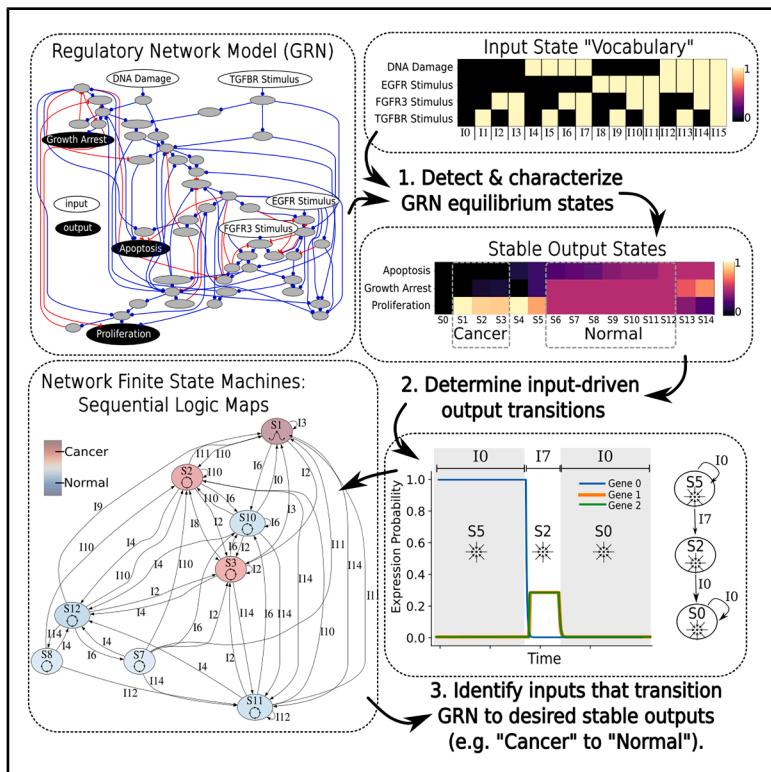


Harnessing the analog computing power of regulatory networks with the Regulatory Network Machine

Graphical abstract



Authors

Alexis Pietak, Michael Levin

Correspondence

alexis.pietak@gmail.com (A.P.),
michael.levin@tufts.edu (M.L.)

In brief

Systems biology; Computer science;
Bioengineering

Highlights

- Developed GRN analysis tool called the Regulatory Network Machine (RNM)
- RNMs map input-driven transitions between stable equilibrium states of GRNs
- RNMs reveal sequential logic in GRNs and intelligent behaviors
- RNMs identify strategies for multiple biomedical/technological applications



Article

Harnessing the analog computing power of regulatory networks with the Regulatory Network Machine

Alexis Pietak^{1,3,*} and Michael Levin^{1,2,4,*}

¹Allen Discovery Center, Tufts University, Medford, MA 02155, USA

²Center for Regenerative and Developmental Biology, Tufts University, Medford, MA 02155, USA

³Ionovate Inc., Kingston, ON K7P 0L5, Canada

⁴Lead contact

*Correspondence: alexis.pietak@gmail.com (A.P.), michael.levin@tufts.edu (M.L.)

<https://doi.org/10.1016/j.isci.2025.112536>

SUMMARY

Gene regulatory networks (GRNs) are critically important for efforts in biomedicine and biotechnology. Here, we introduce the Regulatory Network Machine (RNM) framework, demonstrating how GRNs behave as analog computers capable of sophisticated information processing. Our RNM framework encapsulates: (1) a dissipative dynamic system with a focus on GRNs, (2) a set of inputs to the system, (3) system output states with identifiable relevance to biotechnological or biomedical objectives, and (4) Network Finite State Machines (NFSMs), which are maps detailing how the system changes equilibrium state in response to patterns of applied inputs. As an extension to attractor landscape analysis, the NFSMs map the sequential logic inherent in the GRN and, therefore, embody the “software-like” nature of the system, providing easy identification of specific applied interventions necessary to achieve desired, stable biological outcomes. We illustrate the use of our RNM framework in important biological examples, including in cancer renormalization.

INTRODUCTION

Surviving and thriving in highly dynamic, stochastic environments requires living systems to be adept at sensing environmental conditions, processing the information obtained from the senses, and implementing a response that is maximally effective for the organism’s survival. As even single-celled organisms lacking central nervous systems are capable of sophisticated, context-dependent information processing and finely tuned process control in stochastic environments,^{1–5} it becomes critical to understand how these capabilities may be instantiated in terms of biological algorithms and mechanisms.

For example, a yeast cell can sense a plethora of environmental characteristics and respond in surprisingly complex and nuanced ways, including the ability to anticipate future events with survival-enhancing responses.^{5–9} Likewise, single cells in a developing multicellular organism sense transient morphogenic cues and respond by developing permanent and significant changes to gene expression profiles, representing their differentiation into different cell lineages and types.^{10,11} In a mature organism, a transient event such as wounding can initiate tightly regulated, multi-step wound healing^{12–14} or regeneration¹⁵ responses involving sophisticated cascades of activity that restore the initial (healed) state of the organism. In all these cases, the same control schemes take an “experience”—that is, input information obtained from the environment via sensors/receptors—and process that information to engage survival-

appropriate responses through changes in the expression or activity of its own components. Alternatively, under other circumstances dysregulation can occur, wherein an organism’s state transitions from one of “health”, where the structural and functional properties of the organism are stable and life-promoting, to one of “disease”, where disorganization and disruption of physical structures and processes occurs. A precise and detailed understanding of what this transition means, how it happens, and how to effectively move from the state of disease and back to one of health has not been defined for biological systems with outcomes governed by networks including GRNs.

As regulatory networks, GRNs are collectives of molecular regulators, including protein-based transcription factor gene products, which up- or down-regulate each other’s expressions in a defined functional connectivity map.^{16,17} GRNs serve a major role in regulating the behavior of single cells, multicellular tissues, organs, and organisms,^{16–18} and are therefore recognized as critically important targets of inquiry for evolutionary developmental biology,^{19–21} biomedicine,^{18,22} bioengineering,^{23–25} and synthetic biology.^{26–29} It is essential to understand how to predict their behaviors and how to manipulate these networks to induce desired biological outcomes.^{30–35} These networks are well-recognized to have emergent properties,^{36–42} yet their control remains challenging, with several perspectives emerging on their system-level regulation.^{22,43–47} Therefore, frameworks that elucidate and focus our understanding of information processing by GRN are acutely needed in biological and biomedical fields.



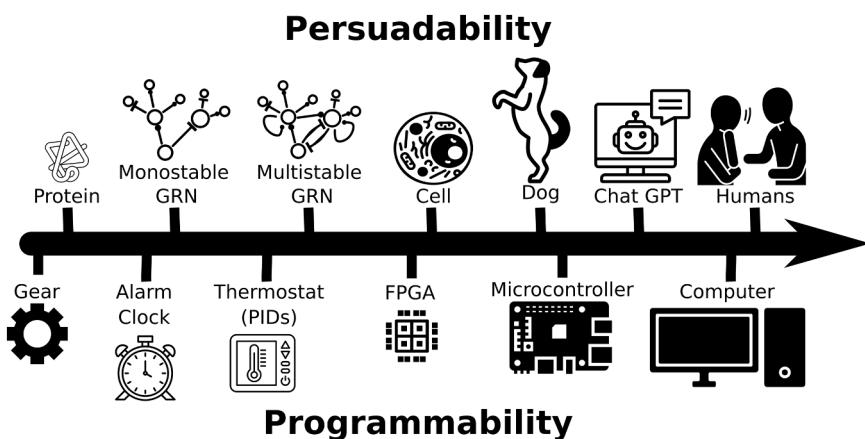


Figure 1. Schematic of the persuadability spectrum

In a persuadable system, simple information signals (e.g., pharmaceuticals, treats, words) can activate inherent competencies in a system to get it to emit a desired behavior, or to perform a desired function. In contrast to persuadability, a programmable system is one in which detailed information is used to precisely specify (i.e., micromanage) how a system should behave in response to different conditions and in general comes at more effort and expense than being able to persuade a system to emit desired behavior.

A gradualist approach considers the essential problem-solving strategies that lifeforms utilized to thrive and survive prior to the evolution of advanced brains. We envision a continuum of advances in the information processing capacities of a system, which we call the *Persuadability Spectrum*,⁴⁸ Figure 1. A system is “persuadable” to the degree to which concise information signals (e.g., pharmaceuticals, food rewards, and words) can activate inherent competencies in a system to get it to emit a desired behavior or to perform a desired function. Highly persuadable systems offer complex, autonomous responses to low information-content trigger prompts. Mechanistic physical systems, such as gears, are present on one extrema of the continuum (Figure 1). If one desires a change of function (“behavior”), these systems cannot be persuaded to change; rather, change in their output can only occur with alteration of their hardware (i.e., by changing their physical make-up). On the other extreme are human beings, whose behavior can be radically changed by a brief communication encoding a rationally or emotionally persuasive argument that alters values, beliefs, and/or feelings (Figure 1). We propose that between these extremes lies a rich panoply of intermediate agents, including biological networks such as GRNs, which can be controlled by signals, stimuli, and training, without needing to directly alter the physical substratum of the agent.⁴⁸

Here, we develop a conceptual framework and related computational system for working with GRNs as persuadable agents. Regulatory networks such as GRNs have classically been assumed to be mechanistic, “clockwork-like” systems requiring direct and permanent changes to network topology via genetic modification to effect stable changes in GRN output.^{49–55} However, gene therapy is still a significant barrier for biomedical interventions; moreover, the inverse problem of discovering *how* to modify a genetic network to achieve a desired set of responses is extremely challenging,⁵⁶ motivating the search for different ways to predict and control GRN behavior. Our assertion is that such *a priori* assumptions are unwarranted and unnecessarily limit advances in our fundamental understanding and productive work in the fields of both basal intelligence and biomedicine. In contrast, we hypothesize GRNs to be “persuadable” which, if true, opens the door to a powerful set of strategies for their manipulation.

Our general view is of regulatory networks, including GRNs, as a class of *dissipative dynamic system* (see Box 1 for definitions), which can exist in one of multiple possible *equilibrium states* (Box 1) at a time, where the collective state of all nodes in the network is a state of the GRN (Figure 2). Persuasion of the GRN system requires the capability to define and apply inputs to the GRN network, which are precisely the concise information signals that can activate inherent competencies in the GRN system and are therefore a requirement for persuasive control of the GRN (Figure 2B). The input signals to the GRN system may be of a chemical (e.g., chemical, biochemical, and morphogen) or physical (e.g., light, pressure, and temperature) nature (Figure 2B). As a dissipative dynamic system, the application of an input signal of sufficient strength and duration causes the GRN to exist as (i.e., occupy) one of its equilibrium states (Figure 2C). The states of the GRN are patterns of gene expression (Figure 2C) which are known to be directly associated with biological outcomes^{55,59} and, therefore, the GRN states are essentially synonymous with important cell and organism states, examples including cell phenotype (e.g., “embryonic stem cell” or “fibroblast”), the health status of the cell (e.g., “cancer” or “normal”), or different physiological modalities (e.g., “stress response” versus “resting response”). The overall technical objective of persuading the GRN system is to identify patterns of input stimuli that can transition the GRN system from the occupation of one (undesirable, e.g., diseased) equilibrium state to another (desirable, e.g., healthy) output state without requiring a modification of the network structure. Moreover, we wish to identify patterns of input stimuli that can make this transition permanent after the input stimulus is removed, which enables avenues of research such as movement beyond the paradigm of “one constant dose” medicine.

We use computational modeling of GRNs to identify the equilibrium state that the GRN system transitions to in response to each input signal applied to the system when it starts in each equilibrium state. We use this set of equilibrium state transitions induced by input signals to build Network-associated Finite State Machines (NFSMs). Finite State Machines (FSMs) are universal models of sequential logic⁶⁰ (Box 1) and, therefore, the NFSM of a GRN is a mapping of the sequential logic that the GRN system can exhibit. Note that while an NFSM may appear superficially similar to a GRN, the NFSM is not a regulatory network model like a GRN; rather, it is a map detailing how an

Box 1. Glossary of commonly used terms

Analog computing: A computational method that uses continuous physical phenomena, such as electrical voltages or mechanical movements, to represent and manipulate data. Unlike digital computing, which represents information in discrete bits, analog computing operates on continuous signals. This approach can be highly efficient for certain types of problems, particularly those involving differential equations and simulations of dynamic systems.

Asymptotically stable: A system is considered asymptotically stable if, given an initial condition sufficiently close to an equilibrium point, the system's state will converge to that equilibrium point as time approaches infinity. In simpler terms, it means that the system will eventually settle down to an equilibrium state, regardless of small disturbances. This concept is crucial in various fields, including control theory, physics, and engineering, where it is used to analyze the stability of systems and design controllers to stabilize them.

Attractor: A set of states toward which a system tends to evolve. It is a sort of "magnetic" point or region in the system's phase space that pulls the system's trajectory toward it. Attractors can be simple, like a single point (a fixed-point attractor), or more complex, like a closed loop (a periodic attractor). In chaotic systems, attractors can be intricate geometric shapes known as strange attractors.

Context: The circumstances that form the setting for an event, statement, or idea, and in terms of which it can be fully understood and assessed. It encompasses the surrounding environment, background information, and relevant factors that influence meaning and interpretation. In the realm of language, context helps determine the specific meaning of words and phrases. In broader terms, it refers to the social, cultural, historical, or political setting within which something occurs. Understanding context is crucial for effective communication, problem-solving, and decision-making.

Compute: To determine something, often using mathematical methods or a computer. It involves processing information and performing operations to arrive at a specific result. For example, one might compute the area of a circle, the average of a set of numbers, or the solution to a complex equation.

Context ratio: Used in the estimate of the Intelligence Potential of a Regulatory Network Machine, the context ratio (CR) is the total number of non-trivial input contexts (N Contexts) scaled by the number of input states, where a non-trivial input context is an input-specified state subspace containing more than one equilibrium (i.e., a multistable sub-state).

Dissipative dynamic system: A system that loses energy over time, leading to a reduction in its overall activity. This energy loss can be due to various factors such as friction, heat dissipation, or other forms of energy transfer. As a result, the system's trajectories converge toward a smaller region in its phase space, often referred to as an attractor. Dissipative systems are common in nature and engineering. Examples include mechanical systems with friction, electrical circuits with resistance, and biological systems that lose energy through metabolism.

Direction field: An array of directions defined at each point in state space that indicates the trajectory/time evolution of the system from that specific state point (Figure 1). When the system is at any point/state in the state space, the direction field is akin to a force field that directs the evolution of the system to another state. The direction field of the state space is akin to a leaf (system state) moving along on the currents of a river (direction field). The direction field allows for visualization and understanding of the different characteristics of the dynamic system, including attractors and limit cycles (cyclic attractors).

Equilibrium: A state of balance or stability. In a system at equilibrium, there is no net change over time. Equilibrium can be either static (no change) or dynamic (continuous change without a net change).

Finite state machine: A mathematical model of computation used to design algorithms. It is a simple machine with a finite number of states. At any given time, the machine exists in exactly one of these states. When the machine receives an input, it transitions from its current state to another state or remains in the same state. This transition is determined by a set of rules defined for the FSM. The machine can also produce an output based on its current state and the input it receives. FSMs are widely used in computer science and engineering to model various systems, from simple traffic lights to complex computer programs.

Gene regulatory network: A complex system of interconnected genes and their regulatory elements. Regulatory elements, such as transcription factors, which are in turn proteins expressed by genes, control the expression of other genes, determining when and how much of a particular protein is produced. GRNs are essential for cellular processes and the development of organisms.

Hierarchical incoherence: A calculated parameter of a network that gives insight to its topology, where low hierarchical incoherence means that all its vertices have approximately the same hierarchical level and that the network is therefore influenced by a large percentage of its vertices. In contrast, low hierarchical incoherence means that there are distinct hierarchical levels and that the network is therefore influenced by a small percentage of its vertices.

Hopfield network: A type of recurrent ANN that can be used as an associative memory. It consists of a single layer of interconnected neurons, where each neuron is connected to every other neuron. These connections have weights that are adjusted during training. The network operates by iteratively updating the state of each neuron based on the weighted sum of its inputs. This process continues until the network reaches a stable state, which represents a stored memory pattern. Hopfield networks are known for their ability to recall patterns even when presented with noisy or incomplete input. They have applications in various fields, including pattern recognition, optimization, and content-addressable memory.

Intelligence potential: For a Regulatory Network Machine, the intelligence potential (IP) estimates the capability of the network to output diverse behaviors in response to stimuli, maintaining memory of state, and emitting context-dependent behaviors. The IP of a network is the sum of the Multistate Ratio (MR) and the Context Ratio (CR).

Metastable: A state of a system that is not in its lowest energy state but is stable enough to persist for a relatively long time. It is like a ball balanced on a hilltop: it is not the lowest point (the valley), but it can stay there unless disturbed. A small disturbance can cause the system to transition to a more stable state. For example, a diamond is metastable at room temperature and pressure: it is not the most stable form of carbon, but it can exist for a very long time without changing.

Multistate ratio: Used in the estimate of the Intelligence Potential of a Regulatory Network Machine, the multistate ratio (MR) is the total number of output states, scaled by the number of input states.

Osmoadaptation: The ability of an organism to survive and acclimatize to changes in its environmental osmolarity. Osmoadaptation involves physiological sensors (typically surface receptors) that alert the system to changes in osmolarity and transduce the signal to activate appropriate

cellular responses. Activated cellular responses ultimately activate appropriate actuators/effectors that achieve the goal state of appropriate directed water flow across the cell, thereby maintaining cell volume homeostasis against the osmotic challenge.

Persuasion: As we use the term here, persuasion involves using communication techniques to transition a system such that it adopts a particular state or takes a specific action.

Setpoint control: A control strategy used to maintain a system at a desired value. It involves measuring the current state of the system, comparing it to the desired setpoint, and then adjusting the system's inputs to minimize the difference between the two.

State space: Commonly used in dynamical systems and control theory, *state space*, also known as *phase space*, is a conceptual metaphor to physical space where all possible states of the system are represented as points in the state space. The evolution of the system in time progresses along trajectories in the state space, which helps to visualize and characterize the full characteristics of the system. A continuous, spatial representation for the states of a system facilitate the use of conceptual tools related to navigation and problem-solving by agents with various degrees of context-sensitive control.

Turing machine: A formalism for computation that clearly separates the concept of a machine (a device with discrete states and a known, deterministic transition between those states) and data (an information stream that both guides the machine's behavior and is itself modified by the machine). The Turing machine metaphor introduced the general concept of programmability (and the distinction between software and hardware) because its function can be radically altered by the information content of the input data without needing to change the design or components of the machine.

equilibrium state of the GRN system can be reached given a pattern of input stimuli applied to the system when it starts from a specific initial equilibrium state. Importantly, we have found that many systems do not have a 1:1 correspondence between an equilibrium state and the application of an input signal, and that, in contrast, the NFSM highlights previously undetectable path-dependencies, path-irreversibilities, modes of system behavior (e.g., stable versus oscillatory), cycles, and unreachable equilibrium states occurring under patterns of applied inputs, which all underpin “intelligent” behavior exhibited by a GRN-regulated cell system.

Taken together, our complete computational framework is called the Regulatory Network Machine (RNM) and is designed for optimal persuasive control of GRN. The RNM consists of: (1) a dissipative dynamic system, which may be a real-world system queried experimentally or a system studied using computational modeling (here, we focus on computationally modeled GRNs); (2) a set of input states that can be applied to the dissipative dynamic system, which have the capacity to alter the behavior of the system; (3) system equilibrium states with identifiable relevance to biomedical or biotechnological objectives; and (4) NFSMs comprising the input signal-induced transitions between equilibrium states of the dissipative dynamic system (here, the equilibrium states of the GRN).

