are counted from $E + 2$ because the input gene (1) and the ESM receptor genes (E) do not have incoming edges. A fraction of these edges is used specifically for connections with the input gene. At most, there are $E + N_r = N - (1 + E)$ edges from the input gene (as the other $1 + E$ genes do not have incoming edges). The number of these edges is $L \cdot (E + N_r)$, where $0 \leq L \leq 1$ is a parameter. Additional information about the RBN construction within cells can be found in S1 Section in [S1 File](#).

Cells communicate with each other using ESMs, which are implemented using the diffusion mechanism in Biocellion. Here we assume that molecule diffusion is a much faster process than gene regulation and so diffusion is implemented with a steady state PDE. A single simulation time step is sufficient for both the input signal and ESMs to reach the steady state. Following this, the RBN for each cell is updated. For each ESM molecule m , diffusion in the environment follows [Eq \(2\)](#) where C_i^m is the concentration of ESM molecule m at voxel i , β is

the diffusion coefficient, α is the degradation rate, η_i^m is the secretion rate of molecule m by the cell at voxel i , and V_c is voxel volume (all voxels have the same volume).

This PDE is numerically calculated [82, 83]. We also define λ as the effective interaction distance, where $\lambda = \sqrt{\beta/\alpha}$. The θ parameter is the threshold for registering ESM presence in the cell's voxel. It is defined as a factor of the ESM concentration in a cell's voxel when the cell is producing the ESM molecule at basal rate and this concentration is numerically calculated. The same value of θ is used for all the ESM. For Proof of Concept and Layered Community, the ESM communication parameters: α (1.0/45.0), β (5.0), λ (15.0, derived from α and β), η (1.0 basal, 5.0 active), and θ (11.5) have the same values as in the previous work done by Echlin et al [57]. For Sensitivity Analysis, β and η are the same as for Proof of Concept and Layered Community, whereas λ and θ parameter ranges are informed by S6 Fig in S1 File. Based on information from previous work [84], the tested parameter values explore scenarios where:

- Intercellular communication does not occur, because the receptor is always on, always off, or only responds to self-generated signal.
- Intercellular communication occurs, where the receptors either respond to neighbor-generated signal, self-generated signal, or a combination of the two.

For each simulation time step, actions are done in the following order (as implemented in Biocellion):

1. For each ESM, the secretion rate (η_i^m) in Eq (2) is updated to active or basal, depending on whether the corresponding secreter gene is on or off.
2. Input signal Dirichlet boundary value is set to 100,000 if the current input signal value is 1, and to 0 otherwise.
3. With the above two steps setting the variables, the PDEs are now solved and molecule concentrations in every voxel stabilized.
4. Input gene and receptor genes are updated according to the environment, i.e. if the concentration of the corresponding molecule is above the threshold, these genes turn on.
5. The secreter and reservoir genes are updated according to their update functions / truth tables. Since this step happens after the input and receptor genes are updated, their new state is used in the update functions. For the other secreter and reservoir genes that are used as inputs into the update functions, the values from the previous timestep are used.

Linear regression

Linear regression analysis was done using the LassoCV function (with *selection* = "random", *tol* = 0.05 parameters for faster convergence) from the Python language scikit-learn library [85]. Each simulation starts off with $T_w = 100$ warm-up time steps, included in the total $T = 1000$ time steps, during which the simulation runs as normal, but the outputs are not used for linear regression. The covariates of linear regression are all reservoir gene states of all cells belonging to the output layers, which are collectively denoted here as g^O . The number of output layers (O) is a parameter and the layers are selected sequentially from the community, starting at the side opposite of the input signal source. Additionally, we define the parameter d which is the delay in time steps between the objective function ground truth output at time t , y_b , and g^O which is taken from time $t + d$, denoted as $g^O(t + d)$. This delay parameter lets the information propagate through the system for d timesteps before linear regression is performed. This is necessary in Layered Community, where it takes two time steps for the signal

