State-dependent information processing in gene regulatory networks - Draft version

Marçal Gabalda-Sagarra, Lucas Carey, and Jordi Garcia-Ojalvo

Department of Experimental and Health Sciences, Pompeu Fabra University, Barcelona, Spain

October 7, 2014

Abstract

Single cells have the potential and the necessity to process the information they receive from their environment. In particular, they commonly need to process temporal information obtained simultaneously from multiple inputs. In addition, response to multiple temporally ordered inputs is evolvable under laboratory conditions, suggesting that genetic networks are constructed to enable organisms to integrate novel information over time. However, the logic used by cellular regulatory networks to perform such complex information processing tasks is not understood. Here we show that gene regulatory networks are consistent with a computation paradigm known as reservoir computing (RC), and that this network structure enables single cells to process temporal information. A core subnetwork of genes (the reservoir) encodes and classifies complex time-varying information in a state-dependent manner. Because the state of the reservoir can then be decoded by a single layer of readout genes, allowing cells to process temporal information and efficiently learn new complex environmental conditions. In support of claim, we analyzed transcription factor networks from a variety of organisms, and found that their topology is compatible with RC. We identified the reservoir cores of the regulatory networks, and tested them using the memory-demanding NARMA prediction task, used as a standard benchmark for RC systems in machine learning. Our results show that the gene regulatory networks perform, and significantly better than other, more constrained topologies reported to work as RC. Interestingly, we find that, in real biological subnetworks, the information processing capacity of of the subnetwork is not strongly dependent on on the number of genes that receive input from the environment. Therefore, reservoir computing is an efficient way to for cells to process information without needing to increase the number of genes or the structure of the network. This is in contrast to other network configurations.

Keywords: Reservoir computing, gene regulatory network, information processing, echo state networks, liquid-state machine

1 Introduction

All living beings, from the simplest unicellular organism to humans, survive by constantly processing the information that they survey from their environment. By detecting threats and opportunities in the world around them, organisms can decide their next steps to maximize their success. This process occurs already at the level of single cells, and involves a complex network of thousands of interacting elements that dynamically adapt the cell's function to both its environment [1] and, in some casesm its predicted future environment.

The traditionally described response of cells to changes in external conditions is homoeostasis. That is a resistance of the cellular state to change despite environment alterations. Thus, in principle cellular responses should aim to directly counteract changes in their local surrounding. This fundamental mechanism of cell resilience relies on biochemical processes such as signal transduction, gene expression, transport, and protein degradation, which are not instantaneous. Hence, cells need a certain amount of time to build a response to a given external change [2]. Given this, there is a potential benefit for cells in being capable to process certain temporal information and to infer future external determinants. In other words, the ability to efficiently anticipate changes in the environment would represent a significant improvement in the capacity of adaptation of the cell.

From the point of view of a single cell, some environmental changes are stochastic, in that they are not predictable based on the current or past environment. In many cases, however, the current environment does carry some information about the likelihood of future environments. For example, periodic changes in the environment can be forecasted through molecular oscillators or cellular clocks, as shown by the adaptation to circadian light-dark cycles in cyanobacteria [3, 4]. Another situation arises when an event is usually preceded by another, such as the case of a temperature increase being followed by oxygen depletion in *Escherichia coli* as it is ingested by mammals [5]. Similarly, enterobacteria encounter sequential changes in sugars as they pass through the intestinal tract and yeast experience a progression of different stresses during alcoholic fermentation [6]. Furthermore, experimental evolution studies have shown that predictive environmental sensing can be evolved in relatively short periods of time in a laboratory setting [7]. In all these cases the mentioned microorganisms possess mechanisms to take advantage of the predictable behaviour of their environment and anticipate it.

Moreover, beyond their ability to predict immediate environmental changes, recent studies have shown microbes posses both short-term and long-term memory that influence cellular decisions. The stress response of *Bacillus subtilis* depends not only on the condition in which it is currently growing, but also on the past growth condition [8]. Having any sort of record of previous history –i.e. memory– could putatively help bacteria, as any other living being, to infer future environment behavior from past events.

Even though some of the environment-anticipation mechanisms mentioned above –e.g. molecular oscillators– may involve only a handful of cellular elements, the large complexity of interactions among diverse types of molecules such as DNA, RNA, proteins and metabolites is what makes the cell able to accurately adapt to the environment conditions [9, 10]. In that sense, the networks of interactions that regulate the cell encode not only the direct response mechanisms of the cell but also the paradigm that the cell uses to integrate and process environmental signals [1].

We are interested in the capability of cellular regulatory networks to integrate complex inputs, and specially their possibility to process complex temporal information. To do so, it is necessary to consider what kind of network architectures may support these processing requirements. Since the computational power of networks is evident in neuroscience and machine learning, we deem it appropriate to review some of the network layouts described in these fields.

Feed-Forward Neural Network (FFN)

Also known as *multi-layer perceptron*, it is a widely used network organisation in machine learning. The processing elements, named neurons, are organised in linear layers letting the information travel unidirectionally from one layer to the next one. This structure is neatly modular, allowing the use different types of neurons and simplifying the process of training the weights of the connections for a given task. On the other hand, its principal limitation is that it

cannot describe time: the output of the system computed from a given input does not depend at all on previous nor following inputs. In summary, feed forward networks can be easily trained to process spatially complex inputs but cannot process temporal information [11, 12].

