

Reprogramming multistable monotone systems with application to cell fate control

Rushina Shah^{1*}, Domitilla Del Vecchio¹

¹ Department of Mechanical Engineering, Massachusetts Institute of Technology, Cambridge, Massachusetts, USA * rushina@mit.edu

ABSTRACT

Multistability is a key property of dynamical systems modeling cellular regulatory networks implicated in cell fate decisions, where, different stable steady states usually represent distinct cell phenotypes. Monotone network motifs are highly represented in these regulatory networks. In this paper, we leverage the properties of monotone dynamical systems to provide theoretical results that guide the selection of inputs that trigger a transition, i.e., reprogram the network, to a desired stable steady state. We first show that monotone dynamical systems with bounded trajectories admit a minimum and a maximum stable steady state. Then, we provide input choices that are guaranteed to reprogram the system to these extreme steady states. For intermediate states, we provide an input space that is guaranteed to contain an input that reprograms the system to the desired state. We then provide implementation guidelines for finite-time procedures that search this space for such an input, along with rules to prune parts of the space during search. We demonstrate these results on simulations of two recurrent regulatory network motifs: self-activation within mutual antagonism and self-activation within mutual cooperation. Our results depend uniquely on the structure of the network and are independent of specific parameter values.

I. INTRODUCTION

¹Multistability is encountered in several models of cell differentiation and development. In particular, core gene regulatory networks (GRNs) that control cell fate decisions are traditionally modeled as multistable dynamical systems, with stable steady states representing cell phenotypes [2]-[9]. Under this framework, cell differentiation, which is the process by which cells convert from one type to another, can be viewed as the state of the dynamical system moving from one stable steady state to another. Artificially converting cells from one phenotype to another using external inputs typically involves up- or down-regulating the transcription factors of these core gene regulatory networks. Which factors to regulate, as well as whether to up- or down-regulate them, is typically decided via experimental trial-and-error [10]-[13], and often, the efficiency of the cell conversion process is quite low (about 1% for induced pluripotent stem cell reprogramming [14]). It has

been theoretically shown in the case of reprogramming to pluripotent stem cells, that “intuitive” inputs, such as up-regulating factors that are higher in the target state and down-regulating those that are lower in the target state, might be ineffective [2]. Theoretical and computational work that could guide this experimental process could therefore be crucial for reducing the time and number of experiments, as well as for increasing the efficiency of inducing cell fate transitions.

Several studies have analyzed multistable dynamical models of GRNs controlling cell fate decisions. In [2], [6]-[8], specific GRNs are analyzed and recommendations for reprogramming these systems are made. In [15], [16], boolean models of GRNs are analyzed computationally to identify the key transcription factors that, when perturbed, destabilize the undesired steady state and induce a transition to the desired steady state. In [3], parameter regimes for two-node motifs of mutual antagonism and mutual cooperation are found that result in bistability and multistability. These works either rely on a specific choice of biologically reasonable parameters, or on computationally sampling parameters for the network. However, many GRNs responsible for cell-fate decisions belong to the class of monotone dynamical systems for which there is rich theoretical work [17]-[19]. This work could be leveraged to make more general recommendations for reprogramming strategies, without relying on brute-force search in parameter space. The classical works on monotone systems [17], [18] present results on the stability and limit-sets of these systems, among others. The work in [19] extends this theory for the case of controlled monotone systems. In [20]-[22], multistable systems with positive feedback are analyzed, such that locations and stability of steady states can be found using the input-output characteristic of the resulting monotone system when the feedback loop is open.

In this paper, we leverage these theoretical results to provide parameter-independent strategies for choosing inputs to reprogram monotone systems. In Section II, we present motivating examples of network motifs that are highly represented in GRNs that control cell-fate decisions and can be modeled as monotone dynamical systems. In Section III, we summarize some key results from the literature on monotone systems, and provide a formal definition of reprogramming. In Section IV, we show that the set of stable steady states of monotone systems must have a minimum and a maximum. We then show that, based on the graphical structure of the network, we can determine which nodes must receive a positive input and which must receive a negative input, so that for large enough inputs of this type, the system is guaranteed to be reprogrammed to extremal states. Next, in Section V, we provide results

¹A preliminary version of this work can be found in [1].

for the remaining non-extremal (intermediate) stable steady states, to eliminate inputs that are guaranteed not to reprogram the system to the intermediate stable steady states. Then, we present an input space that, for any desired stable steady state, is guaranteed to contain an input that reprograms the system to that state. We present guidelines to design finite-time search procedures to search for such an input, and apply our theoretical results to prune the input space while searching. We demonstrate these results using two ubiquitous network motifs which control cell fate at several points along the cell development process. In Section VI, we apply these results to a 3-node network motif, and demonstrate how the system can be reprogrammed to the desired stable steady state using these strategies. Section VII presents the conclusions and directions for future work.

II. MOTIVATING EXAMPLES

As motivating examples, we consider three network motifs, which are highly represented in gene regulatory networks (GRNs) involved in cell fate determination. The first is a motif where two nodes are each self-activating while mutually repressing each other (Fig. 1a). The second motif is one where two nodes are self-activating while also activating each other (Fig. 1b), and the third motif is one with three nodes that are self-activating while also activating each other (Fig. 1c). The first network motif is found in gene regulatory networks controlling lineage specification of hematopoietic stem cells [23]-[34]. The second and third motifs are found in gene regulatory networks involved in maintenance of the pluripotent stem cell state [35]-[41].

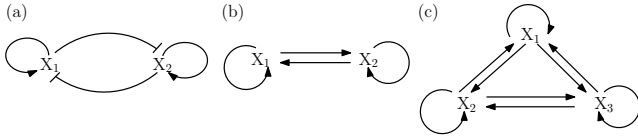


Fig. 1: Examples of network motifs found in GRNs involved in cell fate determination [23]-[41]. Here, arrows “ \rightarrow ” represent activation (positive interaction) and arrows “ \dashv ” represent repression (negative interaction). (a) Mutual antagonism network, where two nodes mutually repress one another while self-activating. (b) Two-node mutual cooperation network, where two nodes mutually activate one another while also self-activating. (c) Three-node mutual cooperation network, where each node activates the others and itself.

The dynamics of these network motifs have been commonly modeled through ordinary differential equations (ODEs) as follows:

$$\dot{x}_i = h_i(x) - \gamma_i x_i + q(x_i, w_i), \quad (1)$$

with $x = (x_1, \dots, x_n) \in X \subset \mathbb{R}_+^n$ representing the state of the network, and x_i the concentration of protein x_i . The regulatory function $h_i(x)$, called Hill-function [42], [43], [44], captures the effect of the proteins x_1, \dots, x_n of the network on the production rate of protein x_i . If protein x_j activates (represses) x_i , that is $x_j \rightarrow x_i$ ($x_j \dashv x_i$), then $h_i(x)$ increases (decreases) with x_j . Further, $h_i(x)$ is strictly positive and bounded from above for all x . The constant γ_i is the decay rate constant due to degradation or dilution. Here, $w_i = (u_i, v_i) \in \mathbb{R}_+^2$ is

an external input stimulation that can be applied during the reprogramming process, with $q(x_i, w_i) = u_i - v_i x_i$, where u_i is a positive stimulation achieved by over-expression of protein x_i [45], [46], and v_i is a negative stimulation achieved by enhanced degradation of protein x_i [2]. For example, for the mutual antagonism network motif of Fig. 1a, we have $n = 2$, with

$$\begin{aligned} h_1(x) &= \frac{\beta_1 + \alpha_1 (x_1/k_1)^{n_1}}{1 + (x_1/k_1)^{n_1} + (x_2/k_2)^{n_2}}, \\ h_2(x) &= \frac{\beta_2 + \alpha_2 (x_2/k_3)^{n_3}}{1 + (x_2/k_3)^{n_3} + (x_1/k_4)^{n_4}}, \end{aligned} \quad (2)$$

indicating that the production rate of x_1 increases with x_1 and decreases with x_2 , and the production rate of x_2 increases with x_2 and decreases with x_1 . Here, β_1, β_2 are the rate constants of leaky expressions of the species, α_1, α_2 are the activation rate constants, k_1, k_2, k_3, k_4 are the apparent dissociation constants, and n_1, n_2, n_3, n_4 are the Hill function coefficients. The nullclines and steady states for this system for a particular set of parameters and no input ($w = \mathbf{0}$) are shown in Fig. 2a. For these parameters, the system has three stable steady states S_1, S_2 and S_3 . When this system models the hematopoietic stem cell network, x_1 and x_2 represent the concentrations of proteins PU.1 and GATA1, the steady state S_2 represents the hematopoietic stem cell state, whereas S_1 and S_3 represent more specialized states, namely, the erythrocyte and myeloid cell lineages, respectively [23]-[34].

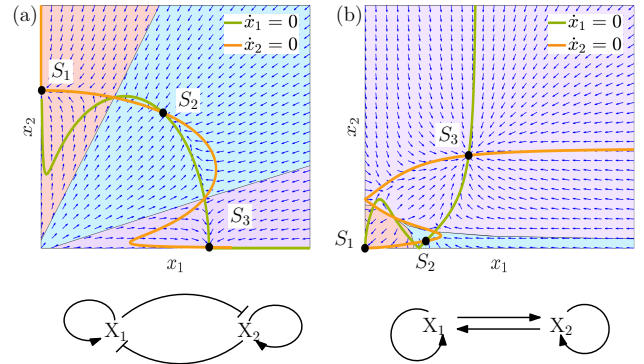


Fig. 2: Nullclines of the ODEs modeling the two-node network motifs given by (1), (2), (3) when unstimulated ($w = \mathbf{0}$). Filled circles represent the stable steady states S_1, S_2 and S_3 . Blue arrows show vector field (\dot{x}_1, \dot{x}_2) . Regions of attraction of the stable steady states S_1, S_2 and S_3 are shown by the coral, blue and purple shaded regions, respectively. (a) Nullclines for the system of ODEs modeling the mutual antagonism network, with $h_i(x)$ given by (2). Parameters: $\alpha_1 = 5\text{nMs}^{-1}$, $\beta_1 = 3\text{nMs}^{-1}$, $\alpha_2 = 5.2\text{nMs}^{-1}$, $\beta_2 = 4\text{nMs}^{-1}$, $n_1 = n_2 = n_3 = n_4 = 2$, $k_1 = k_2 = 1\text{nM}$, $\gamma_1 = \gamma_2 = 0.2\text{s}^{-1}$. (b) Nullclines for the system of ODEs modeling the two-node mutual cooperation network, with $h_i(x)$ given by (3). Parameters: $\eta_1 = \eta_2 = 10^{-4}\text{nMs}^{-1}$, $a_1 = 2\text{nMs}^{-1}$, $b_1 = 0.25\text{nMs}^{-1}$, $c_1 = 2.5\text{nMs}^{-1}$, $a_2 = 0.18\text{nMs}^{-1}$, $b_2 = 2\text{nMs}^{-1}$, $c_2 = 2.5\text{nMs}^{-1}$, $n_1 = n_2 = n_3 = 2$, $k_1 = k_2 = k_3 = 1\text{nM}$, $\gamma_1 = \gamma_2 = 1\text{s}^{-1}$.