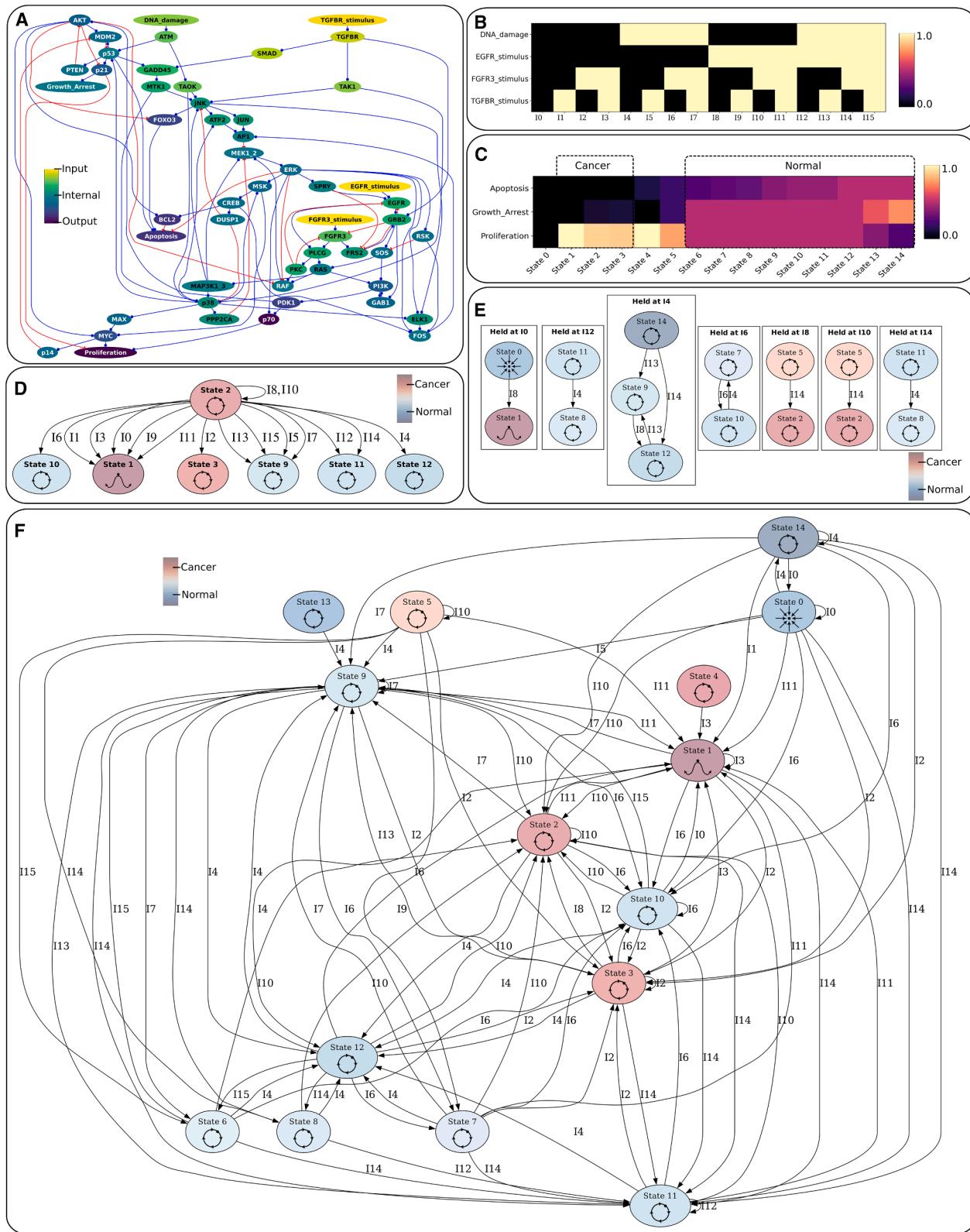
Ultimately, while it is well-known that regulatory networks can perform functions such as state-switching, memory of state, counting, and timing,^{39,41,42} it remains unclear how these networks can perform the context-dependent, multi-step, information processing and event-driven process control seen in real living organisms. Moreover, a formalized understanding of what it means to “persuade” a regulatory network, and how to optimally and systematically achieve such persuasion, does not yet exist. Here, we use our RNM framework to specifically illustrate various biological cases, including what it means for a biological system to perform analog “computations” to arrive at survival-appropriate responses in a stochastically changing environment; how, given a regulatory network, an RNM model can be constructed and used to predict prompts that elicit desired behaviors from the system and can be used to inform disease treatment; and how some

RNMs elucidate characteristics of smart, persuadable systems and can pinpoint the origins and implications of context-dependent behaviors. Overall, our work is an extremely tractable minimal system offering a route to both move beyond classical paradigms of biomedicine (e.g., “one constant dose” and “single factor, single treatment” doctrines) and to understand how larger kinds of proto-cognitive systems may emerge from smaller components.

RESULTS

While our RNM framework will work with any dissipative dynamic system, here, we focus on GRN due to the extensive biomedical and biotechnological applications. The GRN can be modeled using any method that returns the equilibrium state(s) of the system that occur in response to a stimulus/perturbation. We use two complementary methods to computationally model GRNs: (1) as continuous systems described by coupled nonlinear differential equations, which offers a high level of accuracy, yet requires more intensive computation¹⁷; and (2) as simplified, yet highly computationally efficient, Boolean networks,^{61,62} where a Boolean model of a regulatory network simplifies the values that a single node can occupy to be equal to 0 or 1, and the regulatory activities of nodes upon other nodes are handled using logical functions.⁶² Our GRN models are described in detail in the **STAR Methods** section. We directly compare the results of the Continuous and Boolean models when each are used with the same underlying GRN in our RNM framework and find excellent comparability between both modeling techniques (see **Figure S1**; **Table S1**).

The GRN utilized in our RNM may be any network comprising nodes that assume different factor levels, where nodes interact with each other via directed edges representing activation (represented by a blue edge ending in a dot) and inhibition (represented by a red edge ending with a bar) relationships.¹⁷ Note that our RNM framework can utilize any GRN modeling method that returns the equilibrium state of the GRN in response to an applied input/stimulus/perturbation and is not limited to the continuous partial differential or Boolean models we have utilized herein.



(legend on next page)

Dissipative dynamic systems as analog computing agents

In essence, a computational device is one that maps a system from an *input state* to an *output state*, where the internal operations involved in going from the input to the output are described as the *computational process*,⁶³ see **Box 1** and **Figure 3A** for additional details. While there is no formal definition of what constitutes analog computation (**Box 1**), the view we adopt here is of the *dissipative dynamic system* as an analog computer, where computations are understood in terms of what happens in the *state space* of the system in response to inputs as perturbations,⁶³ as shown in **Figure 3A**. In our RNM framework, the dissipative dynamic system is of a chemical nature, comprising dynamically interacting biomolecular, protein, and genetic components represented by a regulatory network (i.e., by a GRN).

Dynamic systems, including GRNs, can be described using *differential equations* estimating how the variables defining the system change in time and their relationships to one another.^{17,64,65} By imagining that the values of each system variable can be represented along an axis, each variable can thereby form a dimension of a metaphorical space, where each point in the space is one possible *state* of the system (**Figure 3A**). This metaphorical space represents an important concept called the *state space* (**Box 1**). At each point in state space, the set of differential equations describing the system specifies a rate of change for each system variable, which can be depicted as a vector showing the direction and magnitude of change at a point in state space. A vector field called the *direction field* (**Box 1**) can thereby be created by computing instantaneous rate of change vectors at each point in the state space (**Figure 3A**). The system's evolution in time from one point of the state space to another is then naturally prescribed by the direction field (**Figure 3A**).

The direction field can be analogized to the currents of a river, where the present state of the system is akin to a leaf afloat on the river's surface and the leaf's trajectory in time depends on the characteristics of the currents that carry it along. The form of the direction field (river's currents) determines how the network's state (leaf floating on one place in the river) will be drawn toward different regions of the state space (river as a whole), providing an indication of how the system state will evolve in time, if it becomes stalled or trapped in certain areas and, if so, how stable this entrapment is. The entrapment of the system in a certain state is formally called an *equilibrium state* (**Box 1**).

Importantly, the direction field leading to the equilibrium can assume different forms, leading to different stability and other characteristics of the equilibrium.

The stability characteristic of an equilibrium describes how the network responds to a *perturbation* that takes it away from the equilibrium. For example, an asymptotically stable equilibrium returns to the equilibrium point—as if drawn back by a force field—even when a moderately strong perturbation occurs (**Figure S2**). In contrast, a meta-stable equilibrium such as a saddle point moves away from the equilibrium when exposed to even a small disturbance (**Figure S2**). Several types of equilibrium dynamics and stability characteristics are commonly observed in the state space of dissipative dynamic systems, including: asymptotically stable point attractors; asymptotically stable limit cycle attractors, which oscillate several times with decreasing amplitude before reaching a stable equilibrium; cyclic limit cycle attractors with sustained oscillations, and meta-stable saddle-point attractors^{65–67} (see **Figure S2** for additional details and examples). Details regarding our algorithms for detecting and characterizing equilibria in Continuous and Boolean GRN models can be found in **STAR Methods**.

Analogous concepts to the above-described state space and direction fields also exist for the simplified Boolean network models, where individual nodes of the regulatory network are restricted to values of 0 or 1, and regulatory interactions between nodes are modeled using logic functions^{68,69} (see **STAR Methods** and **Figure S2**). A Boolean network with N nodes has 2^N total possible states in its state space. Boolean network dynamics are commonly analyzed using state transition graphs (STGs), which are also referred to as the “attractor landscape”.^{68,70–72} The STG or attractor landscape shows how each possible system state spontaneously moves toward the equilibrium states of the system (**Figure S2**). In this, the STG of a GRN modeled as a Boolean network is a simplification of the full direction field obtained from the continuous, partial differentiation model of the GRN (**Figure S2**). Note that for both the STG of Boolean networks and the direction fields of continuous models, the attractor landscape of the system is generated without applying a specific input state to the system, and transitions between states happen spontaneously as non-equilibrium states naturally transition to equilibrium states with the progression of time (**Figure S2**).

Having now reviewed fundamental concepts of dissipative dynamic systems theory, the basic concept of a dissipative

Figure 2. Complete RNM for the MAPK Cancer Cell Fate Network of Grieco et al.⁵⁷

(A–F) The MAPK Cancer Cell Fate GRN (A) has been decomposed into: (1) an input node layer (“DNA damage”, “EGFR stimulus”, “FGFR3 stimulus”, “TGFBR stimulus”, see (B); (2) a network core (all non-input and non-output nodes in A, not shown); and (3) an output node layer (“Apoptosis”, “Growth Arrest”, and “Proliferation” nodes in (C)), representing the output responses of the network that form 15 unique equilibrium states (C). Each input node variable represents an information bit, which taken together, form 16 input “words” where the colorbar indicates expression levels (B). Each equilibrium output state is associated with an equilibrium type (e.g., point attractor, limit cycle, and to a unique expression pattern of all output nodes ((C), where colorbar indicates expression levels). Nodes in (A) are color coded to their hierarchical level.⁵⁸ We further associate output equilibrium states with no apoptosis, low cell-cycle arrest, and high proliferation as indicative of the cancer cell state (i.e., “Apoptosis” = 0, “Growth Arrest” = 0, and “Proliferation” = 1 represents the ideal cancer state with closest state matches indicated in (B)), whereas limit cycle cell states that cycle through proliferation, cell-cycle arrest, and apoptosis expression regimes are indicative of the healthy cell state (B). The pathway analysis of the G-NFSM showing sustained input interventions leading from a cancer-like State 2 to healthy States 9, 10, 11, and 12 is shown in (D), where node color indicates the Euclidean distance of the equilibrium output state from the idealized cancer state. The complete set of E-NFSMs for the system are shown in (E), indicating temporary stimuli can lead to permanent transition to a cancer state (sub-graphs ‘Held at I8’ and ‘Held at I10’ in E), but that there are no temporary stimuli that can treat cancer. The full G-NFSM for the system is shown in (F), revealing all held interventions that allow for access and movement between states.

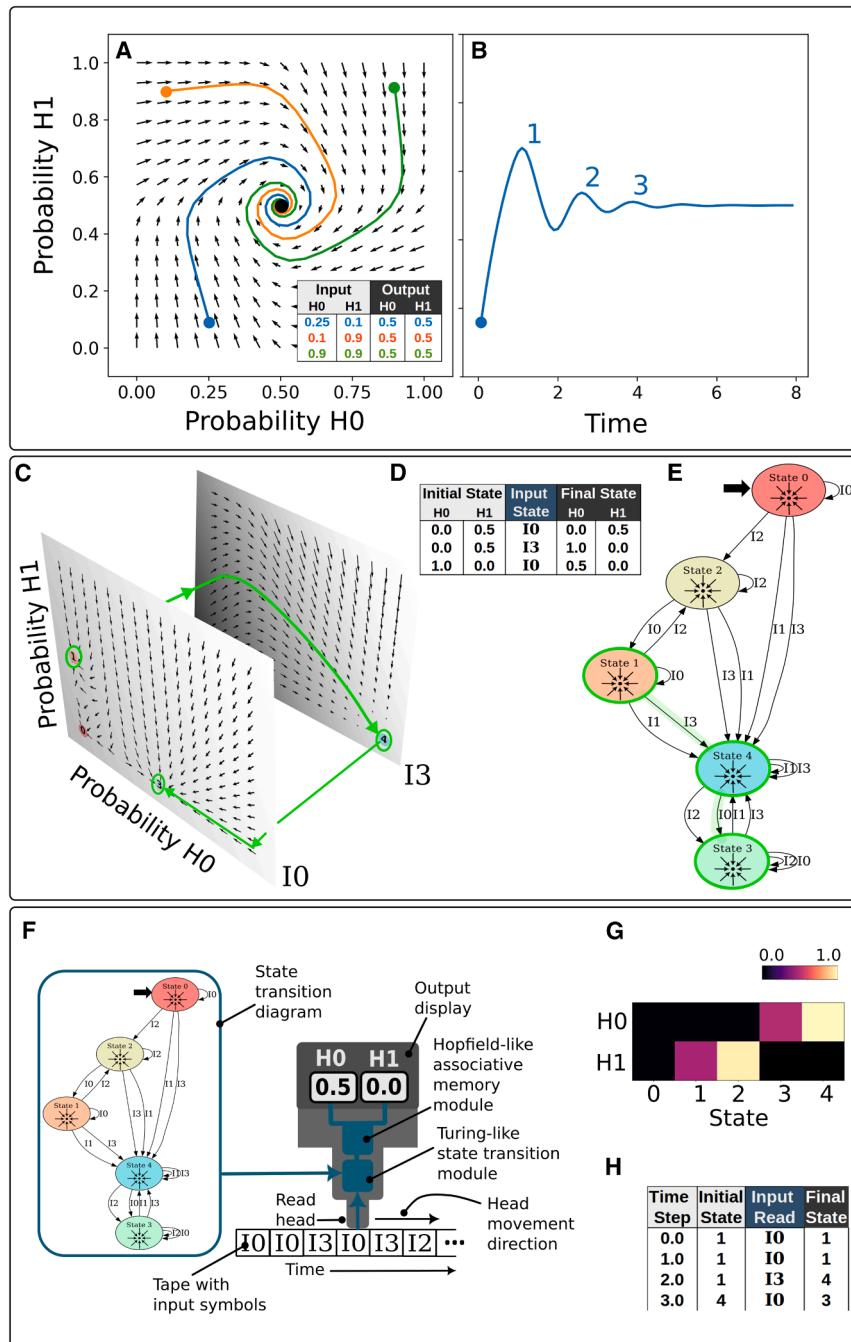


Figure 3. Analog computing uses dynamic systems to perform individual computations (A and B) as well as coordinated computational programs (C and D) that can be analogized to a hybrid Turing-Hopfield digital computing machine (F, G, and H)

Here, dynamic systems with two time-dependent variables, ' H_0 ' and ' H_1 ', are considered. An analog computation is defined as a trajectory in state space, where the *input* is represented by initial conditions (orange, green, and blue circles in (A)), the *computational process* is the direction-field-guided time-evolution of the system (orange, green, and blue trajectories in (A)) and the *output* is the resulting system state (black dot in (A)). While the system in (A) functions as a homeostat, the time-evolution of the same system can be used as a counter that activates a process several times before ceasing (B). Our RNM model goes further to create a computational program using externally set inputs to a dynamic system (' I_0 ' and ' I_3 ' in C) to select subsets of the system's full state space (planes in C). This has the effect of driving analog computations between equilibrium points of two state-space subsets with each change in the applied input state, creating a computational trajectory that spans the two state subspaces (green line in (C)). This allows the dynamic system to occupy multiple different equilibrium points, and to transition between these different equilibria under the influence of an applied input state, which can be organized into a state transition diagram we call the G-NFSM (E). Our RNM model can be analogized to a hybrid Turing-Hopfield digital computing machine (F, G, and H), which, like a classical Turing machine, is fed a tape of input state symbols (F) that are read by a read-head moving to the right in time (F). Yet, like a Hopfield network, the system also has associative memory, such that each state represents a pattern of output variables, which are the computational output (F and G). A detailed comparison of the properties of Turing, Hopfield, and Regulatory Network Machines is summarized in Table 2.

dynamic system as an analog computing agent can be appreciated in its most trivial embodiment by visualizing a system's direction field (Figure 3A). The computational input to the system may correspond to the initial state of the system (orange, green, and blue dots in Figure 3A), where the analog "computation" is performed simply by allowing the system to spontaneously follow the direction field of the state space as a temporal evolution, with the system ultimately becoming entrained by an equilibrium. The set of system variable values associated with the

equilibrium represents the output of the analog computation, and the trajectory taken along the direction field represents the computational process. The orange, green, and blue trajectories in Figure 3A represent the analog computational process for a simple system and the black dot in Figure 3A represents the output state as the single equilibrium point of the system. The direction field is therefore instrumental in understanding how analog computing via dissipative dynamic systems works, and the full "computational strategy" of the dynamic system is

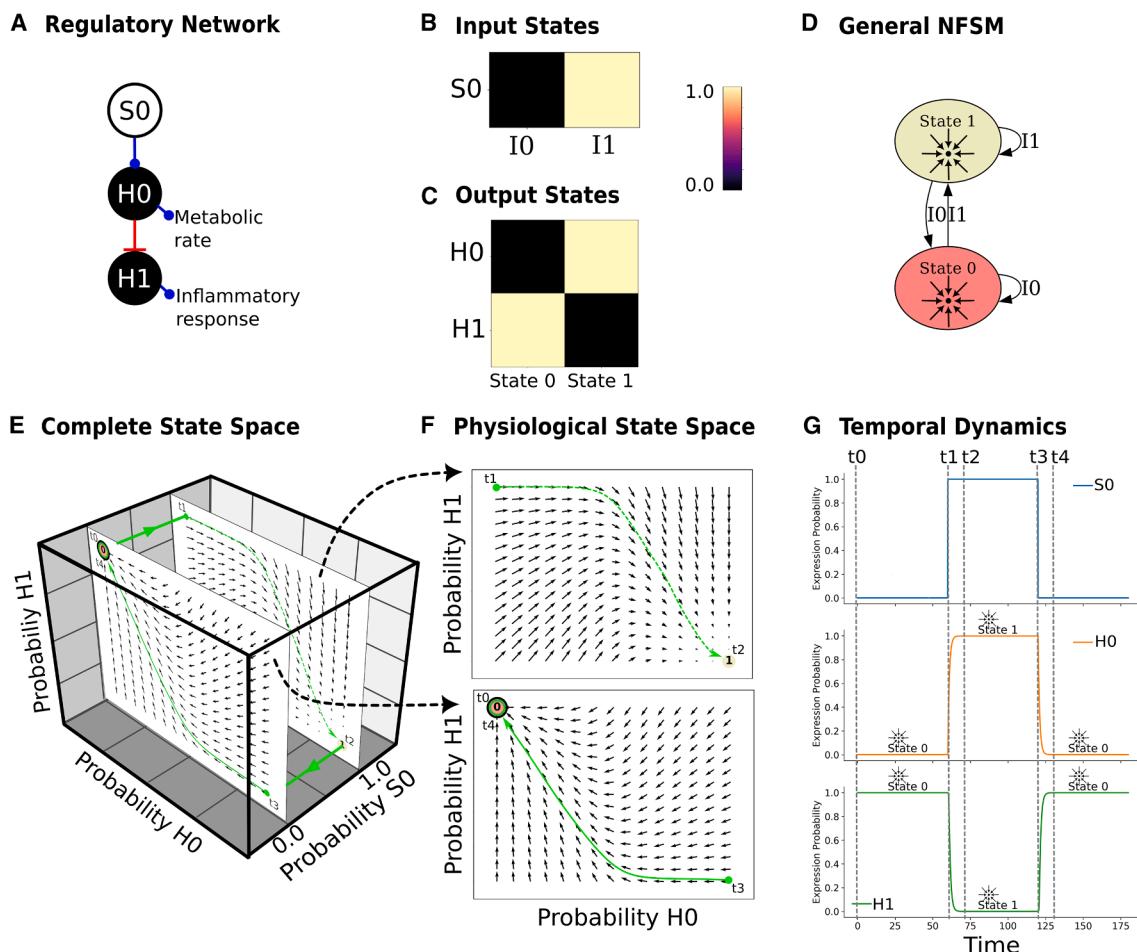


Figure 4. Here, we elucidate the core characteristics of *monostable* RNM computational programs using a simple 3-node regulatory network (A) with a visualizable complete state space (E)

(A–G) The regulatory network studied is shown in panel (A), with input states of the system shown in (B), and the two output states in (C). The G-NFSM for the system is shown in (D). As the full state space of this system can be visualized, we can see how the transitions of the G-NFSM arise in response to applied inputs (E and F). The complete state space of the dynamic system has dimensions made of the probability of H0 expression, the probability of H1 expression, and the probability of the input S0 (E). Setting S0 = 0.0 or S0 = 1.0 selects two regions of the complete state space (two state subspaces), which each have a slice of the full direction field (E and F). The output states, State 0 and State 1 can be seen as individual equilibrium points on each state subspace plane (labeled circles '0' and '1' in E and F). At time $t = 0.0$, the system starts in an initial condition equivalent to state 0 ('t0' in E, F, and G) with input S0 = 0.0 applied and held (equivalent to stating that input state I0 is applied and held); in this configuration the system is in a stable equilibrium and will not change in time (G). At time $t1 = 60$, the input state is changed to I1 (equivalent to changing S0 = 1.0), which injects the system into the second state subspace (see E), where it is no longer in equilibrium (see t1 of E and F). The system follows the direction field of the new state subspace plane to a new equilibrium (see t2 of E, F, and G). As there is only one equilibrium state in each state subspace plane, the system output state is reversible to the original condition when the input state is changed back to I0 (see t3 and t4 of E, F, and G).

embodied in the form of the direction field (Figure 3A). Note that as the STGs of Boolean networks are analogous to direction fields, the spontaneous transitions indicated by the STG also represent the input-output relations and computational strategy of the system (Figure S2).