Recurrent Neural Network (RNN)

RNNs are specially relevant —compared with FFN— due to their robustness, the capability to process temporal information and the ability to model highly nonlinear systems [13]. In this kind of networks there are no hierarchies in the way neurons are organized, making no special distinction between input, output or hidden nodes (i.e. intermediate nodes). As consequence there exist recurrences in the information flow inside the network and thus any node can potentially affect any other. The direct effect of this is that after a given input arrives to the system, it produces a complex dynamical perturbation that will only gradually disappear. This behaviour — called *fading memory* — gives RNNs the capability to process temporal information. In other words, the response of the network will be shaped in a complex manner by the interaction between external input and the internal state of the network. Nevertheless, the main drawback of RNNs is also a consequence of the recurrence of its connections: modifying the weight of a single link may affect the behaviour of the whole system. This makes the training process impossible to achieve by gradient-descend methods, and in general computationally too expensive [11].

Reservoir Computing (RC)

Reservoir Computing (also called Liquid State Machines [14] and Echo State Networks [15]) is based on a central recurrent reservoir of hidden nodes, analogous to a RNN, with fading memory. The transient state of the reservoir encodes the recent history of the system. In contrast to standard RNN, however, the output nodes are placed in a feedforward structure that is affected unidirectionally by the reservoir and is capable of decoding its transient dynamics. The advantage of RC is that it describes the possibility of using a RNN without the need of adapting the weights of the internal connections of the network. Instead, only the links towards and between the output layers need to be trained to learn a new task [11](Figure 1). Thus, the approach based on RC simplifies notably the training of the RNN and makes it more meaningful as a biological model.

Although the RC models were initially described for highly complex and recurrent systems such as neuronal networks, relatively simple topologies suffice as long as they are recurrent [16]. Additionally, although the minimum requirements of RC in terms of number of nodes have not been studied in detail, reservoirs with biologically reasonable sizes of tenths of nodes were reported to be functional [16].

We therefore hypothesised that information processing by cellular regulatory networks might follow a RC paradigm, which would enable cells to efficiently integrate and learn on temporal information using relatively small numbers of genes. This would imply that part of these networks should serve as recurrent reservoirs with fading memory, influenced by the input signals and affecting a non-recursive readout structure. As a consequence, cellular regulatory networks would have the capability to process temporally complex inputs.

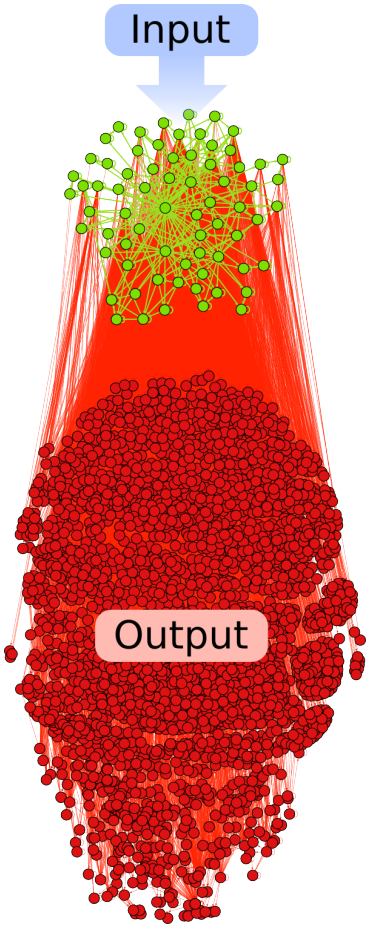

Figure 1: Representation of the Reservoir Computing network architecture. The input signals reach some nodes of the reservoir, perturbing their state. The dynamics of the reservoir, a group of nodes with recurrent connections, encodes recent input history (green circles). Downstream, a readout structure of nodes organized in a strictly feed-forward manner is trained to interpret the state of the reservoir (red circles). Only the links towards readout nodes need to be considered in the learning process. The nodes in this figure correspond to the reservoir and readout nodes identified in the Ecocyc gene regulatory network (Table 2).

2 Methods

2.1 Cellular regulatory networks

We used transcription factor networks already available in the bibliography. Data for *Bacillus subtilis* was obtained from DBTBS [17]. Data for *Escherichia coli* was extracted from EcoCyc [18]. In this case the sigma factors were included in the network as transcription factors. Data for *Saccharomyces cerevisiae* was obtained from YEASTRACT [19]. The gene regulatory network for *Drosophila melanogaster* was obtained from the modENCODE initiative [20]. Data for *Homo sapiens* was extracted from the ENCODE project [21].

2.2 Network pruning

The networks were pruned in order to select their minimal recursive subgraph, i.e. a subgraph with exclusively all the nodes and edges that form recursive structures and the nodes that

interconnect this structures. This subgraph has (i) the full RC-computational capabilities, which is given by the recurrent parts of the graph [16] and (ii) less complexity than the full network, in terms of numbers of nodes and edges. The pruning methodology consisted in removing iteratively those nodes that had either in-degree or out-degree equal to zero until no more nodes could be deleted.

2.3 Control networks

As reference reservoirs the following topologies were used:

- Echo State Network – fixed mean degree (kESN): random network with the only constrain of having the same mean degree ($2 \times n_{edges}/n_{nodes}$) as the problem topology.
- Echo State Network – fixed fraction of links (fESN): random network with the only constrain of having the same fraction of existing links over all possible links (n_{edges}/n_{nodes}^2) as the problem topology.
- Simple Cycle Reservoir (SCR): a directed circular graph, which is the simplest network topology reported to work as RC[16].

Note that for control networks with the same number of nodes as the problem topology, kESN is equal to pESN. This is not the case, however, when the number of nodes changes.

2.4 Computational performance

To assess the computational capabilities of the topologies analysed they were used to build reservoirs. Then a downstream output node was trained to perform a standard NARMA task. All the simulations were produced using the *Oger toolbox* [22].