For the two-node mutual cooperation network motif of Fig. 1b, we have $n = 2$,

$$\begin{aligned} h_1(x) &= \frac{\eta_1 + a_1 (x_1/k_1)^{n_1} + b_1 (x_2/k_2)^{n_2} + c_1 (x_1 x_2 / k_3^2)^{n_3}}{1 + (x_1/k_1)^{n_1} + (x_2/k_2)^{n_2} + (x_1 x_2 / k_3^2)^{n_3}}, \\ h_2(x) &= \frac{\eta_2 + a_2 (x_1/k_1)^{n_1} + b_2 (x_2/k_2)^{n_2} + c_2 (x_1 x_2 / k_3^2)^{n_3}}{1 + (x_1/k_1)^{n_1} + (x_2/k_2)^{n_2} + (x_1 x_2 / k_3^2)^{n_3}}, \end{aligned} \quad (3)$$

so that the production rates of x_1 and x_2 both increase with x_1 , and with x_2 . The nullclines and steady states for this system for a particular set of parameters and no input ($w = \mathbf{0}$ in (1)) are shown in Fig. 2b. For these parameter values, this motif results in three stable steady states S_1, S_2 and S_3 . When this system is used to model the pluripotency network, x_1 and x_2 can be regarded as the concentrations of the Nanog protein and the Oct4-Sox2 hetero-dimer, respectively, the state S_2 represents the undifferentiated pluripotent state, S_1 corresponds to the trophectoderm state and S_3 corresponds to the primitive endoderm state [35]-[41].

The biological problem of reprogramming a cell to a desired phenotype or cell state can be formulated as the mathematical problem of finding an input $w = (w_1, \dots, w_n)$ that, when applied transiently, can trigger a transition in the state of system (1) to a desired stable steady state. We will refer to this mathematical problem as “reprogramming” in the rest of the paper. To reprogram a multi-stable system to a desired stable steady state, say S_d , we use transient input simulations as follows. We apply a constant input $w = (w_1, \dots, w_n)$, if it exists, such that the trajectory of the stimulated system enters the region of attraction of S_d . After the trajectory has entered the region of attraction of the target state S_d , the stimulation is removed, and the trajectory of the unstimulated system converges to S_d so that the system is reprogrammed to it.

In this paper, leveraging the monotone nature of the systems’ dynamics, we provide sufficient conditions for the existence of such constant inputs and a finite-time search procedure to find them. While for 2D systems with known parameters, nullcline analysis can address this question, when system dimension is higher or parameters are not known, a more general approach is required. Our approach relies only on structural information and does not require parameter values.

III. SYSTEM DESCRIPTION AND PROBLEM DEFINITION

In this section, we provide the definition and some key properties of monotone dynamical systems, and mathematically define the problem of reprogramming.

A. Background: Monotone systems

First, we provide some notations and definitions for monotone systems, and then summarize properties of such systems that we leverage in this paper.

Definition 1. A partial order “ \leq ” on a set S is a binary relation that is reflexive, antisymmetric, and transitive. That is, for all $a, b, c \in S$, the following are true:

- (i) Reflexivity: $a \leq a$.
- (ii) Antisymmetry: $a \leq b$ and $b \leq a$ implies that $a = b$.
- (iii) Transitivity: $a \leq b$ and $b \leq c$ implies that $a \leq c$.

Example. On the set $S = \mathbb{R}^n$, the following are partial orders:

- (i) $x \leq y$ if $x_i \leq y_i$ for all $i \in \{1, \dots, n\}$.
- (ii) $x \leq y$ if $x_i \leq y_i$ for $i \in I_1$ and $x_j \geq y_j$ for $j \in I_2$, where $I_1 \cup I_2 = \{1, \dots, n\}$.

To more easily represent partial orders, we introduce the following notation from [17]. Let $m = (m_1, m_2, \dots, m_n)$, where $m_i \in \{0, 1\}$, and

$$K_m = \{x \in \mathbb{R}^n : (-1)^{m_i} x_i \geq 0, 1 \leq i \leq n\}.$$

K_m is an orthant in \mathbb{R}^n , and generates the partial order \leq_m defined by $x \leq_m y$ if and only if $y - x \in K_m$. We write $x <_m y$ when $x \leq_m y$ and $x \neq y$, and $x \ll_m y$ when $x \leq_m y$ and $x_i \neq y_i, \forall i \in \{1, \dots, n\}$.

We consider the system Σ_w of the form:

$$\dot{x} = f(x, w), \quad (4)$$

where $x \in X \subset \mathbb{R}_+^n$ and $w \in W \subset \mathbb{R}_+^{2n}$. Let the flow of system Σ_w starting from $x = x^0$ be denoted by $\phi_w(t, x^0)$ for $t \in \mathbb{R}$, with $\phi_{wi}(t, x^0)$ being its i^{th} component. If there exists a sequence of points $\{t_n\}$ such that $\lim_{n \rightarrow \infty} t_n \rightarrow \infty$ and $\lim_{n \rightarrow \infty} \phi_w(t_n, x^0) = x'$ for some $x^0 \in X$, we call x' an ω -limit point of Σ_w . The set of all ω -limit points of Σ_w for a given x^0 is represented by $\omega_w(x^0)$. The flow of the system with $w = 0$ is denoted by $\phi_0(t, x^0)$. A domain X is said to be p_m -convex if $ax + (1 - a)y \in X$ whenever $x, y \in X$, $0 < a < 1$, and $x \leq_m y$ [17]. Then, we give the following main definition of this paper, adapted from [17] for f dependent on a parameter w .

Definition 2. System Σ_w is said to be a *monotone system* with respect to K_m if domain X is p_m -convex and

$$(-1)^{m_i + m_j} \frac{\partial f_i}{\partial x_j}(x, w) \geq 0, \forall i \neq j, \forall x \in X, \forall w \in W \cup \{0\}. \quad (5)$$

A monotone system can be recognized by its graphical structure. Consider a graph G , whose nodes correspond to the states of the system, and two nodes $i \neq j$ are connected by an edge only if at least one of $\frac{\partial f_i}{\partial x_j}, \frac{\partial f_j}{\partial x_i}$ has a non-zero value somewhere in X . We say Σ_w is *sign-stable* if $\frac{\partial f_i}{\partial x_j}$ for all $i \neq j$ has the same sign for all $x \in X$ and all $w \in W \cup \{0\}$, and *sign-symmetric* if $\frac{\partial f_i}{\partial x_j} \frac{\partial f_j}{\partial x_i} \geq 0$, for all i, j and for all $x \in X$ and $w \in W \cup \{0\}$. Then, an edge between nodes i, j is a *positive edge* if $\frac{\partial f_i}{\partial x_j} \geq 0$ and $\frac{\partial f_j}{\partial x_i} \geq 0$, and is a *negative edge* if $\frac{\partial f_i}{\partial x_j} \leq 0$ and $\frac{\partial f_j}{\partial x_i} \leq 0$. Then Σ_w is monotone in X if and only if for every closed loop in G , the number of negative edges is even [17]. For the networks considered in Section II, activation edges between two nodes are positive edges, and repression edges between two nodes are negative edges. The network of mutual antagonism and the 2-node and 3-node networks of mutual cooperation are sign-symmetric and sign-stable. The graph G constructed for the two-node networks as described above has no closed loops (with a negative edge connecting the two nodes for the mutually antagonistic network and a positive edge connecting the two nodes for the mutually cooperative network), and therefore these networks are monotone dynamical systems. The two-node network of mutual cooperation is a monotone system with respect to the partial order $m = (0, 0)$, and the two-node network of mutual antagonism is a monotone system with respect to the partial order $m = (0, 1)$.

For convenience, we include Proposition 5.1 from [17] here, adapted for f dependent on a parameter w , stated as a Lemma. This lemma states that the flow of a monotone system preserves the ordering on the initial condition.

Lemma 1. [17] Consider system Σ_w . Let X be p_m -convex and f be a continuously differentiable vector field on X such that (5) holds. Let $<_r$ denote any one of the relations \leq_m , $<_m$, \ll_m . If $x <_r y$, $t > 0$ and $\phi_w(t, x)$ and $\phi_w(t, y)$ are defined, then $\phi_w(t, x) <_r \phi_w(t, y)$ for all $w \in W \cup \{0\}$.

B. Problem definition

We consider a dynamical system Σ_w of the form (4), where state $x \in X \subseteq \mathbb{R}_+^{n \times 2}$ and constant input $w \in \mathbb{R}_+^{n \times 2}$. Subscripts denote indices, so that $x = [x_1, x_2, \dots, x_n]$, $f = [f_1, f_2, \dots, f_n]$, and $w = [w_1, \dots, w_n]$. For every $i \in \{1, \dots, n\}$, $w_i \in \mathbb{R}_+^2$ such that $w_i = (u_i, v_i)$, where input u_i is the positive stimulation on state i , i.e., for all $x \in X, w \in W$ $\frac{\partial f_i(x, w)}{\partial u_i} \geq 0$ and $\frac{\partial f_j(x, w)}{\partial u_i} = 0$ for all $j \neq i$; and v_i is the negative stimulation on state i , i.e., for all $x \in X, w \in W$ $\frac{\partial f_i(x, w)}{\partial v_i} \leq 0$ and $\frac{\partial f_j(x, w)}{\partial v_i} = 0$ for all $j \neq i$.

The unstimulated system is $\Sigma_0 : \dot{x} = f(x, 0)$. Let the set of stable steady states of Σ_0 be \mathbb{S} , and let there be p such isolated stable steady states. For every $S \in \mathbb{S}$, let $\mathcal{R}_0(S)$ denote its region of attraction, i.e., $\mathcal{R}_0(S) := \{x_0 \in X \mid \lim_{t \rightarrow \infty} \phi_0(t, x_0) = S\}$. We denote the i^{th} component of steady state S by S_i .

We wish to reprogram the system Σ_0 to a desired stable steady state $S_d \in \mathbb{S}$, i.e., find constant input w_d such that the trajectory of system Σ_{w_d} converges inside $\mathcal{R}_0(S_d)$. Then, once the input is removed, the trajectories of system Σ_0 starting from inside $\mathcal{R}_0(S_d)$ converge to the desired stable steady state S_d , so that Σ_0 is reprogrammed to S_d . We formally define two concepts of reprogramming, depending on the set of initial conditions starting at which Σ_0 can be reprogrammed to the desired stable steady state S_d .

Definition 3. System Σ_0 is said to be *strongly reprogrammable* to state $S_d \in \mathbb{S}$ by input w_d if w_d is such that $\omega_{w_d}(x_0) \subseteq \mathcal{R}_0(S_d)$, for all $x^0 \in X$.

Definition 4. System Σ_0 is said to be *weakly reprogrammable* from some state $x^0 \in X$ to state $S_d \in \mathbb{S}$ by input w_d if w_d is such that $\omega_{w_d}(x_0) \subseteq \mathcal{R}_0(S_d)$.

In summary, *weak reprogrammability* deals with reprogramming a system from a given initial state x^0 to a desired stable steady state S_d , as in Fig. 3, where the stimulated system Σ_w has an asymptotically stable steady state $x_d \in \mathcal{R}_0(S_d)$ such that $\lim_{t \rightarrow \infty} \phi_{w_d}(t, x^0) = x_d$. *Strong reprogrammability* deals with reprogramming the system from any initial state to the desired stable steady state S_d , such as would be the case if x_d in Fig. 3 were a globally asymptotically stable steady state of Σ_w , such that the trajectory of Σ_w from any $x^0 \in X$ would converge to x_d .