While Figure 3A shows an example of an individual analog computation from an unstable to an equilibrium point, to work with regulatory networks as persuadable information-processing agents, a more sophisticated tool is required. Specifically, we require a framework that can specify how individual analog computations can come together into what we could refer to as an analog computational program. A crucial difference be-

tween standard attractor landscape analysis and our RNM framework is that our framework identifies and maps non-spontaneous, input signal-driven analog computations between two equilibrium states of the dissipative dynamic system, whereas attractor landscape analysis works with spontaneous transitions between *all* possible states of the system. In a Boolean model this means an attractor landscape model works with all 2^N states, which may be prohibitively large (for $N = 53$ GRN nodes this is 9×10^{15} states), whereas our RNM framework refines the focus to only equilibrium states, which tend to number less than 100, even in complex networks, and represent the time-stable output of the GRN system.

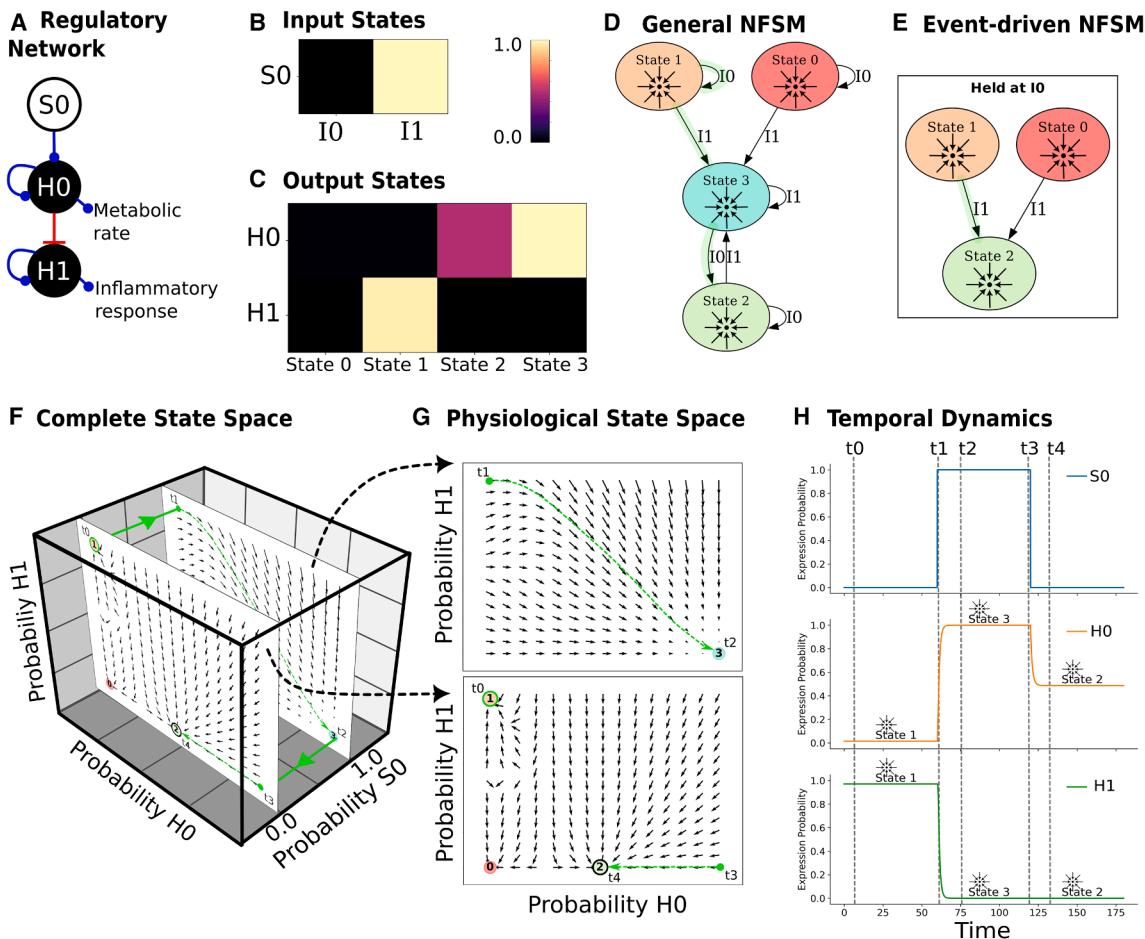


Figure 5. Here, we elucidate the core characteristics of *multistable RNM* using a simple 3-node regulatory network (A) with a visualizable complete state space (E)

(A–H) The regulatory network is shown in (A), the two input states in (B), and the four output states in (C). The G-NFSM is shown in (D). We found that in some cases this system responds to transient application of input states with permanent changes to output state and therefore also has an E-NFSM (E). As the full state space of this system can be visualized, we can see how the transitions of the G-NFSM and E-NFSM arise in response to a sequence of applied inputs (F and G). The complete state space of the dynamic system has dimensions made of the probability of H0 expression, the probability of H1 expression, and the probability of the input S0 (E). Setting S0 = 0.0 or S0 = 1.0 selects two regions of the complete state space (two state subspaces), where each plane contains a functional slice of the full direction field (F and G). The multiple output states are individual equilibrium points on each state subspace plane (labeled circles '0', '1', '2' and '3' in E and F). Note that plane selected by S0 = 0.0 (input state I0) contains three unique equilibrium points and is therefore a multistable state subspace. At time t = 0.0, the system starts in an initial condition equivalent to State 1 ('t0' in F, G, and H) with input S0 = 0.0 (I0) applied and held; in this configuration the system is in a stable equilibrium and will not change in time (H). At time t1 = 60, the input state is changed to I1 (equivalent to changing S0 = 1.0), which injects the system into the second state subspace (see F), where it is no longer in equilibrium (see t1 of F and G). The system follows the direction field of the new state subspace plane to a new equilibrium, which corresponds to State 3 (see t2 of F, G, and H). At t3 S0 = 1.0 (input I0) is reapplied and the system is re-injected into the original state subspace, however, the system output tracks to a new output State 2 due to its new location in the state subspace and the presence of multiple equilibria altering the direction fields (see t3 and t4 of F, G, and H).

Regulatory Network Machines: A tool to work with regulatory networks as persuadable agents

Here, we develop the basic concepts underlying the RNM concept, with two simple examples of RNMs shown in Figures 4 and 5. The RNM comprises multiple elements including: a regulatory network (Figures 4A and 5A), a set of input states representing the “vocabulary” that can be utilized to “persuade” the system into modes of activity (Figures 4B and 5B), a set of stable output states representing the set of inherent competencies (the behavioral repertoire) that may be activated

by the vocabulary through persuasion (Figures 4C and 5C), and the NFSM maps specifying how each element of the system’s input vocabulary can be used to persuade the system from one stable behavioral pattern to another (Figures 4D, 5D, and 5E). The NFSM of a regulatory network are maps showing if transitions between equilibrium states are possible and, if so, precisely how the system can be persuaded into a specific stable output state from an initial stable state using the available input state vocabulary. Note that the NFSM also predicts the stable behavioral dynamics that are emitted by the system when it is

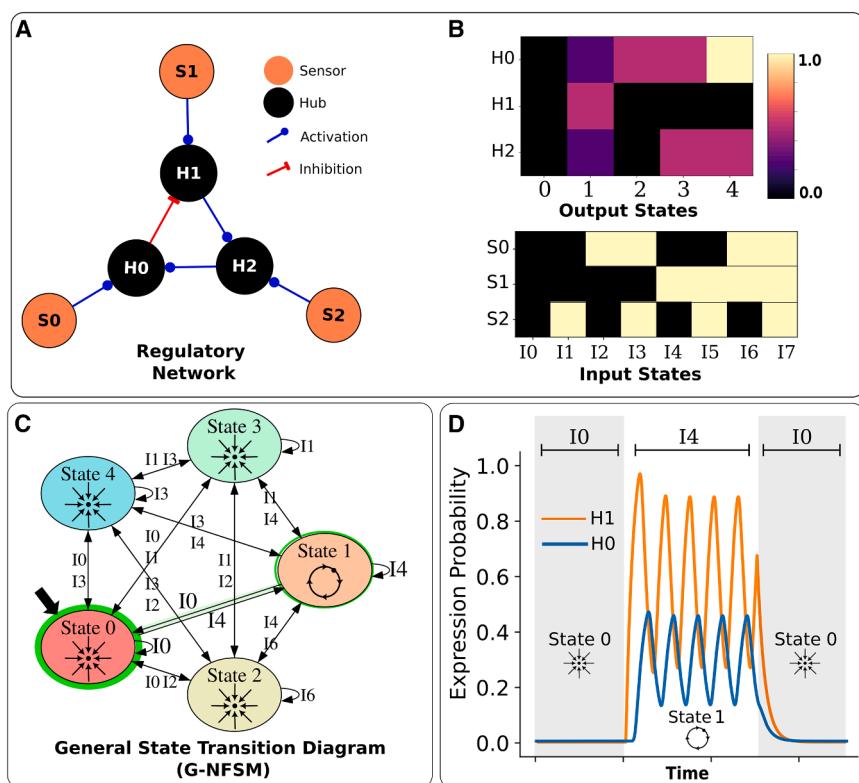


Figure 6. NFSM predict the emission of categorically different behaviors associated with certain input states, creating a map of both levels and modes of system behavior

(A–D) Here, a cyclic regulatory network (A) with five unique output states in association with eight input states (B) forms a monostable NFSM with a limit cycle attractor as output State 1 (where all other states have point attractors) (C). Examining its behavior over time, starting the system in State 0 and holding with input state I0, then changing to input state I4, leads to a transition from a monotonically stable pattern of gene expression to a genetic oscillator, while input state I4 is applied and held (D). The genetic oscillation ceases when the input state is reverted to I0 (D). The behavior of the system is predicted by the G-NFSM, which shows the point attractor of the start state, State 0 (indicated by a black arrow and bold green halo in (C) predicted to transition to the limit cycle of State 1 (indicated by thin green halo in (C) via a transition induced by I4, and to proceed back from State 1 to State 0 via a transition induced by input I0 (C). The transitions are shown highlighted in light green (C).

in a particular equilibrium state (i.e., which kind of equilibrium the direction field pattern generates), allowing for the persuasive activation of not only monotonic patterns of gene expression but also of dynamic behaviors such as genetic oscillators, which arise naturally from the nonlinear system dynamics (Figure 6).

A key aspect of our RNM framework is the assertion that the network is not an isolated entity with homogeneous node types; rather, we ultimately need to consider heterogeneous networks that are interfaced with an environment via sensors endowing the network with the ability to observe/experience/sample the environment, and where the network can change state to adopt a response to the environment (see comprehensive example in Figure 2) with other good examples from the literature.^{57,73,74} We thereby distinguish between *input*, *internal*, and *output* nodes of the system, which may all involve different types of agents acting in the network (Figure 2). This heterogeneity, wherein levels of some internal nodes of the network (e.g., gene expression levels) may be influenced by sensing agents such as *trans-membrane receptors* and, in turn, where the levels of internal nodes exert an influence on the environment via potentiating outputs, is a natural aspect of most biological systems.^{74–77} For example, most signaling cascades involve *trans-membrane receptors* that sense environmental variables (i.e., variables external to the network) such as morphogens, growth factors, and physical properties such as light, temperature, or pressure (the inputs), which are capable of influencing genetic expression (the state of internal network nodes) to elicit survival-enhancing responses (output nodes) in relation to environmental variable values.^{66,67,73,74,75} For maximum comparability with digital

computational systems and simplicity, regardless of whether our model is discrete or continuous, we fix the value of each input node to binary values of 0 or 1, representing a high or low presence of the respective factor; therefore, the input state space of our model is discrete rather than continuous. However, this is an artificial restriction and the value of input nodes can assume continuous levels of resolution, as needed.

An essential aspect of our RNM framework is that the application of an input signal selects a *state subspace* or alternatively an *attractor sub-landscape*, that contains a direction field and associated equilibrium states unique to the specific input signal. For the case of Boolean networks, the state subspace associated with a specific input is represented by an STG instead of a direction field. This concept of subspace selection by application of a specific input state can be visualized in the simple systems of Figures 4E and 5F. While the complete state space of our RNM framework is spanned by dimensions created by values of *all* nodes of the network, the input nodes become instrumental in selecting modules of possible system output (Figures 4 and 5).

This use of input states to select state sub-spaces allows for a natural representation of how the network functions as an information processing entity capable of emitting non-trivial behavior sequences in response to patterns of sensed experience and allows us to formalize two major functions (visualized in Figures 4 and 5). Primarily, the application of an input signal changes the selection of the state subspace and, therefore, the input changes the direction field and the possible equilibrium states present in the subspace. When the input signal is applied and held for a sufficient duration, the system will spontaneously track to one of the equilibria in the state subspace, and while there are no further changes to the nature of the input signal, the system will remain

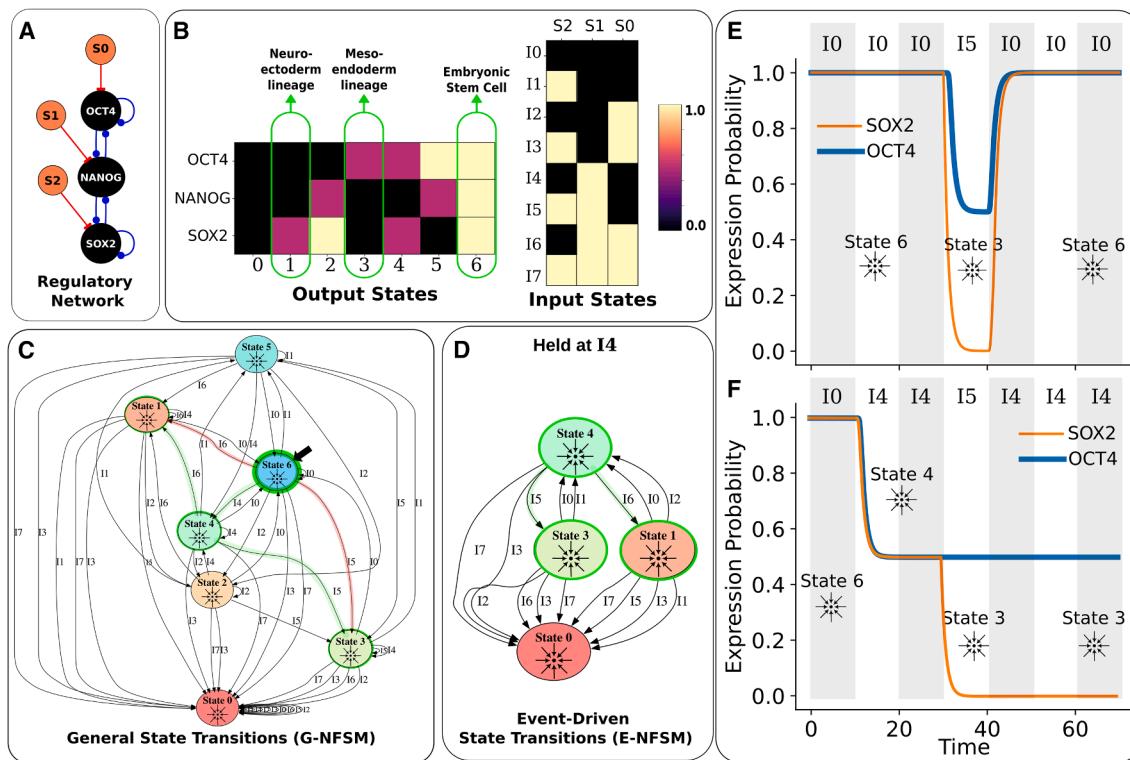


Figure 7. Stem cell differentiation in response to transient exposure to morphogens is a context-dependent, event-driven state transition in a multistable RNM system

(A–F) The ESC regulatory network modeled in this example is shown in (A). High levels of NANOG, OCT4, and SOX2 are known to correspond with undifferentiated ESC (corresponding to output State 6, (B), whereas differentiation toward a neuroectoderm lineage occurs when levels of NANOG and OCT4 drop, leaving SOX2 high (corresponding to output State 1, panel B), and differentiation toward a mesoendoderm lineage occurs when levels of NANOG and SOX2 drop, leaving OCT4 high (corresponding to output State 3, (B)). Our RNM analysis of the ESC network in (A) indicates it is a multistable system with seven unique output states (B), both a G-NFSM (C) and an E-NFSM (D), and a high average Intelligence Potential of 1.64 (see “Full Chain” network of Table 1). The biological observation of high NANOG, OCT4, and SOX2 in undifferentiated ESC indicates that the biological system starts in State 6 (marked with a black block arrow and bold green highlight in (C)). The G-NFSM suggests that both States 1 and 3 are accessible from State 6 via application of input I6 and I5, respectively (trajectories highlighted in pink in (C)). However, State 6 exists in monostable context I0; therefore, while transient application of I6 or I5 leads to transition of the system to State 1 or State 3, it is not a permanent change, and the system reverts to State 6 once input I6 or I5 are removed (E). The E-NFSM shows us that it is only in context I4 that a permanent change of the system from State 6 to States 1 or 3 can be achieved from a transient application of the same I6 or I5 inputs, with respective paths highlighted in green (C and D). In the time evolution of the system shown in F, we see that by first switching to State 4 with the application of I4, a transient exposure to I5 now leads to a permanent switch of the system to State 3 even with a return to the original held context established by I4 (F).

in that equilibrium. However, when there is a change in the input signal applied to the system, the selected state subspace changes, and the system is injected from its equilibrium location in the original state subspace to a corresponding location in the new state subspace (this can be visualized in the low-dimensional systems of Figures 4E and 5F). Yet, with this change in the input signal, the location of the system in the new state subspace may no longer be an equilibrium and the system will spontaneously track to a new equilibrium, depending on the nature of the direction field in the new state subspace (Figures 4E and 5F). This means that a change in the input state initiates an analog computational process as a trajectory that traverses both within and between two subspaces (see the green line in Figures 4E and 5F) and allows for the input state to act as the agent that induces a transition between two system states. This also allows the input signal to access different modes of analog computation within each subspace, as the computational strategy of the sys-

tem is embedded in the morphology of the subspace’s direction field (Figure 3A).

