

Coevolution of functional flow processing networks

Pablo Kaluza^{1,2,a}

¹ National Scientific and Technical Research Council & Faculty of Exact and Natural Sciences, National University of Cuyo, Padre Contreras 1300, 5500 Mendoza, Argentina

² Abteilung Physikalische Chemie, Fritz-Haber-Institut der Max-Planck-Gesellschaft, Faradayweg 4-6, 14195 Berlin, Germany

Received 20 January 2017 / Received in final form 13 March 2017

Published online 2 May 2017

© The Author(s) 2017. This article is published with open access at Springerlink.com

Abstract. We present a study about the construction of functional flow processing networks that produce prescribed output patterns (target functions). The constructions are performed with a process of mutations and selections by an annealing-like algorithm. We consider the coevolution of the prescribed target functions during the optimization processes. We propose three different paths for these coevolutions in order to evolve from a simple initial function to a more complex final one. We compute several network properties during the optimizations by using the different path-coevolutions as mean values over network ensembles. As a function of the number of iterations of the optimization we find a similar behavior like a phase transition in the network structures. This result can be seen clearly in the mean motif distributions of the constructed networks. Coevolution allows to identify that feed-forward loops are responsible for the development of the temporal response of these systems. Finally, we observe that with a large number of iterations the optimized networks present similar properties despite the path-coevolution we employed.

1 Introduction

Networks play a fundamental role in cell biology and are responsible for a variety of principal cell functions, including genetic expression, metabolism and signal transduction [1,2]. Therefore, full understanding of functionality and dynamics of real bionetworks is a major challenge. Once this is achieved, broad perspectives for manipulation, control and modification of existing biological networks will open. This is the promise of genetic engineering and, generally, of synthetic biology [3].

In contrast to synthetic biology which deals with real biological systems and their physical implementation, constructive biology (see, e.g., Kaneko [4]) operates with idealized models. By choosing elementary mathematical building blocks, which are much simplified abstractions of actual biological elements, one tries to design artificial systems which exhibit the same kinds of functions as their real biological counterparts. As a result, it may be possible to conclude what system properties are generic and what are specific for the details of particular biological organisms. Examples of such approach are shown in reference [4]. Note that methods of constructive biology can also be used to analyze problems of biological evolution [5], including evolution of bionetworks.

When biological organisms evolve, they progressively acquire new functionality. This means that the networks, which are responsible for the functions, have to evolve accordingly. As the functions become more complex, the

underlying networks coevolve, proceeding through a sequence of definite structural changes. Essentially, the aim of our study was to analyze this process using a simple model system.

Flow networks, considered in the present publication, can be viewed as toy models of intracellular signal transduction [6]. The flows are supplied to input nodes, they propagate through a network of distribution nodes connected by pipes and finally arrive at the output nodes, specific for each selected input. It has been shown that, by using evolutionary optimization methods, functional networks, that generate definite output patterns, can be designed [7]. Moreover, networks with high robustness against deletion of links or nodes and introduction of quenched noise have been obtained; statistical analysis of the properties of the designed networks has been performed [8,9]. Additionally, networks which generate programmable time-dependent responses have also been considered [10].

Note that, although evolutionary computer algorithms have been used in such previous studies, the focus has always been on the properties of the final designed networks, not on the evolution process. Here, we want to analyze in detail how the evolution proceeds in a flow network whose functionality changes progressively in time.

Our research yields three major results: (i) the evolution towards a functional network shows a dramatic improvement akin to a phase transition at some moment during the evolution process. The transition can be seen in the time evolution of the output error, the motif distribution

^a e-mail: pkaluza@mendoza-conicet.gob.ar

of the selected networks and in their other statistical measures. After that transition, the statistical properties of the networks that are selected show only small incremental changes; (ii) the network evolution can be monitored through the time evolution of its motif distributions. Characteristic changes in the motif distribution reflect the onset of temporal delays in the target pattern and of increasing spatial complexity; (iii) the networks selected via coevolution are to a large degree path independent: they have the same statistical properties whether the target pattern changes on a time scale of a few hundred iterations or on a time scale of tens of thousands of iterations. This seems to be an important and remarkable property: evolution to a network that is optimally adapted to a given function does not require that this function be stable in time nor do intermediate stages of the network have to be optimally adapted to the intermediate stages of the required function to end up in a network that is optimally adapted to the final function.

This work is organized as following: in Section 2 we present the flow processing networks with time-dependent responses, the different coevolutions of the target patterns and the evolutionary algorithm for the network construction. In Section 3 we present the main results obtained by numerical simulations. Finally in Section 4 we present the results and the discussion.

2 Models

We consider in this work the model of flow processing networks with time-dependent responses presented previously by the authors in reference [10]. We reproduce in this second section the model definition and its main characteristics. Additionally, we present the three different coevolutions for the target patterns starting from simple functionality and evolving to complex tasks in the final stage. Finally, we describe the annealing algorithm we employ for the construction of the functional networks.

2.1 Network architecture

In the considered model, networks transport fluxes from input nodes to output nodes. They have N nodes with a layered-like structure with N_{in} input nodes in the first layer, M middle nodes in the second layer, and, N_{out} output nodes in the third layer. Figure 1 shows an example of such a network.

Connections between the nodes have some restrictions. Input nodes can be connected only to middle nodes. Middle nodes can be connected between them and to output nodes. Output nodes have only input connections coming from middle nodes. In all cases, we allow only one connection with the same direction between two nodes. Self-connections are not allowed. All links have the same capacity (weight). Thus, the network architecture is completely characterized by the adjacency matrix \mathbb{A} , with the elements $A_{ij} = 1$ if link from j to i exists and $A_{ij} = 0$ otherwise.