to propagate from the first to the last, third community layer. For that reason, d is 2 in that analysis. The input to linear regression is matrix X , where columns correspond to covariates/genes and rows to time steps. If X_i is the i^{th} row of X and w the objective function window size, then for each time step t , $T_W + w + d \leq t \leq T$, $X_{t-(T_W+w+d)+1} = g^O(t)$. Additionally, the ground truth array Y for the objective function f is calculated for each time step t , $T_W + w \leq t \leq T - d$, on the input sequence window: $T_{t-(T_W+w)+1} = f(V_{t-w+1}, V_{t-w+2}, \dots, V_t)$. The rows of X and Y are divided into training (75%) and test dataset (25%) randomly. LassoCV is ran on the training dataset and the accuracy of the model is assessed on the test dataset.

Sensitivity analysis

For Sensitivity Analysis, Latin Hypercube Sampling (LHS) [86] is used to generate 10,000 sets of parameter samples, each simulated 5 times. For each sampled parameter, LHS divides the parameter value range into 10,000 equally sized intervals and takes a randomly selected value uniformly at random from each interval. This ensures an overall fairly uniform sampling which would not be guaranteed by brute force sampling. The order of selected values is then randomly shuffled for each parameter.

In order to calculate rank correlation in Sensitivity Analysis for each parameter individually, the effect of other parameters has to be cancelled out. The parameters and output are first rank-transformed to ignore non-linear effects and keep only monotonicity. Then, for every rank transformed parameter r_i^x and output r^y , the following linear models are built that estimate the parameter and output from other parameters:

$$\hat{r}_i^x = c_0 + \sum_{j=1, j \neq i}^k c_j \cdot r_j^x \quad \hat{r}^y = b_0 + \sum_{j=1, j \neq i}^k b_j \cdot r_j^x$$

Pearson correlation is then calculated between the residuals, which have linear effects of other parameters removed: $\text{corr}(r_i^x - \hat{r}_i^x, r^y - \hat{r}^y)$. The plots showing correlation between each varied parameter and accuracy can be seen in S7 Fig in [S1 File](#), and partial correlation after rank transformation in S8 Fig in [S1 File](#). The distribution of accuracies for Sensitivity Analysis simulation runs can be seen in S9 Fig in [S1 File](#). The fixed parameters for Sensitivity Analysis, found in S2 Table in [S1 File](#) were chosen such that the accuracy distribution would be evenly spread out. If too many simulations were concentrated at 0.5 or 1.0 accuracies, the Sensitivity Analysis results would be less reliable.

Miscellaneous

Plots in this work were generated using the ggplot2 [87] and seaborn [88] libraries.

The simulation code and data produced can be found online at github.com/vlad0x00/multicell-rc and the archived version is available at Zenodo (doi: [10.5281/zenodo.7212658](https://doi.org/10.5281/zenodo.7212658)).

Supporting information

S1 File. Contains all the supporting tables and figures.
(PDF)

Acknowledgments

The authors wish to acknowledge Canada's Michael Smith Genome Sciences Centre, Vancouver, Canada for computing services. The GSC's IT infrastructure is supported by WestGrid (www.westgrid.ca) and Compute Canada (www.computecanada.ca). M.E. acknowledges

support from Tampere University, Tampere, Finland. I.S. and B.A. gratefully acknowledge support from the ISB.

Author Contributions

Conceptualization: Boris Aguilar, Ilya Shmulevich.

Data curation: Vladimir Nikolić.

Formal analysis: Vladimir Nikolić.

Investigation: Vladimir Nikolić.

Methodology: Vladimir Nikolić, Moriah Echlin, Boris Aguilar, Ilya Shmulevich.

Project administration: Boris Aguilar, Ilya Shmulevich.

Resources: Boris Aguilar.

Software: Vladimir Nikolić, Boris Aguilar.

Supervision: Moriah Echlin, Boris Aguilar, Ilya Shmulevich.