We make the following assumptions on system Σ_w :

Assumption 1. The function $f(x, w)$ takes the form $f(x, w) = [f_1(x, w_1), f_2(x, w_2), \dots, f_n(x, w_n)]$ such that $f_i(x, w_i = (u_i, v_i)) = h_i(x) - \gamma_i x_i + u_i - v_i x_i$, where

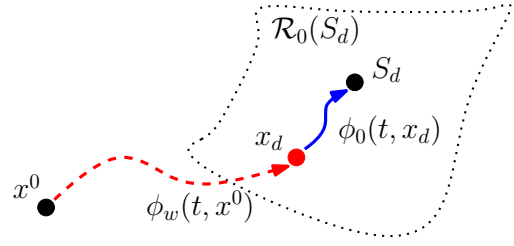


Fig. 3: Reprogramming to a desired stable steady state S_d . The red dashed arrow represents the trajectory of the stimulated system Σ_w and blue solid arrow represents the trajectory of the unstimulated system Σ_0 . The stimulated system Σ_w has an asymptotically stable steady state x_d such that $\lim_{t \rightarrow \infty} \phi_w(t, x^0) = x_d$, and $x_d \in \mathcal{R}_0(S_d)$ so that the system Σ_0 is reprogrammed to S_d under input w .

$h_i(x) \in C^1$, $\gamma_i > 0$ is a constant, and there exists an $H_{iM} > 0$ such that $0 < h_i(x) < H_{iM}$ for all $x \in X$.

Note that the above assumption is consistent with the properties of the dynamics of GRN modules as described in Section II.

Assumption 2. System Σ_w is a monotone system with respect to some K_m .

Assumption 3. There exists an $\epsilon > 0$ such that for any $w \in B_\epsilon(0)$, any steady state $S(w)$ of Σ_w is a locally unique and continuous function of w .

IV. EXTREME STABLE STEADY STATES

In this section, we show that under Assumptions 1 and 2, the set of stable steady states \mathbb{S} of Σ_0 has a maximum and a minimum stable steady state, and further provides inputs that are guaranteed to strongly reprogram Σ_w to these extreme stable steady states.

Lemma 2. Under Assumptions 1 and 2, the set of stable steady states \mathbb{S} of system Σ_0 has a minimum $S_{\min} = \min(\mathbb{S})$ and a maximum $S_{\max} = \max(\mathbb{S})$ with respect to the partial order \leq_m .

Proof. Proof of this Lemma is given in Appendix B. ■

To find inputs that are guaranteed to reprogram Σ_0 to the minimum or maximum steady states, we first define the following two types of inputs. These input types are such that, for every state x_i , the input $w_i = (u_i, v_i) \in \mathbb{R}_+^2$ is such that either $u_i \geq 0$ and $v_i = 0$, or $u_i = 0$ and $v_i \geq 0$. For a system that is monotone with respect to a partial order $m \in \{0, 1\}^n$, the value $m_i \in \{0, 1\}$ determines whether state i is given a positive input or a negative input as follows.

(i) Input of **type 1**: An input of type 1 satisfies the following: for all $i \in \{1, \dots, n\}$, if $m_i = 0$ then $u_i \geq 0$ and $v_i = 0$ (positive or no simulation) and if $m_i = 1$ then $u_i = 0$ and $v_i \geq 0$ (negative or no simulation). Further, at least one node is given an input not 0.

(ii) Input of **type 2**: An input of type 2 satisfies the following: for all $i \in \{1, \dots, n\}$, if $m_i = 1$ then $u_i \geq 0$ and $v_i = 0$ (positive or no simulation) and if $m_i = 0$ then $u_i = 0$ and $v_i \geq 0$ (negative or no simulation). Further, at least one node is given an input not 0.

(iii) Input of **type 3**: An input of type 3 is any input such that every node i either has $u_i \geq 0$ and $v_i = 0$, or $v_i \geq 0$ and $u_i = 0$, but is not an input of type 1 or 2.

For inputs of type 1 and type 2, Theorem 1 provides guarantees for strongly reprogramming Σ_0 to the minimum steady state S_{\min} and the maximum steady state S_{\max} .

Theorem 1. *Under Assumptions 1 and 2, a sufficiently large input of type 1 ensures that Σ_0 is strongly reprogrammable to the maximum stable steady state S_{\max} , and a sufficiently large input of type 2 ensures that Σ_0 is strongly reprogrammable to the minimum stable steady state S_{\min} .*

Proof. Consider a $\underline{w} = (\underline{w}_1, \dots, \underline{w}_n)$ where $\underline{w}_i = (\underline{u}_i, \underline{v}_i)$ such that $\underline{u}_i = 2(1 - m_i)H_{iM}$, and $\underline{v}_i = m_i \left(\frac{H_{iM}}{\min_{S \in \mathbb{S}}(S_i)} - \gamma_i \right)$. Then, using Lemma 1 of Appendix B, we have that for $m_i = 0$, $\lim_{t \rightarrow \infty} x_i(t) \geq \max_{S \in \mathbb{S}}(S_i)$ for all $x_i(0)$. Using Lemma 2 of Appendix B, we have that for $m_i = 1$, $\lim_{t \rightarrow \infty} x_i(t) \leq \min_{S \in \mathbb{S}}(S_i)$ for all $x_i(0)$. Note that if x, y are such that for a state where $m_i = 0$, $x_i \leq y_i$, and for a state where $m_i = 1$, $x_i \geq y_i$, then $x \leq_m y$. Thus, $\omega_{\underline{w}}(x_0) \geq_m \max(\mathbb{S})$, $\forall x_0$ and $\forall u \geq \underline{w}$ (element-wise) with an input of type 1. By monotonicity, if $z \geq_m \max(\mathbb{S})$, $\omega_0(z) = \{\max(\mathbb{S})\}$. Thus, $\omega_{\underline{w}}(x_0) \subset \mathcal{R}_0(\max(\mathbb{S})) \forall x_0$. Thus, Σ_0 is strongly reprogrammable to $\max(\mathbb{S})$.

Consider a $\underline{w} = (\underline{w}_1, \dots, \underline{w}_n)$ where $\underline{w}_i = (\underline{u}_i, \underline{v}_i)$ such that $\underline{u}_i = 2m_i H_{iM}$, and $\underline{v}_i = (1 - m_i) \left(\frac{H_{iM}}{\min_{S \in \mathbb{S}}(S_i)} - \gamma_i \right)$. Then, using Lemma 1, we have that for $m_i = 1$, $\lim_{t \rightarrow \infty} x_i(t) \geq \max_{S \in \mathbb{S}}(S_i)$ for all $x_i(0)$. Using Lemma 2, we have that for $m_i = 0$, $\lim_{t \rightarrow \infty} x_i(t) \leq \min_{S \in \mathbb{S}}(S_i)$ for all $x_i(0)$. Using the same reasoning as above, we have that $\omega_{\underline{w}}(x_0) \leq_m \min(\mathbb{S})$, $\forall x_0$ and $\forall u \geq \underline{w}$ (element-wise) with an input of type 2. Under Lemma 1, if $z \leq_m \min(\mathbb{S})$, $\omega_0(z) = \{\min(\mathbb{S})\}$. Thus, $\omega_{\underline{w}}(x_0) \subset \mathcal{R}_0(\min(\mathbb{S})) \forall x_0$. Thus, Σ_0 is strongly reprogrammable to $\min(\mathbb{S})$. ■

Example. Consider again the two-node network motifs of mutual antagonism and mutual cooperation shown in Figs. 1a and 1b, which are monotone with respect to the partial orders $m = (0, 1)$ and $m = (0, 0)$, respectively, thus satisfying Assumption 2. Their dynamics (1), with Hill-functions given by (2) and (3), also satisfy Assumption 1. Thus, these systems satisfy the hypotheses of Lemma 2 and Theorem 1. As expected under Lemma 2, the set of stable steady states of these systems have a minimum S_1 and a maximum S_3 , as shown in Figs. 2a and 2b. Note that these minima and maxima are defined with respect to the partial order m . For example, consider the two-node network motif of mutual antagonism. The partial order for this system is defined by $m = (0, 1)$, that is $x \leq_m y \iff x_1 \leq y_1, x_2 \geq y_2$. The steady-state S_1 (Fig. 2a) is then the minimum stable steady state for this system since it has the lowest value of x_1 and the highest value of x_2 , and the steady-state S_3 is similarly the maximum stable steady state with the highest value of x_1 and the lowest value of x_2 . For the two-node network of mutual cooperation, the partial order is defined by $m = (0, 0)$, that is $x \leq_m y \iff x_1 \leq y_1, x_2 \leq y_2$. The steady-state S_1 (Fig. 2b) is then the minimum stable steady state for this system with

the lowest levels of x_1 and x_2 , and S_3 is the maximum stable steady state with the highest levels of x_1 and x_2 .

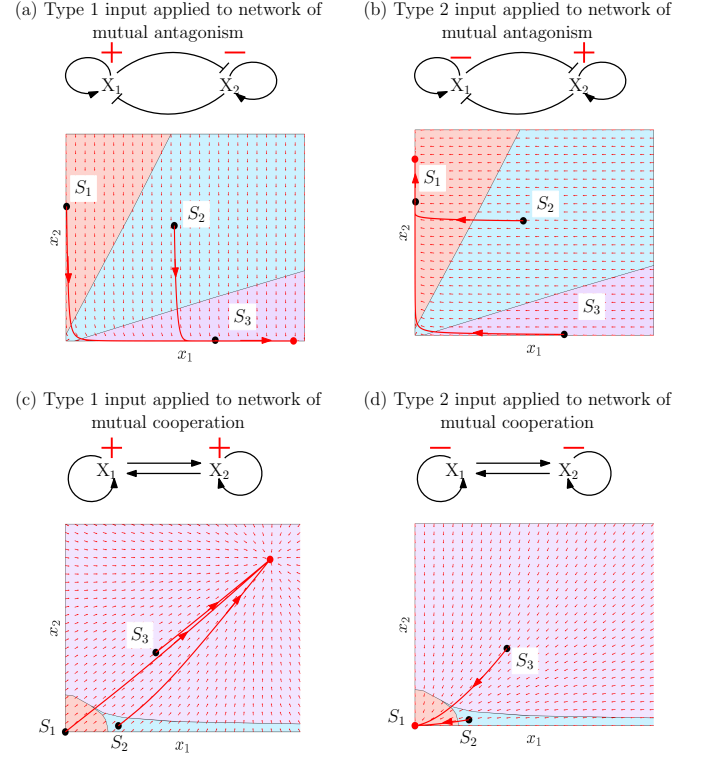


Fig. 4: Inputs of type 1 and 2 applied to the two-node networks of mutual antagonism and mutual cooperation as in (1). Resulting vector fields and stable steady states of the stimulated system Σ_w are shown via red arrows and solid red circles, respectively. The regions of attraction of S_1 , S_2 and S_3 (solid black circles) for the unstimulated system Σ_0 are shown in the background in coral, blue and purple, respectively. Parameters of the unstimulated systems are the same as those of Fig. 2. (a) An input of type 1, with $u_1 = 3\text{nMs}^{-1}$, $v_2 = 10\text{s}^{-1}$ is applied to the mutual antagonism network. This results in a globally asymptotically stable steady state in the region of attraction of S_3 , and thus the input of type 1 strongly reprograms the system to the maximum steady state S_3 . (b) An input of type 2, with $v_1 = 10\text{s}^{-1}$, $u_2 = 2\text{nMs}^{-1}$ is applied to mutual antagonism network. This results in a globally asymptotically stable steady state in the region of attraction of S_1 , and thus the input of type 2 strongly reprograms the system to the minimum steady state S_1 . (c) An input of type 1, with $u_1 = 2\text{nMs}^{-1}$, $u_2 = 1.8\text{nMs}^{-1}$ is applied to the mutual cooperation network. This results in a globally asymptotically stable steady state in the region of attraction of S_3 , and thus the input of type 1 strongly reprograms the system to the maximum steady state S_3 . (d) An input of type 2, with $v_1 = 4\text{s}^{-1}$, $v_2 = 6\text{s}^{-1}$ is applied to the mutual cooperation network. This results in a globally asymptotically stable steady state in the region of attraction of S_1 , and thus the input of type 2 strongly reprograms the system to the minimum steady state S_1 .

