Breakdown of modularity in complex networks

Sergi Valverde*1,2,3
1ICREA-Complex Systems Lab, Universitat Pompeu Fabra, Dr Aiguader 88, 08003 Barcelona, Spain
2Institute of Evolutionary Biology, Psg Barcelona, UPF-CSIC, Barcelona
3European Centre for Living Technology, S. Marco 2940, 30124, Venice, Italy

I. INTRODUCTION

Complex networks pervade the evolution and organisation of a wide range of systems, from cellular or brain webs to technological graphs. Their structure has important consequences for their stability, resilience and fragility. Some particular properties of these networks are very common, such as the presence of modular organisation (1–4). In modular webs, different subsets of nodes display a higher integration among them than with the rest of the system. This feature allows some degree of segregation between different parts of the network and has been suggested to be a prerequisite for the evolvability of biological systems. In technology, modularity defines a clear division of tasks and it is an explicit design target. However, many natural and artificial systems experience a breakdown in their modular pattern of connections, which has been associated to failures in hub nodes or the activation of global stress responses. In spite of its importance, no general theory of the breakdown of modularity and its implications has been advanced yet. Here we propose a new, simple model of network landscape where it is possible to exhaustively characterise the breakdown of modularity in a well-defined way. We found that evolution cannot reach maximally modular networks under the presence of functional and cost constraints, implying the breakdown of modularity is an adaptive feature.

Keywords: Modularity, Phenotype Network, Boolean Function, Evolution.

The presence of modular organisation is a common property of a wide range of complex systems, from cellular or brain networks to technological graphs. Modularity allows some degree of segregation between different parts of the network and has been suggested to be a prerequisite for the evolvability of biological systems. In technology, modularity defines a clear division of tasks and it is an explicit design target. However, many natural and artificial systems experience a breakdown in their modular pattern of connections, which has been associated to failures in hub nodes or the activation of global stress responses. In spite of its importance, no general theory of the breakdown of modularity and its implications has been advanced yet. Here we propose a new, simple model of network landscape where it is possible to exhaustively characterise the breakdown of modularity in a well-defined way. We found that evolution cannot reach maximally modular networks under the presence of functional and cost constraints, implying the breakdown of modularity is an adaptive feature.

Keywords: Modularity, Phenotype Network, Boolean Function, Evolution.

I. INTRODUCTION

Complex networks pervade the evolution and organisation of a wide range of systems, from cellular or brain webs to technological graphs. Their structure has important consequences for their stability, resilience and fragility. Some particular properties of these networks are very common, such as the presence of modular organisation (1–4). In modular webs, different subsets of nodes display a higher integration among them than with the rest of the system. This feature allows some degree of segregation between different parts of the network and has been suggested to be a prerequisite for the evolvability of biological webs (5). Within the context of technological evolution, modular structures have been often proposed as a target for engineering design.

Modules are also expected to play a key role in providing a source of specialisation, while their proper interconnection guarantee integration at the system-level scale. Both are needed in order to sustain proper functionality and we need to understand both how modules are generated and how their disconnection leads to functional decay. An illustrative example is provided by brain network topology or the so called connectome (6). Connectomics has been a major breakthrough in pushing forward a new approach to brain disease where both brain areas and their connectivity patterns become integrated in a single picture. Under this view, neurological disorders including Alzheimer’s disease or schizophrenia to challenged healthy cognition, such as in sleep or awareness, can be understood in terms of faulty intermodular communication (7–10). This failure can lead to the so called breakdown of modularity (BM) first proposed in (11). It involves a transition from high modularity to low modularity. Similar patterns can be found in other areas, but no general theory of this phenomenon and its implications has been advanced yet.