2.4.1 Reservoir dynamics

The networks analysed were used to build reservoirs with the same topology. The nodes were updated synchronously according to

$$x_{i,t+1} = tanh(v_i z_t + \sum_{j=1}^{n} w_{ji} x_{j,t}) \tag{1}$$

where z_t is the system input at time t; $x_{i,t}$ is the state of the ith node of the reservoir at time t; n is the number of nodes in the reservoir; w_{ji} is the weight of the link from node jth to node ith; and v_i is the weight of the link from the input to the ith node. The values of the vector V are randomly chosen to be either -0.05 or 0.05. At the same time, the values of the $n * n$ matrix W are reals drawn from $unif[-1, 1]$ if the link exists and 0 otherwise. Additionally, the W matrix was normalized to have a spectral radius of 0.9 to assure the echo state property [12, 15].

2.4.2 Output node

A single output node performing a ridge regression from the state of all the nodes of the reservoir was trained to predict the output of the selected task. Ridge regression is a method often used in RC. It allows to estimate the parameters of a linear regression when the predictor variables are very strongly correlated, as in this kind of problems [23].

$$W = YX^T(XX^T + \gamma^2 I)^{-1} \tag{2}$$

where X is a matrix with the $x_{i,t}$ values for all i and t, Y is a matrix with all expected outputs, I is the identity matrix and γ is a regularization parameter that needs to be adjusted. After several trials with different values of $0 \leq \gamma \leq 1$ our system gave the most consistent and better predictions with $\gamma = 0$, thus approximating a simple regression node (results not shown).

2.4.3 NARMA prediction task

The systems were trained to predict the *10th order Nonlinear AutoRegressive Moving Average* (NARMA) function and then evaluated in doing so. The NARMA system is a discrete time system introduced by **(author?)** [24] used as one of the standard RC performance tests. The values of input $s(t)$ samples uniform random numbers from the interval $[0, 0.5]$ and output is given by

$$\begin{aligned} y(t+1) =& 0.3y(t) + 0.05y(t)\sum_{i=0}^{9} y(t-i) \\ &+ 1.5s(t-9)s(t) + 0.1 \end{aligned} \tag{3}$$

Thus, output at time t depends on both input at time t and previous input and output history. In general, modelling this system is difficult, due to the nonlinearity and potential long memory.

2.4.4 training and evaluation

For our tests we used 9 NARMA datasets of 1000 time steps to train the system, in the learning phase. An additional time series of the same length was used in the test phase, to assess the prediction capability of the system.

In all cases the performance of the reservoir was measured by means of the Normalised Root Mean Square Error (NRMSE) between the prediction $\hat{y}$ and the actual output from the NARMA equation y:

$$\text{NRMSE} = \frac{\sqrt{\frac{1}{N}\sum_{t=1}^{N}(y_t - \hat{y}_t)^2}}{\text{sd}(y)} \tag{4}$$

2.5 Stress signal inputs

In order to build more realistic rules regarding which nodes of the reservoir are reached by the input signal we used a variety of well-characterised stress signals. The annotations available on the Ecocyc database [18] of all the nodes were in the Ecocyc core were manually screened. For a given stress type signal each node that was likely to receive the input signal was given a subjective certainty value. A node was considered to be likely affected by an input signal if the expression of the gene or the activity of the protein is modified by the stress through other mechanisms than the transcriptional regulations already represented in the Ecocyc core network. The same classification was done for signals of stresses of different nature: antibiotics, lack of oxigen, osmotic stress, extreme pH, oxidative stress lack of nutrients, extreme temperatures and an additional category considering all other stress types together. Supplementary Table 1 reports the number of nodes affected by every stress category with a certainty above a given threshold.

When performing the stress signal input experiments, the V input weights were assigned with the following rule: $v_i = 0.05$ if the activity of the ith gene or the protein increases in the presence of the stress, $v_i = -0.05$ if the effect was the opposite, and $v_i = 0$ there was no

effect described. In the case of the extreme temperatures positive v_i was assigned to nodes whose activity increased under high temperature (or that decreased under low temperature) and negative v_i to the ones that are known to react in the opposite manner. For the extreme pH stress, positive weights were associated with high activity under acidic media or decreased activity in basic media, and negative values to the genes with the opposite dynamics.

3 Results

Transcription factor networks from various databases for different organisms were analysed (see *methods*). Table 1 summarizes some of the descriptors of the networks built. Additionally, Figure 2 presents the degree distribution of each network in a log-log plot. It can be seen from the degree distribution that all of them have some sort of non-trivial structure, and are far from random or regular networks.

Table 1: Characteristics and topological properties of the analysed transcription factor networks

	Nodes	**Edges**	**Self loops**	**Average degree**
Whole graph				
DBTBS	921	1381	54	3.00
Ecocyc	3243	8373	128	5.16
YEASTRACT	6725	201972	197	60.1
modENCODE	9440	231179	0	49.0
ENCODE	16356	163272	28	20.0
Core graph				
DBTBS	13	30	7	4.62
Ecocyc	70	317	55	9.06
YEASTRACT	289	9046	195	62.6
modENCODE	486	23470	0	96.6
ENCODE	207	1434	26	13.9

Despite the complexity and large size of the networks, the subgraphs containing recurrent connections are the ones considered to be relevant for the computational capabilities of a reservoir [16]. Thus, the networks where pruned to remove all the strictly feed-forward nodes (see *methods*). The resulting subgraph is what will be referred from now on as *core* or *reservoir*. The topology of such cores is schematized in Figure 3. In addition, Table 1 summarizes some of the descriptors of the cores of the networks.