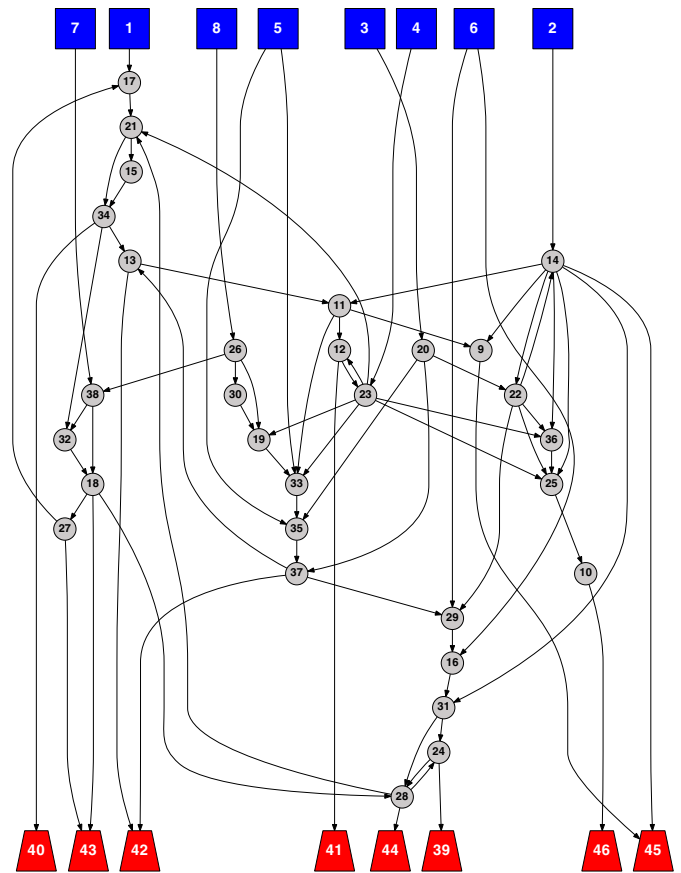


Fig. 1. Example of a flow processing network with the total number of nodes $N = 46$, divided into $N_{in} = 8$ input nodes (blue squares), $M = 30$ middle nodes (gray circles), and $N_{out} = 8$ output nodes (red trapezoids). Fluxes applied to the input nodes $i = 1, 2, \dots, 8$ are processed by the middle nodes and delivered, with different delays, to the output nodes $i = 39, 40, \dots, 46$.

Processing of flows is performed in the middle nodes. A node collects all its incoming fluxes at a given time moment and sends them instantaneously, at the same time moment, through its outgoing connections, splitting the total received flux in equal proportions among them. Time is discrete and one time step is always needed for a flux to pass over a link, from one node to another. Thus, the total flux x_i received by the node i at time $t + 1$ is given by a sum of the fluxes sent at the previous time t from all those nodes j which are connected to the node i . Explicitly, we have

$$x_i(t + 1) = \sum_{j=1}^N A_{ij} \frac{x_j(t)}{\sum_{k=1}^N A_{kj}} + I_{ext}^i(t). \quad (1)$$

Here, we have included external fluxes $I_{ext}^i(t)$ which can be applied only to the input nodes ($i = 1, \dots, N_{in}$). Note that fluxes cannot be accumulated in the network nodes and material is not stored there. The set of linear equations (1) determines the dynamics of the considered flow processing model.

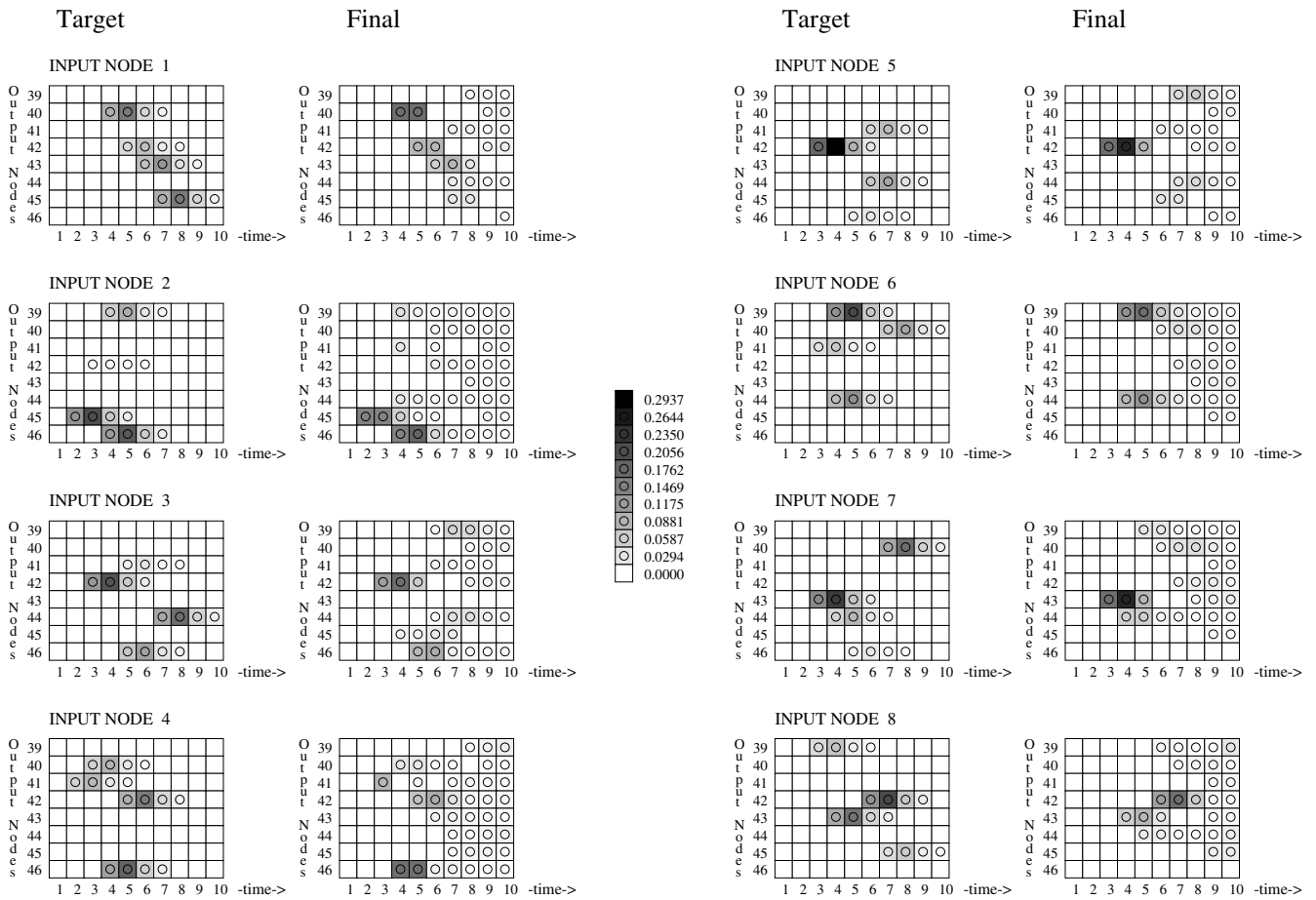


Fig. 2. Target (left columns) and final (right columns) input-output patterns $D_{ij}^0(t)$ and $D_{ij}(t)$. These patterns correspond to the construction of the network shown in Figure 1. Gray-scale coding is used to display matrix elements $D_{ij}(t)$. Each diagram displays the time-dependent responses of all output nodes in response to the initial activation of a different input node.

If an input node i is activated by an external input flux I_{ext}^i at the initial time $t = 0$, the flux propagates through the network and parts of it arrive at different delayed time moments to the output nodes j , with the delays determined by the length of the network paths between the given input and output nodes. The network performance is thus characterized by a time-dependent response matrix $\mathbb{D}(t)$. Its element $D_{ji}(t)$ specifies the flux arriving in the node j at time t if the input node i has been activated by unit flux $I_{ext}^i = 1$ at the initial time $t = 0$. Since the dynamics is linear, network responses following activation of different input nodes are independent and can be separately investigated.