Validation: Vladimir Nikolić, Moriah Echlin.

Visualization: Vladimir Nikolić, Moriah Echlin.

Writing – original draft: Vladimir Nikolić, Moriah Echlin.

Writing – review & editing: Vladimir Nikolić, Moriah Echlin, Boris Aguilar, Ilya Shmulevich.

References

1. Fisher MJ, Paton RC, Matsuno K. Intracellular signalling proteins as ‘smart’ agents in parallel distributed processes. *Biosystems*. 1999; 50(3):159–171. [https://doi.org/10.1016/S0303-2647\(99\)00006-4](https://doi.org/10.1016/S0303-2647(99)00006-4) PMID: 10400267
2. Pinto D, Mascher T. (Actino)Bacterial “intelligence”: using comparative genomics to unravel the information processing capacities of microbes. *Current Genetics*. 2016; 62(3):487–498. <https://doi.org/10.1007/s00294-016-0569-3> PMID: 26852121
3. Lan G, Tu Y. Information processing in bacteria: memory, computation, and statistical physics: a key issues review. *Reports on Progress in Physics*. 2016; 79(5):052601. <https://doi.org/10.1088/0034-4885/79/5/052601> PMID: 27058315
4. Ray SK, Valentini G, Shah P, Haque A, Reid CR, Weber GF, et al. Information Transfer During Food Choice in the Slime Mold *Physarum polycephalum*. *Frontiers in Ecology and Evolution*. 2019; 7. <https://doi.org/10.3389/fevo.2019.00067>
5. Brumley DR, Carrara F, Hein AM, Yawata Y, Levin SA, Stocker R. Bacteria push the limits of chemotactic precision to navigate dynamic chemical gradients. *Proceedings of the National Academy of Sciences*. 2019; 116(22):10792–10797. <https://doi.org/10.1073/pnas.1816621116> PMID: 31097577
6. Sarris M, Sixt M. Navigating in tissue mazes: chemoattractant interpretation in complex environments. *Current Opinion in Cell Biology*. 2015; 36:93–102. <https://doi.org/10.1016/j.ceb.2015.08.001> PMID: 26355911
7. Reid CR, Latty T, Dussutour A, Beekman M. Slime mold uses an externalized spatial “memory” to navigate in complex environments. *Proceedings of the National Academy of Sciences*. 2012; 109(43):17490–17494. <https://doi.org/10.1073/pnas.1215037109> PMID: 23045640
8. Cardelli L, Csikász-Nagy A. The Cell Cycle Switch Computes Approximate Majority. *Scientific Reports*. 2012; 2(1). <https://doi.org/10.1038/srep00656> PMID: 22977731
9. Kobayashi TJ, Kamimura A. Theoretical Aspects of Cellular Decision-Making and Information-Processing. In: *Advances in Experimental Medicine and Biology*. Springer New York; 2011. p. 275–291. Available from: https://doi.org/10.1007/978-1-4419-7210-1_16.
10. Bowsher CG, Swain PS. Environmental sensing, information transfer, and cellular decision-making. *Current Opinion in Biotechnology*. 2014; 28:149–155. <https://doi.org/10.1016/j.copbio.2014.04.010> PMID: 24846821

