

## Research



**Cite this article:** Kadelka C, Wheeler M, Veliz-Cuba A, Murrugarra D, Laubenbacher R. 2023 Modularity of biological systems: a link between structure and function. *J. R. Soc. Interface* **20**: 20230505.  
<https://doi.org/10.1098/rsif.2023.0505>

Received: 29 August 2023

Accepted: 5 October 2023

### Subject Category:

Life Sciences—Mathematics interface

### Subject Areas:

systems biology, biomathematics

### Keywords:

decomposition theory, structure and function of networks, Boolean networks

### Author for correspondence:

Claus Kadelka

e-mail: [ckadelka@iastate.edu](mailto:ckadelka@iastate.edu)

<sup>†</sup>These authors contributed equally to the study.

# Modularity of biological systems: a link between structure and function

Claus Kadelka<sup>1,†</sup>, Matthew Wheeler<sup>2,†</sup>, Alan Veliz-Cuba<sup>3</sup>, David Murrugarra<sup>4</sup> and Reinhard Laubenbacher<sup>2</sup>

<sup>1</sup>Department of Mathematics, Iowa State University, Ames, IA, USA

<sup>2</sup>Department of Medicine, University of Florida, Gainesville, FL, USA

<sup>3</sup>Department of Mathematics, University of Dayton, Dayton, OH, USA

<sup>4</sup>Department of Mathematics, University of Kentucky, Lexington, KY, USA

CK, 0000-0002-5712-8529; RL, 0000-0002-9143-9451

This paper addresses two topics in systems biology, the hypothesis that biological systems are modular and the problem of relating structure and function of biological systems. The focus here is on gene regulatory networks, represented by Boolean network models, a commonly used tool. Most of the research on gene regulatory network modularity has focused on network structure, typically represented through either directed or undirected graphs. But since gene regulation is a highly dynamic process as it determines the function of cells over time, it is natural to consider functional modularity as well. One of the main results is that the structural decomposition of a network into modules induces an analogous decomposition of the dynamic structure, exhibiting a strong relationship between network structure and function. An extensive simulation study provides evidence for the hypothesis that modularity might have evolved to increase phenotypic complexity while maintaining maximal dynamic robustness to external perturbations.

## 1. Introduction

Building complicated structures from simpler building blocks is a widely observed principle in both natural and engineered systems. In molecular systems biology, it is also widely accepted, even though there has not emerged a clear definition of what constitutes a simple building block, or module. Consequently, it is not clear how the modular structure of a system can be identified, why it is advantageous to an organism to be composed of modular components, and how we could take advantage of modularity to advance our understanding of molecular systems [1–3]. In the (graph-theoretic) network representation of molecular systems, such as gene regulatory networks or protein–protein interaction networks, a module is typically considered to be a ‘highly’ connected region of the graph that is ‘sparsely’ connected to the rest of the graph, otherwise known as a community in the graph. Graph theoretic algorithms that depend on the choice of parameters, and the specific definition of ‘highly’ and ‘sparsely’ are typically used to define modules [4,5]. Similar approaches are used for identifying modules in co-expression networks based on clustering of transcriptomics data [6].

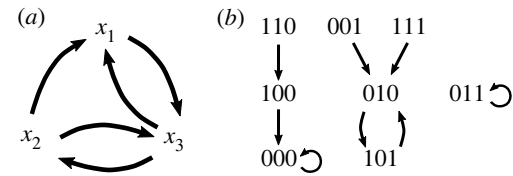
A major limitation of this approach to modularity is that it focuses entirely on a static representation of gene regulatory networks and other systems. However, living organisms are dynamic, and need to be modelled and understood as dynamical systems. Thus, modularity should have an instantiation as a dynamic feature, as advocated in [7]. The most common types of models employed for this purpose are systems of ordinary differential equations and discrete models such as Boolean networks and their generalizations, providing the basis for a study of dynamic modularity. In recent years, there have been an increasing number of papers that take this point of view. Jimenez *et al.* [8] argue

that dynamic modularity may be independent of structural modularity, and they identify examples of multi-functional circuits in gene regulatory networks that they consider dynamically modular but without any underlying structural modularity. A similar argument is made in [9] by analysing a small gene regulatory network example. For another example of a similar approach see [10].

The literature on how modularity might have evolved and why it might be useful as an organizational principle cites as the most common reasons robustness, the ability to rapidly respond to changing environmental conditions, and efficiency in the control of response to perturbations [2,11,12]. An interesting hypothesis has been put forward in [3], namely that a modular organization of biological structure can be viewed as a symmetry-breaking phase transition, with modularity as the order parameter.

This literature makes clear that research on the topic of modularity in molecular systems, both structural and dynamic, would be greatly advanced by clear definitions of the concept of module, both structural and dynamic. This would in particular help to decide whether and how structural and dynamic modularity are related, and it would provide a basis on which to distinguish between dynamic modularity and multi-stationarity of a dynamic regulatory network. To be of practical use, such a theory should include algorithms to decompose a dynamic network into structural and/or dynamic modules. At the same time, it would be of great practical value, for instance for synthetic biology, to understand how systems can be composed from modules that have specific dynamic properties.

The search for such algorithms has led us to look for guidance to mathematics, as a complement to biology. After all, if the dynamic mathematical models that are widely used to encode gene regulatory networks are appropriate representations, and if modularity is indeed an important feature of such networks, then it should be reflected in the model structure and dynamics. Choosing the widely used modelling framework of Boolean networks, we asked whether it is possible to identify meaningful concepts of modularity that, ideally, link both the structural and dynamic aspects. Modularity is fundamentally about connectivity. The central dynamic instantiation of connectivity is the feedback loop, which we, therefore, choose as the defining feature. The concept of module we propose is structural, in terms of special subgraphs of the (directed) graph of dependencies of network nodes. These subgraphs, called strongly connected components (SCCs), are maximal with respect to the property that every node is connected to every other node in the subgraph through a directed path. In other words, none of the nodes in the SCC are involved in feedback loops that are not entirely within the SCC. These types of decomposition-based approaches are by no means novel and have been employed by computer scientists for developing faster, more efficient algorithms for finding and enumerating attractors. For instance, tree decompositions have been used to find fixed points and attractors of nested canalizing networks [13], and SCCs have been used to enumerate the attractors of an asynchronous network, in a manner very similar to our own approach [14]. Our aim is to highlight the structure that these types of decompositions, in particular that of SCCs, place on the attractor landscape, and to investigate the implications for the modelled biological systems.



**Figure 1.** Wiring diagram and state space of the Boolean network  $F = (f_1, f_2, f_3) = (x_2 \wedge \neg x_3, x_3, \neg x_1 \wedge x_2)$ . (a) The wiring diagram encodes the dependency between variables. Subnetworks are defined on the basis of the wiring diagram. For example, the subnetwork  $F|_{\{x_2, x_3\}}(x_1, x_2, x_3) = (x_3, \neg x_1 \wedge x_2)$  is the restriction of  $F$  to  $\{x_2, x_3\}$  and contains external parameter  $x_1$ . (b) The state space is a directed graph with edges between all states and their images. This graph, therefore, encodes all possible trajectories and attractors. Here,  $F$  has two steady states, 000 and 011, and one limit cycle, (010, 101), so  $\mathcal{A}(F) = \{000, 011, (010, 101)\}$ .

The main result of this paper is that this structural decomposition of the model into modules induces a similar decomposition of model dynamics, explicitly linking the dynamics of the structural modules in a mathematically clearly specified way. This theorem links structural and dynamic modularity, and provides an example of how network structure influences network function. We provide an important application of this theorem to network control by showing that, in order to control a network, it is sufficient to control its modules, and we provide an application of this result to a published cancer signalling network. This result is important both for applications to e.g. medicine and might provide a candidate for a mechanism that allows organisms to quickly respond to changes in their external environment. We also discuss our results in the context of published Boolean network models of regulatory networks and provide specific instantiations of our decomposition theorem. Finally, we address the question as to why evolution should favour modularity as a structural and dynamic feature. We carry out an extensive simulation study that provides evidence for the hypothesis that modularity enables phenotypic complexity while maintaining maximal robustness to external perturbations.

## 1.1. Boolean networks

For the purpose of this article, we will focus on the class of Boolean networks as a modelling paradigm. Recall that a Boolean network  $F$  on variables  $x_1, \dots, x_n$  can be viewed as a function on binary strings of length  $n$ , which is described coordinate-wise by  $n$  Boolean update functions  $f_i$ . Each function  $f_i$  uniquely determines a map

$$F_i: \{0, 1\}^n \rightarrow \{0, 1\}^n, \quad F_i(x_1, \dots, x_n) = (x_1, \dots, f_i(x), \dots, x_n),$$

where  $x = (x_1, \dots, x_n)$ . Every Boolean network defines a canonical map, where the functions are synchronously updated,

$$F: \{0, 1\}^n \rightarrow \{0, 1\}^n, \quad F(x_1, \dots, x_n) = (f_1(x), \dots, f_n(x)).$$

In this paper, we only consider this canonical map, i.e. we only consider synchronously updated Boolean network models. Two directed graphs can be associated with  $F$  (see figure 1 for an example). The *wiring diagram* (also known as dependency graph) contains  $n$  nodes corresponding to the  $x_i$ , and has a directed edge from  $x_i$  to  $x_j$  if  $f_j$  depends on  $x_i$ . The *state space* of  $F$  contains as nodes the  $2^n$  binary strings, and has a directed edge from  $\mathbf{u}$  to  $\mathbf{v}$  if  $F(\mathbf{u}) = \mathbf{v}$ . Each connected component of the

state space gives an *attractor basin* of  $F$ , which consists of a directed loop, *the attractor*, as well as trees feeding into the attractor. Attractors can be steady states (also known as fixed points) or limit cycles. Each attractor in a biological Boolean network model typically corresponds to a distinct phenotype [15]. The set of attractors of  $F$ , denoted  $\mathcal{A}(F)$ , contains all attractors, i.e. all minimal subsets  $C \subseteq \{0, 1\}^n$  satisfying  $F(C) = C$ . Note that a limit cycle of length  $k$  represents  $k$  trajectories. For example, the 2-cycle (010, 101) in figure 1 represents (010, 101, 010, ...) and (101, 010, 101, ...). This distinction becomes important later, when decomposing the dynamics of Boolean networks.

## 2. Results

### 2.1. A structural definition of modularity for Boolean networks

Given a Boolean network  $F$  and a subset  $S$  of its variables, we can define a subnetwork of  $F$ , denoted  $F|_S$ , as the restriction of  $F$  to  $S$ . If some variables in  $S$  are regulated by variables not in  $S$ , then we require these regulations to be included in  $F|_S$ . In this case, the subnetwork is a Boolean network with external parameters. For the example in figure 1, the subnetwork  $F|_{\{x_2, x_3\}}$  contains  $x_1$  as external parameter because  $x_1$  regulates  $x_3$ . If the variables in  $S$  form a SCC (that is, (i) every pair of nodes in  $S$  (excluding possible external parameters) is connected by a directed path and (ii) the inclusion of any additional node in  $S$  will break this property), we call the subnetwork a *module*.