2.2 Target patterns

Our aim is to construct networks with prescribed response patterns described by given target matrices $\mathbb{D}^0(t)$. They are constructed as follows: we require that the activation of an input node always produces only the activation of K output nodes which are randomly chosen for each input node i . The response of node j to activation of node i

starts with delay τ_{ji} and has intensity α_{ji} . All responses have, however, the same duration and temporal shape.

Hence, the temporal response of node j to activation of node i at time $t = 0$ is given by function $D_{ji}^0(t) = 0$ if $t \leq \tau_{ji}$ and $D_{ji}^0(t) = C_i \alpha_{ji} p(t - \tau_{ji})$ if $t > \tau_{ji}$. The response intensities α_{ji} are chosen as independent random numbers with a uniform distribution between 0 and 1. Delays τ_{ji} are also randomly and independently chosen from the interval between 1 and τ_{max} , the maximal possible response time delay. Thus, the responses are always localized within a time interval from 0 to T , where the maximal response time T is the sum of the response duration and the maximal delay time τ_{max} . The normalization constant C_i is determined by the condition that the integral total response should be unity, i.e. $\sum_{j=1}^{N_{out}} \int_{t=0}^T D_{ji}(t) dt = 1$. This yields $C_i = (\sum_j \alpha_{ji})^{-1}$.

In our numerical study, networks with $N_{in} = 8$ input nodes and $N_{out} = 8$ output nodes are considered. The target responses involve activation of $K = 4$ output nodes. The maximal response time is $T = 10$ and the maximal time delay is $\tau_{max} = 6$. The response function $p(t)$ is the same for all output nodes, it is chosen as a pulse of four time steps with $p(0) = 0.3$, $p(1) = 0.5$, $p(2) = 0.15$

and $p(3) = 0.05$. Figure 2 presents a target pattern for the network shown in Figure 1. For each input node the matrix in the left side indicates the target response $\mathbb{D}^0(t)$ that is required, and the matrix at in the right side shows the actual output pattern $\mathbb{D}(t)$ of the network.

2.3 Coevolution of target patterns

In the previous subsection we have presented the target pattern as a static object. The main goal of this work, is however, to develop different possible evolutions for the target pattern. The aim is to start from simple responses and increasing their complexities until reach the a final pattern $\mathbb{D}^0(t)$ that we have already shown before. For a simple response we mean that an input node only must activate few output nodes (less than K), or it just needs to generate a temporal response with fewer steps than the function $p(t)$. We have designed three possibles coevolutions (paths) for the target patterns. Our objective is to study the properties of the constructed networks when we increase the target pattern complexity.

2.3.1 Distributive-temporal coevolution (DT)

In this case of pattern coevolution we want, firstly, to distribute the input flux between the output nodes, and after that, to develop the temporal responses $p(t)$. Since we work with networks that ideally activate $K = 4$ output nodes by each input node, and the temporal response are pulses of 4 time steps of duration, we design a pattern evolution consisting on 16 stages. These stages can be seen in Figure 3 for the case of activation of one input node.

We observe that in first stage only one output node is activated at the corresponding time delay. At stage number 4, the four output nodes are activated with its corresponding time delay and relative intensities. Next stage, number 5, starts to develop the temporal response activating the second time steps of the first pulse. This process continues until all the pattern is completed. It is important to note that the target pattern is always normalized to the unit since the total flow must be conserved.

2.3.2 Temporal-distributive evolution (TD)

In this second case, we develop initially the temporal response for each output node in a sequential way, thus, the distribution among the output nodes is a secondary effect. Similarly to the previous case, the temporal-distributive coevolution has 16 stages according with the parameter values we consider.

Figure 4 shows an example of this pattern evolution for an input node. As we see in this figure, during the initial four stages the first pulse is generated. After that, the second one is developed during the next four stages, and this process is repeated for the K pulses.

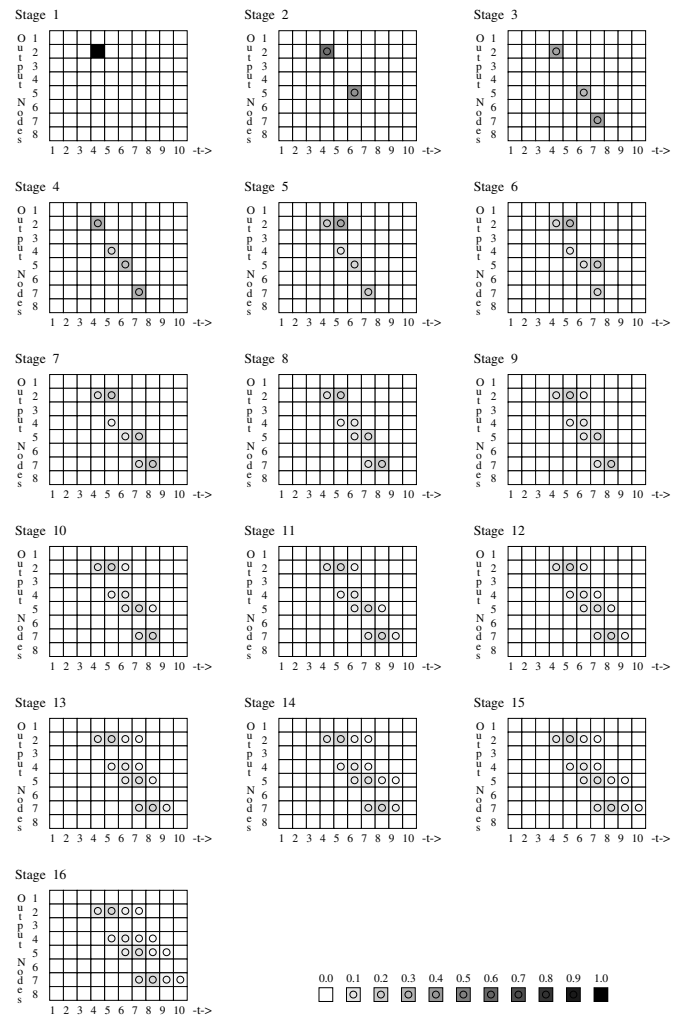


Fig. 3. Distributive-temporal evolution (DT) of the target pattern. This evolution has 16 stages. The figure presents the target output pattern of an input node. There are 8 output nodes and $T = 10$ temporal steps.

2.3.3 Pulse replication coevolution (RE)

In this last case, an initial pulse reproduces itself (copy), and the new copies get different relative intensities and delays during the following stages. For this coevolution of the target pattern we have 10 stages.

Figure 5 presents an example of this pattern coevolution. In the first stage, we have a complete pulse in the output node 2. In the second stage, the initial pulse has a copy on the output node 5. In third stage, the new pulse in node 5, get its own delay. Finally, at the fourth stage, the new pulse gets its own relative intensity. This process is repeated for the following two new copies of the original pulse.