Despite the small size of the core subgraphs relative to the whole network their location is central. Table 2 groups the nodes of each network depending on whether they are part of the reservoir, they are placed downstream of it or neither. As shown, the vast majority of the nodes are placed downstream and thus directly affected by the core, forming what we would call the *readout*. It is worth to note again that by definition there are no recurrences outside the reservoir and thus all these readout nodes do not affect back the reservoir. Furthermore, the structure of the readout is rather simple, with most of the nodes being terminal which means

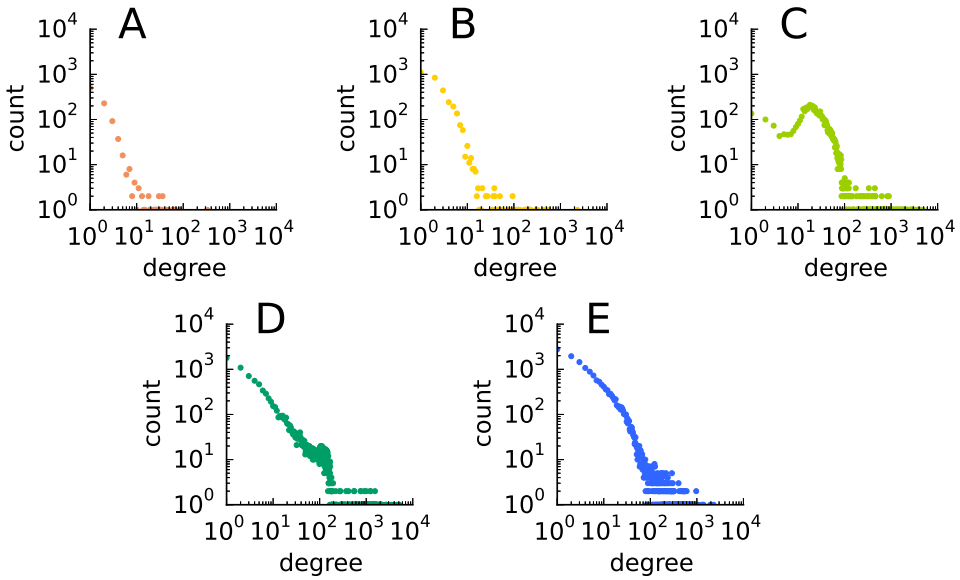

Figure 2: Degree distribution of the analysed gene regulatory networks on a log-log scale. Each plot corresponds to one of the networks: DBTBS (A), Ecocyc (B), YEASTRACT (C), modENCODE (D) and ENCODE (E).

that they do not affect any other node.

Table 2: Recount of nodes in each location of a Computational Reservoir. Terminal nodes are the subset of the readout nodes with out degree 0

System	**Reservoir nodes**	**Readout nodes (Terminal)**	**Other**
DBTBS	13	537 (490)	371
EcoCyc	70	3133 (3023)	43
YEASTRACT	289	6436 (6418)	0
modENCODE	486	8795 (8721)	159
ENCODE	207	13497 (13449)	2652

The computational capabilities of the network cores was assessed by means of a NARMA prediction task. We built reservoirs with the topologies extracted from the network cores to check if they were able to encode the recent input history. A readout node was trained to predict the output of the NARMA datasets from the state of the reservoir. Figure 4 shows a representative snapshot of the test phase of each core topology. It can be observed that the precision of the prediction tends to improve with larger cores.

Nonetheless, a more quantitative measure of the predictive performance of a reservoir topology was obtained through the Normalized Root Mean Squared Error (NRMSE) between the prediction and the actual NARMA output. Figure 5 shows the median NRMSE achieved by reservoirs with the topology of the biological cores. Performance of control reservoirs within a range of sizes are also shown. It can be observed that in all cases the biological cores perform as well as the random fESN and kESN control networks with the same number of nodes. While all control networks show similarly worse performances for sizes under 40 nodes, for larger net-

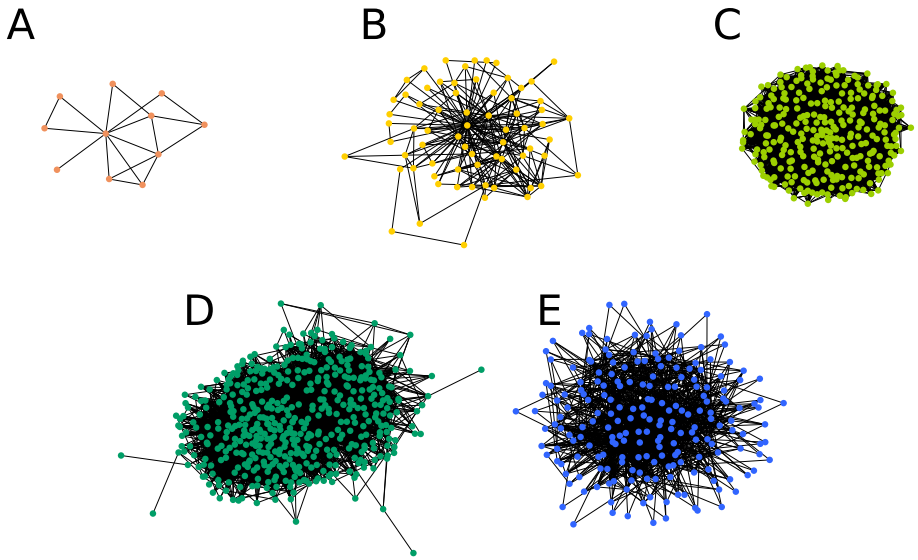

Figure 3: Topology of the subgraph of the recurrent core of each network: DBTBS (A), Ecocyc (B), YEASTRACT (C), modENCODE (D) and ENCODE (E). Every node in this subgraphs has both out-degree and in-degree larger than 0.

work sizes the recurrent but highly structurally constrained SCR is outperformed by the less structured fESN and kESN. In fact, differences between SCR and the ESN variants increase with size within the interval analysed.