The wiring diagram of any Boolean network  $F$  is either strongly connected or it consists of a collection of SCCs where connections between two SCC point in only one direction. Let  $W_1, \dots, W_m$  be the SCCs of the wiring diagram, with  $Y_i$  denoting the set of variables in SCC  $W_i$  (note  $\cup_i Y_i = Y$  and  $Y_i \cap Y_j = \emptyset$  for  $i \neq j$ ). Then, the *modules* of  $F$  are  $F|_{Y_1}, \dots, F|_{Y_m}$ , the restrictions of  $F$  to the  $Y_i$ . By setting  $W_i \rightarrow W_j$  if there exists at least one edge from a node in  $W_i$  to a node in  $W_j$ , we obtain a directed acyclic graph

$$Q = \{(i, j) | W_i \rightarrow W_j\}, \quad (2.1)$$

which describes the connections between the modules of  $F$ .

As we will show later, any Boolean network can be decomposed into modules and this structural decomposition implies a decomposition of the network dynamics, which is of practical utility. The main question to be answered at this point, though, is whether there exists biological evidence that our concept of modularity and the structural and dynamic decomposition theory that follows does in fact reflect reality.

### 2.2. Modularity in expert-curated biological networks

A recent study investigated the features of 122 distinct published, expert-curated Boolean network models [16]. Analysing the wiring diagrams of these models, we found that almost all of them (113, 92.6%) contained at least one feedback loop and thus at least one non-trivial SCC/module (which contains more than one node). The nine models that only contained single-node SCCs mainly describe signalling pathways. Thirty models (24.6%) contained even more than one non-trivial SCC, with one Influenza A virus replication model possessing 11 [17]. The directed acyclic graph structure (equation (2.1)) of these models varied widely (figure 2). While the average connectivity of a network was not correlated with the number of

non-trivial SCCs ( $\rho_{\text{Spearman}} = -0.08$ ,  $p = 0.37$ ), network size was positively correlated ( $\rho_{\text{Spearman}} = 0.37$ ,  $p < 10^{-4}$ ). The same trends persisted when considering the binary variable ‘multiple non-trivial SCCs’ (multi-variable logistic regression: connectivity  $p = 0.07$ , size  $p = 0.002$ ).

Modules are subnetworks that carry out key control functions in a cell. It would, therefore, not be surprising if there was a selection bias among systems biologists to focus their attention on such modules. Larger networks are still challenging to build and analyse since an accurate formulation of a biological network model requires a substantial amount of data for a careful inference and calibration of the update rules by a subject expert [18–21]. For this reason most published expert-curated models might focus on one specific cellular function of interest and contain, therefore, only one non-trivial SCC. Assuming that a principled method for predetermining the modular structure of a biological system existed, one interesting application of this modular decomposition would be to allow Boolean inference algorithms to use this decomposition to focus on one module at a time reducing the complexity of the problem.

### 2.3. Modularity confers phenotypical robustness and a rich dynamic repertoire

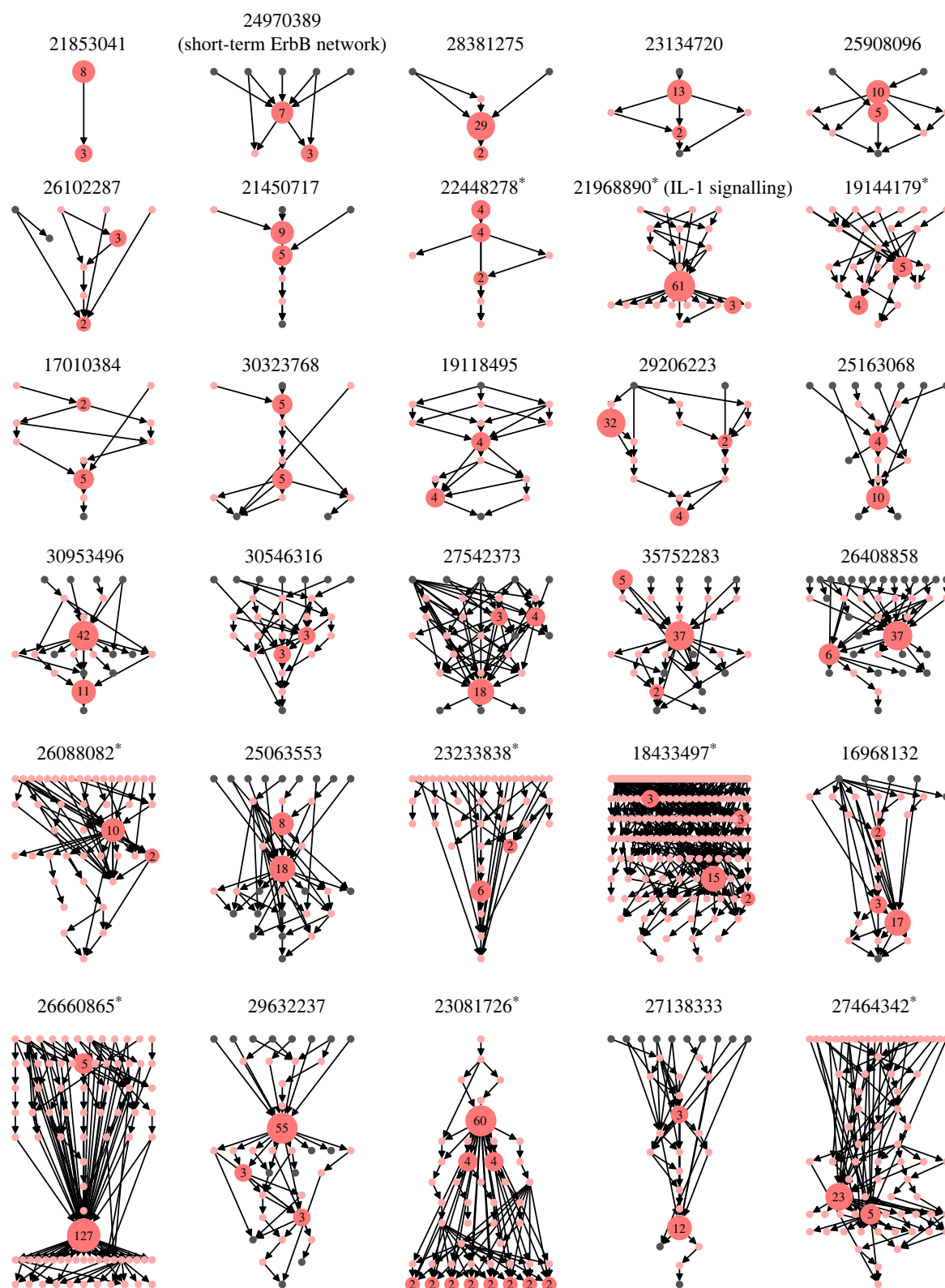
To provide additional evidence that SCCs form biologically meaningful modules, we performed a computational study which shows that the presence of several modules confers robust phenotypes and a rich dynamic repertoire, both desirable features for an organism.

Biological networks must harbour multiple phenotypes, allowing the network to dynamically shift from one attractor to another based on its current needs. This shift is typically mitigated by external signals. Many evolutionary innovations are the result of newly evolved attractors of gene regulatory networks (GRNs) [22,23]. The number of attractors of a Boolean network, therefore, describes its *dynamical complexity*.

Furthermore, biological networks need to robustly maintain a certain function (i.e. phenotype) in the presence of intrinsic and extrinsic perturbations [24,25]. At any moment, these perturbations may cause a small number of genes to randomly change their expression level. For a Boolean GRN model, this corresponds to an unexpressed gene being randomly expressed, or vice versa. The robustness of the network describes how a perturbation on average affects the network dynamics. One popular robustness measure for Boolean networks (BNs), the Derrida value, describes the average Hamming distance between two states after one synchronous update according to the Boolean network rules, given that the two states differed in a single node [26]. Due to the finite size of the state space, any state of a BN eventually transitions to an attractor, which corresponds to a distinct biological phenotype. Thus, while the Derrida value is a meaningful robustness measure, a more phenotype-focused measure describes how frequently a small perturbation (e.g. a single node flip) forces the network to transition to a different attractor. We, therefore, measure the *phenotypical robustness* of a Boolean network  $F : \{0, 1\}^n \rightarrow \{0, 1\}^n$  by

$$r(F) = \frac{1}{n2^n} \sum_{\mathbf{x} \in \{0,1\}^n} \sum_{i=1}^n \mathbf{1}[A(\mathbf{x}) = A(\mathbf{x} \oplus \mathbf{e}_i)] \quad (2.2)$$

$$= \frac{1}{n2^{n-1}} \sum_{\mathbf{x}, \mathbf{y} \in \{0,1\}^n, \|\mathbf{x}-\mathbf{y}\|=1} \mathbf{1}[A(\mathbf{x}) = A(\mathbf{y})] \in [0, 1]. \quad (2.3)$$



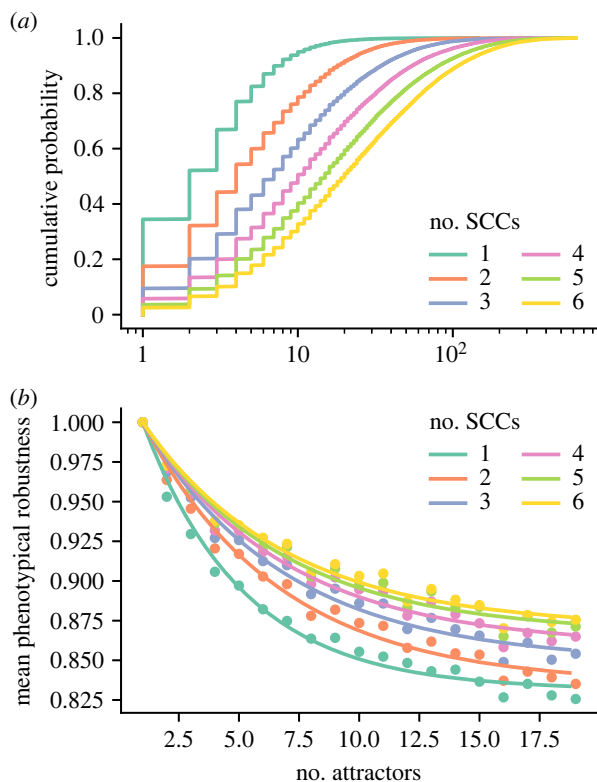
**Figure 2.** Modular decomposition of all published expert-curated Boolean gene regulatory network models with more than one non-trivial module. Each model is labelled by the Pubmed ID of its source. Each red non-trivial module is labelled by its size, i.e. the number of nodes contained in the module. Trivial modules consist of one node only. They are coloured grey if they are input or output nodes, i.e. nodes without incoming or outgoing edges, respectively. Otherwise, they are coloured pink. For models with more than 40 modules, input and output modules are omitted for clarity, indicated by \* after the Pubmed ID. An arrow from module  $X$  to module  $Y$  indicates that some node in  $X$  regulates some node in  $Y$ . The directed acyclic graph of the multi-cellular pancreatic cancer model, analysed in figure 6, is shown in row 4, column 4 (Pubmed ID 35752283).