11. Bassel GW. Information Processing and Distributed Computation in Plant Organs. *Trends in Plant Science*. 2018; 23(11):994–1005. <https://doi.org/10.1016/j.tplants.2018.08.006> PMID: 30219546
12. Martinez-Corral R, Liu J, Prindle A, Süel GM, Garcia-Ojalvo J. Metabolic basis of brain-like electrical signalling in bacterial communities. *Philosophical Transactions of the Royal Society B: Biological Sciences*. 2019; 374(1774):20180382. <https://doi.org/10.1098/rstb.2018.0382> PMID: 31006362
13. Marre O, Botella-Soler V, Simmons KD, Mora T, Tkačik G, Berry MJ. High Accuracy Decoding of Dynamical Motion from a Large Retinal Population. *PLOS Computational Biology*. 2015; 11(7): e1004304. <https://doi.org/10.1371/journal.pcbi.1004304> PMID: 26132103
14. Peak D, West JD, Messinger SM, Mott KA. Evidence for complex, collective dynamics and emergent, distributed computation in plants. *Proceedings of the National Academy of Sciences*. 2004; 101(4):918–922. <https://doi.org/10.1073/pnas.0307811100> PMID: 14732685
15. Ben-Jacob E, Levine H. Self-engineering capabilities of bacteria. *Journal of The Royal Society Interface*. 2005; 3(6):197–214. <https://doi.org/10.1098/rsif.2005.0089>
16. Underhill DM, Goodridge HS. Information processing during phagocytosis. *Nature Reviews Immunology*. 2012; 12(7):492–502. <https://doi.org/10.1038/nri3244> PMID: 22699831
17. Agliari E, Annibale A, Barra A, Coolen ACC, Tantari D. Retrieving infinite numbers of patterns in a spin-glass model of immune networks. *EPL (Europhysics Letters)*. 2017; 117(2):28003. <https://doi.org/10.1209/0295-5075/117/28003>
18. Singh V. Recent advances and opportunities in synthetic logic gates engineering in living cells. *Systems and Synthetic Biology*. 2014; 8(4):271–282. <https://doi.org/10.1007/s11693-014-9154-6> PMID: 26396651
19. Rubens JR, Selvaggio G, Lu TK. Synthetic mixed-signal computation in living cells. *Nature Communications*. 2016; 7(1). <https://doi.org/10.1038/ncomms11658> PMID: 27255669
20. Gardner TS, Cantor CR, Collins JJ. Construction of a genetic toggle switch in *Escherichia coli*. *Nature*. 2000; 403(6767):339–342. <https://doi.org/10.1038/35002131> PMID: 10659857
21. Siuti P, Yazbek J, Lu TK. Engineering genetic circuits that compute and remember. *Nature Protocols*. 2014; 9(6):1292–1300. <https://doi.org/10.1038/nprot.2014.089> PMID: 24810038
22. Roquet N, Soleimany AP, Ferris AC, Aaronson S, Lu TK. Synthetic recombinase-based state machines in living cells. *Science*. 2016; 353(6297):aad8559–aad8559. <https://doi.org/10.1126/science.aad8559> PMID: 27463678
23. Doshi J, Willis K, Madurga A, Stelzer C, Benenson Y. Multiple Alternative Promoters and Alternative Splicing Enable Universal Transcription-Based Logic Computation in Mammalian Cells. *Cell Reports*. 2020; 33(9):108437. <https://doi.org/10.1016/j.celrep.2020.108437> PMID: 33264624
24. Matsuura S, Ono H, Kawasaki S, Kuang Y, Fujita Y, Saito H. Synthetic RNA-based logic computation in mammalian cells. *Nature Communications*. 2018; 9(1). <https://doi.org/10.1038/s41467-018-07181-2> PMID: 30451868
25. Green AA, Kim J, Ma D, Silver PA, Collins JJ, Yin P. Complex cellular logic computation using ribocomputing devices. *Nature*. 2017; 548(7665):117–121. <https://doi.org/10.1038/nature23271> PMID: 28746304
26. Courbet A, Amar P, Fages F, Renard E, Molina F. Computer-aided biochemical programming of synthetic microreactors as diagnostic devices. *Molecular Systems Biology*. 2018; 14(4). <https://doi.org/10.15252/msb.20177845>
27. Zhang Y, Kojima T, Kim GA, McNERney MP, Takayama S, Styczynski MP. Protocell arrays for simultaneous detection of diverse analytes. *Nature Communications*. 2021; 12(1). <https://doi.org/10.1038/s41467-021-25989-3> PMID: 34588445
28. Goñi-Moreno A, Amos M, de la Cruz F. Multicellular Computing Using Conjugation for Wiring. *PLoS ONE*. 2013; 8(6):e65986. <https://doi.org/10.1371/journal.pone.0065986> PMID: 23840385
29. Macía J, Posas F, Solé RV. Distributed computation: the new wave of synthetic biology devices. *Trends in Biotechnology*. 2012; 30(6):342–349. <https://doi.org/10.1016/j.tibtech.2012.03.006> PMID: 22516742
30. Grozinger L, Amos M, Gorochofski TE, Carbonell P, Oyarzún DA, Stoof R, et al. Pathways to cellular supremacy in biocomputing. *Nature Communications*. 2019; 10(1). <https://doi.org/10.1038/s41467-019-13232-z> PMID: 31748511
31. Boo A, Ellis T, Stan GB. Host-aware synthetic biology. *Current Opinion in Systems Biology*. 2019; 14:66–72. <https://doi.org/10.1016/j.coisb.2019.03.001>
32. Ceroni F, Boo A, Furini S, Gorochofski TE, Borkowski O, Ladak YN, et al. Burden-driven feedback control of gene expression. *Nature Methods*. 2018; 15(5):387–393. <https://doi.org/10.1038/nmeth.4635> PMID: 29578536