Note that with these three pattern coevolutions, the final target patterns are the same for the three cases, and they are also the same to the one used in our previous works [10]. Particularly, for the the cases of pattern coevolutions distributive-temporal and temporal-distributive, they share for the first and the last stages the same target patterns.

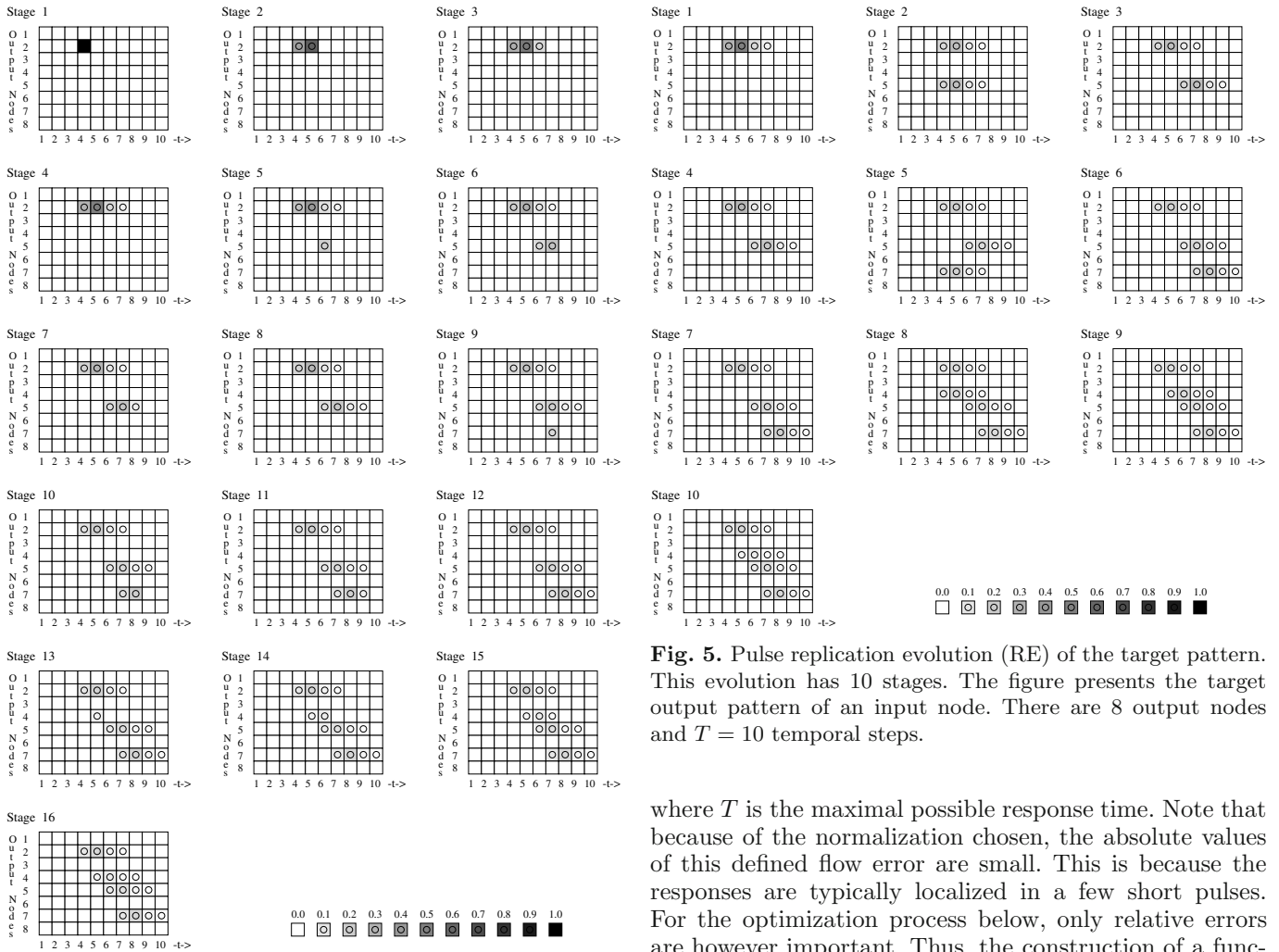


Fig. 4. Temporal-distributive evolution (TD) of the target pattern. This evolution has 16 stages. The figure presents the target output pattern of an input node. There are 8 output nodes and $T = 10$ temporal steps.

Fig. 5. Pulse replication evolution (RE) of the target pattern. This evolution has 10 stages. The figure presents the target output pattern of an input node. There are 8 output nodes and $T = 10$ temporal steps.

2.4 Construction of functional networks

In this work our task is to construct networks which are functional, i.e. networks that can reproduce prescribed output patterns. For these constructions we use the algorithm developed in reference [10]. We consider a situation where the number of available middle nodes is relatively small and, thus, most of the nodes need to be involved in generation of several different responses. In this case, no simple rational construction is possible and evolutionary optimization algorithms should be applied.

First, we introduce the flow error. The flow error $\epsilon(G)$ of a network G , having a response pattern $\mathbb{D}^G(t)$, with respect to a given target response pattern $\mathbb{D}^0(t)$ is defined like

$$\epsilon(G) = \frac{1}{2TN_{in}} \sum_{i=1}^{N_{in}} \sum_{j=1}^{N_{out}} \sum_{t=1}^T (D_{ji}^0(t) - D_{ji}^G(t))^2, \quad (2)$$

where T is the maximal possible response time. Note that because of the normalization chosen, the absolute values of this defined flow error are small. This is because the responses are typically localized in a few short pulses. For the optimization process below, only relative errors are however important. Thus, the construction of a functional network with the prescribed response pattern $\mathbb{D}^0(t)$ can be seen as an optimization problem. An approximate approach to the problems of complex combinatorial optimization, such as the traveling salesman problem, is provided by the method of simulated annealing [11]. Note that this method have been also employed successfully to optimize functionalities in genetic networks [12] and systems of phase oscillators [13]. A variant of this method is employed below.

To construct a network with the chosen performance $\mathbb{D}^0(t)$, an optimization process is run. Each iteration consists of the same steps:

1. Take a network G with a flow error ϵ .
2. Apply an evolutionary mutation to G , obtaining a new network G' with a new flow error ϵ' .
3. Calculate the difference $\Delta\epsilon = \epsilon' - \epsilon$.
4. If $\Delta\epsilon \leq 0$, accept the mutation and replace G by G' . If $\Delta\epsilon > 0$, accept the mutation with probability $\exp(\Delta\epsilon/\epsilon\sigma)$.
5. Return to step 1.

The evolution is started with a random initial network G_{ini} and continued for a fixed number of iterations¹.

¹ For a review on complex networks see Albert and Barabási [14].