We also identify two distinct classes of RNM: (1) a *monostable* class where there is only one equilibrium state in each subspace (see Figure 4) and (2) a *multistable* class where there is at least one subspace that contains multiple equilibria (see Figure 5). As we shall demonstrate in examples, multistability endows RNM with the capacity to exhibit behavior that depends on experience (e.g., Figure 7) and to “recognize” context (e.g., Figure 8). If the system is monostable (Box 1 and Figure 4), meaning there is only one equilibrium state in each of the state subspaces, then there is necessarily a 1:1 correspondence between the input signal applied to the system and the equilibrium state occupied, as the direction field in the state subspace will always track to the single equilibrium state independent of the starting position of the system in the state space (Figure 4 and Document S1). However, if at least two of the system’s state subspaces are

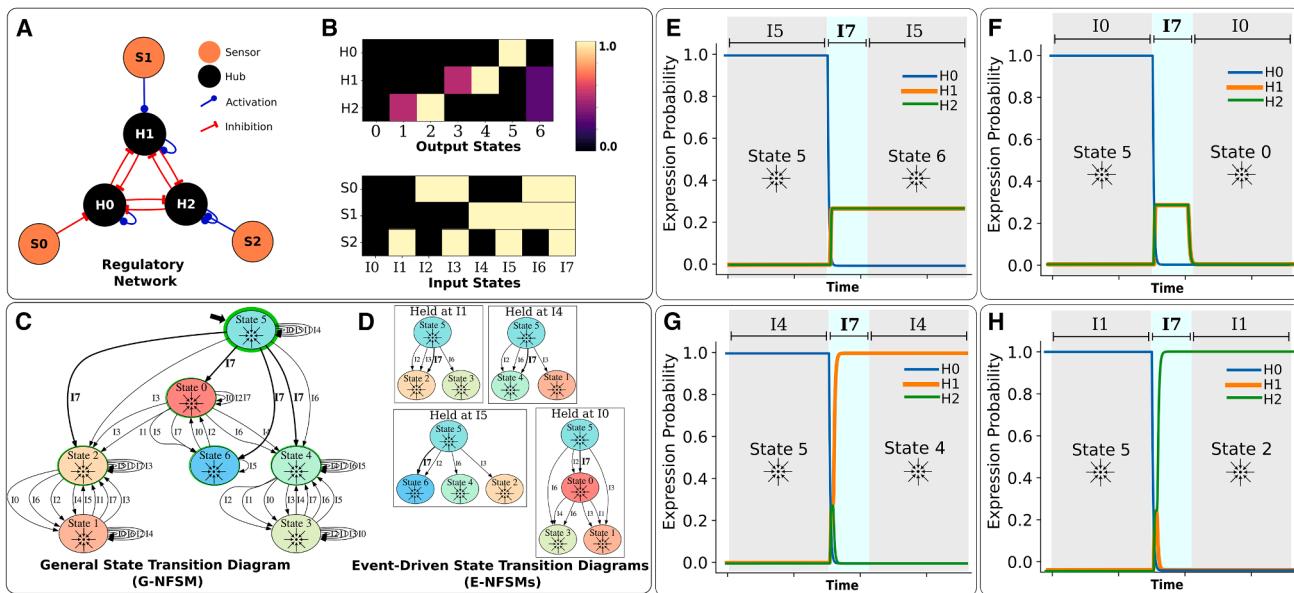


Figure 8. Multiple, apparently identical, systems can show dramatically different responses to the same intervention yet are deterministically described within the RNM framework

(A–H) The regulatory network modeled in this example is shown in (A), which was found to have 7 unique output states (B) and to have a G-NFSM (C) and multiple E-NFSM (D). The G-NFSM shows four transitions from State 5 to four unique states (State 0, State2, State 4, and State 6) happening with the application of the same input state I7 (bold arrows of C). The E-NFSMs show four different input contexts (for systems with a baseline-normal held input state of I0, I1, I4, and I5) where State 5 transitions to another state under input I7 (bold arrows of (D)). In time-course studies, we find that systems with a held input state of I5 (E), I0 (F), I4 (G) and I1 (H) are initially stable in State 5, as predicted by the G-NFSM and E-NFSMs. Subsequent transient application of input I7 to each of the four systems leads to four different final states (E, F, G, and H), as predicted by the E-NFSMs (D).

multistable (Box 1 and Figure 5), exhibiting more than one equilibrium state in the subspace, then upon change of input signal the location of the system in the new, multistable state space may track to one of several possible equilibria, depending on the location of the newly injected point in the new state space (Figure 5F). The nature of path-dependency and context-dependent behavior in the system's behavior is founded on these subspace multi-stabilities, as tracking to a specific equilibrium state in a multistable subspace can introduce irreversibilities, as it may no longer be possible to return to specific areas of other multistable state sub-spaces (see Figure 5). Therefore, multistable systems enable non-trivial transitions between different equilibria in response to different sequences of applied inputs and are the origin of more complex behavioral patterns (Figure 5).

We further distinguish between inputs that are transient perturbations to the system, which we refer to as *input events* (Box 1), and those input states that are applied and held for a duration in time, which we refer to as the system's *context-defining inputs* (Box 1). By system *context*, we specifically refer to the state sub-space that is selected when an input state is applied and held (e.g., the sub-space planes in Figures 4E and 5F each represent a unique context that is accessed by the applied input state). As each context may comprise radically different direction field patterns and equilibria, this means categorically different computational processes and output responses may become available for each context-defining input (see Figures 5F and 5G). Biologically, we take a context-defining input to represent the case where the applied input state remains

relatively stable in time. For example, a cell in a body experiences “normal” physiological conditions pertaining to a regulated body temperature, levels of various salts in the extracellular media, and so on. In contrast, we distinguish an *input event* to be a transient deviation from this baseline context of “normal conditions”, for example, the temporary incidence of a very hot environment can generate hyperthermia and dehydration, altering various parameters from what can be considered the physiological baseline, or a morphogen may be present for only a relatively short period of time during development. In our RNM framework, when an input state is transiently applied, yet is then returned to the original input state, the transient input is taken to act as a *perturbing event*, which may possibly change the state of the system by removing the system from one equilibrium in a multi-stable sub-space, temporarily transitioning the system to an alternative equilibrium in a new sub-space, yet upon returning the original multistable subspace the original equilibrium cannot be accessed and therefore a new equilibrium is occupied, leading to an irreversibility (Figure 5).

The progression of the regulatory network system in time is governed by a set of nonlinear partial differential equations (see STAR Methods), where the full output of the system is the value of all internal nodes when the system nears an equilibrium. Note that since only certain nodes interact directly with an external process, for simplicity and conciseness, we categorize nodes with zero out-degree as the set of output nodes (Figure 2). This convention is commonly followed in the literature in studies using regulatory networks to work with experimental biological

systems.^{57,74} We emphasize that as the network may comprise many output nodes, each unique, stable network equilibrium state represents a categorically different pattern of activity (e.g., gene expression) and, therefore, a categorically different response and action taken by the network (Figure 2). The set of outputs may represent a variety of outcomes such as the categorically different gene expressions of differentiated cell types in an organism (e.g., the different gene expressions of fibroblasts, stem cells, and neurons), different metabolic modes (e.g., ketosis versus glycolysis), or other modes such as the physiological stress state versus the resting state response.^{7,9,78,79}

Working from the basis of our RNM framework, we have developed another tool we call the NFSM that maps the emitted behaviors of the GRN system in response to applied inputs and is therefore a map of the sequential logic program of the GRN. We distinguish between two different types of NFSM: (1) the general NFSM (G-NFSM) that maps all possible transitions from each equilibrium state and each applied input signal to induce a transition to a new equilibrium state and (2) the event-driven NFSM (E-NFSM), which shows *irreversible* state transitions occurring for *transiently applied* input signals in a context-defining input (Figure 5E). Please see the **STAR Methods** for the operational steps to create the G-NFSM and E-NFSM from any GRN (or, more generally, any dissipative dynamic system). Once the NFSMs associated with the GRN have been generated, pathway analyses can be performed on the NFSM to identify “persuasive” routes to accessing desired output states from an initial state (e.g., to move from a “disease” equilibrium state to one of “health”). Cycles analysis on the NFSMs can identify multi-stage processes in the system and how to engage them (e.g., a wound-healing multi-step process). Please see **STAR Methods** for steps to utilize the G-NFSM and E-NFSM in biomedical and biotechnological applications.

Having outlined our RNM framework for working with regulatory networks as persuadable information processing entities, we next examine several cases, including those of real biological networks, to demonstrate the utility of the analog computing and RNM frameworks.

Biological systems use analog methods to “compute” correct responses to environmental challenges

We first explored a case-study of yeast osmoadaptation,^{7,80–82} where we show how a biological system successfully uses analog computing to solve an environmental challenge. We also looked at identifying fundamental differences between how biological systems and human engineers solve a challenge, such as maintaining an important variable at a target value (a set-point control problem), as recognizing these differences is the first step in improving our ability to work more effectively with biological systems through better understanding of their fundamental operations.

Human engineers use devices such as proportional-integral-derivative (PID) controllers that employ digital computation to perform a function such as set-point control.^{83,84} A PID controller requires a physical memory to store information, a physical central processing unit (CPU) to perform digital computations, a software program that provides specific instructions on what to read/write to memory and which digital calculations are required to produce an output, connections to take input from

a physical sensor that measures the process variable, and connections to an effector device that can receive computed output from the PID controller to influence some external process that changes the level of the process variable.⁸⁴ The PID controller uses these elements to solve a *set-point control* problem by: (1) using the sensor to measure the present state of the process variable, (2) using *digital computing* to calculate the error signal as the difference between the measured value of the process variable and the desired set-point value that is stored in physical memory of the controller, (3) using its physical memory to store a history of the error signal, and (4) using the error signal and its value stored at past times to compute an output signal in terms of the PID algorithm using digital computation and an algorithm specified by software.⁸⁴ The PID’s algorithm computes output as the sum of a component proportional to the error signal, and numerically estimated integrals and derivatives involving the error signal. Finally, the PID controller sends the digitally computed output signal to the effector to attempt to alter the value of the process variable in a way that maintains it at the set point value.

To illustrate the fundamental differences between human engineered PID controllers and biological systems, we considered yeast cell osmoadaptation as our case study – a classic, well-studied example of biological set-point control.^{80–82,85,86} Details of our osmoadaptation model can be found in **Document S1**. Osmoadaptive set-point control provides an excellent example of an analog computational system naturally “computing” the correct response required to maintain cell volume (vol_{cell}) at a target value for a wide range of environmental conditions (Figure 9). Here, the same physical process of osmotic-pressure-induced cell volume change is further compared in an inanimate vesicle (Figure 9A) and in an osmoadapted yeast cell (Figure 9B). The dimensions of the systems’ complete state spaces are the internal cell osmolytes (n_i), cell volume (vol_{cell}), and environmental osmolyte concentrations (m_o) (Figures 9C–9H). In the yeast cell, the osmotic process interacts with a regulatory network involving a sensor for cell volume via membrane strain (PhoQ), which conveys information regarding cell volume through to the HOG-MAPK signaling pathway, thereby altering levels of intracellular glycerol, which contributes to and, therefore, allows the cell to change, total internal cell osmolytes (Figure 9B).

We see that due to the intrinsic dynamics of the system, for each initial system state (an example initial state is given by the green dot in Figures 9F and 9H), the system dynamics naturally “compute” a new state (large green dot in Figures 9F and 9H) by simply evolving in time under the system dynamics (green trajectory of Figures 9F and 9H).

The direction field of the nonliving vesicle maintains a surface of equilibrium states (shown as the dotted black line in Figures 9E and 9F), meaning there is a range of possible equilibrium vesicle volumes for different environmental osmolyte concentrations and starting system states; therefore, there is no way the vesicle can maintain a single target volume in the face of different environmental osmolarities. In contrast, the yeast system’s direction field is in the form of an attractive limit cycle with a single equilibrium state appearing as a central point corresponding to the system’s desired target volume (Figures 9G and 9H, where equilibrium/target volume are indicated by the large green dot). Moreover, visualizing the yeast system’s complete state space in 3D, we

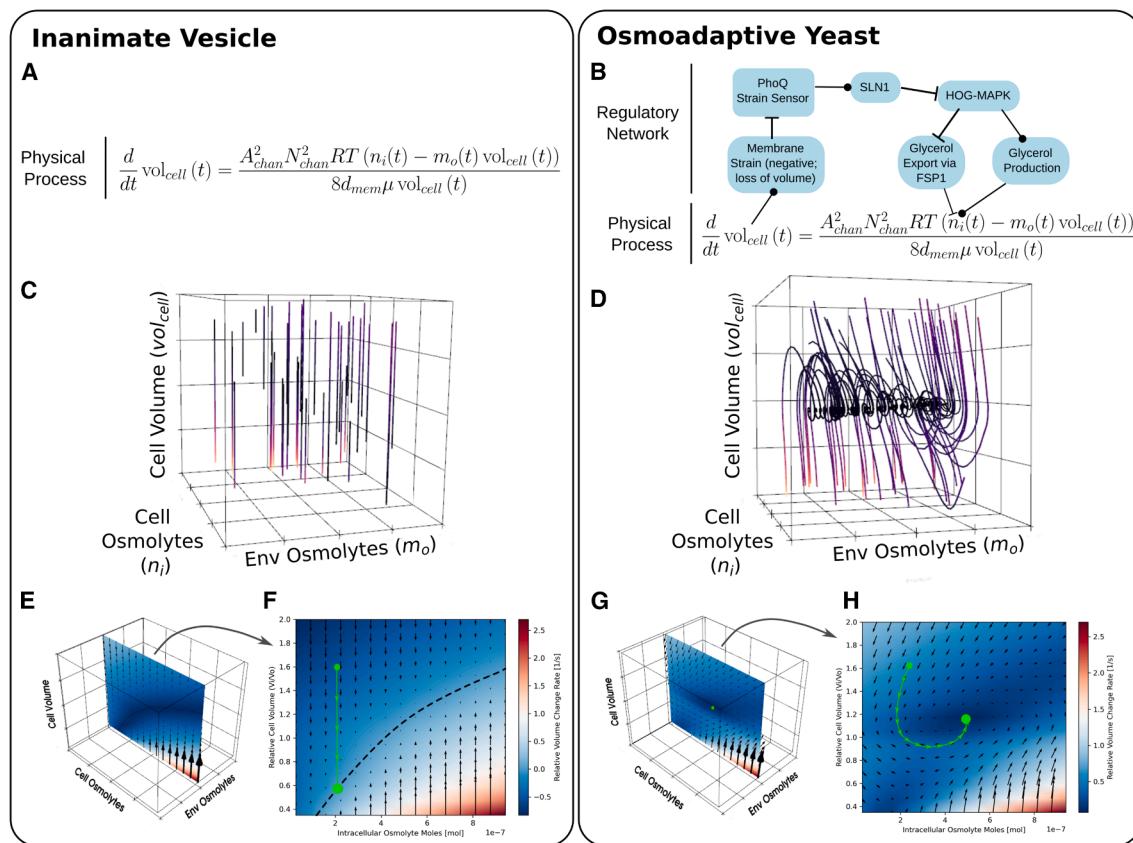


Figure 9. Osmoadaptive set-point control is an analog computational system naturally “computing” the correct response required to maintain cell volume (vol_{cell}) at a target value for a wide range of environmental conditions

(A–H) Here, the same physical process of osmotic-pressure-induced cell volume change is compared in an inanimate vesicle (A) and in an osmoadapted yeast cell (B). The dimensions of the systems’ full state spaces are the internal cell osmolytes (n_i), cell volume (vol_{cell}), and environmental osmolyte concentrations (m_o) (C–H). In the yeast cell, the osmotic process interacts with a regulatory network involving a sensor for cell volume via membrane strain (PhoQ), which conveys information regarding cell volume through to the HOG-MAPK signaling pathway, thereby altering levels of intracellular glycerol, which contributes to, and therefore allows the cell to change, total internal cell osmolytes (B). For each initial system state (an example initial state is given by the small green dot in (F) and (H)) the system dynamics naturally “compute” a new state (large green dot in (F) and (H)) by simply evolving in time under the system dynamics (green trajectory of (F) and (H)). The direction field of the inanimate vesicle maintains a surface of equilibrium states (shown as the dotted black line in (E) and (F), meaning there is a range of possible equilibrium state vesicle volumes for different environmental osmolyte concentrations and starting system states. In contrast, the yeast system’s direction field is in the form of an attractive limit cycle with a single equilibrium state appearing as a central point corresponding to the system’s desired target volume ((G) and (H)), where equilibrium state/target volume are indicated by the large green dot). In 3D, the direction field for the yeast system is cylindrically-shaped, meaning the attractive limit cycle will converge to the consistent target cell volume for a wide range of environmental and internal osmolyte concentrations, thereby demonstrating that the system has excellent control of cell volume (B).

see that the direction field for the yeast system is cylindrically shaped, meaning the attractive limit cycle will converge to the consistent target cell volume for a wide range of environmental and internal osmolyte concentrations, thereby demonstrating that the system has excellent control of cell volume (Figure 9B).

In striking contrast to how a digital PID controller works, we see from the above example that by using analog computing, the biological system *has no need for a physical memory* to store the set-point value; rather, the set-point value is an inherent aspect of the system dynamics, and can be considered to be “stored” abstractly in the system’s direction field as the equilibrium point of a limit cycle attractor (i.e., the large green circle terminating the green trajectory in Figures 9G and 9H). The biological system also has no need for a CPU to

perform computations of an error signal or output variable, nor for physical memory to store a history of the error signal, nor for any software to specify a computational algorithm. Instead, as can be recognized by visualizing the direction fields of the osmoadapted yeast model (Figures 9D–9H), that the system dynamics naturally utilize the present state of the biological system, in combination with the sensed process variable, to spontaneously evolve in a manner that naturally embodies an appropriate response to an environmental challenge such as high extracellular osmolarity.

We note that in contrast to a digital computation, the above example also illustrates how analog computations embody sophisticated modules of holistic functional competency. However, while the analog computing responses of the biological

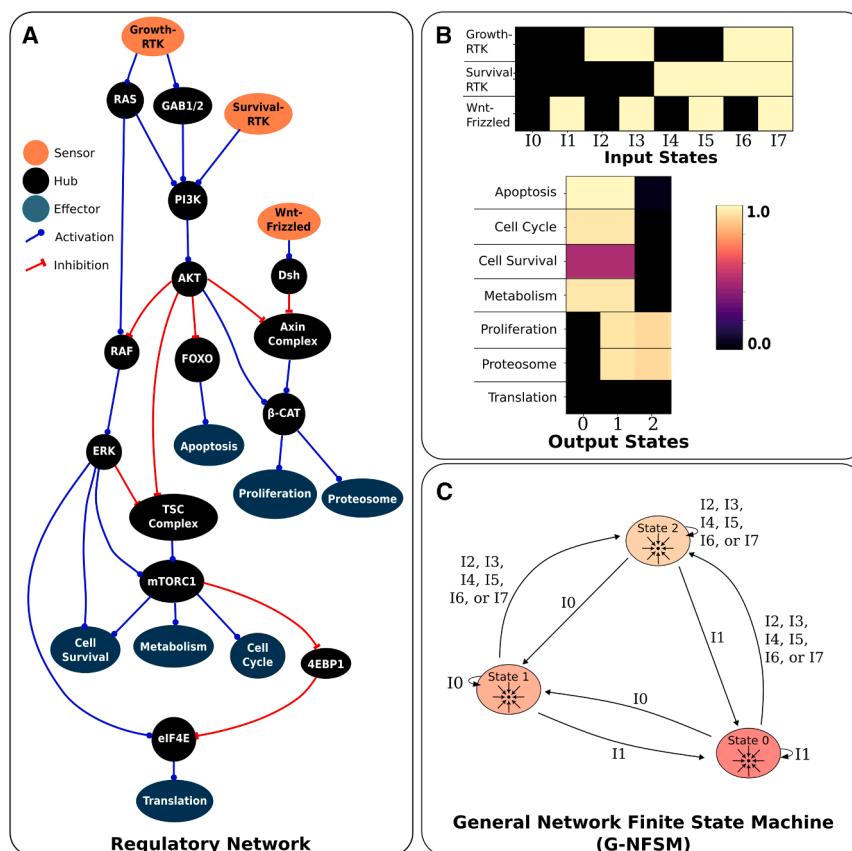


Figure 10. The G-NFSM is useful as a detailed set of instructions specifying which inputs engage the specific output states of signaling networks important in disease processes, such as the cross-regulation network between PAM, RAS/ERK, and Wnt/β-catenin (adapted from Figure 7 of Glaviano et al.⁷⁴)

(A–C) The regulatory network in A specifies activation/inhibition interactions between biologically relevant input, hub, and effector nodes, making it a suitable candidate for our RNM framework. A set of input states were defined from the input nodes (B), and three unique output states were returned from our analysis, which are shown with respect to effector node levels in C. Our analysis also determined that the dynamic system for this regulatory network is monostable, with a 1:1 reversible mapping between each input and output state, leading to a fully connected G-NFSM (C), no E-NFSM on account of its monostability, and a low Intelligence Potential of 0.8 (see ‘AKT Net’ in Table 1). The G-NFSM specifies the exact input states (as labels on transition arrows) that will result in the desired output state (C), which are intrinsic to the system and can be accessed without having to alter the network topology.

system are remarkable due to their ability to function without requiring a physical memory storage strategy nor a CPU, a key difficulty in working with analog computing is the inherent difficulty to change the function of the system (i.e., to reprogram existing computational capabilities). To re-program the characteristics of a specific analog computation (e.g., to alter the set-point value of the cell’s target volume) requires editing the system’s direction field, which in turn necessitates a change to the biological hardware. This is entirely unlike the ease of inputting a new set-point value into a PID’s software module, as it is unclear how to perform the required editing to achieve the desired results.

We therefore note that, as this model of osmoadaptation has only one stable state corresponding to the set-point target value, at the level of individual functions and behaviors (i.e., at the level of individual analog computations), this system is technically neither programmable nor persuadable to alternative behavioral outcomes. However, we next consider systems with dynamics that allow for multiple equilibrium states associated with different inputs to see how input stimuli can be used to select analog computing modules from a meta-network of possibilities (e.g., the NFSM), which we highlight as the act of using persuasion on regulatory networks taken from real-world biological systems.

The G-NFSM is a map guiding persuasion of regulatory networks

We next demonstrate how the G-NFSM is highly useful as a detailed set of instructions capable of specifying which inputs

engage the specific output states of signaling networks important in disease processes. We chose the cross-regulation regulatory network between PI3K/AKT/mTor, RAS/ERK, and Wnt/β-catenin as an example.⁷⁴ The regulatory network shown in Figure 10A, adapted from Figure 7 of ref.⁷⁴, specifies activation/inhibition interactions between biologically relevant input, hub, and effector nodes, making it a suitable candidate for our RNM framework. A set of input states were defined from the input nodes of the network (Figure 10B), and three unique output states were returned from our equilibrium search analysis, which are shown in Figure 10C.

Our analysis also determined that the dynamic system for the PI3K/AKT/mTor, RAS/ERK, and Wnt/β-catenin network is monostable, which means there is a 1:1, fully reversible mapping between each input and output state, as monostability means there is only one output state in each state sub-space (see Figure 4), leading to a fully connected G-NFSM (Figure 10C), no E-NFSM on account of its monostability, and a low Intelligence Potential of 1.0 (see ‘AKT Net’ in Table 1). Our analysis indicates that despite the appearance of apparent complexity in the signaling network, any regulatory network that does not have nodes engaged in cycles of interaction is necessarily monostable as mathematically there is only one solution possible for each set of input values (see Document S1). For a monostable network, there is no dependency of state transition on the history of inputs; therefore, any state can be reached from any other state with the application of the appropriate input.