Further, under Theorem 1, the mutually cooperative and mutually antagonistic systems are guaranteed to be strongly reprogrammed to their maximum and minimum steady states using sufficiently large inputs of type 1 and type 2, respectively. Since the mutual antagonism network is monotone with respect to $m = (0, 1)$, applying a positive input on x_1 ($u_1 \geq 0, v_1 = 0$) and a negative input on x_2 ($u_2 = 0, v_2 \geq 0$) constitutes an input of type 1. When such an input is sufficiently large, it is guaranteed to strongly reprogram this

system to its maximum steady state S_3 . This is shown in Fig. 4a, where a large input of type 1 results in a globally asymptotically stable steady state (shown by a red filled circle) in the region of attraction of S_3 . Thus, trajectories of the system under this input from every initial condition converge to this globally asymptotically stable steady state. Once the state of the system has entered the region of attraction of S_3 , and the input is removed, the trajectory of the unstimulated system converges to S_3 , so that the unstimulated system is reprogrammed to S_3 . Similarly, applying a sufficiently negative input on x_1 ($u_1 = 0, v_1 \geq 0$) and a sufficiently large positive input ($u_2 \geq 0, v_2 = 0$) on x_2 (an input of type 2), results in a globally asymptotically stable steady state in the region of attraction of S_1 , the minimum steady state of the system. Thus a sufficiently large input of type 2 strongly reprograms this system to S_1 as shown in Fig. 4b. For the mutual cooperation network, which is monotone with respect to $m = (0, 0)$, a sufficiently large positive input on both nodes ($u_1, u_2 \geq 0, v_1 = v_2 = 0$, an input of type 1) strongly reprograms the system to the maximum steady state S_3 , and a sufficiently large negative input on both nodes ($u_1 = u_2 = 0, v_1, v_2 \geq 0$, an input of type 2) strongly reprograms the system to the minimum steady state S_1 . This is illustrated in Figs. 4c and 4d.

Thus, a general n -dimensional monotone dynamical system has a minimum and a maximum stable steady state, by Lemma 2. Further, the network structure of such a system can be used to determine the partial order m with respect to which it is monotone. Based on m , inputs can be determined that are guaranteed, when sufficiently large, to strongly reprogram the system to the minimum or maximum stable steady states by Theorem 1.

V. INTERMEDIATE STABLE STEADY STATES

The previous section provided inputs, based on the network structure of Σ_0 , that are guaranteed to reprogram Σ_0 to its extreme stable steady states. In this section, we address the problem of reprogramming Σ_0 to its non-extremal, or intermediate, stable steady states. First, we show via Theorem 2 that inputs of type 1 and type 2 are not good candidates to reprogram Σ_0 to its intermediate stable steady states. Next, we consider inputs of type 3, and show via examples that this type of input is not guaranteed to reprogram the system to the desired intermediate stable steady state. Finally, we provide results that can be used to prune the input space while searching for inputs that reprogram a monotone system to its intermediate stable steady states.

A. Reprogramming to intermediate stable steady states using type 1, type 2 or type 3 inputs

The following results show that inputs of type 1 and type 2 cannot strongly reprogram Σ_0 to an intermediate steady state (Theorem 2), and that inputs of type 1 and type 2 may not be able to weakly reprogram Σ_0 to an intermediate steady state (Theorem 3).

Theorem 2. *Under Assumptions 1 and 2, for any input of type 1 (type 2), system Σ_0 is not strongly reprogrammable to any steady state $S \neq S_{\max}$ ($S \neq S_{\min}$).*

Proof. Consider the extended system $\Sigma'_w: \dot{x} = f(x, w), \dot{w} = 0$. Notice that for any input w_0 of type 1, $w_0 \geq_{m \times -m} 0$. Note that the following initial conditions are ordered, i.e., $(\max(\mathbb{S}), 0) \leq_{m \times m \times -m} (\max(\mathbb{S}), w_0)$. Since $(\max(\mathbb{S}), 0)$ is a steady state of the extended system, by the monotonicity of the extended system (Proposition III.2 of [19]), we have that $\max(\mathbb{S}) \leq_m \phi_{w_0}(t, \max(\mathbb{S}))$, under Lemma 1. Hence, $\omega_{w_0}(\max(\mathbb{S})) \geq_m \max(\mathbb{S})$.

We now consider the system $\Sigma_0: \dot{x} = f(x, 0)$, starting at an initial condition $z \geq_m \max(\mathbb{S})$. By the monotonicity of Σ_0 , we have that $\omega_0(z) \geq_m \max(\mathbb{S})$, under Lemma 1. Since $\omega_0(z) \subseteq \mathbb{S}$, we have that $\omega_0(z) = \{\max(\mathbb{S})\}$. Thus, for any $z \geq_m \max(\mathbb{S}), z \in \mathcal{R}_0(\max(\mathbb{S}))$. Thus, $\omega_{w_0}(\max(\mathbb{S})) \subseteq \mathcal{R}_0(\max(\mathbb{S}))$. That is, for the system Σ_w with an input of type 1, any trajectory starting at $\max(\mathbb{S})$ will converge to a steady state in the region of attraction (for Σ_0) of $\max(\mathbb{S})$. Thus, Σ_0 is not strongly reprogrammable to any steady state other than $\max(\mathbb{S})$, since there exists an x_0 such that $\omega_w(x_0) \notin \mathcal{R}_w(S)$, for all $S \neq \max(\mathbb{S})$.

The proof for type 2 inputs ($w_0 \leq_{m \times -m} 0$) and the steady state $\min(\mathbb{S})$ follows similar to above. ■

Theorem 3. *Consider two steady states $S, S_d \in \mathbb{S}$, and let Σ_w satisfy Assumptions 1 - 3. Then, there exist a pair $w', w'' \in \mathbb{R}_+^{2n}$ such that for an input of type 1 (type 2) with $w \leq w'$ or $w \geq w''$, Σ_0 is not weakly reprogrammable from S to S_d if $S_d \neq S_{\max}$ ($S_d \neq S_{\min}$).*

Proof. Consider w with w close to 0. Under Assumption 3, $x(w)$ is a locally unique solution to $f(x, w) = 0$; furthermore $x(w)$ is a continuous function of w . Therefore, for w sufficiently close to 0, we will have that $x(w)$ is close to S . We can thus pick w small enough such that $x(w)$ is in the region of attraction of S . Therefore, there is an input w' sufficiently close to zero such that if $w \leq w'$, the system is not reprogrammed from S to S_d . The fact that there exists a w'' sufficiently large that if $w \geq w''$, the system is not reprogrammed to S but in fact to S_{\max} (or S_{\min}) for an input of type 1 or type 2 follows from Theorem 1. ■

According to this theorem, if an input of type 1 (type 2) is too large or too small, it cannot weakly reprogram Σ_0 to an intermediate (non-extremal) steady state. Further, depending on the parameters of Σ_0 , w' and w'' could be very close, and in fact, it is possible that $w' \geq w''$, in which case, no input value of type 1 (or type 2) could weakly reprogram the system from S to S_d . This situation is demonstrated for the mutual cooperation network below.

Example. We seek to weakly reprogram the network of mutual cooperation from stable steady state S_1 to the intermediate, stable steady state S_2 , as shown in Fig. 5. From this figure, we see that as the positive inputs u_1 and/or u_2 are increased, the stable steady state in the blue region (region of attraction of the desired state S_2) disappears before the stable steady state in the coral region (region of attraction

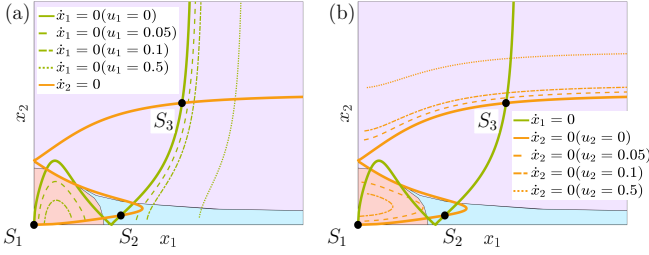


Fig. 5: Input of type 1 applied to the network of mutual cooperation to attempt to weakly reprogram it from S_1 to S_2 (big black circles). Increasing u_1 and/or u_2 (the positive input on nodes x_1 and x_2 , respectively) changes the shape of the nullclines $\dot{x}_1 = 0$ and/or $\dot{x}_2 = 0$ such that the stable steady state in the blue region ($\mathcal{R}_0(S_2)$) disappears before the stable steady state $a^*(w)$ (small red circles) in the coral region ($\mathcal{R}_0(S_1)$). Finally, for large u_1 and/or large u_2 , only a stable steady state in the purple region ($\mathcal{R}_0(S_3)$) remains.

of S_1). We call the stable steady state of system Σ_w in the coral region $a^*(w)$. Finally, when u_1 and/or u_2 are sufficiently large, the stable steady state in the coral region also disappears, leaving only a globally asymptotically stable steady state in the purple region. We call the stable steady state of system Σ_w in the purple region $b^*(w)$. Then, for any non-zero input w of type 1, that is, with $u_1, u_2 \geq 0$ and $v_1 = v_2 = 0$, the trajectory of the stimulated system Σ_w starting from the stable steady state S_1 converges to $a^*(w)$, if it exists, or to $b^*(w)$ once $a^*(w)$ disappears. After the input is removed, the trajectory of the unstimulated system starting from $a^*(w)$ converges to S_1 (since $a^*(w) \in \mathcal{R}_0(S_1)$), and the trajectory of the unstimulated system starting from $b^*(w)$ converges to S_3 (since $b^*(w) \in \mathcal{R}_0(S_3)$). Thus, for any input of type 1, the system starting at S_1 either converges back to S_1 (for a small input of type 1), or is reprogrammed to S_3 (for a large input of type 1). Thus, there is no input of type 1 that can weakly reprogram the system from S_1 to S_2 .

Thus, inputs of type 1 and type 2 are not good candidates to reprogram Σ_0 to a desired intermediate steady state. In the next example, we apply inputs of type 3 to the two-node system of mutual antagonism to demonstrate that these may be promising.