33. Cardinale S, Arkin AP. Contextualizing context for synthetic biology—identifying causes of failure of synthetic biological systems. *Biotechnology Journal*. 2012; 7(7):856–866. <https://doi.org/10.1002/biot.201200085> PMID: 22649052
34. Amos M. Population-based microbial computing: a third wave of synthetic biology? *International Journal of General Systems*. 2014; 43(7):770–782. <https://doi.org/10.1080/03081079.2014.921001>
35. Solé RV, Macia J. Expanding the landscape of biological computation with synthetic multicellular consortia. *Natural Computing*. 2013; 12(4):485–497. <https://doi.org/10.1007/s11047-013-9380-y>
36. Adamala KP, Martin-Alarcon DA, Guthrie-Honea KR, Boyden ES. Engineering genetic circuit interactions within and between synthetic minimal cells. *Nature Chemistry*. 2016; 9(5):431–439. <https://doi.org/10.1038/nchem.2644> PMID: 28430194
37. Hennig S, Rödel G, Ostermann K. Artificial cell-cell communication as an emerging tool in synthetic biology applications. *Journal of Biological Engineering*. 2015; 9(1). <https://doi.org/10.1186/s13036-015-0011-2> PMID: 26265937
38. Regot S, Macia J, Conde N, Furukawa K, Kjellén J, Peeters T, et al. Distributed biological computation with multicellular engineered networks. *Nature*. 2010; 469(7329):207–211. <https://doi.org/10.1038/nature09679> PMID: 21150900
39. Tamsir A, Tabor JJ, Voigt CA. Robust multicellular computing using genetically encoded NOR gates and chemical ‘wires’. *Nature*. 2010; 469(7329):212–215. <https://doi.org/10.1038/nature09565> PMID: 21150903
40. Mogas-Díez S, Gonzalez-Flo E, Macía J. 2D printed multicellular devices performing digital and analogue computation. *Nature Communications*. 2021; 12(1). <https://doi.org/10.1038/s41467-021-21967-x> PMID: 33723265
41. Guiziou S, Mayonove P, Bonnet J. Hierarchical composition of reliable recombinase logic devices. *Nature Communications*. 2019; 10(1). <https://doi.org/10.1038/s41467-019-08391-y> PMID: 30692530
42. Macia J, Manzoni R, Conde N, Urrios A, de Nadal E, Solé R, et al. Implementation of Complex Biological Logic Circuits Using Spatially Distributed Multicellular Consortia. *PLOS Computational Biology*. 2016; 12(2):e1004685. <https://doi.org/10.1371/journal.pcbi.1004685> PMID: 26829588
43. Manicka S, Levin M. Modeling somatic computation with non-neural bioelectric networks. *Scientific Reports*. 2019; 9(1). <https://doi.org/10.1038/s41598-019-54859-8> PMID: 31819119
44. Joesaar A, Yang S, Bögels B, van der Linden A, Pieters P, Kumar BVVSP, et al. DNA-based communication in populations of synthetic protocells. *Nature Nanotechnology*. 2019; 14(4):369–378. <https://doi.org/10.1038/s41565-019-0399-9> PMID: 30833694
45. Urrios A, Macia J, Manzoni R, Conde N, Bonforti A, de Nadal E, et al. A Synthetic Multicellular Memory Device. *ACS Synthetic Biology*. 2016; 5(8):862–873. <https://doi.org/10.1021/acssynbio.5b00252> PMID: 27439436
46. Urrios A, Gonzalez-Flo E, Canadell D, de Nadal E, Macia J, Posas F. Plug-and-Play Multicellular Circuits with Time-Dependent Dynamic Responses. *ACS Synthetic Biology*. 2018; 7(4):1095–1104. <https://doi.org/10.1021/acssynbio.7b00463> PMID: 29584406
47. Basu S, Gerchman Y, Collins CH, Arnold FH, Weiss R. A synthetic multicellular system for programmed pattern formation. *Nature*. 2005; 434(7037):1130–1134. <https://doi.org/10.1038/nature03461> PMID: 15858574
48. Li X, Rizik L, Kravchik V, Khoury M, Korin N, Daniel R. Synthetic neural-like computing in microbial consortia for pattern recognition. *Nature Communications*. 2021; 12(1). <https://doi.org/10.1038/s41467-021-23336-0> PMID: 34035266
49. Gorochofski TE, Hauer S, Kreft JU, Marucci L, Stillman NR, Tang TYD, et al. Toward Engineering Biosystems With Emergent Collective Functions. *Frontiers in Bioengineering and Biotechnology*. 2020; 8. <https://doi.org/10.3389/fbioe.2020.00705> PMID: 32671054
50. Jaeger H. Adaptive Nonlinear System Identification with Echo State Networks. In: *Proceedings of the 15th International Conference on Neural Information Processing Systems*. NIPS’02. Cambridge, MA, USA: MIT Press; 2002. p. 609–616.
51. Maass W, Natschläger T, Markram H. Real-Time Computing Without Stable States: A New Framework for Neural Computation Based on Perturbations. *Neural Computation*. 2002; 14(11):2531–2560. <https://doi.org/10.1162/089976602760407955> PMID: 12433288
52. Tanaka G, Yamane T, Héroux JB, Nakane R, Kanazawa N, Takeda S, et al. Recent advances in physical reservoir computing: A review. *Neural Networks*. 2019; 115:100–123. <https://doi.org/10.1016/j.neunet.2019.03.005> PMID: 30981085
53. Dambre J, Verstraeten D, Schrauwen B, Massar S. Information Processing Capacity of Dynamical Systems. *Scientific Reports*. 2012; 2(1). <https://doi.org/10.1038/srep00514> PMID: 22816038