The G-NFSM specifies the exact input signals (as labels on transition arrows) that will result in each possible stable output

Table 1. Estimating the intelligence potential (IP) of various regulatory networks

Network	N Nodes	N Edges	N Cycles	Max In-Degree	Hier. Incoh.	N Output States	N Contexts	Multist. Ratio	Context Ratio	Intel. Poten.
AKT Net	24	30	0	3	0.27	6.36	0.00	0.8	0.00	0.80
Chain	6	5	0	2	0.24	6.50	0.00	0.81	0.00	0.81
Double Chain	6	7	2	3	1.37	6.27	0.91	0.78	0.11	0.90
Cycle	6	6	1	2	1.00	7.20	0.8	0.90	0.10	1.00
Double Cycle	6	9	5	3	2.00	7.18	1.55	0.90	0.19	1.09
Full Cycle	6	12	8	4	2.75	6.09	2.73	0.76	0.34	1.10
Self-loop Cycle	6	9	4	3	1.50	8.10	4.00	1.01	0.50	1.51
Self-loop Chain	6	8	3	3	0.58	7.30	5.20	0.91	0.65	1.56
Full Chain	6	10	5	4	1.95	7.40	5.70	0.93	0.71	1.64
hESC Net	12	56	997	10	156.60	10.45	3.18	1.31	0.40	1.70

The IP of a network is defined by the sum of the Multistate Ratio (MR) and the Context Ratio (CR). The MR is defined as the total number of output states (N Output States), scaled by the number of input states (N Input States = 8). A non-trivial input context is defined as an input-specified state subspace containing more than one equilibrium; a system with nontrivial input contexts is multistable and, therefore, capable of maintaining memory of state and emitting context-dependent behaviors. The CR is defined as the total number of non-trivial input contexts (N Contexts) scaled by the number of input states. By definition, a monostable network has at most one equilibrium per input context and can therefore have a maximum number of output states equal to the number of input states, for a maximum MR = 1.0. In contrast, multistability allows multiple states to be present in any input context, leading to MRs greater than 1.0. State degeneracy (the case where the same state is present in multiple input contexts) leads to MRs less than 1.0. A multi-stable system can have a maximum number of non-trivial input contexts equal to the number of input states for a maximum CR of 1.0. All networks studied in the table have eight input states. The intelligence potential of the system is loosely related to the number of cycles and hierarchical incoherence of the regulatory network, but unrelated to the total number of nodes, edges, or the maximum in-degree of nodes in the network. The form of networks included in this analysis are shown in [Figure S7](#), and analysis was performed on 15 unique networks of each class, each with different randomly generated edge interaction types (e.g., different unique combinations of activators and inhibitors).

equilibrium state ([Figure 1C](#)), and therefore, the G-NFSM shows precisely which input signal needs to be applied and held to obtain a desired emitted behavior. Note that the equilibrium states are intrinsic to the system and can be accessed without having to alter the network topology, therefore, this network is persuadable. We see that for this PI3K/AKT/mTor, RAS/ERK, and Wnt/β-catenin cross-regulation network, the G-NFSM of the RNM provides the necessary information to persuade the system to any desired output equilibrium state. For example, if state 0 represents a disease state while state 1 represents one of health, we can see that “persuading” the system from one of disease to one of health requires applying and holding input state I0. By using the RNM approach, we also determine that since the system is monostable and there is a 1:1 correspondence between an input signal and the resulting output equilibrium, there is no possibility in the case of this network for a transient intervention to effect a permanent change in output (i.e., one-constant-dose therapy is required to maintain the desired output state). Moreover, we recognize that the results apply to all monostable networks, which includes all regulatory networks that are directed acyclic graphs lacking cycles ([Document S1](#)).

However, our preliminary analysis determined that not all networks are monostable, and that multistability—the property of a state sub-space containing multiple equilibrium states (see [Figure 5](#))—endows an RNM with higher-order features such as the ability to have future behavior depend on past experiences and the ability to exhibit different behaviors to the same input signal depending on context. Therefore, our next biological case study examines the multistable OCT4-SOX2-NANOG tran-

scription factor network involved in direct lineage specification of embryonic stem cells (ESC).^{79,87,88}

Features of smart, persuadable regulatory networks

Here, we show a simple real-world example of how an RNM can parameterize stem cell differentiation in response to a *transient exposure* to a morphogen in a context-dependent, event-driven state transitions ([Figure 7](#)). While multistability has previously been suggested to account for the cell differentiation process,^{88,89} RNMs provide a framework enabling organized, systematic work with this concept ([Figure 7](#)). In essence, we are showing how multistable RNMs can have future behavior depend on their past experience to maintain a long-lasting, or even permanent, change to their outputs and future responses after a transient experience.

The transcription factors OCT4, SOX2, and NANOG are well known to form a tightly interacting network motif responsible for lineage specification in embryonic stem cells (ESC),⁷⁹ with the specific ESC regulatory network modeled in this example shown in [Figure 7A](#). High levels of NANOG, OCT4, and SOX2 are known to correspond with undifferentiated ESC (corresponding to output State 6, [Figure 7B](#)),⁷⁹ whereas differentiation toward a neuroectoderm lineage occurs when levels of NANOG and OCT4 drop leaving SOX2 high (corresponding to output State 1, [Figure 7B](#)).⁷⁹ Differentiation toward a mesoendoderm lineage occurs when levels of NANOG and SOX2 drop, leaving OCT4 high (corresponding to output State 3; [Figure 7B](#)).⁷⁹ In this example, application of different input states corresponds to the presence of morphogens that actuate the input effects.⁷³ Our RNM analysis of the ESC network in (A) indicates it is a multistable system with seven unique output states ([Figure 7B](#)), both a

G-NFSM (Figure 7C) and E-NFSMs (Figure 7D), and a high average Intelligence Potential of 1.64 (see “Full Chain” network of Table 1).

The biological observation of high NANOG, OCT4, and SOX2 in undifferentiated ESC indicates that undifferentiated ESC must start in State 6 (marked with a black block arrow and bold green highlight in Figure 7C). The G-NFSM suggests that both States 1 and 3 are accessible from State 6 via application of input I6 and I5, respectively (trajectories highlighted in pink in Figure 7C). However, State 6 exists in monostable context I0; therefore, while transient application of I6 or I5 leads to transition of the system to State 1 or State 3, it is not a permanent change, and the system reverts to State 6 once input I6 or I5 are removed (Figure 7E).

From the E-NFSM, we see that a non-trivial input context exists for input I4 when it is held as the “normal” baseline input for the system (Figure 7D). The E-NFSM shows us that it is only in context I4 that a permanent change of the system from State 6 to States 1 or 3 can be achieved from a transient application of the same I6 or I5 inputs, with respective paths highlighted in green (Figures 7C and 7D). In the time evolution of the system shown in Figure 7F, we see that by first switching to State 4 with the application of I4, a transient exposure to I5 now leads to a permanent switch of the system to State 3 even with a return to the original held context established by I4 (Figure 7F). Similarly, a system first switched from State 6 to State 4 with the application of I4 with a transient exposure to I6 will permanently switch to State 1, even with a return to the original held context established by I4.

This example illustrates the importance of baseline context in multistable systems by showing how the same morphogen can only successfully induce differentiation in the correct context. Moreover, the RNM specifies precisely how to use input states to set different contexts to see desired permanent results. Moving beyond the case of stem cell differentiation, this example also has important implications for medicine, as it demonstrates how a transient course of a medicine can offer successful long-term treatment of certain diseases by effecting permanent changes in a multistable system.

RNMs pinpoint an origin of highly variable biological responses

In our next example, we show how the RNM framework can help us understand how multiple –apparently identical– systems can show dramatically different responses to the same intervention. The regulatory network modeled in this example is shown in Figure 8A, which was found to have 7 unique output states (Figure 8B) and to have a G-NFSM (Figure 8C) and multiple E-NFSM (Figure 8D). The G-NFSM shows four transitions from State 5 to four unique states (State 0, State2, State 4, and State 6) happening with the application of the same input state I7 (bold arrows of Figure 8C). The E-NFSMs show four different input contexts (for systems with a baseline-normal held input state of I0, I1, I4, and I5) where State 5 transitions to another state under input I7 (bold arrows of Figure 8D). In time-course studies, we find that systems with a held input state of I5 (Figure 8E), I0 (Figure 8F), I4 (Figure 8G) and I1 (Figure 8H) are initially stable in State 5, as predicted by the G-NFSM and E-NFSMs. Subsequent transient application of input I7 to each of the four systems

leads to four different final states (Figures 8E–8H), as predicted by the E-NFSMs (Figure 8D).

This example shows that it is possible to use the RNM framework to identify nonspecific elements of the system (i.e., the combination of input factors S0, S1, and S2 that create the different contexts I0, I1, I4, and I5 in the Figure 8 example) that lead to categorically different responses to the same perturbing event/intervention (i.e., I7 in the Figure 8 example). This is significant, as a real-life situation may entail multiple patients who appear to have similar levels of key factors in their blood work (therefore being equivalent to the four systems shown here that start in State 5, see Figure 8) and yet these patients may each show dramatically different responses to a pharmaceutical treatment. The RNM framework opens the possibility to identify and control for these apparently extraneous factors (i.e., the fact that the four systems are under different input contexts) to minimize undesirable variation in biological responses to an intervention.

RNMs elucidate strategies in cancer renormalization therapy

Our results detail how RNM analysis can be used to identify specified treatment strategies to transition cells from a disease state back to a healthy state. When applied to cancer, this concept of inducing a transition between diseased to healthy cell states is known as *cancer renormalization*.^{70,71} We used the “MAPK Cancer Cell Fate” Boolean network model from,⁵⁷ (see Figure 2A), sourced from the Cell Collective model database, as an experimentally verified Boolean GRN model that specifies input nodes (“DNA damage”, “EGFR stimulus”, “FGFR3 stimulus”, “TGFBR stimulus”, see Figure 2B) and output nodes (“Apoptosis”, “Growth Arrest”, and “Proliferation” nodes in Figure 2C), in addition to 46 internal nodes with numerous internal feedback cycles (Figure 2A). Each input node variable represents an information bit, which taken together, form 16 input states, labeled I0 through I15 (Figure 2B). Our RNM analysis found that 15 unique equilibrium output states (labeled State 0 through State 14, Figure 2C; a heatmap with all internal node expressions of the GRN in the 15 states is shown in Figure S3) were possible for all of the possible applied inputs and determined the relationships between the equilibrium output state transitions and the applied inputs using techniques to generate both E-NFSMs (Figure 2E) and G-NFSM (Figure 2F). Interestingly, all equilibrium output states in the system are predicted to be limit cycles (except for States 0 and 1), which would spontaneously and cyclically pass through different modalities of a cell cycle rather than remain fixed as a monotonic gene expression.

The descriptive nature of the output nodes in this model allows for easy definitions of the ideal cancer state as one in which apoptosis is inhibited but proliferation is maximized (i.e., Cancer State = “Apoptosis” = 0, “Growth Arrest” = 0, and “Proliferation” = 1), versus a normal cell state which can cycle through all of the outputs (i.e., Normal State = “Apoptosis” = 0.33, “Growth Arrest” = 0.33, and “Proliferation” = 0.33). From the definition of the idealized cancer state, we calculated the Euclidean distance between each of the 15 equilibrium output states and the ideal cancer state, finding that States 1, 2, and 3 were closest to the ideal cancer state, that States 4 and 5

Table 2. Feature comparison between classical Turing machines, Hopfield networks, and regulatory network machines (RNMs), which are a hybrid model between Turing machines and Hopfield networks

	Turing Machine	Hopfield Network	RNM
States represent machine output?	No. States are abstract placeholders defining a step in a computational algorithm. States do not represent an output of the machine.	Yes. States are patterns of machine output representing memories recalled by the system in association with an input stimulus.	Yes. Output states are patterns representing memories recalled by the system in association with an input stimulus. Output states may represent patterns of genetic expression related to modes of system behavior.
Has a State Transition Diagram?	Yes. The system's computational program is embodied by a state transition diagram that instructs the machine to move to different states depending on what symbols it reads on its tape.	No. The Hopfield network is a model of associative memory, where the association between input and output is a 1:1 mapping. ⁹¹	Yes. Multistable RNMs have sophisticated relationships between inputs and outputs such that a transition to the next state depends on the history of inputs, thereby requiring a state transition diagram to map the possible evolution pathways.
Has a start state?	Yes. The Turing machine has a start state and its activity evolves in time.	No. As a model of content-addressable associative memory, the Hopfield network has no need for a start state.	Yes. RNMs have a start state and their activity evolves in time.
Has a moving read head?	Yes. A Turing machine has a head that can read and write to the tape and the head can be instructed to move left or right.	No. As a model of content-addressable associative memory, the Hopfield Network has no need for a moving read head.	Yes. An RNM has a head that can read from its tape and move to the right in time. Note: the RNM head is read-only and can only move right.
Has a tape of input symbols?	Yes. A Turing machine has a tape of input symbols that drive state transitions, as prescribed by its state transition diagram.	No. A Hopfield network has no input tape. However, it does have a language of input symbols that retrieve output states via associative memory.	Yes. A Turing machine has a tape of input symbols that drive state transitions. This is analogous to the biological system emitting behavior in response to experienced input.
Has a stop/halt state?	Yes. Machine halts when an answer is computed, delivering the output of its computation. This output is written to the tape, or is the symbol on the tape, when the machine halts.	No. As a model of content-addressable associative memory, the Hopfield Network has no need for a stop/halt state.	Not really. The RNM halts only when the organism dies.

were akin to a pre-cancerous state, and that the remaining States 6 through 14 could be interpreted as normal, healthy cells demonstrating both programmed proliferation and cell death. Pathway analysis of the G-NFSM map allows for automatic identification of persistent therapies that can be trialed to persuasively transition cancer cells back to a healthy state. The pathway analysis of the G-NFSM showing sustained input interventions leading from a cancer-like State 2 to healthy States 9, 10, 11, and 12 is shown in Figure 2D, indicating that while inputs I8 and I10 sustain a cancer state in this model, inputs I4 through I7 and I12 through I14 are all candidate therapeutic interventions.

The E-NFSMs of Figure 2E provide insights into whether it is possible for transient therapies to induce permanent outcomes in this model system. E-NFSM results show that under a sustained exposure to held inputs (baseline context) I8 or I10, a pre-cancerous State 5 irreversibly proceeds to a cancerous State 2 or 3 after a temporary exposure to various input events (Figure 2E). These results imply that once cancer develops, there is no way to persuade the cell to return to a normal state using a transient intervention without a change in network topology. However, we learned from G-NFSM analysis of this model that cancer is maintained only in the I8 and I10 contexts, which feature high EGFR in isolation, or EGFR and FGFR3 with no other stimuli. The role of I8 and I10 in sustaining cancer states is consistent with the results discovered by Grieco et al. in working with this same Boolean network model.⁵⁷ However, our G-NFSM also makes clear predictions for a wide array of cancer renormalization strategies, including the application of I4, I5, I6, I7, I12, I13, and I4 (see Figure 2D with input states shown in Figure 2B). Overall, we show how the G-NFSM and E-NFSMs of our RNM framework provide a clear and comprehensive map of the sequential logic embedded in the dynamics of a complex GRN dynamic system, leading to clear and efficient exposure of treatment strategies and possibilities that remain hidden in other analysis techniques.

DISCUSSION

Here, we presented and developed the conceptual and computational framework of RNMs, thereby enabling work with GRNs as persuadable entities capable of context-dependent information processing and behavioral output that depends on past experiences.

Notably, our RNM framework builds upon a well-known analysis method used to study regulatory networks called *Attractor Landscape Analysis*.^{70,71} Attractor Landscape Analysis is a technique, typically used with Boolean network models,^{68–71} that maps out spontaneous transitions from non-stable states to equilibrium states of a network for a *single* applied input state. While the STG of an attractor landscape analysis may superficially appear similar to our NFSMs, there are crucial differences to be noted. The STG shows how any starting state of the system naturally and spontaneously moves toward different kinds of equilibrium states in the attractor landscape defined when one input state is applied to the system (Figure S2), and the STG is a discrete version of the continuous direction field concept (Figure S2). There are no labels on the edges of the STG, as there is nothing that drives the transition between states; these transi-

tions happen naturally and spontaneously due to internal system dynamics. In our RNM framework, the application of an input state to the network is also associated with an attractor landscape (here, an attractor sub-landscape or state sub-space); however, we see that different input states generate categorically different attractor landscape sub-spaces (Figures 4E and 5F visualize this in two simple examples). In contrast to the STG, the NFSM maps *non-spontaneous* transitions between equilibria within and between different attractor sub-spaces, where transitions occur only as different input signals are applied to the system (Figure S2). In contrast to the STG, edge labels in the NFSM specify the input signal that is applied to induce the transition between two equilibrium states, and the states of the NFSM represent only equilibrium states and exclude all other states. By dealing only with the stable equilibrium states, our NFSM also offers a major advantage over attractor landscape models that work with all 2^N states, which may be prohibitively large (for $N = 53$ GRN nodes this is 9×10^{15} states), whereas our RNM framework refines the focus to only equilibrium states, which tend to number less than 1000 (in the examples presented here, to less than 20), even in complex networks. Thus, the NFSM produces a map of the effects that patterns of input have on the stable outputs of the system, where, as we have seen in biological case examples, path-dependencies and cycles in the NFSM are crucial for supporting advanced behaviors.

Through our RNM framework, we have shown how a regulatory network with fixed topology can take different “experiences” as input information obtained from the internal/external environment and use analog computation to process this information to engage a range of responses through changes in the expression or activity of its own nodes, which are represented by output equilibrium states. The RNM framework thereby demonstrates how a GRN can be understood to be a persuadable system in which concise information signals (i.e., input signal states) can activate inherent competencies (i.e., output states) to get the system to emit a desired behavior or perform a desired function. In this framework, we can thereby contextualize “health” and “disease” as different possible output states that an organism may occupy, where the NFSMs of the RNM framework provide maps indicating which specific interventions are most likely to result in transitions between health and disease states, and how the intervention needs to be applied (e.g., constantly, or if a transient course of treatment will effect a permanent change of state). Here, this process was demonstrated in detail using the MAPK Cancer Cell Fate network (Figure 2).

In working with various biological case studies, we have identified system multistability as a key criteria distinguishing “smart” RNMs that can learn from experience (or alternatively, that can change permanently after a transient event, Figure 7) and perform context-dependent behaviors where different outcome behavior is observed from systems in the same initial state but experiencing different baseline conditions (Figure 8) from those that have more simplistic 1:1 relationships between inputs and outputs (Figure 10). The RNM framework thereby shows how, if we have accurate regulatory network models, we can enhance biomedicine in two ways. First, it suggests that we can use RNM with accurate regulatory network models to move beyond the “one-constant-dose” paradigm in medicine

through the identification of specific factors in multistable RNM systems that allow the system to respond permanently to a transient intervention (Figure 7). Second, the RMN framework opens the possibility to identify and control for apparently extraneous factors existing in input contexts to minimize undesirable variation in patient responses to an intervention (Figure 8).

We have also provided some illustrations of the analog computing characteristics of biological systems controlled by regulatory networks, which highlight characteristics that are in striking contrast to how human-made digital systems, such as PID controllers, function (Figure 9). By shifting our focus to see a biological system as one that employs analog computing, we see that the biological system has no need for a physical memory to store information, such as the set-point value of a homeostat; rather, the set-point value is an inherent aspect of the system dynamics, and can be considered to be stored abstractly in the system's direction field as the equilibrium point of a limit cycle attractor (i.e., the large green circle terminating the green trajectory in Figures 9G and 9H). The biological system also has no need for a CPU to perform algorithmic computations of system variables such as error signals and the magnitude of behavioral responses. Instead, the system dynamics naturally utilize the present state of the biological system and sensed environmental variables to spontaneously evolve in a manner that naturally embodies an appropriate response to an environmental challenge such as high extracellular osmolarity (Figure 9).

Computationally, our RNM model can be analogized to a hybrid Turing-Hopfield digital computing machine (Figures 3F–3H) which, like a classical Turing machine,^{60,90} is fed a tape of input state symbols (Figure 3F) that are read by a read-head moving to the right in time (Figure 3F). The tape represents the sensed experience of the dynamic system in time. Like a classical *Turing machine*, the RNM can store a memory of state and runs a program specified by a state transition diagram (Figure 3F), which specifies which state to progress to, given its present state and the input symbol read from the tape (Figure 3F and Box 1). Yet, like a *Hopfield network*,⁹¹ the system also has associative memory, such that each state represents a pattern of output variables, which are the computational output (Figures 3F and 3G and Box 1). The result is a progression of initial and final system output states with each input state applied in time (Figure 3H). A detailed comparison of the properties of Turing, Hopfield, and RNMs is summarized in Table 2.