In spite of its importance, BM has received little attention and it is not well-understood. One important reason of this is connected with the difficulties associated to understanding the mapping between structure (genotype) and function (phenotype) in evolved networks. This is specially difficult when dealing with a property as modular structure, and the need for understanding how and when modular networks are expected to evolve and how optimality is tied to modular architecture. For example, it has been suggested that networks evolved under "modularly varying goals" must be modular (17). Specifically, computational experiments showed that optimal networks are non-modular whenever the goal was kept fixed or under randomly varying goals (with no common subgoals). However, Clune and co-workers have shown that modular networks evolve even in the presence of fixed and modular input-output mappings (18). Here modular patterns would be the byproduct of a cost-dependent selection process. More generally, when dealing with evolving networks, an important question is the role played by modular structures in enhancing robust functionalities and how modular structures are associated to evolvable designs. In other words: is the landscape of modularity associated to Boolean functions smooth? Are optimal modular solutions always tied to efficient functions and evolvable architectures?

In order to address these limitations, a simple case study that can be systematically explored would be desirable. Here we propose such a toy model of network landscapes where it is possible to exhaustively characterise BM in a well-defined way. In this context, it is worth noting that BM seems to be a common feature of computational systems (11; 14). Because computation-
related networks can be seen as instances of functional Boolean processes performed on well-defined circuits, a
minimal case study can help to gain insight into the role played by modular architecture. Specifically, we consider the set of minimal Boolean feed-forward networks implementing the \(2^3 = 256\) Boolean functions \(f_\mu\) with 3 input variables, i.e. the mapping
\[
f_\mu : \Sigma^3 \to \Sigma
\]
where \(\Sigma = \{0, 1\}\) is the Boolean state space. Such kind of Boolean representation has been widely used in the study of evolved networks and in different contexts, including cellular circuits (12) or pattern-forming genetic circuits (13). A systematic exploration requires necessarily a limitation of the combinatorial space to be analysed. However, relevant computational spaces and specific cases can be observed even in the simplest networks (19). Our analysis suggests that the optimisation of specific input-output mappings is not always compatible with highly modular structures and how the BM might be an adaptive feature.

II. FEED-FORWARD BOOLEAN NETWORKS

The model used here is based on Boolean logic (20), which has been used in the modelling and analysis of the flow of information in natural and artificial systems, such as gene regulatory networks (15; 16). A Boolean function can be represented using a truth table, functional forms and networks and it is worth noting that, despite the Boolean picture is a necessarily simple, neural and genetic networks display nonlinear functional responses that ultimately involve an almost all-or-none behaviour. The interactions between these representations reveal the presence of functional constraints in the organization of complex systems (see below). Our truth tables give the value for the function \(f_\mu(a, b, c)\) for each possible combination of the inputs \(a, b,\) and \(c\). The function is identified by its designation number
\[
X = f_\mu(1, 1, 1)f_\mu(1, 1, 0)...f_\mu(0, 0, 0)
\]
that is, the binary sequence of all function values (see Figure 1A). We can achieve more readable (but ambiguous) expressions using functional forms. The full disjunctive normal form (Figure 1A top) is the sum of elementary products (terms) corresponding to input combinations on which \(f_\mu\) is true (minterms). For example, the minterm \(abc\) represents the combination 111, \(ab\neg c\) represents 110 and so on.

A common network representation of a Boolean function is the simple feed-forward Boolean network (FFBN) without no feedback loops. We focus on a subset of the function space \(f_\mu : \Sigma^3 \to \Sigma\) involving all the FFBNs that compute single-output Boolean functions \(f_\mu : \{0, 1\}^3 \to \{0, 1\}\) with \(\nu = 3\) input variables and one output. The FFBN is a directed graph in which all the nodes carry the labels of negative-AND (NAND) gates while input nodes carry the labels of input variables.

Formally, a directed network \(G = (V, E)\) consists of a set of nodes \(v_i \in V\) and a set of edges \((v_i, v_j) \in E\). The adjacency matrix \(A = [A_{ij}]\) has elements such as \(A_{ij} = 1\) if there is link \((i, j) \in E\) and \(A_{ij} = 0\) otherwise. The size of the network is the number of nodes \(N = |V|\) (logic gates) that it contains. The number of links is the sum
\[
L = \sum_{ij} A_{ij}
\]
In the following, we will study the undirected version of the FFBN. The network is undirected if for each edge
(v_i, v_j) \in E$ there is another edge $(v_j, v_i) \in E$. An undirected network has $m = L/2$ edges. We also define the degree $k_i = \sum_j A_{ij}$ as the number of edges attached to the $v_i$ node. The average degree of the network

$$
\langle k \rangle = \frac{1}{N} \sum_i k_i = \frac{L}{N}
$$

measures the overall connectivity in the system and it is one of the main network parameters.