Furthermore, to make sure that the results observed are representative for the whole network, we compared the results obtained using either the core or the whole network as the reservoir. Due to computing limitations, only the EcoCyc network was tested. As shown in Figure S1, the median performance of the whole network is equal to the one of its core. This way we also corroborated that the computation capacity really lies on the recurrent structures of the network.

Finally, we tested more realistic input setups to corroborate the results found. To that end we worked with the Ecocyc core network. Only the nodes that represent genes affected by a stress signal were receiving the input signal. Input circuits for different types of inputs were build on top of the Ecocyc core and each one of them was confronted to the NARMA predictive task. The number of nodes from the reservoir affected by each stress signal is shown in Table 3. As shown on Figure 6 for all the input circuits the cases the Ecocyc GRN topology performed better than the SCR and worse than the ESN. Additionally, it can be observed that the different stress signal input circuits allow the system to achieve different levels of performance. It is worth noting that the differences between each stress signal simulation and the *shuffled* control results are much lower. This fact and the tendency shown in the NARMA results with respect to input size shown in Figure 7 suggest that the differences among the stress signal circuits depend on the input size.

4 Discussion

In this paper we investigate the hypothesis that regulatory networks are able to function as a Reservoir Computing (RC) system. The presence of a subgroup of nodes with recurrent connections would improve the potential of the network to integrate multiple complex inputs in

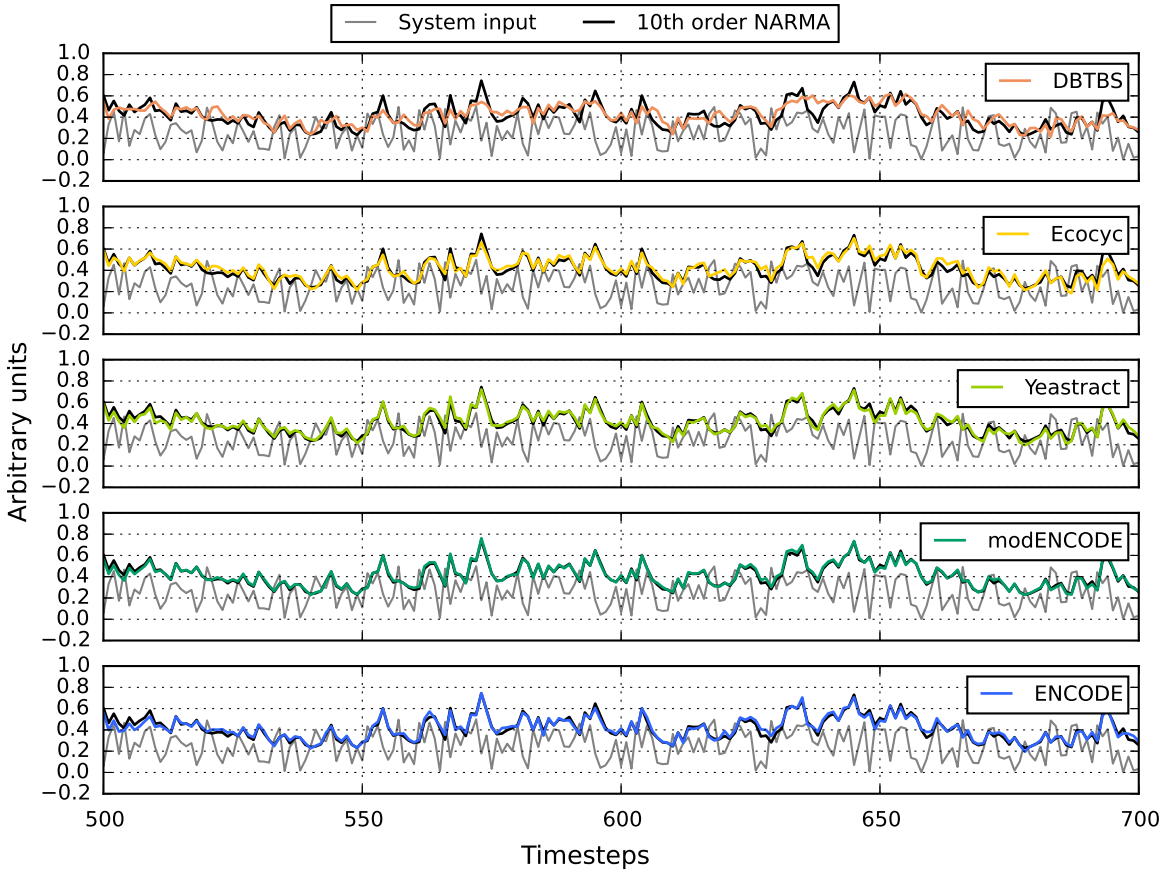

Figure 4: Representative snapshot of the test phase of a 10th order NARMA prediction task. For each biological network studied, the topology of its core was used to build a resevoir, and a readout node was trained to predict an output from the state of this resevoir. The different lines correspond to the random input of the system, the actual output of the NARMA function and, for each case, the output predicted by the readout node.

general, and allow the processing of temporal information in particular. On the other hand, in contrast to a fully recurrent network, the RC topology would facilitate the learning of new environmental conditioning. The reason is that learning by changing weights outside the recurrent reservoir is less likely to modify the comportment of the whole network. In other words, it would make easier the apparition of mutations that affect the behaviour of a given mechanism without necessarily provoking major changes in the rest of the organism.