Our RNM framework is also related to connectivist computational approaches to intelligence that utilize artificial neural networks (ANNs), in particular to Physics Informed Neural Networks⁹² (PINNs) and Reservoir Computing.⁹³ Similar to GRNs, an ANN consists of a network of nodes that interact with each other via connecting edges, where the ANN processes information by transforming an input signal into an output signal. Some ANNs learn a set input-output relationship contained in a dataset through training algorithms, which alter the ANN network topology (i.e., change the connecting edges between nodes). A PINN is an ANN that has been partially trained on differential equations representing the laws of physics (e.g., the neural network may be trained on the Navier-Stokes equations describing fluid dynamics). This allows the PINN to embed the knowledge of the physics (e.g., Navier-Stokes equations) and to thereby capture

the correct solution in a specific problem area (e.g., a fluid dynamics problem) with a minimal amount of new training data. Working from the analog computing perspective developed herein, the GRN of our RNM framework could also be said to be naturally trained (i.e., the GRN topology is naturally shaped) with regards to knowledge of physical systems through the processes of a biological organism's continuous physical experience and evolution through natural selection, which is the "programming" that enables effective analog computing responses to environmental challenges (e.g., osmolarity shocks). Our framework also parallels Reservoir Computing, which is based on the concept that input stimuli perturb the state of a multistable, network-based computational reservoir, sending the reservoir to certain attractor states. Unlike in traditional ANNs, but similar to our RNM framework, in reservoir computing the edges of the reservoir network do not change, and the system relies on the memory and other complex dynamics of its multistable "computational reservoir" to achieve information processing results.

While here we have focused on GRNs, our RNM approach has a high degree of general applicability (i.e., universality). The RNM framework has only three general dependencies to function: (1) the system of study must be a dissipative dynamic system, (2) the system must be able to accept inputs and produce stable outputs, and (3) a systematic methodology must be followed to generate all of the transitions of the NFSM of the RNM (see [STAR Methods](#) for details). Examples of regulatory networks that are dissipative dynamic systems that will work with our RNM framework are:

- (1) Transcription factor networks
- (2) Metabolic and chemical reaction networks
- (3) Bioelectric networks
- (4) Neural networks
- (5) Ecological food webs
- (6) Human economic systems
- (7) Mixed signaling/regulatory networks comprising a variety of entities

Overall, the RNM framework provides the essential elements required to understand a regulatory network as a persuadable entity, where for any given regulatory network, an RNM details what specifically needs to be done to persuade a system to desired modes of activity. The RNM framework also determines the limitations to persuading any given system by indicating the level of function at which it is necessary to reprogram the system by changing its hardware (i.e., alter its physical make up). In this, there are numerous applications of our RNM framework in biomedical and bioengineering. The RNM can be used to address drug habituation (the loss of pharmacological efficacy over time), where the G-NFSM can be used to identify new strategies to achieve a "healed" state from the habituated state using an alternative pharmacological strategy. Also, the RNM can work to solve drug sensitization (functional pharmaceuticals that cannot be given for therapeutically necessary time periods because their side effects build over time), where the E-NFSM of our RNM can be used to identify if and how transient applications of pharmaceuticals can induce permanent outcomes. In

regenerative medicine and injury, the G-NFSM and E-NFSMs can be used to determine how to persuade GRNs to trigger complex repair cascades. As we have shown here, in cancer the G-NFSM and E-NFSM can be used to identify strategies to shift complex, system-level behavioral profiles toward cooperativity with tissue homeostasis. There are also applications in bioengineering where the G-NFSM and E-NFSM can be used to identify strategies to achieve desired outcomes, such as increased yeast resistance to ethanol concentrations to improve bio-ethanol yields.

By working with regulatory networks as analog computational systems, and with the introduction of our RNM framework, we have shown how to generate maps evidencing the “software-like” nature of a regulatory network, providing easy identification of the specific interventions necessary to achieve desired outcomes. Importantly, we have shown how system-level outcomes such as cancer normalization can be induced in a biological system without requiring genetic rewiring, and in cases that are too complex to directly micromanage. Our RNM approach determines the behavior and capabilities of regulatory networks, elucidating their innate computational capabilities and ascertaining the interventions that will provide the most control for the least amount of effort (or side effects). Ultimately, the view of regulatory networks as analog computers described by RNMs expands horizons in our understanding and treatment of disease.

Limitations of the study

As with all frameworks in network science, the results of RNM analysis are only as good as the regulatory network model on which the RNM is based, so care must be taken to utilize good starting GRN models. Obtaining accurate regulatory network models for multistable dynamic systems may be highly dependent on experimental design and the inference method; thus, in future work, we will examine aspects of experimental design and direct network inference methods^{94–97} that best identify regulatory networks in multistable systems. However, it is worth highlighting that as the accuracy of our RNM framework does indeed hinge on the accuracy of an underlying regulatory network model, we have undertaken significant efforts to select regulatory networks that are suitable for the RNM, have already undergone experimental verification, and also have relevance to important biological or biomedical cases, these are: (1) the well-known HOG-MAPK signaling network of yeast osmoadaptation^{80–82,98}; (2) the cross-regulation network between PAM, RAS/ERK, and Wnt/β-Catenin, which is known to govern many processes including cancer transformation⁷⁴; (3) the NANOG/OCT4/SOX2 network that has undergone extensive experimental study by many research groups for its role in stem cell differentiation and cancer^{79,87,88,99}; and (4) the MAPK Cancer Cell Fate network.⁵⁷

Regarding the computational efficiency and capabilities of our RNM framework, here, we have utilized two different computational models of GRN. We have employed continuous models of GRN described by nonlinear differential equations, which maximize bio-realism and accuracy,¹⁷ yet do not offer computationally efficient modeling capacity. However, we have also used Boolean network computational models of the GRNs, which offer exceptional computational efficiency yet may sacrifice ac-

curacy in GRNs with large numbers of cycles and high hierarchical incoherence.^{61,62} We have found that Boolean networks are up to 400 times faster at computing GRN outputs than our continuous model utilizing differential equations (Table S1), with excellent comparison to continuous models for GRNs (Figure S1). In future work, we aim to identify parallelization techniques to improve on the computational efficiency of the continuous partial differential equation model of the GRN.

We view this work as a complement to efforts to understand gene regulatory networks (GRNs) as computational and proto-cognitive media,^{5,41,42,100,101} and to help explain remarkable and still poorly understood ways in which tissues can adaptively respond to novel stressors.¹⁰² Next steps involve developing a deeper understanding of programmability in this computational medium, applying it to a wider set of biomedically relevant networks, integrating it with other kinds of networks, such as neural and non-neural bioelectric circuits,^{103–105} using it to guide ongoing experiments in closed-loop systems to control living cells’ responses to drugs and other stimuli,^{106–109} and exploring the dynamics of evolution performed over a substrate with such computational properties.¹¹⁰ These analyses could be an important step toward moving from an almost exclusive emphasis on micromanaging biological hardware (via genomic editing and promoter engineering) to an exploitation of aspects of decision-making, memory, and autonomy in the material which would enable more powerful control strategies.^{111–113} We especially envision a roadmap in which a better understanding of these unexpected features of pathways lead to biomedical therapies that exploit timing and patterns of diverse stimuli to go beyond what single-dose, continuous use of pharmacological interventions can achieve, to overcome habituation, side effects, and unpredictable variance in efficacy.¹¹⁴

RESOURCE AVAILABILITY

Lead contact

Requests for further information and resources should be directed to, and will be fulfilled by, the lead contact, Michael Levin (michael.levin@tufts.edu).

Materials availability

This study did not generate new unique reagents.

Data and code availability

- Data: Synthetic data used in RNM frameworks can be generated and downloaded for continuous models from: https://github.com/betsee/cellnition/blob/main/ipynb/Tutorial1_ContinuousNFSM_v1.ipynb and for Boolean models from: https://github.com/betsee/cellnition/blob/main/ipynb/Tutorial2_BooleanNFSM_v1.ipynb.
- Code: The software code developed to create RNM frameworks is available for use as the Python API, *Cellnition*, which enables others to explore the application and investigation of regulatory networks in terms of the RNM framework. All original code has been deposited at the following GitHub repository and as of the date of publication is publicly available from: <https://github.com/betsee/cellnition>.
- Any additional information required to reanalyze the data reported in this paper is available from the [lead contact](#) upon request.

ACKNOWLEDGMENTS

The authors gratefully acknowledge support for this work from the Templeton World Charity Foundation grant TWCF0606 and M.L. for a sponsored research agreement from Astonishing Labs. The authors thank computer scientist Cecil

Curry for helpful discussions and Julia Poirier for her assistance with manuscript submission.

AUTHOR CONTRIBUTIONS

Conceptualization, A.P. and M.L.; methodology, A.P.; software, A.P.; validation, A.P., investigation, A.P.; formal analysis, A.P.; resources, M.L.; data curation, A.P.; writing – original draft, A.P.; writing – review and editing, A.P. and M.L.; visualization, A.P.; supervision, M.L.; and funding acquisition, M.L.

DECLARATION OF INTERESTS

M.L. is scientific co-founder and minor shareholder of Astonishing Labs, a company seeking to develop advances in regenerative medicine based on proto-cognitive properties of gene-regulatory networks. Astonishing Labs provides a sponsored research agreement to Tufts University to support this research. M.L. also has an associate faculty appointment at Harvard's Wyss Institute. M.L. and A.P. are listed as co-inventors on a provisional patent application covering the work reported here.

STAR METHODS

Detailed methods are provided in the online version of this paper and include the following:

- KEY RESOURCES TABLE
- METHOD DETAILS
 - General computational strategy
 - GRN base functions
 - Combining multiple regulatory interactions
 - State space equilibrium search and characterization
 - Network Finite State Machine build procedure
- STEPS FOR APPLYING RNM ANALYSIS TO BIOMEDICAL APPLICATIONS
 - Estimating the intelligence potential of regulatory networks
 - Directed acyclic GRNs: Evidence for Guaranteed Mono-stability and fully analytical solutions
 - Osmoadaptation model mathematics
- ADDITIONAL RESOURCES

SUPPLEMENTAL INFORMATION

Supplemental information can be found online at <https://doi.org/10.1101/j.isci.2025.112536>.

Received: December 2, 2024

Revised: February 27, 2025

Accepted: April 24, 2025

Published: April 28, 2025

REFERENCES

1. Lyon, P. (2006). The biogenetic approach to cognition. *Cogn. Process.* **7**, 11–29.
2. Lyon, P. (2015). The cognitive cell: bacterial behavior reconsidered. *Front. Microbiol.* **6**, 264.
3. Baluška, F., and Levin, M. (2016). On Having No Head: Cognition throughout Biological Systems. *Cogn. Sci.* **7**, 902.
4. Gershman, S.J., Balbi, P.E., Gallistel, C.R., and Gunawardena, J. (2021). Reconsidering the evidence for learning in single cells. *Elife* **10**, e61907.
5. Katz, Y., Springer, M., and Fontana, W. (2018). Embodying probabilistic inference in biochemical circuits [Internet]. Preprint at arXiv. <http://arxiv.org/abs/1806.10161>.
6. Shen, W., Gao, Z., Chen, K., Zhao, A., Ouyang, Q., and Luo, C. (2023). The regulatory mechanism of the yeast osmoregulation under different glucose concentrations. *iScience* **26**, 105809.
7. Babazadeh, R. (2014). Integrative analysis of osmoregulation in yeast *Saccharomyces cerevisiae*. Gothenburg: Department of Chemistry and Molecular Biology (University of Gothenburg). [Roja Babazadeh].
8. Gasch, A.P., Spellman, P.T., Kao, C.M., Carmel-Harel, O., Eisen, M.B., Storz, G., Botstein, D., and Brown, P.O. (2000). Genomic Expression Programs in the Response of Yeast Cells to Environmental Changes. *Mol. Biol. Cell* **11**, 4241–4257.
9. Karlgren, S., Pettersson, N., Nordlander, B., Mathai, J.C., Brodsky, J.L., Zeidel, M.L., Bill, R.M., and Hohmann, S. (2005). Conditional Osmotic Stress in Yeast. *J. Biol. Chem.* **280**, 7186–7193.
10. Hernández, R., Jiménez-Luna, C., Perales-Adán, J., Perazzoli, G., Melguizo, C., and Prados, J. (2020). Differentiation of Human Mesenchymal Stem Cells towards Neuronal Lineage: Clinical Trials in Nervous System Disorders. *Biomol. Ther.* **28**, 34–44.
11. Babloyantz, A., and Hiernaux, J. (1975). Models for cell differentiation and generation of polarity in diffusion-governed morphogenetic fields. *Bull. Math. Biol.* **37**, 637–657.
12. Peake, M.A., Caley, M., Giles, P.J., Wall, I., Enoch, S., Davies, L.C., Kipping, D., Thomas, D.W., and Stephens, P. (2014). Identification of a transcriptional signature for the wound healing continuum. *Wound Repair Regen.* **22**, 399–405.
13. Cooper, L., Johnson, C., Burslem, F., and Martin, P. (2004). Wound healing and inflammation genes revealed by array analysis of “macrophage-less” PU.1 null mice. *Genome Biol.* **6**, R5.
14. Deonarine, K., Panelli, M.C., Stashower, M.E., Jin, P., Smith, K., Slade, H., B., Norwood, C., Wang, E., Marincola, F.M., and Stroncek, D.F. (2007). Gene expression profiling of cutaneous wound healing. *J. Transl. Med.* **5**, 11.
15. Jhamb, D., Rao, N., Milner, D.J., Song, F., Cameron, J.A., Stocum, D.L., and Palakal, M.J. (2011). Network based transcription factor analysis of regenerating axolotl limbs. *BMC Bioinf.* **12**, 80.
16. Boi, L. (2011). Plasticity and Complexity in Biology: Topological Organization, Regulatory Protein Networks, and Mechanisms of Genetic Expression. <https://direct.mit.edu/books/edited-volume/3780/chapter/124419/Plasticity-and-Complexity-in-Biology-Topological>.
17. Karlebach, G., and Shamir, R. (2008). Modelling and analysis of gene regulatory networks. *Nat. Rev. Mol. Cell Biol.* **9**, 770–780.
18. Singh, A.J., Ramsey, S.A., Filtz, T.M., and Kioussi, C. (2018). Differential gene regulatory networks in development and disease. *Cell. Mol. Life Sci.* **75**, 1013–1025.
19. Ten Tusscher, K.H., and Hogeweg, P. (2011). Evolution of networks for body plan patterning; interplay of modularity, robustness and evolvability. *PLoS Comput. Biol.* **7**, e1002208.
20. Peter, I.S., and Davidson, E.H. (2011). Evolution of Gene Regulatory Networks that Control Embryonic Development of the Body Plan. *Cell* **144**, 970–985.
21. Uller, T., Moczek, A.P., Watson, R.A., Brakefield, P.M., and Laland, K.N. (2018). Developmental Bias and Evolution: A Regulatory Network Perspective. *Genetics* **209**, 949–966.
22. Shin, D., Gong, J.R., Jeong, S.D., Cho, Y., Kim, H.P., Kim, T.Y., and Cho, K.H. (2025). Attractor Landscape Analysis Reveals a Reversion Switch in the Transition of Colorectal Tumorigenesis. *Adv. Sci.* **12**, 2412503.
23. Liu, R., Liang, L., Choudhury, A., Garst, A.D., Eckert, C.A., Oh, E.J., Winkler, J., and Gill, R.T. (2019). Multiplex navigation of global regulatory networks (MINR) in yeast for improved ethanol tolerance and production. *Metab. Eng.* **51**, 50–58.
24. Kasavi, C., Eraslan, S., Arga, K.Y., Oner, E.T., and Kirdar, B. (2014). A system based network approach to ethanol tolerance in *Saccharomyces cerevisiae*. *BMC Syst. Biol.* **8**, 90.
25. Sun, D., Wu, L., Lu, X., Li, C., Xu, L., Li, H., He, D., Yu, A., Yu, T., Zhao, J., et al. (2025). Engineering transcriptional regulatory networks for improving second-generation fuel ethanol production in *Saccharomyces cerevisiae*. *Synth. Biotechnol.* **10**, 207–217.

26. Velazquez, J.J., Su, E., Cahan, P., and Ebrahimkhani, M.R. (2018). Programming Morphogenesis through Systems and Synthetic Biology. *Trends Biotechnol.* 36, 415–429.
27. Brophy, J.A.N., and Voigt, C.A. (2014). Principles of genetic circuit design. *Nat. Methods* 11, 508–520.
28. Saltepe, B., Bozkurt, E.U., Güngen, M.A., Çiçek, A.E., and Şeker, U.Ö.Ş. (2021). Genetic circuits combined with machine learning provides fast responding living sensors. *Biosens. Bioelectron.* 178, 113028.
29. Krzysztoń, R., Wan, Y., Petreczky, J., and Balázs, G. (2021). Gene-circuit therapy on the horizon: synthetic biology tools for engineered therapeutics. *Acta Biochim. Pol.* 68, 377–383.
30. Peng, W., Song, R., and Acar, M. (2016). Noise reduction facilitated by dosage compensation in gene networks. *Nat. Commun.* 7, 12959.
31. Peng, W., Liu, P., Xue, Y., and Acar, M. (2015). Evolution of gene network activity by tuning the strength of negative-feedback regulation. *Nat. Commun.* 6, 6226.
32. Guye, P., Li, Y., Wroblewska, L., Duportet, X., and Weiss, R. (2013). Rapid, modular and reliable construction of complex mammalian gene circuits. *Nucleic Acids Res.* 41, e156.
33. Slusarczyk, A.L., Lin, A., and Weiss, R. (2012). Foundations for the design and implementation of synthetic genetic circuits. *Nat. Rev. Genet.* 13, 406–420.
34. Beal, J., Lu, T., and Weiss, R. (2011). Automatic compilation from high-level biologically-oriented programming language to genetic regulatory networks. *PLoS One* 6, e22490.
35. Kim, H., and Sayama, H. (2018). How Criticality of Gene Regulatory Networks Affects the Resulting Morphogenesis under Genetic Perturbations. *Artif. Life* 24, 85–105.
36. Solé, R.V., Fernández, P., and Kauffman, S.A. (2003). Adaptive walks in a gene network model of morphogenesis: insights into the Cambrian explosion. *Int. J. Dev. Biol.* 47, 685–693.
37. Villani, M., D'Addese, G., Kauffman, S.A., and Serra, R. (2022). Attractor-Specific and Common Expression Values in Random Boolean Network Models (with a Preliminary Look at Single-Cell Data). *Entropy* 24, 311.
38. Beggs, J.M. (2008). The criticality hypothesis: how local cortical networks might optimize information processing. *Philos. Trans. A Math. Phys. Eng. Sci.* 366, 329–343.
39. Graudenzi, A., Serra, R., Villani, M., Damiani, C., Colacci, A., and Kauffman, S.A. (2011). Dynamical properties of a boolean model of gene regulatory network with memory. *J. Comput. Biol.* 18, 1291–1303.
40. Graudenzi, A., Serra, R., Villani, M., Colacci, A., and Kauffman, S.A. (2011). Robustness analysis of a Boolean model of gene regulatory network with memory. *J. Comput. Biol.* 18, 559–577.
41. Biswas, S., Manicka, S., Hoel, E., and Levin, M. (2021). Gene regulatory networks exhibit several kinds of memory: Quantification of memory in biological and random transcriptional networks. *iScience* 24, 102131.
42. Biswas, S., Clawson, W., and Levin, M. (2022). Learning in Transcriptional Network Models: Computational Discovery of Pathway-Level Memory and Effective Interventions. *Int. J. Mol. Sci.* 24, 285.
43. Csermely, P., Kunscic, N., Mendik, P., Kerestély, M., Faragó, T., Veres, D. V., and Tompa, P. (2020). Learning of Signaling Networks: Molecular Mechanisms. *Trends Biochem. Sci.* 45, 284–294.
44. Perez-Lopez, Á.R., Szalay, K.Z., Türei, D., Módos, D., Lenti, K., Korcsmáros, T., and Csermely, P. (2015). Targets of drugs are generally, and targets of drugs having side effects are specifically good spreaders of human interactome perturbations. *Sci. Rep.* 5, 10182.
45. Kovács, I.A., Mizsei, R., and Csermely, P. (2015). A unified data representation theory for network visualization, ordering and coarse-graining. *Sci. Rep.* 5, 13786.
46. Gyurkó, M.D., Steták, A., Sőti, C., and Csermely, P. (2015). Multitarget network strategies to influence memory and forgetting: the Ras/MAPK pathway as a novel option. *Mini Rev. Med. Chem.* 15, 696–704.
47. Csermely, P., Hódsági, J., Korcsmáros, T., Módos, D., Perez-Lopez, Á. R., Szalay, K., Veres, D.V., Lenti, K., Wu, L.Y., and Zhang, X.S. (2015). Cancer stem cells display extremely large evolvability: alternating plastic and rigid networks as a potential Mechanism: network models, novel therapeutic target strategies, and the contributions of hypoxia, inflammation and cellular senescence. *Semin. Cancer Biol.* 30, 42–51.
48. Levin, M. (2022). Technological Approach to Mind Everywhere: An Experimentally-Grounded Framework for Understanding Diverse Bodies and Minds. *Front. Syst. Neurosci.* 16, 768201.
49. Braun, E., and Marom, S. (2015). Universality, complexity and the praxis of biology: Two case studies. *Stud. Hist. Philos. Biol. Biomed. Sci.* 53, 68–72.
50. Flann, N.S., Mohamadlou, H., and Podgorski, G.J. (2013). Kolmogorov complexity of epithelial pattern formation: the role of regulatory network configuration. *Biosystems* 112, 131–138.
51. Joachimczak, M., and Wróbel, B. (2009). Complexity of the search space in a model of artificial evolution of gene regulatory networks controlling 3d multicellular morphogenesis. *Adv. Complex Syst.* 12, 347–369.
52. Bizzarri, M., Cucina, A., Conti, F., and D'Anselmi, F. (2008). Beyond the oncogene paradigm: understanding complexity in cancerogenesis. *Acta Biotheor.* 56, 173–196.
53. Li, Q., Wennborg, A., Aurell, E., Dekel, E., Zou, J.Z., Xu, Y., Huang, S., and Ernberg, I. (2016). Dynamics inside the cancer cell attractor reveal cell heterogeneity, limits of stability, and escape. *Proc. Natl. Acad. Sci. USA* 113, 2672–2677.
54. Huang, S., Ernberg, I., and Kauffman, S. (2009). Cancer attractors: a systems view of tumors from a gene network dynamics and developmental perspective. *Semin. Cell Dev. Biol.* 20, 869–876.
55. de Bivort, B., Huang, S., and Bar-Yam, Y. (2007). Empirical multiscale networks of cellular regulation. *PLoS Comput. Biol.* 3, 1968–1978.
56. Lobo, D., Solano, M., Bubenik, G.A., and Levin, M. (2014). A linear-encoding model explains the variability of the target morphology in regeneration. *J. R. Soc. Interface* 11, 20130918.
57. Grieco, L., Calzone, L., Bernard-Pierrot, I., Radványi, F., Kahn-Perlès, B., and Thieffry, D. (2013). Integrative Modelling of the Influence of MAPK Network on Cancer Cell Fate Decision. *PLoS Comput. Biol.* 9, e1003286.
58. Moutsinas, G., Shuaib, C., Guo, W., and Jarvis, S. (2021). Graph hierarchy: a novel framework to analyse hierarchical structures in complex networks. *Sci. Rep.* 11, 13943.
59. Huang, S. (2006). Chapter 14 - Multistability and Multicellularity: Cell Fates as High-Dimensional Attractors of Gene Regulatory Networks. In *Computational Systems Biology* [Internet], A. Kriete and R. Eils, eds. (Burlington: Academic Press), pp. 293–326. <https://www.sciencedirect.com/science/article/pii/B9780120887866500332>.
60. Hopcroft, J., Motwani, R., and Ullman, J. (2001). *Introduction to Automata Theory, Languages, and Computation*, 2nd Edition (Reading, MA: Addison-Wesley).
61. Pušnik, Ž., Mraz, M., Zimic, N., and Moškon, M. (2022). Review and assessment of Boolean approaches for inference of gene regulatory networks. *Heliyon* 8, e10222.
62. Schwab, J.D., Kühlwein, S.D., Ikonomi, N., Kühl, M., and Kestler, H.A. (2020). Concepts in Boolean network modeling: What do they all mean? *Comput. Struct. Biotechnol. J.* 18, 571–582.
63. Siegelmann, H.T., and Fishman, S. (1998). Analog computation with dynamical systems. *Phys. Nonlinear Phenom.* 120, 214–235.
64. Campbell, S.L., and Haberman, R. (2011). *Introduction to Differential Equations with Dynamical Systems* (Princeton University Press), p. 312.
65. Miller, P. (2016). Dynamical systems, attractors, and neural circuits. *F1000Res.* 5, 992.
66. Murakami, K. (2018). A concrete example with multiple limit cycles for three dimensional Lotka–Volterra systems. *J. Math. Anal. Appl.* 457, 1–9.