We will also be interested in measuring the flow of information in the network. The path length is a measure of the distance between nodes in the network. The length $l$ of a path is the number of edges traversed along the path. Let’s define $N_{ij}^{(l)}$ as the number of paths of length $l$ that relate any pair of nodes $v_i$ and $v_j$. Among all the alternative paths, we choose the path of minimal distance (or geodesic path), which defines the shortest path distance $d(i, j)$ or the smallest value of $l$ such that $N_{ij}^{(l)} > 0$. The average path length of a graph (or network diameter) provides a measure of network efficiency and it is defined as follows:

$$
\langle d \rangle = \frac{1}{n(n-1)} \sum_{i,j} d(i, j)
$$

(1)

where the normalization term discounts the influence of nodes on themselves.

III. FUNCTIONAL MODULARITY

The main goal of this paper is to characterise the landscape of Boolean functions associated to our system, and specifically the modularity of neighboring functions and how is the modularity of the circuit implementing each function with its one-bit neighbours. Given a decomposition of the network into a set of subgraphs $C_i$, the degree of modularity $Q$ associated to this partition can be measured as follows (28):

$$
Q = \frac{1}{2m} \sum_{i,j} \left[A_{ij} - \frac{k_i k_j}{2m}\right] \delta(C_i, C_j)
$$

(2)

where $k_i = \sum_j A_{ij}$ is the number of connections attached to the $i$-th node (or node degree), $C_i$ is the partition the $i$-th node belongs to and $\delta(x) = 1$ is $x >= 1$ and $\delta(x) = 0$ otherwise. Here, we use the Louvain method for community detection in order to find the optimal partition of the FFBN that maximizes the modularity value (29).

Our hypothesis is that functional requirements constrain structural modularity, that is, evolution cannot reach maximally modular networks under the presence of functional constraints. In this case, we can define a functional (or phenotype) modularity or upper bound for the network (or genotype) modularity (2). However, several genotypes (FFBNs and functional forms) can be found for the same phenotype. When there is representation ambiguity, we often prefer shorter or minimized forms among all the alternatives. For example, a shorter functional form (Figure 1A bottom) divides the support of $f_\mu$ into groups (red circles and blue squares) that exploit functional symmetries (30; 31).

Similarly, engineers are often concerned with the problem of obtaining the most economical design for electronic circuits. We define functional modularity $Q(X)$ as the modularity of the FFBN with minimal cost $L + N$ or sum of number of logical gates and wires (see Figure 1C). The minimization of Boolean functions is a hard problem (in general) and there is no simple way to obtain the optimal solution (20). Following (21), we perform an exhaustive search to obtain the optimal solution for each of the 256 Boolean functions of 3 variables (see SM for a detailed listing).

IV. PHENOTYPE NETWORK

In order to uncover the relationship between modularity and functional requirements, we will make use of the concept of phenotypic network. A phenotypic network is a graph whose nodes represent (in our case) Boolean functions and where two functions are connected if they differ in only one minterm of the full disjunctive normal form. In this metagraph, each node maps a function onto its genotype (a FFBN). The notion of phenotypic network is derived from the conceptual framework associated to genotypic and phenotypic spaces proposed by several authors (22–27). The hypercube is also related to the above definition. The hypercube $Q_\nu$ is a network of $2^\nu$ nodes represented by binary sequences $s_i = (s_{i1}, s_{i2}, s_{i3}, ..., s_{i\nu})$ where $s_{ik} \in \Sigma = \{0,1\}$. The edges of the hypercube connect nodes whose sequences differ in exactly one bit, i.e., there is an edge $(S_i, S_j) \in Q_\nu$ when $d_H(S_i, S_j) = 1$. Let’s define the Hamming distance $d_H(S_i, S_j)$ between any pair of sequences as follows:

$$
d_H(S_i, S_j) = \sum_{k=1}^{\nu} |s_{ik} - s_{jk}|
$$

(3)

In addition, every Boolean function can be represented as a subgraph of the hypercube (35) (see Figure 1B). Functions in the phenotype network are labelled with its designation number (a binary sequence) and thus, adjacent functions can be formally defined using the Hamming distance between the corresponding designation numbers.