54. Balleza E, Alvarez-Buylla ER, Chaos A, Kauffman S, Shmulevich I, Aldana M. Critical Dynamics in Genetic Regulatory Networks: Examples from Four Kingdoms. *PLoS ONE*. 2008; 3(6):e2456. <https://doi.org/10.1371/journal.pone.0002456> PMID: 18560561
55. Jones B, Stekel D, Rowe J, Fernando C. Is there a Liquid State Machine in the Bacterium *Escherichia Coli*? In: 2007 IEEE Symposium on Artificial Life. IEEE; 2007. Available from: <https://doi.org/10.1109/alife.2007.367795>.
56. Gabalda-Sagarra M, Carey LB, Garcia-Ojalvo J. Recurrence-based information processing in gene regulatory networks. *Chaos: An Interdisciplinary Journal of Nonlinear Science*. 2018; 28(10):106313. <https://doi.org/10.1063/1.5039861> PMID: 30384649
57. Echlin M, Aguilar B, Notarangelo M, Gibbs DL, Shmulevich I. Flexibility of Boolean Network Reservoir Computers in Approximating Arbitrary Recursive and Non-Recursive Binary Filters. *Entropy*. 2018; 20(12). <https://doi.org/10.3390/e20120954> PMID: 33266678
58. Seoane LF. Evolutionary aspects of reservoir computing. *Philosophical Transactions of the Royal Society B: Biological Sciences*. 2019; 374(1774):20180377. <https://doi.org/10.1098/rstb.2018.0377> PMID: 31006369
59. Enel P, Procyk E, Quilodran R, Dominey PF. Reservoir Computing Properties of Neural Dynamics in Prefrontal Cortex. *PLOS Computational Biology*. 2016; 12(6):e1004967. <https://doi.org/10.1371/journal.pcbi.1004967> PMID: 27286251
60. Yamazaki T, Tanaka S. The cerebellum as a liquid state machine. *Neural Networks*. 2007; 20(3):290–297. <https://doi.org/10.1016/j.neunet.2007.04.004> PMID: 17517494
61. Reid D, Barrett-Baxendale M. Glial Reservoir Computing. In: 2008 Second UKSIM European Symposium on Computer Modeling and Simulation. IEEE; 2008. Available from: <https://doi.org/10.1109/ems.2008.74>.
62. George JB, Abraham GM, Singh K, Ankolekar SM, Amrutur B, Sikdar SK. Input coding for neuro-electronic hybrid systems. *Biosystems*. 2014; 126:1–11. <https://doi.org/10.1016/j.biosystems.2014.08.002> PMID: 25110321
63. Ortman RL, Venayagamoorthy K, Potter SM. Input Separability in Living Liquid State Machines. In: Adaptive and Natural Computing Algorithms. Springer Berlin Heidelberg; 2011. p. 220–229. Available from: https://doi.org/10.1007/978-3-642-20282-7_23.