This optimization method represents the Metropolis optimization algorithm where the effective temperature $\theta = \epsilon\sigma$ is proportional to the error ϵ and gradually decreases as the error diminishes, as assumed in the simulated annealing. Thus, σ is the annealing parameter, specifying how rapidly the temperature is decreased during the annealing. A detailed study of this parameter can be found in reference [10]. In the present work we use the optimum values of σ in order to obtain the best optimization performance.

To complete the description of the optimization method, evolutionary mutations should be further specified. In the present study, the link mutation scheme is always employed. A mutation consists in adding a new connection or deleting an existing one from the network. During the evolution, the total number of middle nodes M is preserved. In the final network, some middle nodes may, however, turn out to have no incoming connections. Such disconnected nodes are removed from the final networks, so that the number of middle nodes in a final network may become slightly smaller.

In the present work the particularity, as compared with reference [10], is that the target pattern $\mathbb{D}^0(t)$ evolves together (simultaneously) during the network construction process. Each stage of the pattern coevolution scheme is maintained during a time interval given by the total number of iterations of the optimization process divided the number of stages. Figure 6 shows four evolutions for the classical case (a), and for the three target pattern coevolutions that we have designed previously, the distributive-temporal (b), the temporal-distributive (c), and, the pulse replication (d) ones.

We observe that for the cases with coevolution of the target patterns, the flow errors ϵ decrease in each interval with a fixed target pattern. When a stage changes, we see that the flow error increases rapidly since the network constructed in the previous stage is not longer the best one for the more complex target pattern in the following stage. We find that the final error in the four cases is almost the same at the end of the process of network construction.

3 Numerical study

Following, we present several numerical simulations where ensembles of networks are constructed in order to reproduce target functions that evolve by the different proposed pattern coevolutions. The network ensembles allow to study statistically the structural properties of the networks.

We are interested on the structural changes of the networks during the optimization process. We study the dependence of the structures as a function of the total number of iteration we employ, and as function of the pattern coevolution as well the classic evolution.

3.1 Dependence of the optimizations with the number of iterations

In this subsection we study the characteristics of the constructed networks as a function of the total number of

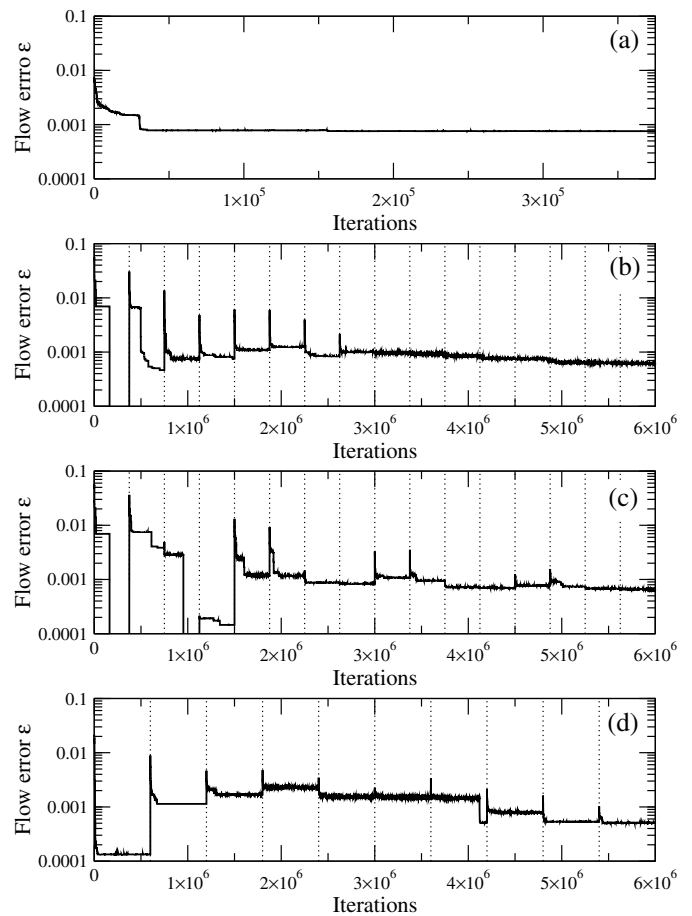


Fig. 6. Flow error ϵ as a function of the number of iterations for different pattern coevolutions: classical (a), distributive-temporal (b), temporal-distributive (c), and replication (d). The vertical dotted lines indicate the stage for each pattern coevolution. The total number of iterations 1×10^6 is equally divided among the pattern coevolution stages. The parameters of the simulations are: $M = 50$ and $\sigma = 1 \times 10^{-3}$.

employed iterations for the four target pattern evolutions and the initial connectivities of the networks.

3.1.1 Classical evolution

We construct several ensembles of 100 networks by using different number of iterations with the classical evolution, i.e., the target pattern is fixed along the optimization. We consider networks with $M = 40$ middle nodes. As initial condition for them we use random networks with different probability of connection $p = 0.01$, $p = 0.1$ and $p = 0.2$. The temperature parameter $\log(\sigma) = -2.75$ (see Ref. [10]).

Figure 7a presents the mean flow error $\langle \epsilon \rangle$ as a function of the total number of iterations for several ensembles of networks. We observe similar behavior for the three network initial connectivities p . For the three cases the mean errors evolve at the end to almost the same value and they follow a similar behavior. Note that at $t = 5 \times 10^4$

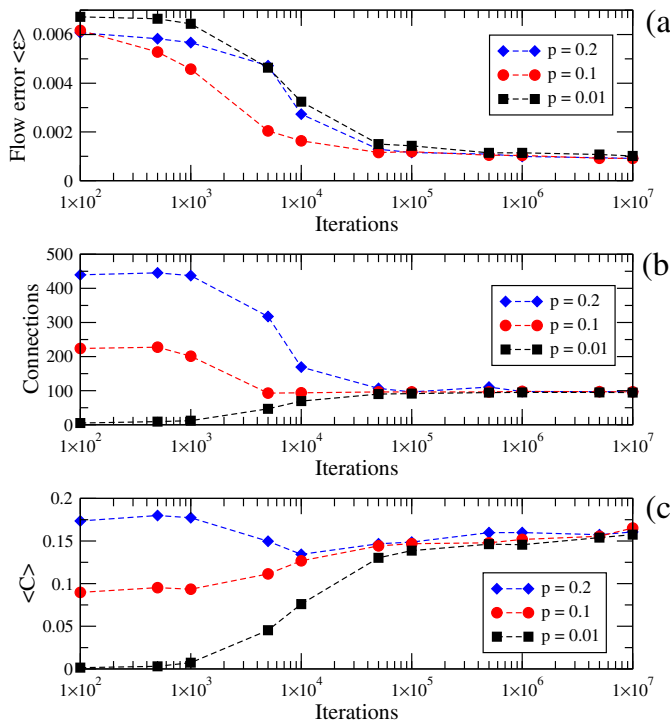


Fig. 7. Properties of the network structures as a function of the total number of iterations for classical constructions with different initial random connectivities. (a) Mean flow error, (b) mean number of connections, and (c) mean clustering coefficient.

the three curves almost coincide despite the initial connectivity. This indicates a kind of phase transition from non-functional networks to functional ones as a function of the number of iterations.