Example. We apply inputs of type 3 to the network of mutual antagonism to reprogram this system to its intermediate steady state S_2 , and present the results in Fig. 6. For this system, a positive input on both nodes ($u_1, u_2 \geq 0, v_1 = v_2 = 0$) or a negative input on both nodes ($u_1 = u_2 = 0, v_1, v_2 \geq 0$) are inputs of type 3. As seen from Fig. 6a, applying a positive input to both nodes results in a globally asymptotically stable steady state in the region of attraction of S_2 , and therefore the system is strongly reprogrammed to S_2 . When a negative input is applied to both nodes (Fig. 6b), the resultant globally asymptotically stable steady state is in the region of attraction of S_2 , and this input also strongly reprograms the system to S_2 .

While potentially more promising than inputs of type 1 and 2, whether inputs of type 3 can reprogram a monotone

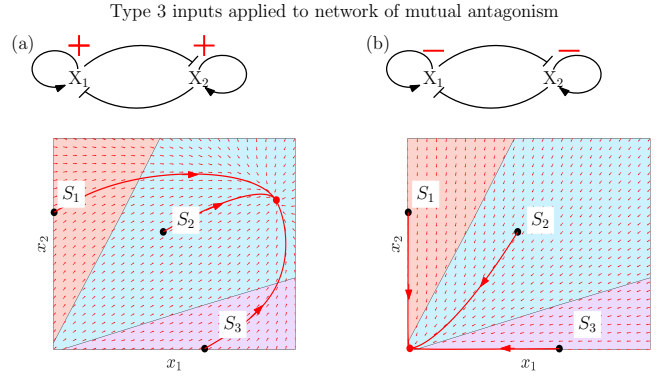


Fig. 6: Inputs of type 3 applied to the two-node network of mutual antagonism as in (1). Resulting vector fields and stable steady states of the stimulated system are shown in red. The regions of attraction of S_1 , S_2 and S_3 for the unstimulated system Σ_0 are shown in the background in coral, blue and purple, respectively. (a) An input of type 3, with $u_1 = 3\text{nMs}^{-1}$, $u_2 = 2\text{nMs}^{-1}$ is applied to the mutual antagonism network. The resulting globally asymptotically stable steady state is in the region of attraction of S_2 , and thus the input of type 3 strongly reprograms the system to the intermediate steady state S_2 . (b) An input of type 3, with $v_1 = 8\text{s}^{-1}$, $v_2 = 12\text{s}^{-1}$ is applied to the mutual antagonism network. This results in a globally asymptotically stable steady state in the region of attraction of S_2 , and thus this input of type 3 strongly reprograms the system to the steady state S_2 .

system to its intermediate steady states depends on the specific parameters. Here, we wish to provide parameter-independent rules for finding an input that is guaranteed to reprogram the system Σ_0 to a desired intermediate steady state.

B. An input space for reprogramming to intermediate steady states

In this section, we describe an input space $W \subset \mathbb{R}_+^{2n}$ that is guaranteed to contain an input that strongly reprograms the system Σ_0 to a desired intermediate steady state. Further, we provide results to guide the search for such inputs in this input space. To this end, we introduce the notion of *static input-state characteristic* [19]:

Definition 5. A controlled system $\dot{x} = f(x, w)$, $x \in X, w \in W$ is said to be endowed with a *static input-state characteristic* $c(\cdot) : W \rightarrow X$ if for each constant $w \in W$ the system has a globally asymptotically stable equilibrium $c(w)$.

Given our system in the form of Assumption 1, we make the two following additional assumptions.

Assumption 4. All first derivatives of the function $h(x)$ are bounded, that is, $\left| \frac{\partial h_i(x)}{\partial x_j} \right| \leq \bar{h}_{Mij} \forall x \in X$ and all $i, j \in \{1, \dots, n\}$.

Assumption 5. For all $i \in \{1, \dots, n\}$, there exists a $c_i > 0$ such that $\frac{\partial h_i(x)}{\partial x_i} + \sum_{j \neq i} \left| \frac{\partial h_i(x)}{\partial x_j} \right| < \gamma_i$ for all $x \in X$ such that $x_i > c_i$.

Note that Assumptions 1 and 4 are naturally satisfied for all gene regulatory networks modeled using the Hill-function representation of gene expression regulation [47]. An easily

checkable case where Assumption 5 is satisfied for state i is when x_i is self-regulating, i.e. $\frac{\partial h_i(x)}{\partial x_i} \neq 0$, and x_i does so without the need to cooperate with other species. In such cases, as x_i becomes large, $\frac{\partial h_i(x)}{\partial x_j}$ approaches 0, and so there exists a c_i large enough where the condition is true.

We now describe the input space, which, as we shall prove, is guaranteed to contain an input that reprograms the system Σ_0 to any desired intermediate steady state S_d . Consider a bounded input space, that is, let u_{mi} be the maximum positive input applied to state x_i such that $0 \leq u_i \leq u_{mi}$, and let v_{mi} be the maximum negative input applied to state x_i such that $0 \leq v_i \leq v_{mi}$. We define $W \subset \mathbb{R}_+^{2n}$ as follows:

$$W = W_1 \times W_2 \times \dots \times W_n = \prod_{i=1}^n W_i, \text{ where} \\ W_i = \{w_i = (u_i, v_i) | 0 \leq u_i \leq u_{mi}, v_i = v_{mi}\} \\ \cup \{w_i = (u_i, v_i) | u_i = u_{mi}, 0 \leq v_i \leq v_{mi}\}. \quad (6)$$

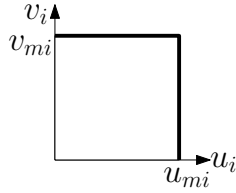


Fig. 7: The set W_i defined in (6) shown by the bold black lines – note that only the uppermost and rightmost boundaries of the box are included in W_i .

The input space W is illustrated in Fig. 7. In essence, any input in this space is such that for any state x_i either we apply: (i) the maximum positive stimulation u_{mi} and a negative stimulation that can vary in $[0, v_{mi}]$, or (ii) the maximum negative stimulation v_{mi} and a positive stimulation that can vary in $[0, u_{mi}]$. Note that this input space is n -dimensional, reduced in dimensionality compared to the original $2n$ -dimensional input space, where u_i and v_i could be varied simultaneously.

The following results are true for the input space above.

Lemma 3. *Under Assumptions 1, 4 and 5, for the input domain W as defined by (6), there exist u_{mi}, v_{mi} sufficiently large for all $i \in \{1, \dots, n\}$ such that the system Σ_w has a static input-state characteristic $c : W \rightarrow X$.*

Proof. Proof using contraction theory [48] provided in Appendix A. ■

Lemma 4. *Let $\delta_i > 0$ be a small constant such that $\delta_i < \min_{S \in \mathbb{S}}(S_i)$, and $B := \{x | \min_{S \in \mathbb{S}}(S_i) - \delta_i \leq x_i \leq \frac{H_{iM}}{\gamma_i}, \forall i \in \{1, \dots, n\}\}$. Under Assumptions 1, 4 and 5, there exist a u_{mi}, v_{mi} sufficiently large for all $i \in \{1, \dots, n\}$ such that, for every $x' \in B$, there exists an input $w' \in W$ such that $c(w') = x'$.*

Proof. Proof using the intermediate value theorem provided in Appendix A. ■

We now present the first main result of this subsection.

Theorem 4. *Let Assumptions 1, 2, 4 and 5 hold, and let input space W be defined by (6). Then, there exist sufficiently large u_{mi}, v_{mi} for all $i \in \{1, \dots, n\}$ such that the following holds:*

- (i) *System Σ_w has a static input-state characteristic $c : W \rightarrow X$.*
- (ii) *For any desired stable steady state S_d of Σ_0 , there exists an $\epsilon > 0$ such that $B_\epsilon(S_d)$ is contained in the image of $c : W \rightarrow X$.*

Proof. (i) Follows directly from Lemma 3.

(ii) We show that for all $S \in \mathbb{S}$, there exists an $\epsilon > 0$ such that $B_\epsilon(S)$ is contained in box B defined in Lemma 4, i.e., $B_\epsilon(S) \subset B$. For this, we see that any equilibrium S of Σ_0 must satisfy: $f_i(S, 0) = h_i(S) - \gamma_i S_i = 0$. Thus, $S_i = \frac{h_i(S)}{\gamma_i}$. Under Assumption 1, $0 < h_i(S) < H_{iM}$. Thus, $0 < S_i < \frac{H_{iM}}{\gamma_i}$. By definition, $S_i \geq \min_{S \in \mathbb{S}}(S_i)$. Thus, for all i , $S_i \in (\min_{S \in \mathbb{S}}(S_i) - \delta_i, \frac{H_{iM}}{\gamma_i})$. Let $\epsilon < \min\{S_i - (\min_{S \in \mathbb{S}}(S_i) - \delta_i), \frac{H_{iM}}{\gamma_i} - S_i\}$. Then, $B_\epsilon(S) \subset B$. Then, from Lemma 4, we have that there exists a $w \in W$ such that $c(w) = x \in B_\epsilon(S) \subset B$. ■

Under Assumptions 1, 2, 4 and 5, and for u_{mi}, v_{mi} sufficiently large, the input space W is guaranteed to contain an input w_d that strongly reprograms Σ_0 to S_d . Note that, for the full space $\prod_{i=1}^n [0, u_{mi}] \times [0, v_{mi}]$, the existence of an input-state characteristic is not guaranteed. In fact, for inputs very close to 0, Σ_w would be multistable, since Σ_0 is multistable. The existence of an input-state characteristic is essential to provide rules that prune the input space in search of an input.

Next, we make use of Theorem 4 and the monotonicity of Σ_0 , to present some results that guide the search for inputs that strongly reprogram the system Σ_0 to a desired stable steady state S_d . For a search procedure, we assume that we are given a simulator (or experimental setup), that is able to simulate the system from a given initial condition and for a given input for a finite pre-set time. The procedure applies inputs for the pre-set time interval, and then checks whether the state of the system, once the input is removed, approaches, in a pre-set finite time, an ϵ -ball around the target state S_d . Since S_d is an asymptotically stable equilibrium point, if the state enters an ϵ -ball around it and ϵ is sufficiently small, then the state will approach S_d . It is therefore useful to introduce the notion of ϵ -reprogramming as follows.

Definition 6. *System Σ_0 is said to be strongly ϵ -reprogrammable to state S_d by tuple (w_d, T_1, T_2) , where $w_d \in \mathbb{R}_+^{2n}$ and $T_1, T_2 \in \mathbb{R}_+$, if $\phi_0(T_2, \phi_{w_d}(T_1, x^0)) \in B_\epsilon(S_d)$, for all $x^0 \in \mathbb{S}$.*

A search procedure would then produce a tuple (w_d, T_1, T_2) that strongly ϵ -reprograms Σ_0 to S_d , given a desired $\epsilon > 0$ and S_d .

Next, we present results for the input space W , based on the definition of ϵ -reprogramming.

Lemma 5. *Let the hypotheses of Theorem 4 hold for Σ_w . For any $\epsilon > 0$ and $S_d \in \mathbb{S}$, there exist a $w_d \in W$ and $T'_1, T'_2 \geq 0$, such that for all $T_1 \geq T'_1$ and for all $T_2 \geq T'_2$ (w_d, T_1, T_2) , strongly ϵ -reprograms Σ_0 to S_d .*

Proof. Proof provided in Appendix B. ■

Next, we present the final result of this section.