To that end, we analysed five gene regulatory networks (GRN) for a representative range of species. All of them, far from being random networks, have a structured degree distribution. In the case of YEASTRACT, the deviation observed in the degree distribution plot (Figure 2C) is thought to be an artefact. Since most of the data in this database comes from compiling a large number of low throughput studies, less central nodes can be expected to be under-represented, as many more studies tend to focus on more influential genes. High throughput studies are less prone to this source of bias, although they may introduce others.

Regarding the topology, the networks analysed are perfectly compatible with a RC functionality. In all cases there is a single group of nodes with recurrent connections and the vast majority of nodes follow a strictly feed-forward organisation downstream of the recurrent part. Furthermore, with no exception the amount of nodes that fall outside of these two groups is

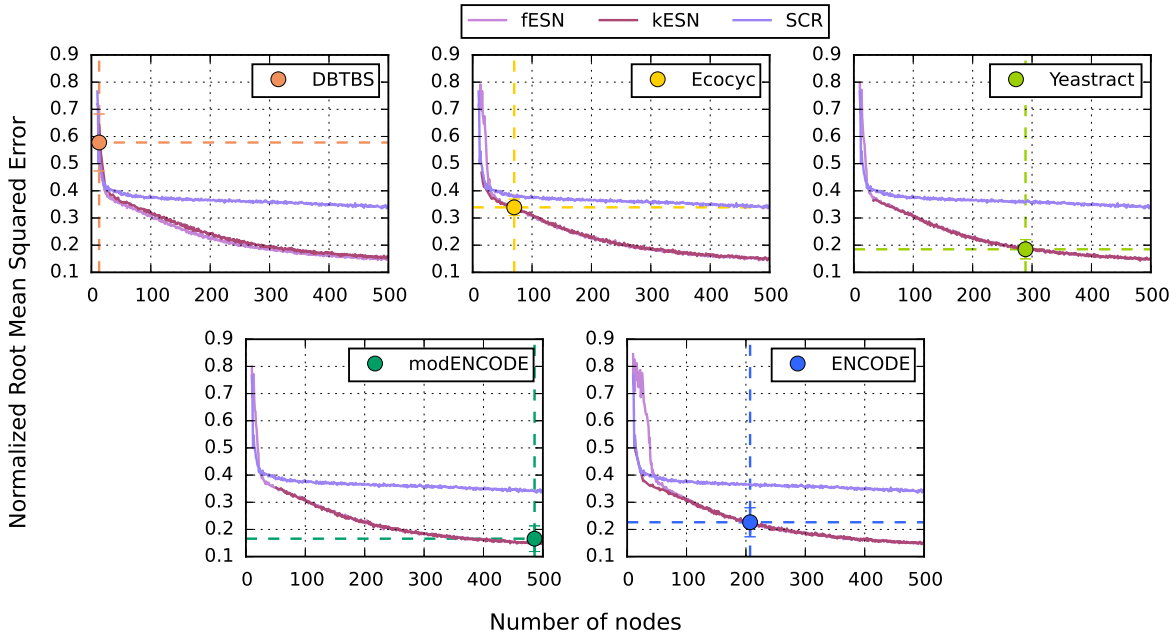

Figure 5: Predicting performance of reservoirs with topologies from the biological cores compared to control topologies. The value represented for each biological network topology corresponds to the median NRMSE value of 10000 trials (with edge weights and data series randomization). Error bars account for standard deviation. The values plotted for each control network (fESN, kESN and SCR) corresponds to the median value of 100 trials for each network size from 10 to 500 nodes. In each case, fESN and kESN are produced taking the fraction of links and the mean degree, respectively, from the biological core they are compared to (see *methods*).

only a small proportion of the total number of genes in the network.

In addition, the topology of the recurrent cores were well capable to integrate temporal information in a memory demanding task such as the 10th order NARMA. According to **(author?)** [16] recurrent structures are a necessary part for a reservoir to be functional. However, the same study points, and we corroborated it here, that unstructured random networks perform better than structurally restricted ones. Random networks are, in fact, the standard reservoir topology option for machine learning applications [13]. Nevertheless, as we found in this study, despite their structured topology, the biological cores perform as well as random networks of the same size. The fact that the topology of GRN is more efficient integrating information than structured recurrent networks like SCR is still true when including realistic rules based on a variety of well-characterized stresses to determine which nodes are directly affected by the input signal.

On the other hand, the differences in performance between the five tested biological networks would be explained by the different number of nodes and cycles. A larger recurrent network would have more complex transient dynamics and a longer fading memory [11].

Finally, for the in-depth studied case of the Ecocyc network, the way different stress signals reach the nodes in the reservoir is shown to allow different levels of efficiency in processing temporal information. The results obtained suggest that these differences in performance are mainly determined by the number of reservoir nodes that the signal of a given stress is reaching and not so much by the identity of those nodes. Nonetheless, it seems perfectly reasonable that varying the number of nodes that receive a given signal is the mechanism that allows evolution

Table 3: Number of nodes of the Ecocyc core that receive directly the input signal for each kind of stress circuit

Stress signal	**Number of nodes**
Any stress	58
Antibiotics	16
Anaerobiosis	6
Osmotic stress	13
Extreme pH	15
Oxidative stress	17
Starvation	30
Temperature	8

to control the precision required to process that signal. It has to be noted, however, that from the results shown in this work it can be deduced that increasing the number of inputs or their localization within the core is a very inefficient way to improve the overall system performance and that the topology of the reservoir is a much stronger determinant.