In order to further investigate this kind of phase transition we present two structural properties of the network ensembles. Figure 7b presents the mean number of connections as a function of the number of iterations for the network ensembles, and for the three different initial connectivities. We observe that after 5×10^4 iterations the number of connections reach similar values for the three cases. After this point, the number of connections is keeping almost constant. This behavior is interesting since the initial networks have a mean value of connections quite different given by $pM(N - 1)$.

In Figure 7c we present the mean clustering coefficient $\langle C \rangle$ [15] of the networks ensembles as a function of the total number of iterations. Similarly to the previous case, the clustering coefficients $\langle C \rangle$ start with different values and they evolve to almost the same final quantity for a large number of iterations.

These three properties show that the networks evolve to the same kind of solution despite the initial connectivity of the networks. We also find a phase-like transition behavior for 5×10^4 iterations. After such as number of iterations all the evolution end with similar functional networks.

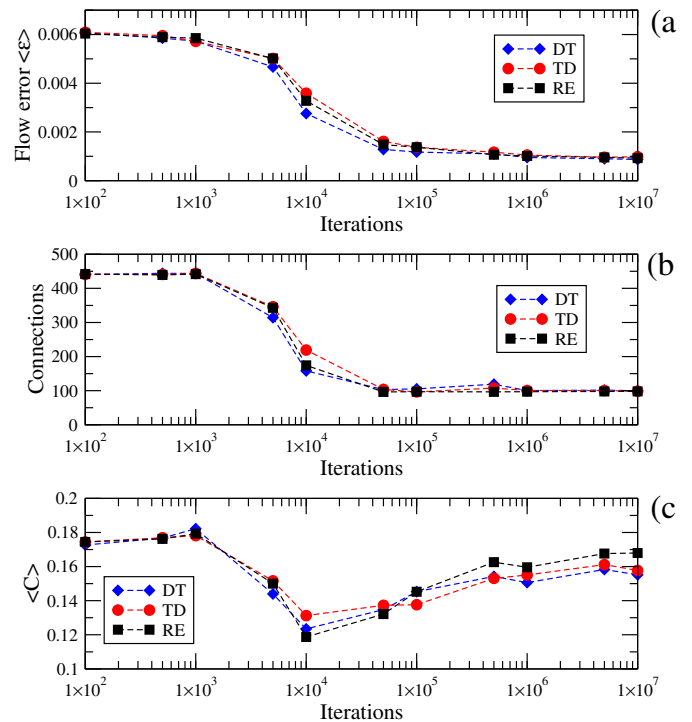


Fig. 8. Properties of the network structures as a function of the total number of iterations for coevolutionary constructions. (a) Mean flow error, (b) mean number of connections, and (c) mean clustering coefficient.

3.1.2 Coevolutions

We consider now the case of coevolutions of the target pattern with different number of iterations. A similar study to the previous one is performed but taking only a connectivity $p = 0.2$ for the initial random networks.

Figure 8a shows the mean flow error as a function of the total number of iterations for the three coevolution cases. We observe that the three curves have similar behavior to those of the classical evolution shown in Figure 7a. Mean values of structural properties, number of connections and clustering coefficients, are shown in Figures 8b and 8c respectively. We observe that for the three cases the curves almost coincide. Similarly to the case of classical evolution, we observe a behavior like a phase transition for 5×10^4 iterations.

3.1.3 Motif distributions

From these studies it results clearly that there is a kind of phase transition approximately for 5×10^4 iterations. After such a number of iterations networks have a well defined structure despite the initial connectivity and the kind of coevolution of the target pattern. In order to understand better this change on the network structure, we compute the motif profiles [16] for the ensembles of networks after different number of total iterations. The computation of the motif profiles is done with the software Mfinder [17]. We consider networks with $M = 40$ middle

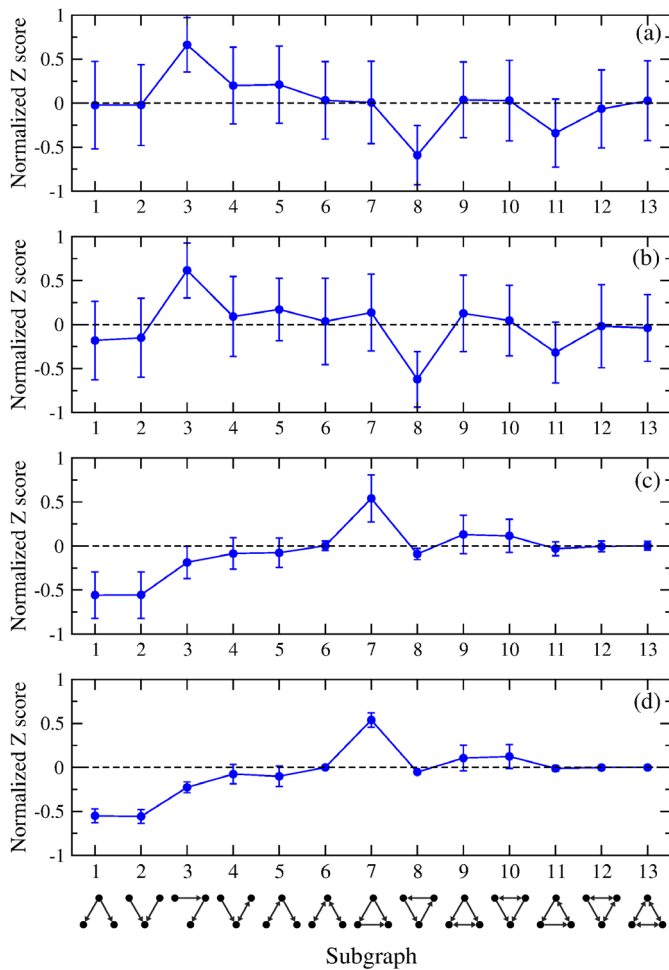


Fig. 9. Mean network motif profiles for ensembles of networks at the end of simulations with different number of iterations. Networks have $M = 40$ middle nodes. The number of iterations are: 1×10^2 (a), 1×10^3 (b), 1×10^4 (c), and, 1×10^5 (d).

nodes and initial connectivity $p = 0.2$. Their construction is performed with classical evolution, i.e., we fix the target pattern along the optimization process.

Figure 9 presents the motif profiles of the networks ensembles as a function of the number of iterations. We observe that the mean motif profiles for few iterations are not well defined and they present strong variations as it can be seen from the dispersions of the mean values in Figures 9a and 9b (1×10^2 and 1×10^3 iterations respectively). We see that for 1×10^4 and 1×10^5 iterations the mean motif profile has almost the same shape with smaller dispersion for the last case (Figs. 9c and 9d).