67. Taylor, R.L.V. (2011). Attractors: Nonstrange to Chaotic. SIAM Undergrad. Res. Online 4, 72–80.
68. Cheng, D., and Qi, H. (2010). State-Space Analysis of Boolean Networks. IEEE Trans. Neural Netw. 21, 584–594.
69. Berntenis, N., and Ebeling, M. (2013). Detection of attractors of large Boolean networks via exhaustive enumeration of appropriate subspaces of the state space. BMC Bioinf. 14, 361.
70. Kim, Y., Choi, S., Shin, D., and Cho, K.H. (2017). Quantitative evaluation and reversion analysis of the attractor landscapes of an intracellular regulatory network for colorectal cancer. BMC Syst. Biol. 11, 45.
71. Cho, S.H., Park, S.M., Lee, H.S., Lee, H.Y., and Cho, K.H. (2016). Attractor landscape analysis of colorectal tumorigenesis and its reversion. BMC Syst. Biol. 10, 96.
72. Davila-Velderrain, J., Martinez-Garcia, J.C., and Alvarez-Buylla, E.R. (2015). Modeling the epigenetic attractors landscape: toward a post-genomic mechanistic understanding of development. Front. Genet. 6, 160. <http://www.ncbi.nlm.nih.gov/pmc/articles/PMC4407578/>.
73. Mossahebi-Mohammadi, M., Quan, M., Zhang, J.S., and Li, X. (2020). FGF Signaling Pathway: A Key Regulator of Stem Cell Pluripotency. Front. Cell Dev. Biol. 8, 79.
74. Glaviano, A., Foo, A.S.C., Lam, H.Y., Yap, K.C.H., Jacot, W., Jones, R.H., Eng, H., Nair, M.G., Makvandi, P., Georger, B., et al. (2023). PI3K/AKT/mTOR signaling transduction pathway and targeted therapies in cancer. Mol. Cancer 22, 138.
75. Jin, C., Samuelson, L., Cui, C.B., Sun, Y., and Gerber, D.A. (2011). MAPK/ERK and Wnt/β-Catenin pathways are synergistically involved in proliferation of Sca-1 positive hepatic progenitor cells. Biochem. Biophys. Res. Commun. 409, 803–807.
76. Baltanás, R., Bush, A., Couto, A., Durrieu, L., Hohmann, S., and Colman-Lerner, A. (2013). Pheromone-Induced Morphogenesis Improves Osmoadaptation Capacity by Activating the HOG MAPK Pathway. Sci. Signal 6, ra26. <https://www.science.org/doi/10.1126/scisignal.2003312>.
77. Guardavaccaro, D., and Clevers, H. (2012). Wnt/β-catenin and MAPK signaling: allies and enemies in different battlefields. Sci. Signal. 5, pe15.
78. Hill, C.B., Jha, D., Bacic, A., Tester, M., and Roessner, U. (2013). Characterization of Ion Contents and Metabolic Responses to Salt Stress of Different Arabidopsis AtHKT1;1 Genotypes and Their Parental Strains. Mol. Plant 6, 350–368.
79. Wang, Z., Oron, E., Nelson, B., Razis, S., and Ivanova, N. (2012). Distinct Lineage Specification Roles for NANOG, OCT4, and SOX2 in Human Embryonic Stem Cells. Cell Stem Cell 10, 440–454.
80. Kühn, C., and Gennemark, P. (2013). Modeling Yeast Osmoadaptation at Different Levels of Resolution. J. Bioinform. Comput. Biol. 11, 1330001.
81. Kühn MC. Modeling and Analysis of Yeast Osmoadaptation in Cellular Context. <https://doi.org/10.18452/16287>.
82. Hohmann, S. (2002). Osmotic Stress Signaling and Osmoadaptation in Yeasts. Microbiol. Mol. Biol. Rev. 66, 300–372.
83. Vu, T.N.L., Lee, J., and Lee, M. (2007). Design of multi-loop PID controllers based on the generalized IMC-PID method with Mp criterion. Int. J. Control Autom. Syst. 5, 212.
84. Li, Y., Ang, K.H., and Chong, G.C.Y. (2006). PID control system analysis and design. IEEE Control Syst. Mag. 26, 32–41.
85. Babazadeh, R., Furukawa, T., Hohmann, S., and Furukawa, K. (2014). Rewiring yeast osmostress signalling through the MAPK network reveals essential and non-essential roles of Hog1 in osmoadaptation. Sci. Rep. 4, 4697.
86. Patel, A.K., Bhartiya, S., and Venkatesh, K.V. (2014). Analysis of osmoadaptation system in budding yeast suggests that regulated degradation of glycerol synthesis enzyme is key to near-perfect adaptation. Syst. Synth. Biol. 8, 141–154.
87. Ogony, J.W., Malahias, E., Vadigepalli, R., and Anni, H. (2013). Ethanol alters the balance of Sox2, Oct4, and Nanog expression in distinct subpopulations during differentiation of embryonic stem cells. Stem Cells Dev. 22, 2196–2210.
88. Facon, P.C., Pardee, K., Kumar, R.M., Li, H., Loh, Y.H., and Wang, X. (2014). Gene Networks of Fully Connected Triads with Complete Auto-Activation Enable Multistability and Stepwise Stochastic Transitions. MacArthur BD. PLoS One 9, e102873.
89. Ghaffarizadeh, A., Flann, N.S., and Podgorski, G.J. (2014). Multistable switches and their role in cellular differentiation networks. BMC Bioinf. 15, S7.
90. Cotteret, M., Greatorex, H., Ziegler, M., and Chicca, E. (2024). Vector Symbolic Finite State Machines in Attractor Neural Networks. Neural Comput. 36, 549–595.
91. Hopfield, J.J. (1982). Neural networks and physical systems with emergent collective computational abilities. Proc. Natl. Acad. Sci. USA 79, 2554–2558.
92. Raissi, M., Perdikaris, P., and Karniadakis, G.E. (2019). Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. J. Comput. Phys. 378, 686–707.
93. Gauthier, D.J., Boltt, E., Griffith, A., and Barbosa, W.A.S. (2021). Next generation reservoir computing. Nat. Commun. 12, 5564.
94. Moyano, T.C., Gutiérrez, R.A., and Alvarez, J.M. (2021). Genomic Footprinting Analyses from DNase-seq Data to Construct Gene Regulatory Networks. In Modeling Transcriptional Regulation: Methods and Protocols [Internet], S. MUKHTAR, ed. (New York, NY: Springer US), pp. 25–46. https://doi.org/10.1007/978-1-0716-1534-8_3.
95. Hu, Y., Ma, S., Kartha, V.K., Duarte, F.M., Horlbeck, M., and Zhang, R. (2023). Single-cell multi-scale footprinting reveals the modular organization of DNA regulatory elements. Preprint at bioRxiv.
96. Sonawane, A.R., DeMeo, D.L., Quackenbush, J., and Glass, K. (2021). Constructing gene regulatory networks using epigenetic data. NPJ Syst. Biol. Appl. 7, 45.
97. Fiers, M.W.E.J., Minnoye, L., Aibar, S., Bravo González-Blas, C., Kalender Atak, Z., and Aerts, S. (2018). Mapping gene regulatory networks from single-cell omics data. Brief. Funct. Genomics 17, 246–254.
98. de Nadal, E., and Posas, F. (2022). The HOG pathway and the regulation of osmoadaptive responses in yeast. FEMS Yeast Res. 22, foac013.
99. Strebinger, D., Deluz, C., Friman, E.T., Govindan, S., Alber, A.B., and Suter, D.M. (2019). Endogenous fluctuations of OCT4 and SOX2 bias pluripotent cell fate decisions. Mol. Syst. Biol. 15, e9002.
100. Katz Y., Goodman N.D., Kersting K., Kemp C., Tenenbaum JB Modeling Semantic Cognition as Logical Dimensionality Reduction. Proceedings of the Annual Meeting of the Cognitive Science Society 30. <https://escholarship.org/uc/item/50r1c7qh>.
101. Katz, Y., and Springer, M. (2016). Probabilistic adaptation in changing microbial environments. PeerJ 4, e2716.
102. Emmons-Bell, M., Durant, F., Tung, A., Pietak, A., Miller, K., Kane, A., Martyniuk, C.J., Davidian, D., Morokuma, J., and Levin, M. (2019). Regenerative Adaptation to Electrochemical Perturbation in Planaria: A Molecular Analysis of Physiological Plasticity. iScience 22, 147–165.
103. Cervera, J., Pietak, A., Levin, M., and Mafe, S. (2018). Bioelectrical coupling in multicellular domains regulated by gap junctions: A conceptual approach. Bioelectrochemistry 123, 45–61.
104. Pietak, A., and Levin, M. (2017). Bioelectric gene and reaction networks: computational modelling of genetic, biochemical and bioelectrical dynamics in pattern regulation. J. R. Soc. Interface 14, 20170425.
105. Pietak, A., and Levin, M. (2016). Exploring Instructive Physiological Signaling with the Bioelectric Tissue Simulation Engine. Bioinforma Comput Biol 4, 55.
106. Zhao, E.M., Zhang, Y., Mehl, J., Park, H., Lalwani, M.A., Toettcher, J.E., and Avalos, J.L. (2018). Optogenetic regulation of engineered cellular metabolism for microbial chemical production. Nature 555, 683–687.

107. Toettcher, J.E., Weiner, O.D., and Lim, W.A. (2013). Using optogenetics to interrogate the dynamic control of signal transmission by the Ras/Erk module. *Cell* **155**, 1422–1434.
108. Toettcher, J.E., Gong, D., Lim, W.A., and Weiner, O.D. (2011). Light-based feedback for controlling intracellular signaling dynamics. *Nat. Methods* **8**, 837–839.
109. Bugaj, L.J., O'Donoghue, G.P., and Lim, W.A. (2017). Interrogating cellular perception and decision making with optogenetic tools. *J. Cell Biol.* **216**, 25–28.
110. Levin, M. (2023). Darwin's agential materials: evolutionary implications of multiscale competency in developmental biology. *Cell. Mol. Life Sci.* **80**, 142.
111. Davies, J., and Levin, M. (2023). Synthetic morphology with agential materials. *Nat. Rev. Bioeng.* **1**, 46–59.
112. Pezzulo, G., and Levin, M. (2016). Top-down models in biology: explanation and control of complex living systems above the molecular level. *J. R. Soc. Interface* **13**, 20160555.
113. Pezzulo, G., and Levin, M. (2015). Re-membering the body: applications of computational neuroscience to the top-down control of regeneration of limbs and other complex organs. *Integr. Biol.* **7**, 1487–1517.
114. Mathews, J., Chang, A.J., Devlin, L., and Levin, M. (2023). Cellular signaling pathways as plastic, proto-cognitive systems: Implications for biomedicine. *Patterns (N.Y.)* **4**, 100737.

STAR★METHODS

KEY RESOURCES TABLE

REAGENT or RESOURCE	SOURCE	IDENTIFIER
Deposited data		
MAPK Cancer Cell Fate Boolean GRN model	Cell Collective; Grieco et al. ⁵⁷	Model 7984: https://research.cellcollective.org/dashboard#module/7984:1/mapk-cancer-cell-fate-network/1
PI3K/AKT/mTor, RAS/ERK, and Wnt/β-catenin cross-regulatory network	This paper; Glaviano et al. ⁷⁴	https://github.com/betsee/cellnition
OCT4-SOX2-NANOG GRN network	This paper; Mossahbei-Mohammadi et al., ⁷³ Wang et al., ⁷⁹ Ogony et al. ⁸⁷	https://github.com/betsee/cellnition
Yeast Osmoadaptation Regulatory Network	This paper; Kuhn et al., ^{80,81} Hohman ⁸²	https://github.com/betsee/cellnition
Software and algorithms		
Cellnition Software Code	This paper	https://github.com/betsee/cellnition

METHOD DETAILS

General computational strategy

The core of each Regulatory Network Machine (RNM) is a dissipative dynamic system, which in our examples is taken to be a Gene Regulatory Network (GRN). Here we utilized computational models of GRNs, and trialed both continuous partial differential equation models (“Continuous Models”) and discrete Boolean network models (“Boolean Models”) of the GRNs. All our models were constructed in the Python language. First, both Continuous and Boolean GRN models were constructed using the symbolic mathematics package Sympy to create fully analytic computational models and, in the Continuous Model case, fully analytic derivatives were computed and used to generate Jacobians and Hessians for use in equilibrium state searches in the GRN model. In the case of the Continuous Models, where possible, these symbolic models were also simplified using analytical techniques to reduce the number of equations, and in some cases, to find complete solutions. For all GRN cases reported herein, except the MAPK Cancer Cell Fate GRN, both Continuous and Boolean models were utilized and the Continuous model results shown in all figures. The MAPK Cancer Cell Fate GRN was the only exception, which was computed using the Boolean method only.

In GRN images, a directed edge between two nodes symbolizes the influence of one node level on the dynamics of another, where the influence can be to activate (blue edge terminating in a circle) or inhibit (red edge terminating in a bar) the growth of the influenced node by modulating the growth term in the influenced node’s change rate function (see the growth term dynamics for H_0 in Figure S4 and S5).

GRN base functions

Continuous GRN models

Continuous GRN modeling conformed to well-known standards commonly used for continuous models of GRN.^{17,62} Descriptions of the parameters utilized in Continuous GRN models are summarized in Table S2. The level of each node of our GRNs represents the probability that the gene is expressed at the time point in question, and therefore, the level of each node spans from 0.0 to 1.0. The level of each node in the network changes in time according to a differential equation involving separate growth and decay terms (H_0 in Figures S4 and S5), where nodes without any influencing input are seen as inputs controlled by external factors (S_0 in Figures S4 and S5). Our GRN models allowed for the option to choose the use of a Hill function based dynamic equation (Figure S4), or a Logistic function based dynamic equation (Figure S5); all models presented in the paper utilized the Hill function model yet little difference was seen between the two function types. Default parameters for models utilized $d_i = 1.0$, $\beta_{ij} = 2.0$, and $n_{ij} = 3.0$ for all node expression probability change rate expressions in a network using the Hill function base, and $d_i = 1.0$, $\beta_{ij} = 0.5$, and $n_{ij} = 15.0$ for expressions using the Logistic function base.

Boolean GRN models

Boolean GRN modeling formalism conformed to well-known standards commonly used for Boolean models of GRN (ref). The level of each node of our Boolean GRNs was discretized and restricted to the values of 0 (False) or 1 (True).⁶² A Boolean GRN with N nodes has 2^N states (note these are not necessarily equilibrium states). Each computation of Boolean GRN state iterates from a previous state, where the next state of the network is calculated directly from the previous state according to logic equations at each network node. An activating interaction of one node (e.g. ‘S0’) upon another node (e.g. ‘H0’) is updated simply as $H0 = S0$, whereas an inhibiting interaction of one node (e.g. ‘S0’) upon another node (e.g. ‘H0’) is expressed in terms of the logical “NOT” operator as $H0 = \text{NOT}$

(S0) or, alternatively, as H0 = 1 – S0. As detailed in the next section, the influence of multiple nodes upon another node (e.g. if nodes ‘S0’ and ‘H0’ both influence ‘H1’), is accomplished using the logical operators “AND” and “OR” to combine multiple influences (e.g. H1 = S0 OR H0 would mean that H1 is activated by the presence of node S0 or by node H0).⁶²

Combining multiple regulatory interactions

In a GRN it’s possible for a single node to be influenced by multiple other nodes, and therefore a strategy must be implemented to combine the effects of multiple influences. A simple example network is shown in Figure S6, where nodes H0 and H1 each have two interactions acting upon them.

In a Continuous GRN model, multiple influences can be combined multiplicatively, which means to have all interaction terms multiply together in the influenced node’s growth term, as shown in Equation 1 for the network of Figure S6. In the Boolean case, this combination type is implemented using the “AND” operator.

$$\begin{bmatrix} \frac{d}{dt} H_0 \\ \frac{d}{dt} H_1 \\ \frac{d}{dt} S_0 \end{bmatrix} = \begin{bmatrix} -H_0 d_0 + \frac{d_0}{(1+(H_0\beta_1)^{-n_1})(1+(S_0\beta_3)^{-n_3})} \\ -H_1 d_1 + \frac{d_1}{(1+(H_1\beta_2)^{-n_2})(1+(H_0\beta_0)^{n_0})} \\ 0 \end{bmatrix} \quad (\text{Equation 1})$$

Alternatively, in a Continuous Model, interactions can be combined additively by adding terms together in the growth term of the influenced node, as shown in Equation 2 for the network of Figure S6. In the Boolean case, this corresponds to combining the terms using an “OR” operator.

$$\begin{bmatrix} \frac{d}{dt} H_0 \\ \frac{d}{dt} H_1 \\ \frac{d}{dt} S_0 \end{bmatrix} = \begin{bmatrix} -H_0 d_0 + \frac{d_0}{2} \left(\frac{1}{1+(2H_0\beta_1)^{-n_1}} + \frac{1}{1+(S_0\beta_3)^{-n_3}} \right) \\ -H_1 d_1 + \frac{d_1}{2} \left(\frac{1}{1+(2H_1\beta_0)^{n_0}} + \frac{1}{1+(2H_1\beta_2)^{-n_2}} \right) \\ 0 \end{bmatrix} \quad (\text{Equation 2})$$

Finally, a mixed scheme can be utilized, wherein inhibitive interactions can be combined in a multiplicative fashion (“AND” operator combination), while activating interactions can be combined in an additive fashion (“OR” operator combination), as shown in Equation 3 for the network of Figure S6.

$$\begin{bmatrix} \frac{d}{dt} H_0 \\ \frac{d}{dt} H_1 \\ \frac{d}{dt} S_0 \end{bmatrix} = \begin{bmatrix} -H_0 d_0 + \frac{d_0}{2} \left(\frac{1}{1+(2H_0\beta_1)^{-n_1}} + \frac{1}{1+(S_0\beta_3)^{-n_3}} \right) \\ -H_1 d_1 + \frac{d_1}{(1+(H_1\beta_2)^{-n_2})(1+(2H_0\beta_0)^{n_0})} \\ 0 \end{bmatrix} \quad (\text{Equation 3})$$

Throughout this work, all models presented utilized the mixed scheme for combining multiple interactions; however, multiplicative, additive, and mixed schemes were all subject to study.