Theorem 5. Let $S_d \in \mathbb{S}$. Let $w^a, w^b \in W$ and $T_1, T_2 > 0$ be such that tuples (w^a, T_1, T_2) and (w^b, T_1, T_2) strongly ϵ -reprogram Σ_0 to $S^a \neq S_d$ and $S^b \neq S_d$, respectively, for $0 < \epsilon < \frac{\min_{S' \in \mathbb{S}} \min_i |S_{di} - S'_i|}{2}$. Then, if there exists a $w' \in W$ such that (w', T_1, T_2) strongly ϵ -reprograms Σ_0 to S_d , and $w^a <_{m \times -m} w' <_{m \times -m} w^b$, then

$$S^a \ll_m S_d \ll_m S^b. \quad (7)$$

Proof. We show this by contradiction as follows. Suppose for some $i \in \{1, \dots, n\}$ where $m_i = 0$ we have that $S_{di} \leq S_i^a$. Since $S^a \neq S_d$, it is then true that $S_i^a - S_{di} \geq \min_{S' \in \mathbb{S}} \min_i |S_{di} - S'_i|$. That is, $S_{di} - S_i^a \leq -\min_{S' \in \mathbb{S}} \min_i |S_{di} - S'_i|$. Let $\phi^a := \phi_0(T_2, \phi_{w^a}(T_1, x^0))$ and $\phi' := \phi_0(T_2, \phi_{w'}(T_1, x^0))$. Since (w^a, T_1, T_2) and (w', T_1, T_2) strongly ϵ -reprogram Σ_0 to S^a and S_d , respectively, we have that $\phi^a \in B_\epsilon(S^a)$ and $\phi' \in B_\epsilon(S_d)$. Then, $\phi'_i - \phi_i^a = (\phi'_i - S_{di}) + (S_{di} - S_i^a) + (S_i^a - \phi_i^a) \leq 2\epsilon - \min_{S' \in \mathbb{S}} \min_i |S_{di} - S'_i|$. Since $\epsilon < \frac{\min_{S' \in \mathbb{S}} \min_i |S_{di} - S'_i|}{2}$, this implies that $\phi'_i - \phi_i^a < 0$, that is $\phi'_i < \phi_i^a$. However, since $w^a <_{m \times -m} w'$, under Assumption 2, it must be that $\phi^a <_m \phi'$. Since $m_i = 0$, it must be that $\phi_i^a < \phi'_i$, which is a contradiction. Thus, for no $i \in \{1, \dots, n\}$ with $m_i = 0$ can we have $S_{di} \leq S_i^a$. It can similarly be shown that for no $i \in \{1, \dots, n\}$ with $m_i = 1$ can we have $S_{di} \geq S_i^a$. Thus, it must be that $S^a \ll_m S_d$.

It can be similarly shown that it must be that $S_d \ll_m S^b$. \blacksquare

Theorem 5 provides a necessary condition that must be satisfied by any input w' that strongly ϵ -reprograms Σ_0 to S_d . This can be leveraged in a search procedure. In Appendix C, we provide an example of a search procedure that looks for an input tuple (w_d, T_1, T_2) that strongly ϵ -reprograms the two-node system of mutual cooperation to its intermediate stable steady state. The search procedure starts with an initial guess for T_1, T_2 and N_{\max} . It iteratively discretizes W for N_{\max} iterations and tries the inputs at the grid-points. Theorem 5 is then used at each iteration to eliminate parts of the input space. Specifically, consider previously tried inputs w^a and w^b that strongly ϵ -reprogram the system to S^a and S^b , respectively. Then, if (7) is not satisfied, we can remove all inputs w' that satisfy $w^a <_m w' <_m w^b$. If N_{\max} iterations are completed without finding a solution, the procedure restarts after increasing T_1, T_2 and N_{\max} . This process is guaranteed to successfully find an input tuple (w_d, T_1, T_2) that strongly ϵ -reprograms the system to the desired steady state.

VI. EXAMPLE APPLICATION

In this section, we reprogram the three-node network motif of mutual cooperation, shown in Fig. 1c. The dynamics of this system are modeled as in (1), where $n = 3$, and

$$\begin{aligned} h_1(x) &= \frac{\eta_1 + a_1 x_1^2 + b_{12} x_2^2 + b_{13} x_3^2 + a_{12} x_1^2 x_2^2 + c_{23} x_2^2 x_3^2}{1 + x_1^2 + d_{12} x_2^2 + d_{13} x_3^2 + e_{12} x_1^2 x_2^2 + k_{23} x_2^2 x_3^2}, \\ h_2(x) &= \frac{\eta_2 + a_2 x_2^2 + b_{21} x_1^2 + a_{21} x_1^2 x_2^2 + b_{23} x_3^2}{1 + x_2^2 + d_{21} x_1^2 + d_{23} x_3^2 + e_{21} x_1^2 x_2^2}, \\ h_3(x) &= \frac{\eta_3 + a_3 x_3^2 + b_{31} x_1^2 + a_{31} x_1^2 x_3^2 + b_{32} x_2^2}{1 + x_3^2 + d_{31} x_1^2 + d_{32} x_2^2 + e_{31} x_1^2 x_3^2}, \end{aligned} \quad (8)$$

indicating that the production rates of x_1, x_2 , and x_3 increase with x_1, x_2 and x_3 . For certain parameter values, this system

of ODEs has 3 stable steady states, as illustrated in Fig. 8. This network motif is commonly used to model the core pluripotency network, where x_1, x_2 and x_3 represent the concentrations of the Oct4, Sox2 and Nanog genes. In this model, given the relative values of the proteins at these steady states, we may view the steady state S_1 as representing the endoderm lineage, S_2 as representing the trophectoderm lineage and S_3 as representing the pluripotent state [35]–[41]. We now search for inputs that reprogram this network to its steady states, using the results of this paper.

We note that this system satisfies Assumption 1, since the dynamics obey the functional form required by this assumption. Further, since $\frac{\partial f_i}{\partial x_j} \geq 0$ for all $i \neq j$, the system is monotone with respect to the partial order $m = (0, 0, 0)$. Thus, this system also satisfies Assumption 2. Then, this system must satisfy Lemma 2, and indeed it does, as seen in Fig. 8b, where the minimum steady state is S_2 , and the maximum steady state is S_1 .

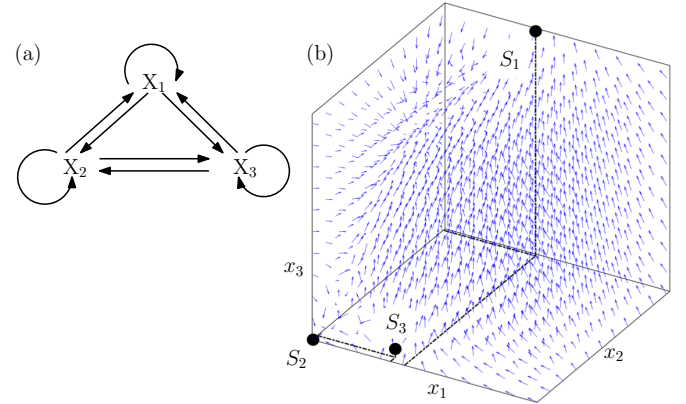


Fig. 8: A specific instance of the three-node network of mutual cooperation. (a) The three-node network motif of mutual cooperation. (b) Steady states and vector-field of system (1) for $n = 3$ and Hill-functions given by (8). Parameters of this system are: $\eta_1 = \eta_2 = \eta_3 = 10^{-4}$, $\gamma_1 = \gamma_2 = \gamma_3 = 1$, $a_1 = 1$, $b_{12} = 0.147$, $b_{13} = 0.073$, $c_{12} = 1.27$, $c_{13} = 0.63$, $d_{12} = 0.67$, $d_{13} = 0.34$, $e_{12} = 0.67$, $e_{13} = 0.34$, $a_2 = 1.6$, $b_{21} = 0.14$, $b_{23} = 0.8$, $c_{21} = 4$, $d_2 = 0.67$, $d_{21} = 1$, $d_{23} = 0.34$, $e_{21} = 1$, $a_3 = 0.816$, $b_{31} = 0.143$, $b_{32} = 1.616$, $c_{31} = 4.08$, $d_3 = 0.34$, $d_{31} = 1$, $d_{32} = 0.67$, $e_{31} = 1$. For this set of parameters, the system has three stable steady states S_1, S_2 , and S_3 , shown by black dots. The vector field $(\dot{x}_1, \dot{x}_2, \dot{x}_3)$ is shown by the blue arrows, and is normalized for visualization.

A. Reprogramming to S_1

We first strongly reprogram this system to the maximum steady state S_1 . Under Theorem 2, a sufficiently large input of type 1 is guaranteed to strongly reprogram this system to the maximum steady state, S_1 . For this system, since $m = (0, 0, 0)$, inputs of type 1 are inputs where a positive stimulation is applied on each node, that is, where $u_i \geq 0$ for $i = 1, 2, 3$ in (1). We apply such an input to the system, and results are shown in Fig. 9. The steady state of the stimulated

system Σ_w under this input is globally asymptotically stable, and lies in the region of attraction of S_1 (Fig. 9b). Once the trajectories of the stimulated system converge to this stable steady state, the input is removed. Then, the trajectory of the unstimulated system Σ_0 , starting from an initial condition in the region of attraction of S_1 , converges to S_1 , and the system is reprogrammed to S_1 (Fig. 9c).

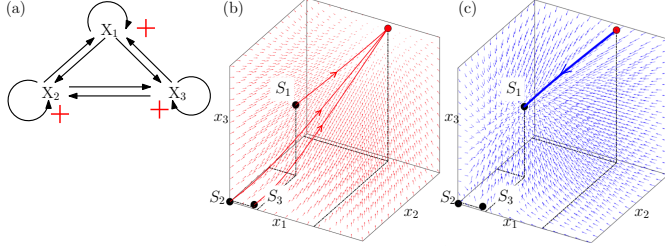


Fig. 9: Input of type 1 applied to the three-node network of mutual cooperation. A sufficiently large input of type 1 is used to strongly reprogram the 3-node mutual cooperation network to its maximum steady state S_1 . (a) An input of type 1 for this system is a positive input on each node. (b) A sufficiently large input of type 1, in this case w such that $w_1 = (2, 0)$, $w_2 = (2, 0)$, $w_3 = (2, 0)$ results in a globally asymptotically stable steady state, shown by the solid red dot. (c) The globally asymptotically stable steady state of the stimulated system is in the region of attraction of S_1 , and once the input is removed, the unstimulated system converges to S_1 .

B. Reprogramming to S_2

Similarly, under Theorem 2, we strongly reprogram the system to its minimum steady state S_2 using an input of type 2. For this system, inputs of type 2 are inputs where all the nodes are given a negative input, that is, $v_i \geq 0$ for $i = 1, 2, 3$ for (1). Under Theorem 2, a sufficiently large input of type 2 is guaranteed to strongly reprogram the system to S_2 . We demonstrate this in Fig. 10. As seen in Figs. 10b and 10c, a large input of type 2 results in a globally asymptotically stable steady state in the region of attraction of S_2 . Thus, under this input, the system is strongly reprogrammed to S_2 in the sense of Definition 3.