64. Snyder D, Goudarzi A, Teuscher C. Computational capabilities of random automata networks for reservoir computing. *Physical Review E*. 2013; 87(4). <https://doi.org/10.1103/PhysRevE.87.042808> PMID: 23679474
65. Bye ET. Investigation of Elementary Cellular Automata for Reservoir Computing; 2016.
66. Nichele S, and MSG. Reservoir Computing Using Nonuniform Binary Cellular Automata. *Complex Systems*. 2017; 26(3):225–246. <https://doi.org/10.25088/ComplexSystems.26.3.225>
67. Babson N, and CT. Reservoir Computing with Complex Cellular Automata. *Complex Systems*. 2019; 28(4):433–455. <https://doi.org/10.25088/ComplexSystems.28.4.433>
68. Olimpio EP, Dang Y, Youk H. Statistical Dynamics of Spatial-Order Formation by Communicating Cells. *iScience*. 2018; 2:27–40. <https://doi.org/10.1016/j.isci.2018.03.013> PMID: 30428376
69. Aguilar B, Gibbs DL, Reiss DJ, McConnell M, Danziger SA, Dervan A, et al. A generalizable data-driven multicellular model of pancreatic ductal adenocarcinoma. *GigaScience*. 2020; 9(7). <https://doi.org/10.1093/gigascience/giaa075> PMID: 32696951
70. Kauffman S. Homeostasis and Differentiation in Random Genetic Control Networks. *Nature*. 1969; 224(5215):177–178. <https://doi.org/10.1038/224177a0> PMID: 5343519
71. Derrida B, Pomeau Y. Random Networks of Automata: A Simple Annealed Approximation. *Europhysics Letters (EPL)*. 1986; 1(2):45–49. <https://doi.org/10.1209/0295-5075/1/2/001>
72. Kauffman S. The origins of order: self-organization and selection in evolution. New York: Oxford University Press; 1993.
73. Marino S, Hogue IB, Ray CJ, Kirschner DE. A methodology for performing global uncertainty and sensitivity analysis in systems biology. *Journal of Theoretical Biology*. 2008; 254(1):178–196. <https://doi.org/10.1016/j.jtbi.2008.04.011> PMID: 18572196
74. Ramilowski JA, Goldberg T, Harshbarger J, Kloppmann E, Lizio M, Satagopam VP, et al. A draft network of ligand–receptor-mediated multicellular signalling in human. *Nature Communications*. 2015; 6(1). <https://doi.org/10.1038/ncomms8866> PMID: 26198319
75. Dow JM. Diffusible signal factor-dependent quorum sensing in pathogenic bacteria and its exploitation for disease control. *Journal of Applied Microbiology*. 2016; 122(1):2–11. <https://doi.org/10.1111/jam.13307> PMID: 27684652
76. Liongue C, Sertori R, Ward AC. Evolution of Cytokine Receptor Signaling. *The Journal of Immunology*. 2016; 197(1):11–18. <https://doi.org/10.4049/jimmunol.1600372> PMID: 27317733