State space equilibrium search and characterization

Our RNM framework requires, ideally all, equilibrium states of the state space to be identified and characterized. For both Continuous and Boolean GRN modeling, state space searches were conducted using a subset of the full network’s nodes, which were selected based on their hierarchical level in the network after excluding the input nodes.

The hierarchical level of a node in a regulatory network is a numerical measure of how much downstream influence the node has on other nodes in the network.⁵⁸ Input nodes have the highest hierarchical level in the network, are also nodes with zero in-degree and levels controlled by factors outside of the network and have the greatest influence on the dynamics of all other nodes of the network.⁵⁸ Conversely, output nodes have the lowest hierarchical level in the network, are nodes with zero out-degree and have no influence on the dynamics of all other nodes in the network.⁵⁸ We found that using only a subset of nodes with the highest hierarchical level (not including the input nodes) as the dimensions of the state space was able to return all the equilibrium states of the whole network. For example, in the 53 node GRN of the MAPK Cancer Cell Fate network model (Figure 2), we were able to return all equilibrium states of the network using only 12 nodes with the highest hierarchical level to form the dimensions of the searched state space (these nodes were ‘TGFBR’, ‘SMAD’, ‘TAK1’, ‘TAOK’, ‘ATM’, ‘FGFR3’, ‘GADD45’, ‘EGFR’, ‘GRB2’, ‘PLCG’, ‘PKC’, and ‘MTK1’, see Figure 2A for the whole GRN and color-coded node hierarchical level). This means the state space search required 2¹² = 4096 individual equilibrium search queries instead of 2⁴⁶ = 70x10¹² queries required if all internal nodes needed to be used to find all equilibrium states, representing an enormous improvement in computational efficiency and feasibility.

For the Continuous models, for each input state assigned to the input nodes, and subsequently, for each trial initial value from the state space search set applied to the remaining nodes of the network, Scipy's root finding algorithm, *fsolve*, was used to find an equilibrium point. A multi-level rounding scheme was used to round solutions to the nearest tenth (i.e. to 0.1), such that gene expressions that differed by more than 10% would be classified as unique solutions. After finding each equilibrium solution, the stability characteristic of each equilibrium was characterized using characteristics of the eigenvalues of the system's Jacobian calculated at the equilibrium point. [Table S3](#) describes how the sign and presence of nonzero imaginary components of the eigenvalues was used to characterize the stability of the equilibrium.

For Boolean models, through comparison with Continuous counterparts ([Figure S1](#) for an example) we determined that the stability characteristic of equilibrium states in Boolean models can be assessed based on the characteristic of repeated motifs when calculating a sequence of states from an initial starting state. We found that a single state repeating is equivalent to an asymptotically stable point attractor, that a motif that persistently cycles between two states is equivalent to a meta-stable saddle-point attractor, and that a motif that persistently cycles between more than two states is a limit cycle. In our Boolean models we did not find any attractive limit cycles (all were oscillating limit cycles) in any GRN models. For a Boolean model with a saddle-point or limit cycle attractor, the value of the equilibrium state was taken to be the average of all state values in the repeating motif, which produced results consistent with the Continuous model. We constructed a motif analyzer in our RNM software to automatically identify the motif and characterize the equilibrium state of the Boolean model.

Network Finite State Machine build procedure

After acquiring a set of unique, stability-characterized equilibrium states (referred to as the matrix M_{states}) using the above-described methods, along with a set of input states, in Continuous models, temporal simulation was employed to determine the state transition network of the NFSM. In Boolean models, a sequence of states was calculated and analyzed for each applied input and starting equilibrium state.

Whether working with a Continuous or Boolean model of a GRN, the G-NFSM and E-NFSM can be created from any GRN (or, more generally, any dissipative dynamic system) through the following five steps:

1. Identify the GRN (or dissipative dynamic system) of interest and construct a realistic computational model of the GRN (or of the dissipative dynamic system).
2. Identify the input nodes, internal nodes, and the output nodes which ideally relate to important biological phenomenon (e.g. cell proliferation, apoptosis, etc). We refer to individual input signals as "I#", where "#" represents an integer identifier.
3. For each applied input signal, I#, use computational methods (see [Supplementary Info S1](#)) to find and characterize all equilibrium states existing for each state sub-space associated with each input signal. We refer to individual equilibrium states as "State #", where "#" represents an integer identifier.
4. Create the edges of the NFSMs by following a methodology of input signal application and equilibrium state detection. Specifically, apply each input signal, I#, to the network (this initially applied input is called the "Held Context I#") and start the system off in each of the equilibrium states ("Initial State #") associated with the applied input signal's sub-space. Then, transiently change the applied input signal to be each member of the input signal set (these are each "Input Event I#"). Detect the equilibrium of the system while the Input Event I# is transiently applied ("Transient State #"). Finally, return the input signal back to the original Held Context I# and detect the equilibrium of the system ("Final State #"). The labeled edges of the G-NFSM are, specifically: Initial State # – *Input Event I#* → Transient State #, and Held State # – *Held Context I#* → Final State. The edges of the E-NFSM are recorded only for the case where the Initial State # is not the same as the Final State #, and are defined in a sub-graph indicating the Held Context I#, specifically: Initial State # – *Input Event I#* → Final State #, and describe a transition under the Held Context I# sub-graph.

STEPS FOR APPLYING RNM ANALYSIS TO BIOMEDICAL APPLICATIONS

To utilize our RNM framework in biomedical or biotechnological applications, the following steps can be followed.

1. Identify phenomenological biological outcomes represented by the different output node states. For example, in the MAPK Cancer Cell Fate GRN model (see [Figure 2](#)), we identified states without apoptosis and only proliferation to be indicative of cancer states, and other states that cycle through proliferation, cell cycle arrest, and apoptosis to be indicative of "normal" cell states ([Figure 2C](#)).
2. Categorize the system's equilibrium output states in terms of the ideal biological outcome states defined in Step 1. For the MAPK Cancer Cell Fate model, this was done by using the Euclidean distance from each equilibrium state and the idealized state to find the closest match ([Figure 2](#)).
3. Use path analysis on the G-NFSM to identify the input states that work to support each of the desired biological outcomes, or alternatively, that can transition the system from an undesirable to desirable state ([Figure 2](#)). These input states are potential therapeutics in the system.

- Examine the E-NFSM to determine if it is possible to move beyond “one constant dose” medicine and use transient application of an input intervention to achieve a permanent transition to a desired biological outcome equilibrium state.

Estimating the intelligence potential of regulatory networks

We developed a metric to estimate the intelligence potential (IP) of regulatory network studied in the RNM framework (Box 1). We define the IP of a network as the sum of two parameters: i) the *Multistate Ratio*, (MR) and ii) the *Context Ratio* (CR). The MR is defined as the total number of output equilibrium states, scaled by the number of input signals. The CR is defined as the total number of non-trivial input contexts (N Contexts) scaled by the number of input signals, where a non-trivial input context is defined as an input-specified state sub-space containing more than one equilibrium (i.e. a multistable state sub-space). A system with non-trivial input contexts is multistable and therefore capable of maintaining memory of state and emitting context-dependent behaviors.

By definition, a monostable network has at most one equilibrium per input context and can therefore have a maximum number of output states equal to the number of input states, for a maximum MR=1.0. In contrast, multistability allows multiple states to be present in any input context, leading to MRs greater than 1.0. The CR of a monostable network is 0.0 by definition as there are no multi-stable sub-spaces. Note that state degeneracy (the case where the same state is present in multiple input contexts) leads to MRs less than 1.0. A multistable system can have a maximum number of non-trivial input contexts equal to the number of input states, for a maximum CR of 1.0.

We studied several network topologies, calculated their average IP, and examined the IP as a function of different network properties (see Table 1). We found that the intelligence potential of a regulatory network is loosely related to the number of *network cycles* and the *hierarchical incoherence* (Box 1, ref.⁹⁴) of the regulatory network, but unrelated to the total number of nodes, edges or the maximum in-degree of nodes in the network. The form of networks included in this analysis are shown in Figure S7, and analysis was performed on 15 unique networks of each class, each with different randomly generated edge interaction types (e.g. different unique combinations of activators and inhibitors).

Directed acyclic GRNs: Evidence for Guaranteed Mono-stability and fully analytical solutions

As GRNs were constructed using symbolic mathematics, it was possible to use computer algebra to look for simplifications to the set of differential equations (i.e. dimensionality reduction), which in some cases lead to the determination of fully analytical solutions. It was found that when using the Hill function base (example shown in Figure S4), any regulatory network that is free of cycles (i.e. networks that are Directed Acyclic Graphs, DAGs) have a single, fully analytical solution determining the steady-state value of each node in terms of input node values. No analytical solutions were found when using the Logistic function base. An example of a network with a complete analytical solution is shown in Figure S8, where the steady-state (equilibrium) expression probability of the two nodes ‘p0’ and ‘p1’ are uniquely specified by the level of input node ‘p2’ and the set of interaction edge parameters β_{ij} and n_{ij} for the network parameterized using Hill functions. For any value of parameters and the state of the input node ‘p2’, there is only one unique solution. We found this to be the case for all DAGs studied.

Osmoadaptation model mathematics

Here we detail our computational model of yeast osmoadaptation. Our model is comprised of a physical component that describes the dynamic change of volume of a cell with water movements mediated by transmembrane osmotic pressure gradients, which can be applied to a variety of cell types including inanimate vesicles, plant cells, and mammalian cells. A second component of our model is a regulatory scheme based on yeast HOG-MAPK osmoadaptation, which interacts with the physical osmotic volume increase to help the cell maintain its volume at a set-point value against stochastic changes in environmental osmolarity. All parameters and variables of our model are summarized in Table S4. In the following we derive our complete osmoadaptation model from first principles.

We begin by defining the osmolarity inside the cell (m_i) in terms of the total moles of osmolytes in the cell (n_i):

$$m_i = \frac{n_i}{vol_{cell}} \quad (\text{Equation 4})$$

The transmembrane osmotic pressure gradient (P_{osmo}) can be defined in terms of the difference in osmolarity between the inside and outside (m_o) of the cell. Note we assume that the osmolarity outside:

$$P_{osmo} = RT(m_i - m_o) \quad (\text{Equation 5})$$

In our model, the cell volume will change with osmotic pressure driven transmembrane water flow. Therefore, we wish to write the osmotic pressure gradient in terms of the cell volume variable by substituting Equations 4 into 5:

$$P_{osmo} = RT \left(\frac{n_i}{vol_{cell}} - m_o \right) \quad (\text{Equation 6})$$

First we consider the case where the present cell volume (vol_{cell}) is less than the undeformed cell volume ($\text{vol}_{\text{cello}}$) and where the cell is wrinkling with volume loss and therefore does not encounter mechanical resistance to volume changes. In this case where $\text{vol}_{\text{cell}} < \text{vol}_{\text{cello}}$, osmotic transmembrane water flow into the cell occurs via flow channels and can be described by:

$$u_{io} = \frac{A_{\text{chan}} N_{\text{chan}} P_{\text{osmo}}}{8\mu d_{\text{mem}}} \quad (\text{Equation 7})$$

Substituting in Equations 6 and 7 we obtain an expression for transmembrane water flow in terms of osmolyte gradients and the present cell volume:

$$u_{io} = \frac{A_{\text{chan}} N_{\text{chan}} RT \left(\frac{n_i}{\text{vol}_{\text{cell}}} - m_0 \right)}{8\mu d_{\text{mem}}} \quad (\text{Equation 8})$$

Multiplying the transmembrane linear flow rate by the area over which water flows (which is equal to the number of water channels multiplied by the cross-sectional area of a water channel) we obtain a volume flow rate, which is equivalent to the rate of cell volume change with transmembrane osmotic water flow:

$$\frac{d}{dt} \text{vol}_{\text{cell}}(t) = A_{\text{chan}} N_{\text{chan}} u_{io} \quad (\text{Equation 9})$$

Substituting in Equations 8 and 9 we obtain the differential equation for the case where $\text{vol}_{\text{cell}} < \text{vol}_{\text{cello}}$:

$$\frac{d}{dt} \text{vol}_{\text{cell}}(t) = \frac{A_{\text{chan}}^2 N_{\text{chan}}^2 RT (n_i - m_0 \text{vol}_{\text{cell}})}{8\mu d_{\text{mem}} \text{vol}_{\text{cell}}} \quad (\text{Equation 10})$$

Steady-state volume for the case where $\text{vol}_{\text{cell}} < \text{vol}_{\text{cello}}$ can be found by solving Equation 10 for vol_{cell} when the vol_{cell} change rate is equal to zero, which yields:

$$\text{vol}_{\text{cell}} = \frac{m_i \text{vol}_{\text{cello}}}{m_0} \quad (\text{Equation 11})$$

For the regime where $\text{vol}_{\text{cell}} >= \text{vol}_{\text{cello}}$ and the cell is changing volume from an elastically deformed state that results in mechanical resistance to stretch the membrane or cell wall as volume expands, we need to consider that the mechanical structure of the cell wall or membrane may resist deforming under an osmotic pressure due to the generation of a mechanical pressure. Hoop stress of a cylindrically shaped cell is one way to account for these mechanical factors, where P_{ind} is the pressure inducing the stress.

In the circumferential direction hoop stress of a cylindrically shaped cell is described by:

$$\sigma_H = \frac{P_{\text{ind}} r_{\text{cello}}}{d_{\text{mem}}} \quad (\text{Equation 12})$$

In the longitudinal (axial) direction hoop stress is described by:

$$\sigma_L = \frac{P_{\text{ind}} r_{\text{cello}}}{2d_{\text{mem}}} \quad (\text{Equation 13})$$

Hoop strain in the circumferential direction is described by:

$$\varepsilon_H = \frac{\sigma_H - \nu \sigma_L}{Y} \quad (\text{Equation 14})$$

While hoop strain in the longitudinal direction is described by:

$$\varepsilon_L = \frac{\sigma_L - \nu \sigma_H}{Y} \quad (\text{Equation 15})$$

Substituting in Equations 12 and 13 into 14 we obtain:

$$\varepsilon_H = \frac{P_{\text{ind}} r_{\text{cello}} (1 - \nu/2)}{Y d_{\text{mem}}} \quad (\text{Equation 16})$$

And substituting in Equations 12 and 13 into 15 we obtain:

$$\varepsilon_L = \frac{P_{\text{ind}} r_{\text{cello}} (1/2 - \nu)}{Y d_{\text{mem}}} \quad (\text{Equation 17})$$

Displacement in the circumferential direction is represented by:

$$d_{H_S} = \varepsilon_H r_{cell0} \quad (\text{Equation 18})$$

And displacement in the longitudinal/axial direction is represented by:

$$d_{L_S} = \varepsilon_L L_{cell0} \quad (\text{Equation 19})$$

An expression for the volume of the cell in terms of the unstrained volume and the circumferential and longitudinal strains is:

$$vol_{cell} = vol_{cell0}(\varepsilon_H + 1)(\varepsilon_L + 1) \quad (\text{Equation 20})$$

Substituting in Equations 16 and 17 into 20 we obtain:

$$vol_{cell} = vol_{cell0} \left(\frac{P_{ind} r_{cell0} (1/2 - \nu)}{Y d_{mem}} + 1 \right) \left(\frac{P_{ind} r_{cell0} (1 - \nu/2)}{Y d_{mem}} + 1 \right) \quad (\text{Equation 21})$$

For simplicity, if we assume the Poisson's ratio is $\nu=0.5$, then:

$$vol_{cell} = vol_{cell0} \left(\frac{3P_{ind} r_{cell0}}{4Y d_{mem}} + 1 \right) \quad (\text{Equation 22})$$

Solving Equation 22 for the inducing pressure, P_{ind} :

$$P_{ind} = \frac{4d_{mem}(vol_{cell} - vol_{cell0})}{3r_{cell0} vol_{cell0}} \quad (\text{Equation 23})$$

In the case where the cell is expanding against structural resistance ($vol_{cell} \geq vol_{cell0}$), the structural pressure P_{ind} (analogous to Turgor pressure) provides a resistance to the osmotic pressure P_{osmo} , possibly limiting osmotic water influx as the cell pressurizes with turgor pressure. Mathematically, this is represented as:

$$u_{io} = \frac{A_{chan} N_{chan} (P_{osmo} - P_{ind})}{8\mu d_{mem}} \quad (\text{Equation 24})$$

Substituting in Equations 23 and 6 into 24, and then substituting the resulting expression for water influx into Equation 9, we obtain the differential equation for the case where $vol_{cell} \geq vol_{cell0}$:

$$\frac{d}{dt} vol_{cell}(t) = \frac{A_{chan}^2 N_{chan}^2 \left(RT \left(\frac{n_i}{vol_{cell}} - m_0 \right) - \frac{4Yd_{mem}(vol_{cell} - vol_{cell0})}{3r_{cell0} vol_{cell0}} \right)}{8\mu d_{mem}} \quad (\text{Equation 25})$$

Solving Equation 25 for the steady state vol_{cell} , which occurs when the change in cell volume is equal to zero, provides the expression:

$$vol_{cell} = \frac{\sqrt{vol_{cell0}} \sqrt{9R^2 T^2 m_0^2 r_{cell0}^2 vol_{cell0} - 24RTY d_{mem} m_0 r_{cell0} vol_{cell0} + 48RTY d_{mem} n_i r_{cell0} + 16Y^2 d_{mem}^2 vol_{cell0}^2} - vol_{cell} (3RTm_0 r_{cell0} - 4Yd_{mem})}{8Yd_{mem}} \quad (\text{Equation 26})$$

The above derivations provide us with cell volume change expressions for a system without an osmoadaptive regulatory network for two regimes: 1) the $vol_{cell} < vol_{cell0}$ regime occurring when the osmotic gradient favors water efflux from the cell to the environment, the cell volume is assumed to shrink without mechanical resistance to decreasing volume and is described by Equation 10, and 2) the $vol_{cell} \geq vol_{cell0}$ regime occurring when the osmotic gradient favors water influx from the environment to the cell, the cell volume is assumed to expand with mechanical resistance created by the properties of the cell membrane or wall which induces mechanical pressure that opposes further water influx, and is described by Equation 25. Steady-state cell volumes for cells with different perimeters (e.g. plant cell wall, yeast cell wall, and mammalian plasma membrane) are shown for models without osmoadaptation in Figure S9.

The final component of our osmoadaptation model incorporated bio-realistic regulatory network-based control components into the osmotic volume change through implementation of a simplified yeast HOG-MAPK pathway.

Firstly, the moles of osmolytes inside the osmoadapted cell was assumed to be the combination of a fixed base level plus the moles of glycerol in the cell:

$$n_i(t) = n_{base} + n_{gly}(t) \quad (\text{Equation 27})$$

The molarity (i.e. moles per liter) of glycerol in the cell was taken to be the product of a decay and growth term, where glycerol decay is mediated by membrane strain activation of glycerol efflux transporters (parameterized by K_1 , b_1 , and ε_{o1} , see Table S4) while

glycerol production is activated by membrane strain via the SLN1 strain sensor activation of the HOG-MAPK pathway (parameterized by K_2 , b_2 , and ε_{o2} , see [Table S4](#)):

$$\frac{d}{dt}m_{gly}(t) = r_{gly} \left(1 - \frac{1 - b_2}{1 + e^{-K_2 \left(\frac{\varepsilon_{o2} + vol_{cell}}{vol_{cell0}} - 1 \right)}} \right) - d_{gly} m_{gly} \left(b_1 + \frac{1 - b_1}{1 + e^{-K_1 \left(\frac{\varepsilon_{o1} + vol_{cell}}{vol_{cell0}} - 1 \right)}} \right) \quad (\text{Equation 28})$$

The response curves for the growth and decay of glycerol production with membrane strain are shown in [Figure S10](#), while the ability of a cell to maintain its volume near a setpoint value against an osmotic challenge is shown in [Figure S11](#).

ADDITIONAL RESOURCES

The methods and results presented in this manuscript can be reproduced and further investigated using the software Application Programming Interface (API) developed for this work available from:

<https://github.com/betsee/cellnition>.

With tutorials demonstrating use of the software API in the creation of NFSMs available from:

<https://github.com/betsee/cellnition?tab=readme-ov-file#tutorials>.

And further documentation for the software API available from:

<https://betsee.github.io/cellnition/>.