Here,  $\mathbf{e}_i$  is the  $i$ th unit vector and  $A(\mathbf{x})$  labels the attractor that state  $\mathbf{x}$  transitions to. Geometrically, if we consider the Boolean hypercube with each vertex in  $\{0, 1\}^n$  labelled by the attractor that the vertex-associated state eventually transitions to, then

$r(F)$  is the proportion of edges, which connect vertices with the same value.

Clearly,  $r(F) = 1$  if a Boolean network  $F$  possesses only a single attractor. Moreover, the expected value,  $\mathbb{E}[r(F)]$ , decreases

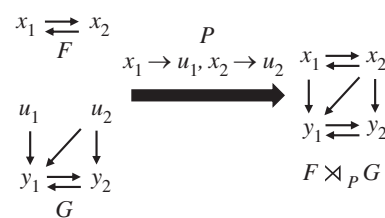




**Figure 3.** Modularity confers dynamical complexity and phenotypical robustness. Sixty-node nested canalizing Boolean networks with a constant in-degree of 3 and with 1–6 modules (i.e. SCCs of the wiring diagram) of equal size were generated (50 000 networks each). For each modular network, a weakly connected directed graph describing the connections between modules, as well as a single edge connecting an upstream with a downstream module were selected uniformly at random. By following the transitions of 500 random initial states to their attractors, the phenotypical robustness and a lower bound for the dynamical complexity (here, number of attractors) were established for each network. (a) Cumulative empirical density function of the number of attractors, stratified by the number of modules or SCCs. (b) The mean phenotypical robustness ( $y$ ) is plotted against the number of discovered attractors ( $x$ ), stratified by the number of modules or SCCs (dots). Since  $y(1) = 1$ , the two-parameter function  $y = \alpha + (1 - \alpha)e^{-k(x-1)}$  is fitted to the means of the number of attractors for  $x = 1, \dots, 19$  (lines).

as the number of attractors of  $F$  increases. This implies that the phenotypical robustness and the dynamical complexity are negatively correlated and that there exists a trade-off when trying to maximize both. It is reasonable to hypothesize that evolution favours robust GRNs that give rise to sufficient variety in the phenotype space. In line with this, we hypothesized that modular networks have higher robustness than non-modular networks with the same dynamical complexity.

To test this hypothesis, we generated Boolean networks with  $N = 60$  nodes, a fixed in-degree of 3, and  $m = 1, \dots, 6$  modules (i.e. SCCs of the wiring diagram) of size  $N/m$ . Since published expert-curated Boolean GRN networks are almost exclusively governed by nested canalizing functions [16], we required all update rules to be of this type. Networks with more modules possessed on average a higher dynamical complexity, quantified here as the number of attractors (figure 3a). At a fixed dynamical complexity, the more modular a network the higher was its average phenotypical robustness (figure 3b). This finding supports the hypothesis that a modular design serves as an evolutionary answer to a multi-objective optimization problem.



**Figure 4.** Semi-direct product of Boolean networks. Wiring diagrams of independent Boolean networks  $F$  and  $G$  (where  $G$  has external parameters) can be combined into  $F \rtimes_P G$ , the semi-direct product of  $F$  and  $G$ . The coupling scheme  $P$  describes which variables of  $F$  take the place of the external parameters and act as inputs to  $G$ .

## 2.4. Structural decomposition of Boolean networks

Thus far, we have described how to define modules as restrictions of Boolean networks and provided evidence that modules defined this way are biologically meaningful. To obtain a successful decomposition theory, we also require the inverse operation of a restriction: a semi-direct product that combines two Boolean networks,  $F$  and  $G$ , such that  $F$  is the upstream module and  $G$  is the downstream module. The coupling scheme  $P$  contains the information which nodes in  $F$  regulate which nodes in  $G$ . We denote the combined Boolean network as  $F \rtimes_P G$  and refer to this as the coupling of  $F$  and  $G$  by the coupling scheme  $P$  or as the semi-direct product of  $F$  and  $G$  via  $P$  (detailed definition in appendix A, §A.1). (The motivation for the term ‘semi-direct product’ comes from the fact that the combination of the two subnetworks is like a product, except that  $F$  acts on  $G$  through  $P$ , which is not the case in an actual product. The term is also used in mathematical group theory, which provided the motivation for our decomposition approach.)

As an example, consider the Boolean networks  $F(x_1, x_2) = (x_2, x_1)G(u_1, u_2, y_1, y_2) = (u_1 \vee (u_2 \wedge y_2), \neg u_2 \wedge y_1)$  where  $G$  possesses two external parameters,  $u_1$  and  $u_2$ . With the coupling scheme  $P = \{x_1 \rightarrow u_1, x_2 \rightarrow u_2\}$ , we obtain the combined nested canalizing network  $F \rtimes_P G: \{0, 1\}^4 \rightarrow \{0, 1\}^4$ ,

$$(F \rtimes_P G)(x_1, x_2, y_1, y_2) = (x_2, x_1, x_1 \vee (x_2 \wedge y_2), \neg x_2 \wedge y_1).$$

At the wiring diagram level, this product can be seen as the union of the two wiring diagrams and some added edges determined by the coupling scheme  $P$  (figure 4).

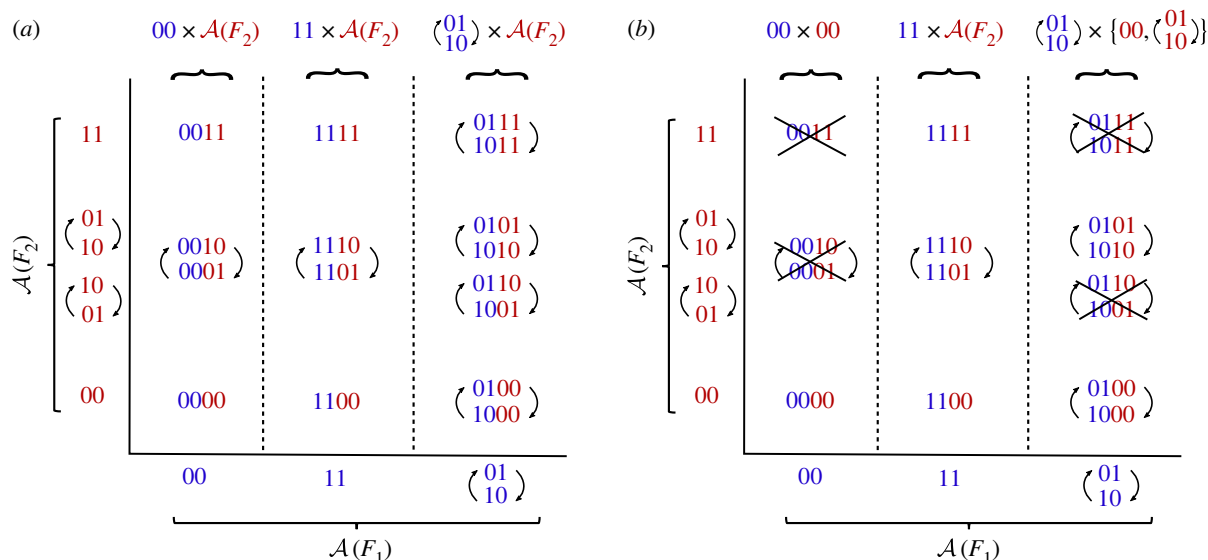
If instead  $G(u_1, u_2, y_1, y_2) = u_1 + u_2 + y_2, u_2 + y_1$  with  $F$  and  $P$  as before, then we obtain the linear network

$$(F \rtimes_P G)(x_1, x_2, y_1, y_2) = (x_2, x_1, x_1 + x_2 + y_2, x_2 + y_1).$$

At the wiring diagram level, this product looks exactly the same (figure 4).

We can prove that every network is either a module or can be decomposed into a semi-direct product of two networks. That is, if a Boolean network  $F$  is not a module (i.e. if its wiring diagram is not strongly connected), then there exist  $F_1, F_2, P$  such that  $F = F_1 \rtimes_P F_2$ , and we call such a network  $F$  decomposable. We can even find a decomposition such that  $F_1$  is a module. By induction on the downstream component  $F_2$ , it follows that any Boolean network is either a module or decomposable into a unique series of semi-direct products of modules. That is, for any Boolean network  $F$ , there exist unique modules  $F_1, \dots, F_m$  ( $m = 1$  if  $F$  is itself a module) such that

$$F = F_1 \rtimes_{P_1} (F_2 \rtimes_{P_2} (\dots \rtimes_{P_{m-1}} F_m)), \quad (2.4)$$



**Figure 5.** Attractors of a Cartesian product and a semi-direct product. (a) The space of attractors of a Cartesian product  $F = F_1 \times F_2$ , with  $F_1(x_1, x_2) = (x_2, x_1)$ ,  $F_2(x_3, x_4) = (x_4, x_3)$ , can be seen as a Cartesian product of  $\mathcal{A}(F_1)$  and  $\mathcal{A}(F_2)$ . To illustrate the different ways to combine attractors of  $F_1$  and  $F_2$ , in the panel we explicitly write  $(01, 10)$  and  $(10, 01)$  for  $F_2$ . (b) In general, the coupling of networks does not behave as a Cartesian product and the space of attractors depends on this coupling. The crossed-out attractors indicate which attractors from the Cartesian product are lost when using a semi-direct product with coupling scheme  $P = \{(x_3, x_2, x_4)\}$ , and  $F_1, F_2$  as in (a).

where this representation is unique up to a reordering, which respects the partial order induced by the directed acyclic graph  $Q$  (equation (2.1)). The collection of coupling schemes  $P_1, \dots, P_{m-1}$  depends on the particular choice of ordering, as well as on the placement of parentheses in the decomposition of  $F$ , which may be rearranged in any associative manner. Appendix A, §A.1 contains the proofs of these theorems.

## 2.5. Dynamic decomposition of Boolean networks

When the variables of a network  $F$  can be partitioned such that  $F = F_1 \rtimes_P F_2 = F_1 \times F_2$  is simply the cross product of two networks  $F_1$  and  $F_2$ , i.e. the coupling scheme  $P = \emptyset$ , then the dynamics of  $F$  can be determined directly from the dynamics of  $F_1$  and  $F_2$ . The dynamics of  $F$  consists of coordinate pairs  $(x, y)$  such that

$$x(t+1) = F_1(x(t)) \quad \text{and} \quad y(t+1) = F_2(y(t)). \quad (2.5)$$

If trajectories  $(x(t))_{t=0}^\infty$  and  $(y(t))_{t=0}^\infty$  have periods  $l$  and  $m$ , respectively, then the periodicity of the trajectory  $(x(t)(t))_{t=0}^\infty(x(t)(t))_{t=0}^\infty$  is the least common multiple of  $l$  and  $m$ . Moreover, the set of periodic points (i.e. attractors) of  $F$  is the Cartesian product of the set of periodic points of  $F_1$  and periodic points of  $F_2$ .