5 Conclusion

Cell need to process information they gather from their environment. Although some specific mechanisms are known, the overall logic of the input integration process in cells and the networks of interactions that regulate them is not defined. The performance of the cores of these networks in the NARMA test confirms the potential of the network cores to process temporal information in a reservoir-computing-like manner.

In this regard, gene regulatory networks showed to work as reservoirs as well as the best performing topologies described. In spite of the fact that the gene regulatory networks have a fixed structured topology, their architecture, and more precisely the one of their cores, is optimal to perform input integration under a Reservoir Computing logic.

Regarding the factors affecting the information-processing efficiency of the system we show that the core topology is the more relevant one. In a second term, the number of nodes receiving the input also affects the computing performance of the system in a significant manner, and finally the identity of the nodes nodes receiving by the input affects marginally the system efficiency.

As a side note, it is important to mention that the processing tasks of a biological network are remarkably different in nature and intensity. The results in tasks addressed by non-standard computations in machine learning need to be much more stably precise.

Acknowledgment

This work was supported by the Ministerio de Economia y Competividad (Spain, project FIS2012-37655). JGO acknowledges support from the ICREA Academia programme.

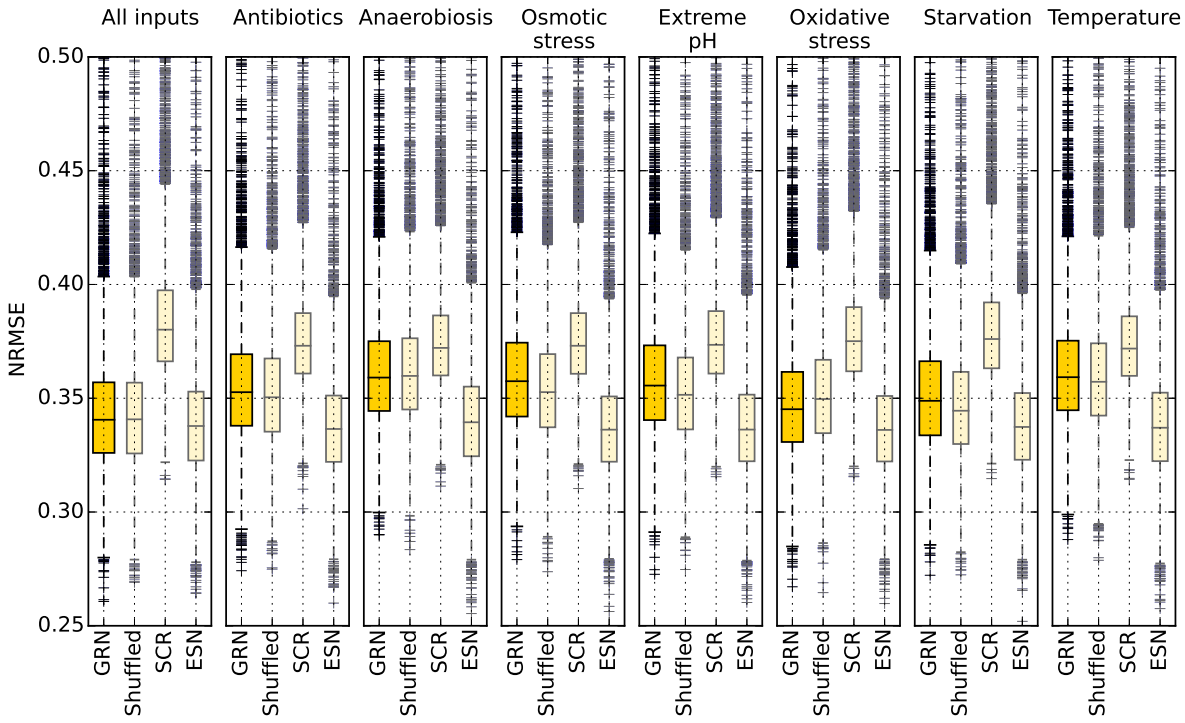

Figure 6: Performance of un the 10th order NARMA task of the Ecocyc recurrent core reciving the input signal through the different stress circuits studied. These are the results of evaluating the computational performance of the Ecocyc core using as input nodes those genes from the reservoir that are known to respond to a specific kind of stress (yellow boxes, *GRN*). For each kind of stress the results of three controls are shown (pale boxes): *Shuffled* is the Ecocyc topology with the same number of input nodes but randomly chosen; *SCR* and *ESN* are the simple circular reservoir and the Erdös-Rény random network with the same number of nodes and the same number of input nodes.

References

[1] A.-L. Barabási and Z. N. Oltvai, "Network biology: understanding the cell's functional organization.," *Nature reviews. Genetics*, vol. 5, pp. 101–13, Feb. 2004.

[2] U. Alon, *An Introduction to Systems Biology: Design Principles of Biological Circuits*. Boca Raton: Chapman and Hall/CRC, 1 ed., July 2006.

[3] S. S. Golden, M. Ishiura, C. H. Johnson, and T. Kondo, "Cyanobacterial Circadian Rhythms," *Annual review of plant physiology and plant molecular biology*, vol. 48, pp. 327–354, June 1997.

[4] T. Mori and C. H. Johnson, "Circadian programming in cyanobacteria.," *Seminars in cell & developmental biology*, vol. 12, pp. 271–8, Aug. 2001.

[5] I. Tagkopoulos, Y.-C. Liu, and S. Tavazoie, "Predictive behavior within microbial genetic networks.," *Science*, vol. 320, pp. 1313–7, June 2008.