C. Reprogramming to S_3

Finally, we reprogram this system to its intermediate stable steady state S_3 . To apply the results of Section V, Assumptions 4 and 5 must be satisfied. Since $h_i(x)$ is a Hill-function, the partial derivatives are bounded, satisfying Assumption 4. Assumption 5 is verified numerically for this system. We apply the search procedure outlined in Appendix C, and described briefly at the end of Section V. Initial guesses for T_1 , T_2 and N_{\max} are chosen arbitrarily. The procedure returns an input tuple (w_d, T_1, T_2) that strongly ϵ -reprograms the system to Σ_0 after trying 443 input tuples. The results are shown in Fig. 11. When the space is searched without using the elimination condition of Theorem 5, 5976 input tuples have to be tried before the solution is found. Note that since the initial T_1 and T_2 were chosen arbitrarily, they were not large enough, and the initial set of iterations failed to return a solution. T_1 and

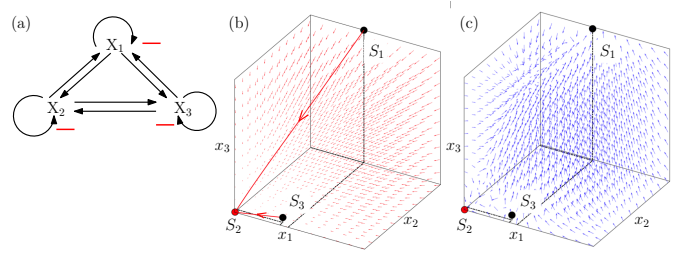


Fig. 10: Input of type 2 applied to the three-node network of mutual cooperation. A sufficiently large input of type 2 is used to strongly reprogram the 3-node mutual cooperation network to its minimum steady state S_2 . (a) An input of type 2 for this system is a negative input on each node. (b) A sufficiently large input of type 2, in this case w such that $w_1 = (0, 100)$, $w_2 = (0, 100)$, $w_3 = (0, 100)$ results in a globally asymptotically stable steady state, shown by the solid red dot. (c) The globally asymptotically stable steady state of the stimulated system is in the region of attraction of S_2 , and once the input is removed, the unstimulated system converges to S_2 .

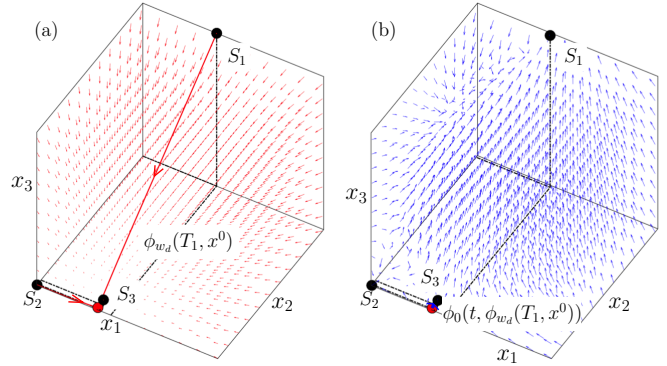


Fig. 11: Input tuple strongly ϵ -reprograms three-node mutual cooperation network to S_3 . The search procedure outlined in Appendix C is used to find an input tuple that ϵ -reprograms the three-node mutual cooperation network to the intermediate stable steady state S_3 . The procedure is initialized with $n = 3$, $u_{m1} = u_{m2} = u_{m3} = 120$, $v_{m1} = 7$, $v_{m2} = v_{m3} = 20$, $S_d = S_3$, $m = (0, 0, 0)$, $\mathbb{S} = \{S_1, S_2, S_3\}$, $\epsilon = 0.01$. We run the procedure with two sets of guesses. The first set is picked arbitrarily with $T_1 = 100$, $T_2 = 100$ and $N_{\max} = 1$. This procedure tries a total of 108 input tuples, and a total simulation time of 56600s, before finding a solution (w_d, T_1, T_2) with $w_{d1} = (0, v_{m1})$, $w_{d2} = (u_{m1}/4, v_{m1})$, $w_{d3} = (0, v_{m3})$, and $T_1 = T_2 = 400$ s. Without the elimination condition, the procedure tries a total of 334 input tuples, with a total simulation time of 201000s. The second set of initial guesses is $T_1 = 10$, $T_2 = 10$ and $N_{\max} = 1$, which tries a total of 443 input tuples, and a total simulation time of 63900s, before finding a solution (w_d, T_1, T_2) , where w_d is the same as found before, and $T_1 = T_2 = 160$ s. Without the elimination condition, the procedure tries a total of 5976 input tuples, with a total simulation time of 908180s. (a) The input w_d is applied to the network as in (1) for time $T_1 = 400$ s. The end points of the trajectories starting at S_1 , S_2 , S_3 and S_4 are close to the globally asymptotically stable steady state of the stimulated system. (b) The input is removed, and the unstimulated system is simulated for time $T_2 = 400$ s. The system converges ϵ -close to desired steady state S_3 .

T_2 were increased in subsequent iterations (as described in Section V and Appendix C). When a larger guess for T_1 and T_2 is used, the same input is found in 108 tries. However, note that since the solution in the first case involves smaller

T_1 and T_2 , the total simulation time in both cases is still about the same. In the first case the total simulation time is 56600s, and the total simulation time in the second case, where T_1 and T_2 were much higher, is 63900s. For both sets of initial guesses, the search procedure terminates after trying far fewer input tuples than if it had not used the elimination condition of Theorem 5.

VII. CONCLUSIONS

In this paper, we presented parameter-independent strategies for choosing inputs to reprogram a monotone dynamical system to a desired stable steady state. Specifically, we considered monotone systems in a form that is commonly found in gene regulatory networks. In these networks, the mathematical problem of reprogramming the system to a stable steady state embodies the practical problem of artificially inducing cell fate transitions. In such a problem, stable steady states correspond to cell fate phenotypes and reprogramming can be performed by using appropriate, biologically feasible input stimulations. Positive input u_i can be applied by increasing the rate of production of protein x_i by infecting the cells with genes that express that protein [49]. Negative input v_i can be applied through the enhanced degradation of the protein, typically achieved using miRNA or shRNA molecules [50].

Our results use the order preserving properties of the flow of a monotone dynamical system and provide criteria for choosing appropriate input strategies depending on whether the target stable state is extremal or intermediate. For extremal stable steady states, one can choose extremal (in an appropriately defined partial order) inputs to reprogram the system to such states. For intermediate stable steady states, we introduce an input set that is guaranteed to have an input that reprograms the system to any desired intermediate steady state. We further define the notion of ϵ -reprogramming for a search procedure. We use this definition to provide results to guide a pruning strategy to decrease the input search space. In practical applications of cell fate reprogramming, often the desired state consists of a range of values due to noise [51], [52], and ϵ could be set such that the ϵ -ball is within this range.

The results presented here serve as a first step in providing a general strategy for reprogramming GRNs to a desired stable steady state. Monotone dynamical systems are highly represented in the core networks that control cell-fate decisions, and in particular, the motifs of mutual antagonism and mutual cooperation shown here are seen at several decision-points along the cell development process [3], [5]. Although network motifs provide a way to abstract the problem of cell fate transition and control, it may be an oversimplification of the biological reality, as has been argued in [53], [54], [55]. Additionally, while in this paper, we have studied the network dynamics using Hill-function models [42], [43], [44], there are other models that describe such networks, such as Boolean models of GRNs [56]. Further, our theoretical results are valid for monotone systems, and while the core GRNs controlling cell fate are monotone, they are often embedded in larger, possibly non-monotone networks. The validity of these results in these scenarios is left for future work.

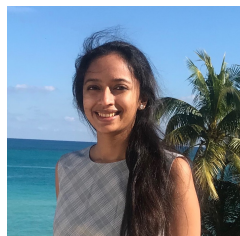
ACKNOWLEDGMENT

The authors would like to thank Theodore W. Grunberg for helping proof-read the technical results, and suggesting shorter proofs for Lemma 4 and Theorem 5. This work was in part supported by NIH/NIBIB Grant Number R01EB024591.

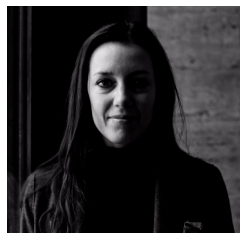
REFERENCES

- [1] Rushina Shah and Domitilla Del Vecchio. Reprogramming cooperative monotone dynamical systems. In *IEEE Conference on Decision and Control*, 2018.
- [2] D. Del Vecchio, H. Abdallah, Y. Qian, and J. J. Collins. A blueprint for a synthetic genetic feedback controller to reprogram cell fate. *Cell systems*, 4(1):109–120, 2017.
- [3] Raúl Guantes and Juan F Poyatos. Multistable decision switches for flexible control of epigenetic differentiation. *PLoS computational biology*, 4(11):e1000235, 2008.
- [4] Dan Siegal-Gaskins, Erich Grotewold, and Gregory D Smith. The capacity for multistability in small gene regulatory networks. *BMC systems biology*, 3(1):96, 2009.
- [5] Joseph X Zhou and Sui Huang. Understanding gene circuits at cell-fate branch points for rational cell reprogramming. *Trends in genetics*, 27(2):55–62, 2011.
- [6] Vijay Chickarmane and Carsten Peterson. A computational model for understanding stem cell, trophectoderm and endoderm lineage determination. *PLoS one*, 3(10):e3478, 2008.
- [7] Mingyang Lu, Mohit Kumar Jolly, Herbert Levine, José N Onuchic, and Eshel Ben-Jacob. MicroRNA-based regulation of epithelial–hybrid–mesenchymal fate determination. *Proceedings of the National Academy of Sciences*, 110(45):18144–18149, 2013.
- [8] Fuqing Wu, Ri-Qi Su, Ying-Cheng Lai, and Xiao Wang. Engineering of a synthetic quadrastable gene network to approach waddington landscape and cell fate determination. *elife*, 6:e23702, 2017.
- [9] Sudin Bhattacharya, Qiang Zhang, and Melvin E Andersen. A deterministic map of waddington’s epigenetic landscape for cell fate specification. *BMC systems biology*, 5(1):85, 2011.
- [10] Qiao Zhou, Juliana Brown, Andrew Kanarek, Jayaraj Rajagopal, and Douglas A Melton. In vivo reprogramming of adult pancreatic exocrine cells to β -cells. *nature*, 455(7213):627, 2008.
- [11] Thomas Vierbuchen, Austin Ostermeier, Zhiping P Pang, Yuko Kokubu, Thomas C Südhof, and Marius Wernig. Direct conversion of fibroblasts to functional neurons by defined factors. *Nature*, 463(7284):1035, 2010.
- [12] Pengyu Huang, Zhiying He, Shuyi Ji, Huawang Sun, Dao Xiang, Changcheng Liu, Yiping Hu, Xin Wang, and Lijian Hui. Induction of functional hepatocyte-like cells from mouse fibroblasts by defined factors. *nature*, 475(7356):386, 2011.
- [13] Xinjian Liu, Fang Li, Elizabeth A Stubblefield, Barbara Blanchard, Toni L Richards, Gaynor A Larson, Yujun He, Qian Huang, Aik-Choon Tan, Dabing Zhang, et al. Direct reprogramming of human fibroblasts into dopaminergic neuron-like cells. *Cell research*, 22(2):321, 2012.
- [14] T. Schlaeger and Dameron. A comparison of non-integrating reprogramming methods. *Nat Biotech*, 33(1):58–63, 2015.
- [15] Isaac Crespo and Antonio del Sol. A general strategy for cellular reprogramming: The importance of transcription factor cross-repression. *Stem Cells*, 31(10):2127–2135, 2013.
- [16] I. Crespo, T. Perumal, W. Jurkowski, and A. Del Sol. Detecting cellular reprogramming determinants by differential stability analysis of gene regulatory networks. *BMC systems biology*, 7(1):140, 2013.
- [17] H. L. Smith. *Monotone dynamical systems: an introduction to the theory of competitive and cooperative systems*. Number 41. American Mathematical Soc., 2008.
- [18] Morris W Hirsch. Systems of differential equations that are competitive or cooperative ii: Convergence almost everywhere. *SIAM Journal on Mathematical Analysis*, 16(3):423–439, 1985.
- [19] David Angeli and Eduardo D Sontag. Monotone control systems. *IEEE Transactions on automatic control*, 48(10):1684–1698, 2003.
- [20] D. Angeli and E.D. Sontag. Multi-stability in monotone input/output systems. *Systems Control Lett*, 51:185–202, 2004.
- [21] D. Angeli, J. E. Ferrell, and E.D. Sontag. Detection of multistability, bifurcations, and hysteresis in a large class of biological positive-feedback systems. *Proc. Natl. Acad. Sci., USA*, 101:1822–1827, 2004.
- [22] German Enciso and Eduardo D Sontag. Monotone systems under positive feedback: multistability and a reduction theorem. *Systems & control letters*, 54(2):159–168, 2005.