For example, the Boolean network  $F(x_1, x_2, x_3, x_4) = (x_2, x_1, x_4, x_3)$  can be seen as  $F = F_1 \times F_2$ , where  $F_1(x_1, x_2) = (x_2, x_1)$  and  $F_2(x_3, x_4) = (x_4, x_3)$ . The sets of attractors of  $F_1$  and  $F_2$  are  $\mathcal{A}(F_1) = \{00, 11, (01, 10)\}$  and  $\mathcal{A}(F_2) = \{00, 11, (01, 10)\}$  (where we omit parentheses around steady states). By concatenating the attractors of  $F_1$  and  $F_2$ , we obtain the attractors of  $F$  (figure 5a). Note that we have two ways of concatenating the limit cycle  $(01, 10)$  of  $F_1$  and the limit cycle  $(01, 10)$  of  $F_2$  to obtain attractors of  $F$ . In general, we have the following equation that formally states that attractors of  $F_1 \times F_2$  are given by concatenating attractors of  $F_1$  and  $F_2$ .

$$\mathcal{A}(F_1 \times F_2) = \mathcal{A}(F_1) \times \mathcal{A}(F_2). \quad (2.6)$$

The computation of the attractors of  $F$  becomes more complicated when  $F$  is slightly modified so that  $F(x_1, x_2, x_3, x_4) = (x_2, x_1, x_2x_4, x_3) = F_1 \rtimes_P F_2$ , where  $F_1$  is as before and  $F_2 = (ux_4, x_3)$  with external parameter  $u$  and coupling scheme  $P = \{x_2 \rightarrow u\}$ . Since the coupling between  $F_1$  and  $F_2$  is no longer empty, not every combination of attractors of  $F_1$  and  $F_2$  will result in an attractor of  $F$  (figure 5b). For example,  $(01, 10) \in \mathcal{A}(F_1)$  and  $(01, 10) \in \mathcal{A}(F_2)$  do give rise to an attractor of  $F$ , while  $(01, 10) \in \mathcal{A}(F_1)$  and  $(10, 01) \in \mathcal{A}(F_2)$  do not. The set of attractors,  $\mathcal{A}(F)$ , is the union of  $00 \times 00$ ,  $11 \times \mathcal{A}(F_2)$  and  $(01, 10) \times \{00, (01, 10)\}$ , and is thus a subset of the attractors of the Cartesian product (figure 5a). This is, however, not always the case but depends on the particular coupling between the networks. Hence, equation (2.6) is not valid in general.

In order to study the dynamics of decomposable networks, we need to understand how a trajectory, which describes the behaviour of an ‘upstream’ network at an attractor, influences the dynamics of a ‘downstream’ network. The trajectory of an ‘upstream’  $m$ -node network  $F_1$  at an attractor  $C_1 = (\alpha_1, \dots, \alpha_r)$  can be described by  $(g(t))_{t=0}^\infty$ , a sequence with elements in  $\{0, 1\}^m$ . This trajectory has period  $r$ , the length of the attractor. The dynamics of the ‘downstream’  $n$ -node network  $F_2$  depend on  $F_1$ . Therefore,  $F_2$  is a *non-autonomous Boolean network*, defined by

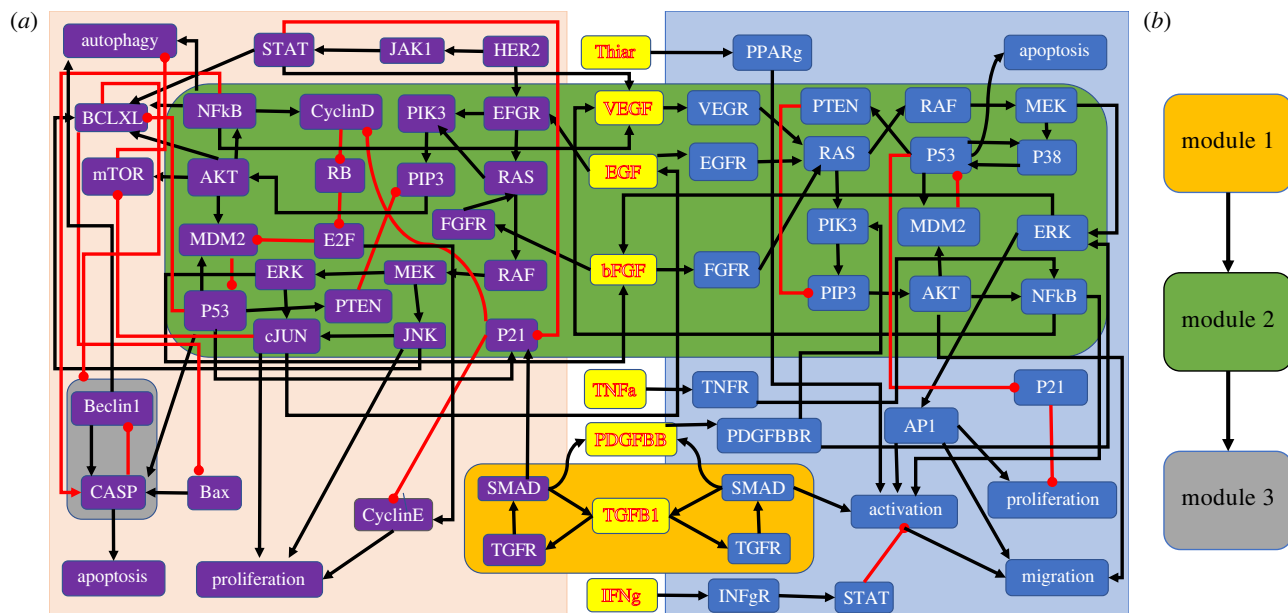
$$y(t+1) = F_2(g(t), y(t)), \quad (2.7)$$

where  $F_2 : \{0, 1\}^{m+n} \rightarrow \{0, 1\}^n$ . Appendix A, §A.2 contains a detailed definition and examples of non-autonomous Boolean networks. To make the dependence of  $F_2$  on the choice of upstream attractor  $C_1 \in C_1$  explicit, we often write  $F_2^{C_1}$  instead of simply  $F_2$ . If  $C_2 = (\beta_1, \dots, \beta_s)$  is an attractor of  $F_2^{C_1}$ , then

$$C_1 \oplus C_2 = ((\alpha_1, \beta_1), (\alpha_2, \beta_2), \dots, (\alpha_{l-1}, \beta_{l-1})) \quad (2.8)$$

is an attractor of the combined network  $F = F_1 \rtimes_P F_2$  of length  $l := \text{lcm}(|C_1|, |C_2|)$ , the least common multiple of  $|C_1|$  and  $|C_2|$ .

Iterating over all attractors of  $F_1$  (that is, all  $C_1 \in \mathcal{A}(F_1)$ ) as well as all attractors of the corresponding non-autonomous



**Figure 6.** A multi-cellular Boolean cancer model [34], which describes the interactions of PCCs (purple nodes), PSCs (blue nodes) and their connecting cytokines (yellow nodes). (a) Wiring diagram describing the regulations between nodes, which are all monotonic, with black and red arrows indicating activation and inhibition, respectively. The non-trivial modules are highlighted by amber, green and grey boxes. (b) Directed acyclic graph describing the connections between the non-trivial modules.

networks  $F_2^{C_1}$  (that is, all  $C_2 \in \mathcal{A}(F_2^{C_1})$ ) yields all attractors of the combined network  $F$ . After the structural decomposition theorem (equation (2.4)), this dynamic decomposition theorem constitutes the second main theoretical result. Mathematically, it can be expressed as

$$\mathcal{A}(F) = \bigsqcup_{C_1 \in \mathcal{A}(F_1)} \bigsqcup_{C_2 \in \mathcal{A}(F_2^{C_1})} C_1 \oplus C_2, \quad (2.9)$$

which can be written as  $\mathcal{A}(F_1) \rtimes_p \mathcal{A}(F_2)$  to highlight the analogy between the structural decomposition of a Boolean network and the decomposition of its dynamics. With this, the dynamic decomposition theorem states  $\mathcal{A}(F_1 \rtimes_p F_2) = \mathcal{A}(F_1) \rtimes_p \mathcal{A}(F_2)$ , which implies a distributive property for the dynamics of decomposable networks. Note that if  $P$  is empty, then  $\mathcal{A}(F_2^{C_1}) = \mathcal{A}(F_2)$  for all  $C_1$  and we recover equation (2.6),  $\mathcal{A}(F_1 \times F_2) = \mathcal{A}(F_1) \times \mathcal{A}(F_2)$ .

The dynamics of a Boolean network  $F$ , which decomposes into modules  $F_1, \dots, F_m$ , can thus be computed from the dynamics of its modules. That is,

$$\mathcal{A}(F) = \mathcal{A}(F_1) \rtimes_{p_1} \left( \mathcal{A}(F_2) \rtimes_{p_2} (\dots \rtimes_{p_{m-1}} \mathcal{A}(F_m)) \right), \quad (2.10)$$

where the placement of the parentheses may be rearranged in any associative manner, just as for the structural decomposition in equation (2.4). Appendix A, §A.2 contains the proof of the dynamic decomposition theorem as well as instructional examples.

## 2.6. Efficient control of decomposable Boolean networks

The state space of a Boolean network grows exponentially in the number of variables. Therefore, the decomposition theorems can reduce the time needed to perform various computations by orders of magnitude for networks with several larger modules. Besides an efficient strategy to compute all attractors of a Boolean network, the structural

decomposition theorem can also be applied to efficiently identify controls of Boolean networks, a topic that has received recent attention [27–29]. Drug developers wonder, for example, which nodes in a gene regulatory network need to be controlled by an external drug to ensure the network transitions to a desired phenotype, typically corresponding to a specific network attractor.

Two types of control actions are generally considered: edge controls and node controls. For each type of control, one can consider deletions or constant expressions, as defined in [30]. The motivation for considering these control actions is that they represent the common interventions that can be implemented in practice. For instance, edge deletions can be achieved by the use of therapeutic drugs that target specific gene interactions, while node deletions represent the blocking of effects of products of genes associated with these nodes [31,32].

A set of controls  $\mu$  stabilizes a Boolean network at an attractor  $C$  when the resulting network after applying  $\mu$  possesses  $C$  as its only attractor. As described in detail in [33], the decomposition into modules can be used to obtain controls for each module, which can then be combined to obtain a control for an entire network. Specifically, for a decomposable network  $F = F_1 \rtimes_p F_2$ , if  $\mu_1$  is a set of controls that stabilizes  $F_1$  in  $C_1$  and  $\mu_2$  is a control that stabilizes  $F_2^{C_1}$  in  $C_2$ , then  $\mu = \mu_1 \cup \mu_2$  is a set of control that stabilizes  $F$  in  $C = C_1 \oplus C_2$ , as long as  $C_1$  or  $C_2$  is a steady state.

A recently published multi-cellular Boolean network model describes the microenvironment of pancreatic cancer cells (PCCs) by modelling the interactions of PCCs, pancreatic stellate cells (PSCs), and their connecting cytokines [34]. This network has 69 nodes, 114 edges and possesses three non-trivial modules (figure 6a). Figure 6b shows the directed acyclic graph, which describes the connections between the modules.