[6] A. Mitchell, G. H. Romano, B. Groisman, A. Yona, E. Dekel, M. Kupiec, O. Dahan, and

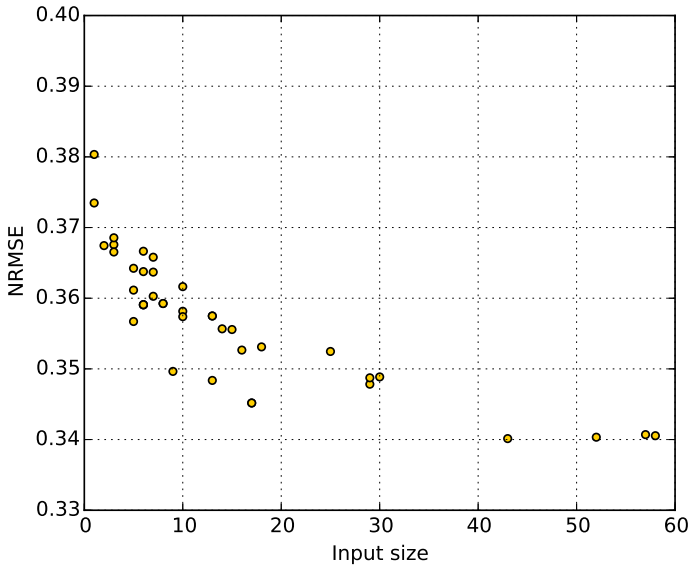

Figure 7: Median performance values obtained with the different input circuits against the input size of each circuit. Input circuits for each stress signal with different subjective certainty thresholds are included (see Suplementary Table 1).

Y. Pilpel, "Adaptive prediction of environmental changes by microorganisms.," *Nature*, vol. 460, pp. 220–4, July 2009.

[7] R. Dhar, R. Sägesser, C. Weikert, and A. Wagner, "Yeast adapts to a changing stressful environment by evolving cross-protection and anticipatory gene regulation.," *Molecular biology and evolution*, vol. 30, pp. 573–88, Mar. 2013.

[8] D. M. Wolf, L. Fontaine-Bodin, I. Bischofs, G. Price, J. Keasling, and A. P. Arkin, "Memory in microbes: quantifying history-dependent behavior in a bacterium.," *PloS one*, vol. 3, p. e1700, Jan. 2008.

[9] J. S. Mattick, "Non-coding RNAs: the architects of eukaryotic complexity.," *EMBO reports*, vol. 2, pp. 986–91, Nov. 2001.

[10] J. Stelling, S. Klamt, K. Bettenbrock, S. Schuster, and E. D. Gilles, "Metabolic network structure determines key aspects of functionality and regulation.," *Nature*, vol. 420, pp. 190–3, Nov. 2002.

[11] D. V. Buonomano and W. Maass, "State-dependent computations: spatiotemporal processing in cortical networks.," *Nature reviews. Neuroscience*, vol. 10, pp. 113–25, Feb. 2009.

[12] M. Lukoševičius and H. Jaeger, "Reservoir computing approaches to recurrent neural network training," *Computer Science Review*, vol. 3, pp. 127–149, Aug. 2009.

[13] D. Verstraeten, B. Schrauwen, M. D'Haene, and D. Stroobandt, "An experimental unification of reservoir computing methods.," *Neural networks*, vol. 20, pp. 391–403, Apr. 2007.

[14] W. Maass, T. Natschläger, and H. Markram, "Real-time computing without stable states: a new framework for neural computation based on perturbations.," *Neural computation*, vol. 14, pp. 2531–60, Nov. 2002.

[15] H. Jaeger, "The "echo state" approach to analysing and training recurrent neural networks-with an erratum note'," tech. rep., Fraunhofer Institute for Autonomous Intelligent Systems, 2001.

[16] A. Rodan and P. Tino, "Minimum complexity echo state network.," *IEEE transactions on neural networks*, vol. 22, pp. 131–44, Jan. 2011.

[17] N. Sierro, Y. Makita, M. de Hoon, and K. Nakai, "DBTBS: a database of transcriptional regulation in Bacillus subtilis containing upstream intergenic conservation information.," *Nucleic acids research*, vol. 36, pp. D93–6, Jan. 2008.

[18] I. M. Keseler *et al.*, "EcoCyc: a comprehensive database of Escherichia coli biology.," *Nucleic acids research*, vol. 39, pp. D583–90, Jan. 2011.

[19] M. C. Teixeira *et al.*, "The YEASTRACT database: an upgraded information system for the analysis of gene and genomic transcription regulation in Saccharomyces cerevisiae.," *Nucleic acids research*, vol. 42, pp. D161–6, Jan. 2014.

[20] S. Roy *et al.*, "Identification of functional elements and regulatory circuits by Drosophila modENCODE.," *Science (New York, N.Y.)*, vol. 330, pp. 1787–97, Dec. 2010.

[21] M. B. Gerstein *et al.*, "Architecture of the human regulatory network derived from ENCODE data.," *Nature*, vol. 489, pp. 91–100, Sept. 2012.

[22] D. Verstraeten and B. Schrauwen, "Oger: Modular Learning Architectures For Large-Scale Sequential Processing," *... of Machine Learning ...*, vol. 13, pp. 2995–2998, 2012.

[23] F. Wyffels, B. Schrauwen, and D. Stroobandt, "Stable output feedback in reservoir computing using ridge regression," *Artificial Neural Networks-ICANN 2008*, vol. 5163, pp. 808–817, 2008.

[24] A. F. Atiya and A. G. Parlos, "New results on recurrent network training: unifying the algorithms and accelerating convergence.," *IEEE transactions on neural networks / a publication of the IEEE Neural Networks Council*, vol. 11, pp. 697–709, Jan. 2000.