For larger number of iterations the motif profiles remain essentially the same as in Figure 9d. This evidence supports the hypothesis of a phase-like transition in the network structures when the number of iterations is larger than 5×10^4 . Note that the motif profile for large number of iterations coincides with the first superfamily of networks [16] as it was previously reported for functional networks [10]. Note that in the first superfamily of networks we find several sensory transcription networks that

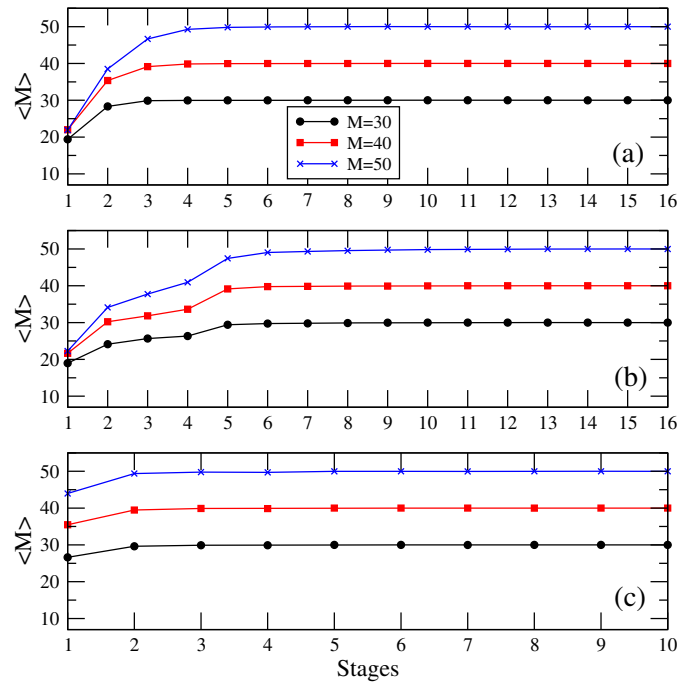


Fig. 10. Number of used nodes at the end of each stage for ensembles of networks with different pattern evolution and network size. (a) Distributive-temporal, (b) temporal-distributive, and (c) replication coevolutions.

control gene expression. Networks ensembles constructed with few iterations not only has big dispersion, they do not fit properly in any superfamily of networks.

3.2 Coevolution with large number of iterations

Our second study consists on the construction of ensembles of networks by using the different prescriptions for the coevolution of the target patterns. In particular, here we employ a large number of iteration in the evolutionary optimization in order to ensure the convergence to good solutions on each stage of the pattern coevolution. This study allows us to understand the structural changes of the networks when the target pattern complexity increases.

We consider networks with different number of middle nodes, with $M = 30$, $M = 40$ and $M = 50$. During the optimization process we keep fixed the number of middle nodes M . The total number of iterations is 1×10^6 and the temperature parameter takes the following different values as a function of the network size (see Ref. [10]): $\log(\sigma) = -2.5$ for $M = 30$, $\log(\sigma) = -2.75$ for $M = 40$, and $\log(\sigma) = -3$ for $M = 50$.

In Figure 10 we present the mean number of the used middle nodes $\langle M \rangle$ at the end of each stage of the target pattern evolution. Although, we keep fixed the number of middle nodes during the whole optimization process, nodes without input connections (and their output links) are not taken into account in order to compute the structural properties at the end of each stage. We observe that for the different pattern coevolutions the number of used middle nodes differ during the first stages.

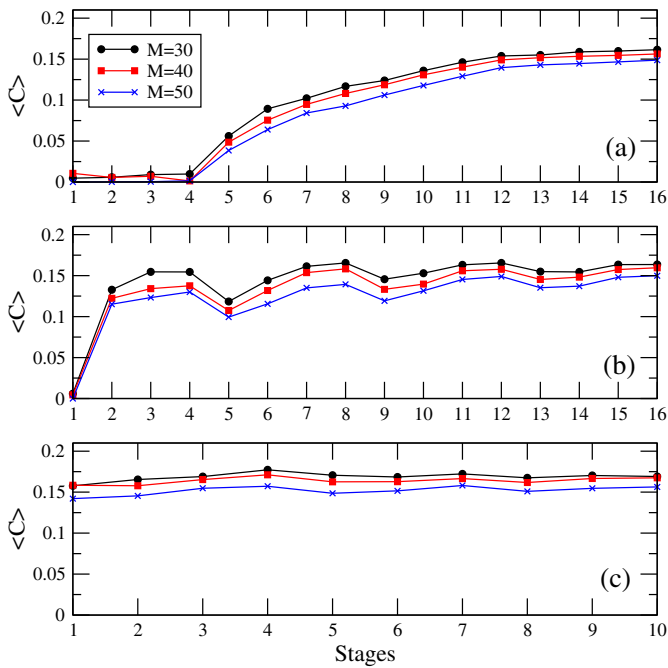


Fig. 11. Clustering coefficient at the end of each stage for ensembles of networks with different coevolution of the target pattern and network size. (a) Distributive-temporal, (b) temporal-distributive, and (c) replication coevolutions.

We observe for the distributive-temporal case that at the end of the first three stages the networks do not use all the available middle nodes to solve the target patterns. However, from the fourth stage to the last one, almost all the middle nodes are used on the network structure. For the temporal-distributive case we find a similar situation where for the first five stage networks do not employ all the available middle nodes during the optimization; and from the sixth stage to the last one, all the middle nodes are used. Finally, for the replication case, networks solutions almost always employ all the middle nodes at the end of each stage. These behaviors are similar for the different networks sized.

Figure 11 presents the mean clustering coefficient $\langle C \rangle$ of the networks ensembles as a function of the stage of the coevolutions of the target patterns for different network sizes. We observe that they are quite different as a function of the target evolution path. In the case of the distributive-temporal coevolution (Fig. 11a) the $\langle C \rangle$ is almost zero for the four first stages, indicating the absence of triangles on the networks structures. After these four stages the $\langle C \rangle$ grows gradually with the following stages. For the second case (Fig. 11b), the temporal-distributive pattern evolution, the $\langle C \rangle$ increases with the development of the temporal responses and each time when a new pulse appears and the flow is distributed, the $\langle C \rangle$ decreases. Finally, for the replication case (Fig. 11c) the mean clustering coefficient $\langle C \rangle$ is always high and it is almost constant for all the stages of the pattern evolution. Note that the final values for a given network size is almost independent from the pattern coevolution we consider.