An effective treatment should induce the cancer cell to undergo apoptosis, which, therefore, represents the desired

attractor of this network. To find a set of controls that stabilizes the network in this attractor, one can exploit the structural decomposition of the network by first controlling the upstream module (module 1), which has four attractors: two steady states and two 3-cycles. This module consists of two feedback loops joined by the node *TGFb1*. It is thus enough to control *TGFb1* to stabilize this module into any of its attractors [35]. Using the methods from Zanudo & Albert [36] or Murrugarra *et al.* [30], the controls of module 2 can be identified. A minimal set of two nodes needs to be controlled to stabilize this module: *RAS* in the pancreatic cell and *RAS* in the stellate cell. After applying these controls, the nodes in the downstream module (module 3) are all already constant and do, therefore, not require additional controls. Using the modular structure of the network, three nodes can be easily identified, which suffice to control the entire network. Notably, this never requires the consideration of the entire network, which saves computation time. Disregarding the decomposition and identifying controls for the whole network instead yields the same minimal set of three controls. However, this may not always be the case. In rare cases, the module-by-module control identification strategy will yield a set of controls that is larger than necessary.

### 3. Discussion

The search for ‘fundamental laws’ has been part of systems biology since its beginning, including features of biological systems that are characteristic of most or all systems of a given type, such as gene regulatory networks. The concept of modularity can be considered as such a feature, and has been studied extensively in several different contexts. Another focus of interest has been the relationship between the structure and function of dynamic networks. The results in this paper in essence provide evidence that modularity is in fact a key feature that connects structure and function of networks.

Systems biology has been a field that is making extensive use of mathematical models as descriptive language and analytic tool. Notions such as dynamic modularity are difficult or impossible to study without the use of mathematical models, as is the relationship between structure and function of networks. A limitation of this approach is of course that published models are partial and simplified representations of the requisite biology, so that caution is required when drawing conclusions. But this approach has yielded useful results in studying motifs in static networks (e.g. [24]). The advantage of a mathematical foundation is that it enables an analytical treatment of concepts that might otherwise have to be studied using heuristics, examples and simulations. This is the essence of our approach in this study. Based on rigorous definitions, we were able to prove the link between structural and functional modularity, as well as the broad application to control of networks. We believe that we have only scratched the surface of results that follow from the mathematical framework we have established. For instance, the flip side of network decomposition is network construction through ‘concatenation’ of modules. This can be done in ways that achieve certain dynamic properties, of potential interest to problems in synthetic biology.

Finally, while we have provided evidence that our concept of structural and functional modularity might have biological

relevance, more work remains to be done. For instance, it would be of interest to investigate the biological features of the individual modules found in the repository of Boolean network models from [16] to investigate whether modules in our definition can be viewed as meaningful biological ‘functional units’. The implications of a functional modular structure also remain to be explored beyond our initial result of control at the modular level. We also believe that many of our results should hold in appropriate form for the modelling framework of ordinary differential equations.

It is also worth observing that our decomposition results do not preclude the existence of emergent properties. Each module is a complex system in itself, capable of exhibiting emergent properties. And as modules perturb other downstream modules, their emergent properties propagate to other modules. Our results simply assert a certain relationship between certain subsystems of the whole system. These subsystems cannot be considered the parts that make up the whole.

## 4. Methods

### 4.1. Meta-analysis of published gene regulatory network models

We used the same repository of 122 published and distinct gene regulatory network models as in [16]. Some of these models include non-essential regulators. That is, a node is included as a regulator in an update rule but a change in this node never affects the update rule. We removed all non-essential regulators from the update rules, before computing for each network the number of genes (i.e. size), the average connectivity, all SCCs, as well as the size of each SCC. From this, we derived the primary metric of interest, the number of non-trivial SCCs. Trivial SCCs consist of one node only. Since SCCs are defined as the largest connected component such that there is a path from every node to every *other* node, it is irrelevant whether the single node in a trivial SCC regulates itself.

The logistic multi-variable regression model, implemented in the Python module `statsmodels.api` is given by

$$\frac{p}{1-p} = e^{\beta_0 + \beta_1 x_1 + \beta_2 x_2}, \quad (4.1)$$

where  $p$  is the probability of a model having multiple non-trivial SCCs, and  $x_1$ ,  $x_2$  are average connectivity and network size.

### 4.2. Generation of Boolean networks for simulation study

To understand the effect of modularity on the phenotypical robustness and the dynamical complexity, we resorted to simulation studies of Boolean networks with a specific structure and a defined number of SCCs (i.e. modules). To reduce the number of potential confounders, we fixed the network size at  $N=60$  and the in-degree of each node at  $n=3$ , which is slightly higher than the average in-degree in published gene regulatory network models [16]. We further considered only nested canalizing update rules (e.g. [37,38] for a definition) since most rules in published gene regulatory networks are of this type [16]. To generate networks with a defined number of  $m \in \{1, 2, \dots, 6\}$  modules, each of which consists of  $N/m$  nodes, we first generated a random directed acyclic graph of  $m$  modules by picking uniformly at random a weakly connected lower triangular binary  $m \times m$ -matrix  $D$  with diagonal entries 1. If  $D_{ij}=1$ , a node in module  $i$  regulates a node in module  $j$ . Otherwise,



there is no connection. To ensure that the number of SCCs was indeed  $m$ , we required each module to be a single SCC. We achieved this by randomly generating wiring diagrams for a module until the wiring diagram was strongly connected (for the sparsest modules (i.e.  $m = 1$ ,  $N/m = 60$ ), this took on average approximately 22 iterations).

### 4.3. Estimating dynamical complexity and phenotypical robustness

The size of the state space of the 60-node Boolean networks used in the computational study prohibits the exhaustive identification of all attractors. To compute all attractors, we could have exploited the decomposition into smaller modules for decomposable networks. However, this does not help with the identification of attractors for non-decomposable networks consisting of a single module of size 60. To avoid introducing any bias by using different methods, we employed the same sampling technique to estimate a lower bound of the number of attractors for each Boolean network. Specifically for each network  $F$ , we generated 500 random initial states  $\mathbf{x}_0 \in \{0, 1\}^{60}$  and continued to synchronously update each  $\mathbf{x}_0$  according to  $F$  (that is,  $\mathbf{x}_{i+1} = F(\mathbf{x}_i)$ ) until a recurring state was found, indicating the arrival at an attractor.

Biologically meaningful attractors ‘attract’ a substantial portion of the state space. With a state space size of  $2^{60}$  and when starting from 500 random initial states, we have a 95% chance of finding an attractor, which attracts 0.6% of the state space and even a 99% chance of finding an attractor, which attracts 0.9% of the state space. Relying on sampling and the resulting lower bound of the number of attractors should, therefore, not limit the validity of our findings.

To estimate the phenotypical robustness, we considered the same 500 random initial states  $\mathbf{x}_0 \in \{0, 1\}^{60}$  and generated for each  $\mathbf{x}_0$  a corresponding state  $\mathbf{y}_0 = \mathbf{x}_0 \oplus \mathbf{e}_i$  by randomly flipping one bit  $i \in \{1, \dots, n\}$  (where  $\mathbf{e}_i$  is the  $i$ th unit vector and  $\oplus$  denotes binary addition). Just as  $\mathbf{x}_0$ , we synchronously updated  $\mathbf{y}_0$  according to  $F$  until it reached an attractor and compared the attractors. As a consequence, all estimated phenotypical robustness values are multiples of  $1/500$ .

**Ethics.** This work did not require ethical approval from a human subject or animal welfare committee.

**Data accessibility.** The source code for the statistical analysis of the expert-curated biological networks and the generation of their directed acyclic graph structure is available from the Github repository: <https://github.com/ckadelka/DesignPrinciplesGeneNetworks> [39]. This repository also contains the rules of all the biological networks.

**Declaration of AI use.** We have not used AI-assisted technologies in creating this article.

**Authors' contributions.** C.K.: conceptualization, formal analysis, methodology, software, visualization, writing—original draft, writing—review and editing; M.W.: conceptualization, formal analysis, methodology, writing—original draft, writing—review and editing; A.V.-C.: conceptualization, formal analysis, methodology, visualization, writing—original draft, writing—review and editing; D.M.: conceptualization, formal analysis, methodology, visualization, writing—original draft, writing—review and editing; R.L.: conceptualization, formal analysis, methodology, writing—original draft, writing—review and editing.

All authors gave final approval for publication and agreed to be held accountable for the work performed therein.

**Conflict of interest declaration.** We declare we have no competing interests.

**Funding.** This work was supported by the Simons foundation (grant nos. 712537 (to C.K.), 850896 (to D.M.), 516088 (to A.V.)); the National Institute of Health (grant no. 1 R01 HL169974-01 (to R.L.)) and the Defense Advanced Research Projects Agency (grant no. HR00112220038 (to R.L.)).

**Acknowledgements.** The authors thank Elena Dimitrova for participating in initial fruitful discussions.

## Appendix A. Mathematical details and supplementary figures

### A.1. Proofs of the structural decomposition theorems

This section contains the proofs of the structural decomposition theorems described in the main text. First, we define in full detail the semi-direct product, used to combine two networks in a hierarchical fashion.

**Definition A.1.** Consider two Boolean networks,

$$F = (f_1, \dots, f_k) : \{0, 1\}^k \rightarrow \{0, 1\}^k,$$

with variables  $x = (x_1, \dots, x_k)$  and

$$G = (g_1, \dots, g_m) : \{0, 1\}^{\ell+m} \rightarrow \{0, 1\}^m,$$

with external inputs  $u = (u_1, \dots, u_\ell)$  and variables  $y = (y_1, \dots, y_m)$ . Let  $\Lambda \subseteq \{1, \dots, k\}$  such that  $|\Lambda| = \ell$  and define  $x_\Lambda := (x_{\lambda_1}, \dots, x_{\lambda_\ell})$ . Then,

$$H = (h_1, \dots, h_{k+m}) : \{0, 1\}^{k+m} \rightarrow \{0, 1\}^{k+m},$$

defines a combined Boolean network by setting

$$h_i(x, y) = \begin{cases} f_i(x) & \text{if } 1 \leq i \leq k, \\ g_{i-k}(x_\Lambda, y) & \text{if } k+1 \leq i \leq k+m. \end{cases} \quad (\text{A } 1)$$

That is, the variables  $x_\Lambda$  act as the external inputs of  $G$ . The corresponding coupling scheme is defined to be

$$P = \{x_{\lambda_1} \rightarrow u_1, x_{\lambda_2} \rightarrow u_2, \dots, x_{\lambda_\ell} \rightarrow u_\ell\}. \quad (\text{A } 2)$$

We denote  $H$  as  $H := F \rtimes_P G$  and refer to this as the coupling of  $F$  and  $G$  by (the coupling scheme)  $P$  or as the semi-direct product of  $F$  and  $G$  via  $P$ .

**Theorem A.2.** If a Boolean network  $F$  is not a module, then there exist  $F_1, F_2, P$  such that  $F = F_1 \rtimes_P F_2$ . Furthermore, we can find a decomposition such that  $F_1$  is a module.

*Proof.* Let  $F = (f_1, \dots, f_n)$  be a Boolean network with variables  $X = \{x_1, \dots, x_n\}$  and assume  $F$  is not a module. Then the wiring diagram of  $F$  is not strongly connected, implying there exists at least one node  $y$  and one node  $x_j \neq y$  such that there exists no path from  $x_j$  to  $y$  in the wiring diagram of  $F$ . Let  $X_2 = \{x_{j_1}, x_{j_2}, \dots, x_{j_m}\}$  denote the set of all such nodes, i.e. the nodes for which there exists no paths to  $y$ . Further, let  $X_1 = X \setminus X_2$  denote the complement set of nodes to  $X_2$ . Note that for every  $x_i \in X_1$ , there exists a path from  $x_i$  to  $y$  but no paths originating from  $X_2$  to  $x_i$ .