77. Ellison D, Mugler A, Brennan MD, Lee SH, Huebner RJ, Shamir ER, et al. Cell–cell communication enhances the capacity of cell ensembles to sense shallow gradients during morphogenesis. *Proceedings of the National Academy of Sciences*. 2016; 113(6). <https://doi.org/10.1073/pnas.1516503113> PMID: 26792522
78. Camley BA. Collective gradient sensing and chemotaxis: modeling and recent developments. *Journal of Physics: Condensed Matter*. 2018; 30(22):223001. <https://doi.org/10.1088/1361-648X/aabd9f> PMID: 29644981
79. Shin YJ, Mahrou B. Modeling collective & intelligent decision making of multi-cellular populations. In: 2014 36th Annual International Conference of the IEEE Engineering in Medicine and Biology Society. IEEE; 2014. Available from: <https://doi.org/10.1109/embc.2014.6943597>.
80. Chivasa S, Goodman HL. Stress-adaptive gene discovery by exploiting collective decision-making of decentralized plant response systems. *New Phytologist*. 2019; 225(6):2307–2313. <https://doi.org/10.1111/nph.16273> PMID: 31625607
81. Kang S, Kahan S, McDermott J, Flann N, Shmulevich I. Biocellion: accelerating computer simulation of multicellular biological system models. *Bioinformatics*. 2014; 30(21):3101–3108. <https://doi.org/10.1093/bioinformatics/btu498> PMID: 25064572
82. Maire T, Youk H. Molecular-Level Tuning of Cellular Autonomy Controls the Collective Behaviors of Cell Populations. *Cell Systems*. 2015; 1(5):349–360. <https://doi.org/10.1016/j.cels.2015.10.012> PMID: 27136241
83. Berg HC. *Random Walks in Biology*. Princeton University Press; 2018.
84. Echlin M. *A Complex Systems Approach to Understanding Cells as Systems and Agents* [PhD dissertation]. University of Washington; 2019.
85. Pedregosa F, Varoquaux G, Gramfort A, Michel V, Thirion B, Grisel O, et al. Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research*. 2011; 12:2825–2830.
86. McKay MD, Beckman RJ, Conover WJ. A Comparison of Three Methods for Selecting Values of Input Variables in the Analysis of Output from a Computer Code. *Technometrics*. 1979; 21(2):239. <https://doi.org/10.1080/00401706.1979.10489755>
87. Wickham H. *ggplot2: Elegant Graphics for Data Analysis*. Springer-Verlag New York; 2016. Available from: <https://ggplot2.tidyverse.org>.
88. Waskom ML. seaborn: statistical data visualization. *Journal of Open Source Software*. 2021; 6(60):3021. <https://doi.org/10.21105/joss.03021>