The only relevant difference between the standard hypercube and the phenotype network is that the same function can be represented with several nodes (genotypes).

Figure 2 maps the space of all reachable Boolean functions with 3 input variables. Even in this case, which is small and might not seem so relevant, several key classes of functions and circuit designs are involved (see below) either as single networks or as part of larger webs. These
maps reveal several interesting features. First, a few functions appear several times in the network because they have more than one minimal FFBNs. For example, the function $f_{10000110}$ accepts three different genotypes with the same minimal cost (they are displayed at the bottom left of Figure 2). Second, the network only considers 80 different functions out of the 256 logical functions of 3 variables. Two functions are equivalent and belong to the same class if one can be obtained from the other by a permutation of the input variables. We have discarded from any consideration functions that are equivalent to any class representative (21) (the SM lists the equivalence classes of functions).

V. ADAPTATION AND THE BREAKDOWN OF MODULARITY

The phenotype network maps the pathways to an eventual BM. Specifically, we can check if it is possible to evolve a less modular target function from any other source function. The distribution of modularity values $P(Q)$ in this space has a well-defined peak with mean $\langle Q \rangle \approx 0.2$ (see Figure 3A). However, the variance displayed by this distribution suggests the possibility that BM is widespread. Specifically, we can find five functions with minimal modularity ($Q(X) \approx 0$) that can be accessed from different neighbourhoods of the phenotypic space (see yellow region in Figure 3B).

There are also highly modular functions surrounded by modular neighbours. To illustrate this behaviour, we have chosen two important functions, namely the multiplexer (see Figure 4A) and the majority function (see Figure 4B). Both circuits have special relevance in both electronic designs and in synthetic biology (25). Each of these functions can be accessed in a few mutation steps from functions with higher modularity, i.e., $Q(10101110) \approx 0.3$ and $Q(00000111) \approx 0.3$, respectively. At least in these two cases, the evolution of useful functions is coupled to a reduction in modularity. These examples suggest how
adaptation might lead to a BM. A decrease of functional modularity takes place when the evolutionary goal is non-separable, i.e., the computation of the output requires the interaction of several inputs. For example, the majority function is a global computation that combines all input variables to obtain the output. Similarly, consciously effortful tasks (like working memory) are expected to break the modularity of neural systems (38).

How can we relate functional characteristics with the breaking of modularity? Figure 3B shows the absence of a clear correlation between the modularity of specific Boolean functions and the average modularity of its nearest neighbours in the phenotype network. Additional network analyses might be helpful to understand this pattern. In a previous study, we have suggested that the BM is related to the small-world behavior of complex networks (11). Software projects have a natural tendency to become disordered structures (41; 48). This degradation is caused by widespread software changes and indirect dependencies between unrelated pieces of code.

These changes in modular organisation have important implications for both engineering and evolved circuit designs. In this context, it was early suggested that software design is an instance of a multi-objective optimization process (36). When designing software, there is a trade-off between efficient communication and separation of functional tasks (i.e., modularization). Indeed, small-sized software architectures are trees (as one should expect from optimization leading to hierarchical structures) but clustering emerges at larger sizes. As the number of components increases, conflicting constraints arise between different components that would prevent the reaching of an optimal solution. A need to exchange information between distant parts of a system can lead to a modularity reduction.