Define  $\Lambda$  to be the subset of indices  $\Lambda = \{\lambda_1, \dots, \lambda_\ell\} \subset \{1, \dots, k\}$  such that for each  $\lambda \in \Lambda$  there exists at least one function  $f_{j_\lambda}$  with  $x_{j_\lambda} \in X_2$  which depends on  $x_\lambda$ .

If  $\Lambda = \emptyset$ , then the sets  $X_1$  and  $X_2$  represent two groups of nodes, which are disconnected in the wiring diagram. Hence the network  $F$  is a Cartesian product of  $F_1$  and  $F_2$ . It follows that  $F = F_1 \rtimes_P F_2$  with  $P = \emptyset$ .

If  $\Lambda \neq \emptyset$ , then for any  $x_i \in X_1$ , the corresponding update function  $f_i$  does not depend on  $X_2$  by construction, as there are no paths from  $X_2$  to  $x_i$ , and we set  $F_1$  to be the restriction of  $F$  to  $X_1$ ,  $(F_1)_i := (F|_{X_1})_i = f_i$ . For any  $x_i \in X_2$ , if the corresponding update function depends on a node  $x_j \in X_1$ , then  $x_j \in \Lambda$  by the definition of  $\Lambda$ . It follows by construction that

any function  $f_i$  then can be written as a Boolean function on  $X_2$  with external inputs from  $x_A$ .

Hence,  $F = F_1 \bowtie_p F_2$ . Note that in the above proof we can choose the node  $y$  such that it belongs to a SCC that receives no edge from any other SCC.  $X_1$  will contain the nodes of this SCC and hence  $F_1$  will be a module. ■

The main structural decomposition theorem follows directly from this:

**Theorem A.3.** For any Boolean network  $F$ , there exist unique modules  $F_1, \dots, F_m$  such that

$$F = F_1 \bowtie_{P_1} (F_2 \bowtie_{P_2} (\dots \bowtie_{P_{m-1}} F_m)), \quad (\text{A } 3)$$

where this representation is unique up to a reordering, which respects the partial order of  $Q$  (equation (2.1)), and the collection of coupling schemes  $P_1, \dots, P_{m-1}$  depends on the particular choice of ordering.

*Proof.* If  $F$  is a module, then  $m = 1$  and the result follows.

If  $F$  is not a module, we use induction on the downstream subnetwork  $F_2$  in theorem A.2 to obtain the result. ■

Consider as an example a network  $F$  with four SCCs,  $F_1, F_2, F_3, F_4$ , where  $F_1$  influences both  $F_2$  and  $F_3$ ,  $F_2$  and  $F_3$  both influence  $F_4$ , but  $F_2$  and  $F_3$  have no influence on each other. The network can first be broken up as  $F = F_1 \bowtie_{P_1} G$  where  $G$  represents the downstream network of  $F_2, F_3, F_4$  and  $P_1$  includes the connections from  $F_1 \rightarrow F_2$  and  $F_1 \rightarrow F_3$ . In turn,  $G$  can be decomposed as  $G = F_2 \bowtie_{P_2} H$ , where  $H$  is the network consisting of  $F_3$  and  $F_4$ , and  $P_2$  denotes the connections from  $F_2 \rightarrow F_4$ . Finally,  $H$  can be decomposed as  $H = F_3 \bowtie_{P_3} F_4$  where  $P_3$  represents the connections from  $F_3 \rightarrow F_4$ . The final decomposition can thus be written as

$$F = F_1 \bowtie_{P_1} (F_2 \bowtie_{P_2} (F_3 \bowtie_{P_3} F_4)).$$

Alternatively, we could have realized the decomposition of  $G$  as  $G = F_3 \bowtie_{P_3} (F_2 \bowtie_{P_2} F_4)$ . The final decomposition then takes the form

$$F = F_1 \bowtie_{P_1} (F_3 \bowtie_{P_3} (F_2 \bowtie_{P_2} F_4)).$$

The ambiguity of choice for decomposing  $G$  arises from the ambiguity of choosing a *total* order for the *partially* ordered set  $Q = \{W_1 \rightarrow W_2, W_1 \rightarrow W_3, W_2 \rightarrow W_4, W_3 \rightarrow W_4\}$ . Both decompositions are equally valid, and the ordering of the modules in each representation respects the partial order  $Q$ .

## A.2. Non-autonomous Boolean networks

This section contains the full definition of non-autonomous Boolean networks, as well as two examples.

**Definition A.4.** A non-autonomous Boolean network is defined by

$$y(t+1) = H(g(t), y(t)), \quad (\text{A } 4)$$

where  $H : \{0, 1\}^{k+m} \rightarrow \{0, 1\}^m$  and  $(g(t))_{t=0}^\infty$  is a sequence with elements in  $\{0, 1\}^k$ . The network, denoted  $H^g$ , is non-autonomous because its dynamics depend on  $g(t)$ .

A state  $c \in \{0, 1\}^n$  is a *steady state* of  $H^g$  if  $H(g(t), c) = c$  for all  $t$ . Similarly, an ordered set with  $r$  elements,

$C = \{c_1, \dots, c_r\}$  is an *attractor* of length  $r$  of  $H^g$  if  $c_2 = H(g(1), c_1)$ ,  $c_3 = H(g(2), c_2), \dots$ ,  $c_r = H(g(r-1), c_{r-1})$ ,  $c_1 = H(g(r), c_r)$ ,  $c_2 = H(g(r+1), c_1), \dots$ . In general,  $g(t)$  is not necessarily of period  $r$  and may even not be periodic.

If  $H(g(t), y) = G(y)$  for some network  $G$  for all  $t$  (that is, it does not depend on  $g(t)$ ), then  $y(t+1) = H(g(t), y(t)) = G(y(t))$  and this definition of attractors coincides with the classical definition of attractors for (autonomous) Boolean networks.

**Example A.5.** Consider the non-autonomous network defined by

$$H(u_1, u_2, y_1, y_2) = (u_2 y_2, y_1),$$

and the two-periodic sequence  $(g(t))_{t=0}^\infty = (01, 10, 01, 10, \dots)$ , which corresponds to a 2-cycle of the upstream 2-node network. If the initial point is  $y(0) = (y_1^*, y_2^*)$ , then the dynamics of  $H^g$  can be computed as follows:

$$y(1) = H(g(0), y(0)) = H(0, 1, y_1^*, y_2^*) = (y_2^*, y_1^*),$$

$$y(2) = H(g(1), y(1)) = H(1, 0, y_2^*, y_1^*) = (0, y_2^*)$$

$$\text{and } y(3) = H(g(2), y(2)) = H(0, 1, 0, y_2^*) = (y_2^*, 0).$$

Thus for  $t \geq 1$ ,  $y(2t) = (0, y_2^*)$  and  $y(2t+1) = (y_2^*, 0)$ . It follows that the attractors of  $H^g$  are given by 00 (one steady state) and (01, 10) (one cycle of length 2). Note that (10, 01) is not an attractor because (10, 01, 10, 01, ...) is not a trajectory for this non-autonomous network. This is a subtle situation that can be sometimes missed when not considering all trajectories a limit cycle represents.

**Example A.6.** Consider the non-autonomous network defined by  $H(u_1, u_2, y_1, y_2) = (u_2 y_2, y_1)$ , as in the previous example, and the one-periodic sequence  $(g(t))_{t=0}^\infty = (00, 00, \dots)$ , which corresponds to a steady state of the upstream 2-node network. If the initial point is  $y(0) = (y_1^*, y_2^*)$ , then the dynamics of  $H^g$  can be computed as follows:

$$y(1) = H(g(0), y(0)) = H(0, 0, y_1^*, y_2^*) = (0, y_1^*)$$

and

$$y(2) = H(g(1), y(1)) = H(0, 0, y_2^*, y_1^*) = (0, 0).$$

Then,  $y(t) = (0, 0)$  for  $t \geq 2$ , and the only attractor of  $H^g$  is the steady state 00.

## A.3. Proof of the dynamic decomposition theorem

For a decomposable network  $F = F_1 \bowtie_p F_2$ , we introduce the following notation for attractors. First, note that  $F$  has the form  $F(x, y) = (F_1(x), F_2(x, y))$  where  $F_2$  is a non-autonomous network. Let  $C_1 = (r_1, \dots, r_m) \in \mathcal{A}(F_1)$  and  $C_2 = (s_1, \dots, s_n) \in \mathcal{A}(F_2^{C_1})$  be attractors of length  $m$  and  $n$ , respectively. Then, the sequence  $((r_i, s_i))_{i=0}^\infty$  has period  $l = \text{lcm}(m, n)$ , so we define the sum (or concatenation) of these attractors to be

$$C_1 \oplus C_2 = ((r_1, s_1), (r_2, s_2), \dots, (r_{l-1}, s_{l-1})). \quad (\text{A } 5)$$

Note that the sum of attractors is not a Cartesian product,  $C_1 \times C_2 = \{(r_i, s_j) \mid \text{for all } i, j\}$ .

Similarly, for an attractor  $C_1$  and a collection of attractors  $A$  we define

$$C_1 \oplus A = \{C_1 \oplus C_2 \mid C_2 \in A\}. \quad (\text{A } 6)$$

Our second main theoretical result shows that the dynamics (i.e. the attractor space) of a semi-direct product can be seen as a type of semi-direct product of the dynamics of the decomposable subnetworks. When applied iteratively, this enables a computation of the attractor space from the attractor space of each module.

**Theorem A.7.** Let  $F = F_1 \rtimes F_2$  be a decomposable network. Then

$$\mathcal{A}(F) = \bigsqcup_{C_1 \in \mathcal{A}(F_1)} C_1 \oplus \mathcal{A}(F_2^{C_1}) = \bigsqcup_{C_1 \in \mathcal{A}(F_1)} \bigsqcup_{C_2 \in \mathcal{A}(F_2^{C_1})} C_1 \oplus C_2. \quad (\text{A.7})$$

*Proof.* Let  $X_1$  and  $X_2$  be the variables of  $F_1$  and  $F_2$ , respectively. Further, let  $\mathcal{C} = \{c_1, \dots, c_l\} \in \mathcal{A}(F)$  be an arbitrary attractor of  $F$  with length  $l$ . We can define  $C_1 = \text{pr}_1(\mathcal{C}) = (\text{pr}_1(c_1), \dots, \text{pr}_1(c_l)) =: (c_1^1, \dots, c_l^1)$  as the projection of  $\mathcal{C}$  onto  $X_1$ , and similarly  $C_2 = \text{pr}_2(\mathcal{C}) =: (c_1^2, \dots, c_l^2)$  as the projection of  $\mathcal{C}$  onto  $X_2$ . By definition,  $F_1$  does not depend on  $X_2$ . Thus,  $F_1(\text{pr}_1(x)) = \text{pr}_1(F(x))$ , and for any  $c_j^1$

$$F_1(c_j^1) = F_1(\text{pr}_1(c_j)) = \text{pr}_1(F(c_j)) = \text{pr}_1(c_{j+1}) = c_{j+1}^1.$$

Iterating this, we find that in general  $F_1^k(c_j^1) = c_{j+k}^1$ , from which it follows that  $C_1 \in \mathcal{A}(F_1)$ .