- [23] J. Walsh, R. DeKoter, H. Lee, E. Smith, D. Lancki, M. Gurish, D. Friend, R. Stevens, J. Anastasi, and H. Singh. Cooperative and antagonistic interplay between pu. 1 and gata-2 in the specification of myeloid cell fates. *Immunity*, 17(5):665–676, 2002.
- [24] I. Roeder and I. Glauche. Towards an understanding of lineage specification in hematopoietic stem cells: a mathematical model for the interaction of transcription factors gata-1 and pu.1. *Journal of theoretical biology*, 241(4):852–865, 2006.
- [25] Y. Arinobu, S. Mizuno, Y. Chong, H. Shigematsu, T. Iino, H. Iwasaki, T. Graf, R. Mayfield, S. Chan, P. Kastner, and K. Akashi. Reciprocal activation of gata-1 and pu. 1 marks initial specification of hematopoietic stem cells into myeloerythroid and myelolymphoid lineages. *Cell Stem Cell*, 1(4):416–427, 2007.
- [26] P. Burda, P. Laslo, and T. Stopka. The role of pu. 1 and gata-1 transcription factors during normal and leukemogenic hematopoiesis. *Leukemia*, 24(7):1249, 2010.
- [27] C. Duff, K. Smith-Miles, L. Lopes, and T. Tian. Mathematical modelling of stem cell differentiation: the pu. 1-gata-1 interaction. *Journal of mathematical biology*, 64(3):449–468, 2012.
- [28] Pu Zhang, Xiaobo Zhang, Atsushi Iwama, Channing Yu, Kent A Smith, Beatrice U Mueller, Salajja Narravula, Bruce E Torbett, Stuart H Orkin, and Daniel G Tenen. Pu. 1 inhibits gata-1 function and erythroid differentiation by blocking gata-1 dna binding. *Blood*, 96(8):2641–2648, 2000.
- [29] Tianhai Tian and Kate Smith-Miles. Mathematical modeling of gata-switching for regulating the differentiation of hematopoietic stem cell. In *BMC systems biology*, volume 8, page S8. BioMed Central, 2014.
- [30] S. Huang, Y. Guo, G. May, and T. Enver. Bifurcation dynamics in lineage-commitment in bipotent progenitor cells. *Developmental biology*, 305(2):695–713, 2007.
- [31] T. Graf and T. Enver. Forcing cells to change lineages. *Nature*, 462(7273):587, 2009.
- [32] R. Monteiro, C. Pouget, and R. Patient. The gata1/pu.1 lineage fate paradigm varies between blood populations and is modulated by *tif1 γ* . *The EMBO journal*, 30(6):1093–1103, 2011.
- [33] L. Doré and J. Crispino. Transcription factor networks in erythroid cell and megakaryocyte development. *Blood*, 118(2):231–239, 2011.
- [34] Pu Zhang, Gerhard Behre, Jing Pan, Atsushi Iwama, Nawarat Wara-Aswapati, Hanna S Radoska, Philip E Auron, Daniel G Tenen, and Zijie Sun. Negative cross-talk between hematopoietic regulators: Gata proteins repress pu. 1. *Proceedings of the National Academy of Sciences*, 96(15):8705–8710, 1999.
- [35] L. Boyer, T. Lee, M. Cole, S. Johnstone, S. Levine, J. Zucker, M. Guenther, R. Kumar, H. Murray, R. Jenner, D. Gifford, D. Melton, R. Jaenisch, and R. Young. Core transcriptional regulatory circuitry in human embryonic stem cells. *Cell*, 122(6):947–956, 2005.
- [36] Laurie A Boyer, Divya Mathur, and Rudolf Jaenisch. Molecular control of pluripotency. *Current Opinion in Genetics & Development*, 16(5):455–462, 2006.
- [37] Jonghwan Kim, Jianlin Chu, Xiaohua Shen, Jianlong Wang, and Stuart H Orkin. An extended transcriptional network for pluripotency of embryonic stem cells. *Cell*, 132(6):1049–1061, 2008.
- [38] Yui-Han Loh, Qiang Wu, Joon-Lin Chew, Vinsensius B Vega, Weitwei Zhang, Xi Chen, Guillaume Bourque, Joshy George, Bernard Leong, Jun Liu, et al. The oct4 and nanog transcription network regulates pluripotency in mouse embryonic stem cells. *Nature genetics*, 38(4):431, 2006.
- [39] H. Abdallah, Y. Qian, and D. Del Vecchio. A dynamical model for the low efficiency of induced pluripotent stem cell reprogramming. In *Submitted to American Control Conference*, 2015.
- [40] Ian Chambers and Simon R Tomlinson. The transcriptional foundation of pluripotency. *Development*, 136(14):2311–2322, 2009.
- [41] Vijay Chickarmane, Carl Troein, Ulrike A Nuber, Herbert M Sauro, and Carsten Peterson. Transcriptional dynamics of the embryonic stem cell switch. *PLoS computational biology*, 2(9):e123, 2006.
- [42] Domitilla Del Vecchio and Richard M Murray. *Biomolecular feedback systems*. Princeton University Press Princeton, NJ, 2015.
- [43] Moises Santillán. On the use of the hill functions in mathematical models of gene regulatory networks. *Mathematical Modelling of Natural Phenomena*, 3(2):85–97, 2008.
- [44] Athanasios Polynikis, SJ Hogan, and Mario di Bernardo. Comparing different ode modelling approaches for gene regulatory networks. *Journal of theoretical biology*, 261(4):511–530, 2009.
- [45] Samantha A Morris and George Q Daley. A blueprint for engineering cell fate: current technologies to reprogram cell identity. *Cell research*, 23(1):33, 2013.
- [46] Thorsten M Schlaeger, Laurence Daheron, Thomas R Brickler, Samuel Entwisle, Karrie Chan, Amelia Cianci, Alexander DeVine, Andrew Ettenger, Kelly Fitzgerald, Michelle Godfrey, et al. A comparison of non-integrating reprogramming methods. *Nature biotechnology*, 33(1):58, 2015.
- [47] D. Del Vecchio and Richard M. M. *Biomolecular Feedback Systems*. Princeton UP, 2014.
- [48] Winfried Lohmiller and Jean-Jacques E Slotine. On contraction analysis for non-linear systems. *Automatica*, 34(6):683–696, 1998.
- [49] K. Takahashi and S. Yamanaka. Induction of pluripotent stem cells from mouse embryonic and adult fibroblast cultures by defined factors. *Cell*, 126(4):663–676, 2006.
- [50] Jun Xu, Yuanyuan Du, and Hongkui Deng. Direct lineage reprogramming: strategies, mechanisms, and applications. *Cell stem cell*, 16(2):119–134, 2015.
- [51] Mukund Thattai and Alexander Van Oudenaarden. Intrinsic noise in gene regulatory networks. *Proceedings of the National Academy of Sciences*, 98(15):8614–8619, 2001.
- [52] Tariq Enver, Martin Pera, Carsten Peterson, and Peter W Andrews. Stem cell states, fates, and the rules of attraction. *Cell stem cell*, 4(5):387–397, 2009.
- [53] Philipp S Hoppe, Michael Schwarzfischer, Dirk Loeffler, Konstantinos D Kokkaliaris, Oliver Hilsenbeck, Nadine Moritz, Max Endeke, Adam Filipczyk, Adriana Gambardella, Nouraz Ahmed, et al. Early myeloid lineage choice is not initiated by random pu. 1 to gata1 protein ratios. *Nature*, 535(7611):299–302, 2016.
- [54] Michael K Strasser, Philipp S Hoppe, Dirk Loeffler, Konstantinos D Kokkaliaris, Timm Schroeder, Fabian J Theis, and Carsten Marr. Lineage marker synchrony in hematopoietic genealogies refutes the pu. 1/gata1 toggle switch paradigm. *Nature communications*, 9(1):1–10, 2018.
- [55] Pablo Navarro, Nicola Festuccia, Douglas Colby, Alessia Gagliardi, Nicholas P Mullin, Wensheng Zhang, Violetta Karwacki-Neisius, Rodrigo Osorno, David Kelly, Morag Robertson, et al. Oct4/sox2-independent nanog autorepression modulates heterogeneous nanog gene expression in mouse es cells. *The EMBO journal*, 31(24):4547–4562, 2012.
- [56] Jan Krumsiek, Carsten Marr, Timm Schroeder, and Fabian J Theis. Hierarchical differentiation of myeloid progenitors is encoded in the transcription factor network. *PLoS one*, 6(8), 2011.



Rushina Shah received her B. Tech. (2014) in Mechanical Engineering from the Indian Institute of Technology, Bombay, and an M.S. (2016) in Mechanical Engineering from the Massachusetts Institute of Technology (MIT). She is currently a PhD candidate at MIT, studying monotone dynamical systems with applications to cellular reprogramming. Her research interests include nonlinear dynamics, optimization and robustness to uncertainty.



Domitilla Del Vecchio received the Ph. D. degree in Control and Dynamical Systems from the California Institute of Technology, Pasadena, and the Laurea degree in Electrical Engineering (Automation) from the University of Rome at Tor Vergata in 2005 and 1999, respectively. From 2006 to 2010, she was an Assistant Professor in the Department of Electrical Engineering and Computer Science and in the Center for Computational Medicine and Bioinformatics at the University of Michigan, Ann Arbor. In 2010, she joined Department of Mechanical Engineering at the Massachusetts Institute of Technology (MIT), where she is currently Professor and member of the Synthetic Biology Center. She is a recipient of the 2016 Bose Research Award (MIT), the Donald P. Eckman Award from the American Automatic Control Council (2010), the NSF Career Award (2007), the American Control Conference Best Student Paper Award (2004), and the Bank of Italy Fellowship (2000). Her research interests include analysis and design of biological networks with applications to systems and synthetic biology.