The minimization of FFBNs is a similar problem. We can check there is a positive correlation between the average path length and the modularity of minimal FFBNs (see Figure 5A). Modularity leads to an enlargement of
the network diameter because there are more pairs of distant nodes across modules than within modules, that is, we can associate the breakdown of modularity to a reduction of the average path length (1) in the network. For example, the addition of a link between the two branches of the multiplexer (see Figure 4A) leads to a sharp decrease of its average path length. Interestingly, highly modular functions are those having the minimal number of dependencies between the variables (and thus low average degree). Highly modular networks have a tree-like architecture (37) while the presence of functional constraints creates a larger density of connections and a more regular (or lattice-like) topology (see Figure 5). That is, the relationship between network efficiency and modularity reflects the trade-off between function and robustness and evolvability.

VI. DISCUSSION

The interplay between fitness and system-level properties such as modularity has been investigated in natural and artificial designs. Simon proposed that nearly decomposable systems composed by independent modules allow faster adaptation to highly fluctuating environments (39; 40). A modular architecture allows independent changes in different parts of the system without affecting the whole. Well-adapted modules are conserved and provide a robust infrastructure for future adaptation. This poses a puzzle because modular architectures cannot be always maintained or reached in artificial evolution. For example, evolutionary algorithms often yield designs that are not decomposable and it is difficult to understand the way these systems work. In software engineering, even if a modular design is provided as initial solution, development rapidly moves to entangled and monolithic solutions (41).

Here we have proposed that the breakdown of modularity takes place because there are changes in the fitness function, e.g., a shift from a well-known environment to a less predictable environment. There are functional constraints to network modularity. For example, tasks involving non-separable input-output mappings like learning a color naming task with interference, evolving a robust metabolic network in a highly fluctuating environment (47) or developing software under constantly changing requirements do not seem to evolve modular networks. Everything is a novelty for a network exposed to a highly fluctuating environment and thus, it makes little sense to maintain costly memories for reusing past information. In a highly fluctuating environment, the only requirement for survival is to issue fast responses and quick adaptations. The lack of memory imposes a strong constraint on the complexity of evolved structures.

The reduction of modularity has to be contrasted with existing theories for the emergence of modularity (51). For example, the breakdown to modularity does not require a change from a modularly changing environment (17) to another static environment. Introducing an additional, temporal dimension in the fitness function is likely to internally decouple the system (and thus creating the possibility of increasing modularity). Moreover, looking at a few case studies does not enable a full understanding of how and when modular networks are expected to evolve. Instead, we have proposed a simple model of network landscape where it is possible to exhaustively characterise the breakdown of modularity in a well-defined way and studied the influence of functional and cost constraints.

The study of biological networks requires the examination of the interactions between modularity, network diameter and function. The phenotype space can be clus-
sified in two types of functions depending if the minimal FFBNs (genotypes) is a sparse or a dense network. Sparse genotypes have treelike topologies and their computation involves minimal input interaction. On the other hand, the modularity of dense networks drops with the increasing number of distant interactions. Some of the sparse networks are also maximally modular in our system, contradicting the intuition that modularity depends on densely intra-connected communities. It can be shown that tree modularity is significant even when they are made of sparse modules (37). Nodes in sparse trees acting as bottlenecks are sufficient to achieve high modularity values. The above suggests that we have to extend our definitions of modularity to take into account different measures of internal network connectivity.

Finally, our results might be useful to understand the limits of the hypothesis put forward by Simon. The analysis of the phenotype network reveals how the breakdown of modularity is more likely to take place from regions of the landscape populated by highly modular functions. Although neutral models suggest that tinkering increases the possibilities to discover such modular designs (49; 50), the structure of the phenotype network suggests that it is not always possible to avoid the breaking of modularity. Sometimes modularity is not so beneficial. The evolution of novelties (52) requires the integration of many different sources of information in order to obtain coherent system changes.

Acknowledgments

I thank Ricard Solé for a careful review of this paper and useful comments and discussions. This work was supported by Botín Foundation by Banco Santander through its Santander Universities Global Division, FIS2016-77447-R from Spain Ministerio de Economía, Industria y Competitividad, AEI/MINEICO/FEDER and UE.

VII. REFERENCES