Next, we consider the non-autonomous network  $F_2^{C_1}$  defined as in definition A.4 where  $y(t+1) = \text{pr}_2 F(g(t), y(t))$ , and  $g(t) = c_t^1$ . If  $y(1) = c_1^2$ , then

$$y(2) = \text{pr}_2 F(g(1), c_1^2) = \text{pr}_2 F(c_1^1, c_1^2) = \text{pr}_2 F(c_1) = \text{pr}_2(c_2) = c_2^2$$

and in general

$$y(k+1) = \text{pr}_2 F(g(k), c_k^2) = \text{pr}_2 F(c_k^1, c_k^2) = \text{pr}_2 c_{k+1} = c_{k+1}^2$$

Hence  $y(l+1) = \text{pr}_2 F(c_l) = \text{pr}_2 c_1 = y(1)$  and thus  $C_2 \in \mathcal{A}(F_2^{C_1})$ . From this we have that  $\mathcal{C} = C_1 \oplus C_2 \in C_1 \oplus \mathcal{A}(F_2^{C_1})$  and thus

$$\mathcal{A}(F) \subset \bigsqcup_{C_1 \in \mathcal{A}(F_1)} C_1 \oplus \mathcal{A}(F_2^{C_1}).$$

Conversely, let  $C_1 \in \mathcal{A}(F_1)$  and  $C_2 \in \mathcal{A}(F_2^{C_1})$ . We want to show that  $C_1 \oplus C_2 \in \mathcal{A}(F)$ . Let  $g(t) = c_t^1$ ,  $y(1) = c_1^2$ , and  $y(t+1) = \text{pr}_2 F(g(t), y(t))$ . Since  $C_2 \in \mathcal{A}(F_2^{C_1})$ , then  $y(t+1) = c_{t+1}^2$  by definition. Let  $N = |C_2|$ . Then

$$\begin{aligned} F(c_k^1, c_k^2) &= (\text{pr}_1 F(c_k^1, c_k^2), \text{pr}_2 F(g(k), y(k))) \\ &= (F_1(c_k^1), F_2^{C_1}(c_k^1, y(k+1))) \\ &= (c_{k+1}^1, c_{k+1}^2). \end{aligned}$$

Thus  $F^N(c_1^1, c_1^2) = F(c_N^1, c_N^2) = (c_1^1, c_1^2)$  and hence  $C_1 \oplus C_2 \in \mathcal{A}(F)$ . It follows that

$$\bigsqcup_{C_1 \in \mathcal{A}(F_1)} C_1 \oplus \mathcal{A}(F_2^{C_1}) \subset \mathcal{A}(F),$$

from which we can conclude that the sets are equal. ■

The following two examples highlight how theorem A.7 enables the computation of the dynamics of a decomposable network from the dynamics of its modules. To match attractors from the upstream module with the attractor spaces of the corresponding non-autonomous downstream networks, it is useful to consider the space of attractors in a specified order: we use parentheses (curly brackets) to denote

an ordered (unordered) space of attractors. If there is no ambiguity, in practice we can use  $\rtimes$  instead of  $\rtimes_p$ .

**Example A.8.** Consider the Boolean network  $F(x_1, x_2, y_1, y_2) = (x_2, x_1, x_2 y_2, y_1)$ . We can decompose  $F = F_1 \rtimes F_2$  where  $F_1(x_1, x_2) = (x_2, x_1)$  is an upstream module and  $F_2(u_2, y_1, y_2) = (u_2 y_2, y_1)$  is a downstream module with external parameter  $x_2$ . To find all attractors of  $F$  by using theorem A.7, we find the attractors of  $F_1$  and the attractors of  $F_2$  induced by each of those attractors. It is easy to see that  $\mathcal{A}(F_1) = \{00, 11, \{01, 10\}\}$  (where we denote steady states  $\mathcal{C} = \{c\}$  simply by  $c$ ).

— For  $C_1 = 00$ , the corresponding non-autonomous network is  $y(t+1) = F_2(0, 0, y(t))$ . If  $y(0) = (y_1^*, y_2^*)$ , then

$$y(1) = F_2(0, 0, y_1^*, y_2^*) = (0, y_1^*)$$

and

$$y(2) = F_2(0, 0, 0, y_1^*) = (0, 0).$$

Thus, the space of attractors for  $F_2^{C_1}$  is

$$\mathcal{A}(F_2^{C_1}) = \{00\}.$$

— For  $C_2 = 11$ , the corresponding non-autonomous network is  $y(t+1) = F_2(1, 1, y(t))$ . If  $y(0) = (y_1^*, y_2^*)$ , then

$$y(1) = F_2(1, 1, y_1^*, y_2^*) = (y_2^*, y_1^*)$$

and

$$y(2) = F_2(1, 1, y_2^*, y_1^*) = (y_1^*, y_2^*).$$

Thus, the corresponding space of attractor is

$$\mathcal{A}(F_2^{C_2}) = \{00, 11, (01, 10)\}.$$

— For  $C_3 = (01, 10)$ , we define  $g(t): \mathbb{N} \rightarrow \{0, 1\}^2$  by  $g(0) = (0, 1)$ ,  $g(1) = (1, 0)$ , and  $g(t+2) = g(t)$ .  $F_2^{C_3}$  is given by  $y(t+1) = F_2(g(t), y(t))$ . If  $y(0) = (y_1^*, y_2^*)$ , then

$$y(1) = F_2(0, 1, y_1^*, y_2^*) = (y_2^*, y_1^*),$$

$$y(2) = F_2(1, 0, y_2^*, y_1^*) = (0, y_2^*),$$

$$y(3) = F_2(0, 1, 0, y_2^*) = (y_2^*, 0)$$

and

$$y(4) = F_2(1, 0, y_2^*, 0) = (0, y_2^*).$$

Then, the corresponding space of attractors is

$$\mathcal{A}(F_2^{C_3}) = \{00, (01, 10)\}.$$

To reconstruct the entire space of attractors for  $F$ , we have

$$\begin{aligned} \mathcal{A}(F) &= \mathcal{A}(F_1) \rtimes \mathcal{A}(F_2) \\ &= (00, 11, (01, 10)) \rtimes (\mathcal{A}(F_2^{C_1}), \mathcal{A}(F_2^{C_2}), \mathcal{A}(F_2^{C_3})) \\ &= 00 \oplus \{00\} \cup 11 \oplus \{00, 11, (01, 10)\} \cup (01, 10) \oplus \{00, (01, 10)\} \\ &= \{0000, 1100, 1111, (1101, 1110), (0100, 1000), (0101, 1010)\}, \end{aligned}$$

which agrees with the space of attractors shown in figure 5b.

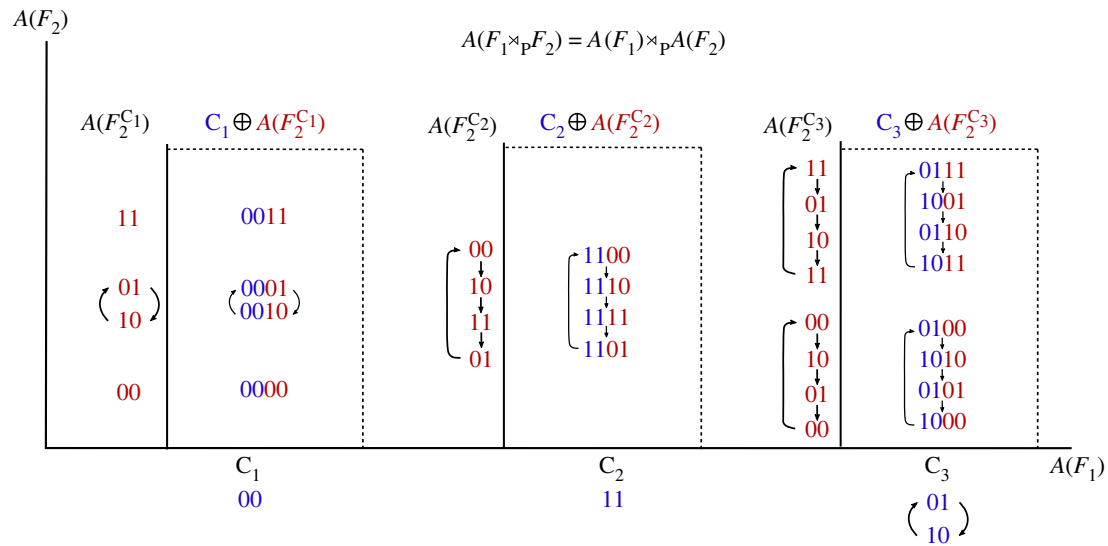
**Example A.9.** Consider the linear Boolean network

$$F(x_1, x_2, y_1, y_2) = (x_2, x_1, x_2 + y_2, y_1).$$

We can decompose  $F = F_1 \rtimes F_2$  into modules  $F_1(x_1, x_2) = (x_2, x_1)$  and  $F_2(u_2, y_1, y_2) = (u_2 + y_2, y_1)$ . The space of attractors of the upstream module  $F_1$  is

$$\mathcal{A}(F_1) = \{00, 11, (01, 10)\}.$$

Using the dynamic decomposition theorem (theorem A.7), we can identify all attractors of  $F$  as follows (see figure 7 for a graphical description).



**Figure 7.** Graphical description of the dynamic decomposition theorem (applied to example A.9). The dynamics of  $F_1 \times_P F_2$  can be seen as a semi-direct product between the dynamics of  $F_1$  and the dynamics of  $F_2$  induced by  $F_1$  via the coupling scheme  $P$ . The dynamics of  $F_2$  induced by attractors of  $F_1$  can vary, and the dynamic decomposition theorem (theorem A.7) shows precisely how to combine all of these attractors.