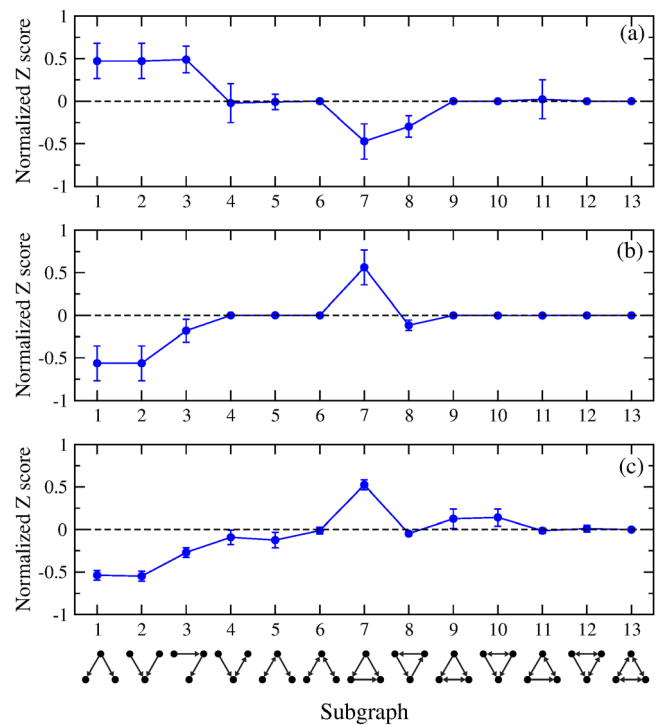


Fig. 12. Motif profiles for network ensembles of stage 4, 5 and 16 of the distributive-temporal coevolution case. Networks have $M = 50$.

These results allow us to relate the existence of triangles with the temporal development of the output signals. In order to find more evidence of this result we have evaluated the motif profiles for the networks ensembles for the temporal-distributive case for the stages fourth, fifth and for the last one. The result is shown in Figure 12 for an ensemble of networks with $M = 50$.

We observe that in stage fourth only the distribution is applied and motif number one, two and three related distribution and with long paths are overrepresented, and motifs seven and eight related with triangles are underrepresented in the motif profile. This first profile can be located in the fourth superfamily of networks where the bipartite model (graphs without triangles) is located. In stage number five, just when the temporal responses start to develop, the motif seven related with triangles is now overrepresented; and on the contrary motifs one, two and three are underrepresented. Finally, the last stage does not differ much from the stage number five and with the motif profile shown in Figure 9d. This fact indicates that the triangles are related with the existence of temporal responses and not because the distribution process among the output nodes.

4 Conclusion and discussion

In this work we have constructed flow processing networks with prescribed functionalities (target patterns). We have designed three pattern coevolutions that gradually increase the complexity of the required network functionality. Particularly, we have studied the evolutions

of the network structures during the construction process for both case: the classic one (fixed target pattern) and when a coevolutionary process acts simultaneously on the target function. Additionally, we have consider different initial connectivities and network sizes.

Our research yields three major results:

- (i) The evolution towards a functional network shows a dramatic improvement akin to a phase transition at some moment during the evolution process. The transition can be seen in the time evolution of the output error, the motif distribution of the selected networks and in their other statistical measures. After that transition, the statistical properties of the networks that are selected show only small incremental changes.
- (ii) The network evolution can be monitored through the time evolution of its motif distributions. Characteristic changes in the motif distribution reflect the onset of temporal delays in the target pattern and of increasing spatial complexity.
- (iii) The networks selected via coevolution are to a large degree path independent: they have the same statistical properties whether the target pattern changes on a time scale of a few hundred iterations or on a time scale of tens of thousands of iterations. This seems to be an important and remarkable property: evolution to a network that is optimally adapted to a given function does not require that this function be stable in time nor do intermediate stages of the network have to be optimally adapted to the intermediate stages of the required function to end up in a network that is optimally adapted to the final function. This result agrees and complement the study shown in the article of Beber et al. [18] where they show the the networks structure is related to the target pattern properties.

We want to remark that this method of coevolution between the target functionality and the system structure can be used in general as a tool to understand their relationship. In effect, in a system with a fixed complex target this relationship can be difficult to elucidate since the final system generates several tasks simultaneously. An example of these systems are genetic regulatory networks with adaptive responses [19,20]. These networks do not only generate adaptive responses, but also they require delays for their onsets. Our coevolutionary method can be applied here in order to develop gradually the delay during the network construction. Thus, the relationship between structure and delay can be better studied.

The author would like to thank Dieter Armbruster for the stimulating discussions regarding to this publication. Author acknowledges financial support from SeCTyP – UNCuyo (project M028 2017-2018) and from CONICET (PIP 11220150100013), Argentina. Open access funding provided by Fritz-Haber-Institute of the Max Planck Society.

Open Access This is an open access article distributed under the terms of the Creative Commons Attribution License (<http://creativecommons.org/licenses/by/4.0>), which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

References

1. U. Alon, *An Introduction to Systems Biology: Design Principles of Biological Circuits* (Chapman and Hall/CRC, 2006)
2. E. Klipp, A. Kowald, C. Wierling, W. Liebermeister, in *Systems Biology* (John Wiley & Sons, 2011)
3. T. Danino, O. Mondragón-Palomino, L. Tsimring, J. Hasty, *Nature* **463**, 326 (2009)
4. K. Kaneko, *Life: An Introduction to Complex Systems Biology* (Springer-Verlag, Berlin, Heidelberg, 2006)
5. J. Swartz, *Nat. Biotechnol.* **19**, 732 (2001)
6. R.A. Bradshaw, E.A. Dennis, *Handbook of Cell Signaling*, 2nd edn. (Academic Press, Amsterdam, 2010)
7. P. Kaluza, M. Ipsen, M. Vingron, A.S. Mikhailov, *Phys. Rev. E* **75**, 015101 (2007)
8. P. Kaluza, A.S. Mikhailov, *Europhys. Lett.* **79**, 48001 (2007)
9. P. Kaluza, M. Vingron, A.S. Mikhailov, *Chaos* **18**, 026113 (2008)
10. P. Kaluza, A.S. Mikhailov, *Eur. Phys. J. B* **85**, 129 (2012)
11. S. Kirkpatrick, C.D. Gelatt, M.P. Vecchi, *Science* **220**, 671 (1983)
12. Y. Kobayashi, T. Shibata, Y. Kuramoto, A.S. Mikhailov, *Phys. Rev. E* **83**, 060901(R) (2011)
13. Z. Levnajić, *Sci. Rep.* **2**, 967 (2012)
14. R. Albert, A. Barabási, *Rev. Mod. Phys.* **74**, 47 (2002)
15. D.J. Watts, S. Strogatz, *Nature* **393**, 6684 (1998)
16. R. Milo, S. Itzkovitz, N. Kashtan, R. Levitt, S. Shen-Orr, I. Ayzenshtat, M. Sheffer, U. Alon, *Science* **303**, 1538 (2004)
17. Software package MFinder is available at www.weizmann.ac.il/mcb/UriAlon
18. M.E. Beber, D. Armbruster, M.-T. Hütt, *Eur. Phys. J. B* **86**, 473 (2013)
19. M. Inoue, K. Kaneko, *PLoS Computational Biology* **9**, e1003001 (2013)
20. P. Kaluza, M. Inoue, *Eur. Phys. J. B* **89**, 156 (2016)