- For  $C_1 = 00$ , the corresponding non-autonomous network is  $y(t+1) = \bar{F}_2(0, 0, y(t))$ . If  $y(0) = (y_1^*, y_2^*)$ , then  $y(1) = \bar{F}_2(0, 0, y_1^*, y_2^*) = (y_2^*, y_1^*)$ . Thus, the space of attractors for  $F_2^{C_1}$  is

$$\mathcal{A}(F_2^{C_1}) = \{00, 11, (01, 10)\}.$$

- Similarly, for  $C_2 = 11$ , we find that the space of attractors for  $F_2^{C_2}$  is

$$\mathcal{A}(F_2^{C_2}) = \{(00, 10, 11, 01)\}.$$

- For  $C_3 = (01, 10)$ , we define  $g(t) : \mathbb{N} \rightarrow X_1$  by  $g(0) = (0, 1)$ ,  $g(1) = (1, 0)$ , and  $g(t+2) = g(t)$ .  $F_2^{C_3}$  is given by  $y(t+1) = F_2(g(t), y(t))$ . If  $y(0) = (y_1^*, y_2^*)$ , then

$$\begin{aligned} y(1) &= (1 + y_2^*, y_1^*), \\ y(2) &= (y_1^*, y_2^* + 1), \\ y(3) &= (y_2^*, y_1^*) \\ \text{and} \quad y(4) &= (y_1^*, y_2^*) = y(0), \end{aligned}$$

and in general for  $t > 0$ ,

$$\begin{aligned} y(4t) &= (y_1^*, y_2^*), \\ y(4t+1) &= (1 + y_2^*, y_1^*), \\ y(4t+2) &= (y_1^*, y_2^* + 1) \\ \text{and} \quad y(4t+3) &= (y_2^*, y_1^*). \end{aligned}$$

It follows that there are only two periodic trajectories in this case:  $(00, 10, 01, 00, 00, 10, 01, 00, \dots)$  and  $(11, 01, 10, 11, 11, 01, 10, 11, \dots)$ , which both have period 4. The corresponding attractor space is

$$\mathcal{A}(F_2^{C_3}) = \{(00, 10, 01, 00), (11, 01, 10, 11)\}.$$

Note that the repetition of certain states is needed to obtain the correct attractors of the full network  $F$ .

To reconstruct the space of all attractors for  $F$ , we have

$$\begin{aligned} \mathcal{A}(F) &= (00, 11, (01, 10)) \times (\mathcal{A}(F_2^{C_1}), \mathcal{A}(F_2^{C_2}), \mathcal{A}(F_2^{C_3})) \\ &= \left\{ \begin{array}{l} 00 \oplus \{00, 11, (01, 10)\} \\ 11 \oplus \{(00, 10, 11, 01)\} \\ (01, 10) \oplus \{(00, 10, 01, 00), (11, 01, 10, 11)\} \end{array} \right\} \\ &= \left\{ \begin{array}{l} 0000, 0011, (0001, 0010), \\ (1100, 1110, 1111, 1101), \\ (0100, 1010, 0101, 1000), \\ (0111, 1001, 0110, 1011) \end{array} \right\}. \end{aligned}$$

The linear network  $F$  possesses thus two steady states, one 2-cycle and three 4-cycles.

## References

- Hartwell LH, Hopfield JJ, Leibler S, Murray AW. 1999 From modular to molecular cell biology. *Nature* **402**, C47–C52. (doi:10.1038/35011540)
- Hernández U, Posadas-Vidales L, Espinosa-Soto C. 2022 On the effects of the modularity of gene regulatory networks on phenotypic variability and its association with robustness. *Biosystems* **212**, 104586. (doi:10.1016/j.biosystems.2021.104586)
- Lorenz DM, Jeng A, Deem MW. 2011 The emergence of modularity in biological systems. *Phys. Life Rev.* **8**, 129–160. (doi:10.1016/j.plrev.2011.02.003)
- Leicht EA, Newman ME. 2008 Community structure in directed networks. *Phys. Rev. Lett.* **100**, 118703. (doi:10.1103/PhysRevLett.100.118703)
- Malliaros FD, Vazirgiannis M. 2013 Clustering and community detection in directed networks: a survey. *Phys. Rep.* **533**, 95–142. (doi:10.1016/j.physrep.2013.08.002)
- Zhang B, Horvath S. 2005 A general framework for weighted gene co-expression network analysis. *Stat. Appl. Genet. Mol. Biol.* **4**, 17. (doi:10.2202/1544-6115.1128)
- Alexander RP, Kim PM, Emonet T, Gerstein MB. 2009 Understanding modularity in molecular networks requires dynamics. *Sci. Signal.* **2**, pe44. (doi:10.1126/scisignal.281pe44)
- Jimenez A, Cotterell J, Munteanu A, Sharpe J. 2017 A spectrum of modularity in multi-functional gene circuits. *Mol. Syst. Biol.* **13**, 925. (doi:10.15252/msb.20167347)



9. Verd B, Monk NA, Jaeger J. 2019 Modularity, criticality, and evolvability of a developmental gene regulatory network. *Elife* **8**, e42832. (doi:10.7554/eLife.42832)
10. Deritei D, Aird WC, Ercsey-Ravasz M, Regan ER. 2016 Principles of dynamical modularity in biological regulatory networks. *Sci. Rep.* **6**, 21957. (doi:10.1038/srep21957)
11. Wagner GP, Pavlicev M, Cheverud JM. 2007 The road to modularity. *Nat. Rev. Genet.* **8**, 921–931. (doi:10.1038/nrg2267)
12. Gilarranz LJ, Rayfield B, Liñán-Cembrano G, Bascompte J, Gonzalez A. 2017 Effects of network modularity on the spread of perturbation impact in experimental metapopulations. *Science* **357**, 199–201. (doi:10.1126/science.aal4122)
13. Akutsu T, Kosub S, Melkman AA, Tamura T. 2012 Finding a periodic attractor of a Boolean network. *IEEE/ACM Trans. Comput. Biol. Bioinf.* **9**, 1410–1421. (doi:10.1109/TCBB.2012.87)
14. Mizera A, Pang J, Qu H, Yuan Q. 2019 Taming asynchrony for attractor detection in large Boolean networks. *IEEE/ACM Trans. Comput. Biol. Bioinf.* **16**, 31–42. (doi:10.1109/TCBB.2018.2850901)
15. Schwab JD, Kühlwein SD, Ikononi N, Kühl M, Kestler HA. 2020 Concepts in Boolean network modeling: what do they all mean?. *Comput. Struct. Biotechnol. J.* **18**, 571–582. (doi:10.1016/j.csbj.2020.03.001)
16. Kadelka C, Butrie TM, Hilton E, Kinseth J, Serdarevic H. 2020 A meta-analysis of Boolean network models reveals design principles of gene regulatory networks. *arXiv*. (<http://arxiv.org/abs/2009.01216>)
17. Madrahimov A, Helikar T, Kowal B, Lu G, Rogers J. 2013 Dynamics of influenza virus and human host interactions during infection and replication cycle. *Bull. Math. Biol.* **75**, 988–1011. (doi:10.1007/s11538-012-9777-2)
18. Pušnik Z, Mraz M, Zimic N, Moškon M. 2022 Review and assessment of Boolean approaches for inference of gene regulatory networks. *Heliyon* **8**, e10222. (doi:10.1016/j.heliyon.2022.e10222)
19. Lee WP, Tzou WS. 2009 Computational methods for discovering gene networks from expression data. *Brief. Bioinform.* **10**, 408–423. (doi:10.1093/bib/bbp028)
20. Pratapa A, Jaliha AP, Law JN, Bharadwaj A, Murali T. 2020 Benchmarking algorithms for gene regulatory network inference from single-cell transcriptomic data. *Nat. Methods* **17**, 147–154. (doi:10.1038/s41592-019-0690-6)
21. Beneš N, Brim L, Huvar O, Pastva S, Šafránek D. 2023 Boolean network sketches: a unifying framework for logical model inference. *Bioinformatics* **39**, btad158. (doi:10.1093/bioinformatics/btad158)
22. Wagner GP. 2014 *Homology, genes, and evolutionary innovation*. Princeton, NJ: Princeton University Press.
23. Halfon MS. 2017 Perspectives on gene regulatory network evolution. *Trends Genet.* **33**, 436–447. (doi:10.1016/j.tig.2017.04.005)
24. Alon U. 2003 Biological networks: the tinkerer as an engineer. *Science* **301**, 1866–1867. (doi:10.1126/science.1089072)
25. Klemm K, Bornholdt S. 2005 Topology of biological networks and reliability of information processing. *Proc. Natl Acad. Sci. USA* **102**, 18 414–18 419. (doi:10.1073/pnas.0509132102)
26. Derrida B, Weisbuch G. 1986 Evolution of overlaps between configurations in random Boolean networks. *J. Phys.* **47**, 1297–1303. (doi:10.1051/jphys:019860047080129700)
27. Borriello E, Daniels BC. 2021 The basis of easy controllability in Boolean networks. *Nat. Commun.* **12**, 1–15. (doi:10.1038/s41467-021-25533-3)
28. Rozum J, Albert R. 2022 Leveraging network structure in nonlinear control. *NPJ Syst. Biol. Appl.* **8**, 36. (doi:10.1038/s41540-022-00249-2)
29. Paul S, Su C, Pang J, Mizera A. 2018 A decomposition-based approach towards the control of Boolean networks. In *Proc. of the 2018 ACM Int. Conf. on Bioinformatics, Computational Biology, and Health Informatics, Washington, DC, 29 August–1 September*, pp. 11–20. New York, NY: ACM. (doi:10.1145/3233547.3233550)
30. Murrugarra D, Veliz-Cuba A, Aguilar B, Laubenbacher R. 2016 Identification of control targets in Boolean molecular network models via computational algebra. *BMC Syst. Biol.* **10**, 94. (doi:10.1186/s12918-016-0332-x)
31. Choi M, Shi J, Jung SH, Chen X, Cho KH. 2012 Attractor landscape analysis reveals feedback loops in the p53 network that control the cellular response to DNA damage. *Sci. Signal.* **5**, ra83. (doi:10.1126/scisignal.2003363)
32. Wooten DJ, Zaňudo JGT, Murrugarra D, Perry AM, Dongari-Bagtzoglou A, Laubenbacher R, Nobile CJ, Albert R. 2021 Mathematical modeling of the *Candida albicans* yeast to hyphal transition reveals novel control strategies. *PLoS Comput. Biol.* **17**, e1008690. (doi:10.1371/journal.pcbi.1008690)
33. Kadelka C, Laubenbacher R, Murrugarra D, Veliz-Cuba A, Wheeler M. 2022 Decomposition of Boolean networks: an approach to modularity of biological systems. *arXiv*. (<http://arxiv.org/abs/2206.04217>)
34. Plaughter D, Murrugarra D. 2021 Modeling the pancreatic cancer microenvironment in search of control targets. *Bull. Math. Biol.* **83**, 1–26. (doi:10.1007/s11538-021-00937-w)
35. Zaňudo JGT, Yang G, Albert R. 2017 Structure-based control of complex networks with nonlinear dynamics. *Proc. Natl Acad. Sci. USA* **114**, 7234–7239. (doi:10.1073/pnas.1617387114)
36. Zanudo JG, Albert R. 2015 Cell fate reprogramming by control of intracellular network dynamics. *PLoS Comput. Biol.* **11**, e1004193. (doi:10.1371/journal.pcbi.1004193)
37. Kauffman S, Peterson C, Samuelsson B, Troein C. 2003 Random Boolean network models and the yeast transcriptional network. *Proc. Natl Acad. Sci. USA* **100**, 14 796–14 799. (doi:10.1073/pnas.2036429100)
38. Li Y, Adeyeye JO, Murrugarra D, Aguilar B, Laubenbacher R. 2013 Boolean nested canalizing functions: a comprehensive analysis. *J. Theor. Comput. Sci.* **481**, 24–36. (doi:10.1016/j.tcs.2013.02.020)
39. Kadelka C, Wheeler M, Veliz-Cuba A, Murrugarra D, Laubenbacher R. 2023 Modularity of biological systems: a link between structure and function. Github repository. (<https://github.com/ckadelka/DesignPrinciplesGeneNetworks>)