iScience, Volume *24*

Supplemental information

Gene regulatory networks exhibit several kinds of memory: quantification of memory in biological and random transcriptional networks

Surama Biswas, Santosh Manicka, Erik Hoel, and Michael Levin

Transparent Methods

Biological GRN models

We used a set of 35 models of GRNs downloaded from an online repository called *Cell Collective* (Helikar et al., 2012), consisting of a maximum of 25 nodes each. Each model is defined as a standard Boolean Network (BN) (Herrmann et al., 2012): a discrete dynamical system whose nodes represent the components of the system (e.g., genes or proteins) that can be in one of two states, namely 1 (ON) or 0 (OFF), and whose edges represent the regulatory interactions (activation/repression) among the nodes, dictating their states (Kauffman et al., 2003). The state of a BN is represented as a vector of the individual gene states, updated synchronously in discrete time-steps: the state of each gene at time $t + 1$ is determined by a Boolean function of the states of its input genes at time t (Shmulevich and Kauffman, 2004). The BNs in the *Cell Collective* database are defined using only the elementary Boolean functions, namely AND, OR and NOT, since any Boolean function can be expressed using some combination of these elementary operators. A BN is simulated by initializing it with some state, then updating it to obtain the next state, and so on, for a specified number of time-steps. When a BN is simulated for a long enough time, it reaches an attractor state. An attractor may consist of a single BN state, known as a "point attractor", or may consist of a set of states that the network cycles through, known as a "cyclic attractor." A BN can have multiple attractors, and different inputs may lead to different attractors (Graudenzi et al., 2011; Groß et al., 2019; Mochizuki et al., 2013; Naldi et al., 2018; Serraa et al., 2007; Shmulevich and Dougherty, 2010; Shmulevich and Kauffman, 2004; Veliz-Cuba et al., 2014; Xiao, 2009). In this work, we compute the memory profile of BNs in a manner that pays attention to its attractor states in order to avoid the effects of the transient dynamics on the analyses. This imposes a limitation on the size of networks considered here because the larger the network, the longer it takes to reach an attractor. This *transient length* to reach an attractor depends on the *Network Size* (the number of nodes in the network) and the *Edge Density* defined as (Number of edges / Total number of possible edges). We found that the transient length (Supplement 14) rises exponentially above 500 time-steps (a practical limit that we chose for this work) for networks of size larger than 25 with a biologically realistic edge density of 10% (Supplement 15). As a result, we restricted ourselves to analysis of BNs of size <=25 to be able to exhaustively analyze all our networks.

Synthetic GRN models for comparison

To evaluate the significance of the memory profiles of the biological GRN models, we generated synthetic Null models for comparison: 1) a set of 3500 BNs obtained by randomizing each GRN 100 times, known as "configuration models"; and 2) a set of 500 random Boolean networks (RBN).

We generated a set of 100 configuration models for each one of the 35 biological GRN models. There are many ways to generate the configuration (Null) models, depending on the null hypothesis that one wishes to consider (Zhai et al., 2018). Since our principal motivation is the idea that memory in GRNs may be mediated by the dynamic relationships among the node's mechanisms, our null hypothesis is therefore that those dynamic relationships don't play a role in mediating memory-related phenomena but are entirely governed by things like edge degree. Therefore, in generating each configuration model we kept the number of nodes and the indegree distribution the same as the original GRN, while randomizing only the inputs to the nodes and the associated Boolean functions. That is, each node in the configuration model has the same number of inputs, but the actual input nodes will be different compared to the original model. Similarly, each Boolean function in the configuration model has the same number of inputs as the original but the Boolean operators are randomly chosen from the set of elementary operators (AND, OR, NOT).

To determine how the memory properties of networks vary with network size in general, we generated five sets of 100 RBNs each, of size 5, 10, 15, 20 and 25 nodes respectively. The edge density was set to $max(10\%\ of\ N^2, N-1)$, as the average edge density of the biological GRNs was found to be ~10%. Unlike the configuration models, we generated an RBN by first randomly choosing unique source-target node pairs and assigning a directed edge between them such that the total number of edges satisfied the specified edge density, and then assigning random Boolean functions to each node. We generated a random Boolean function for a given node as follows. First, we considered the inputs of the node X that may consist of just one input (X itself or some other node,) or more than one input. In the case of the former, the Boolean function may take one of the following forms: 'X =X', 'X = Y' or 'X=~Y', where '~' represents logical NOT (invert) operation. If there are two or more inputs, such as (Y, Z), the Boolean function may take one of the following forms: (Y⊗Z), (~Y⊗Z), (Y⊗~Z) or (~Y⊗~Z), where ⊗ represents a Boolean operator randomly chosen from the list of Boolean operators (AND, OR and XOR). For more than two inputs, the Boolean functions would simply be larger compositions of the above. We then randomly applied NOT operation in the final or intermediate stages of the equation so that 50% of the nodes were affected.

Synthetic GRN models for illustration

To illustrate the phenomenon of memory formation in BNs, we generated 10000 minimal RBNs consisting of 2 and 3 nodes (Figure 5). The process of making the minimal models was same as that of making RBNs except the fact that here we selected a higher edge density randomly in [50, 100]. As we are interested here finding memory in the small networks where the number of nodes is few (2/3), a denser topology is required to produce memory. We first investigated the minimum number of nodes required to form a certain type of memory. In the case of UCS based memory, the minimum requirement was 2 (UCS and R); for all other types of memory it was 3 (UCS, NS/CS, and R). We then fixed the edge density at a random percentage between 50 and 100. We evaluated memory and took as minimal the one which had the fewest edges.

Memory detection

We defined different types of memories, characterized by a specific number and timing of the stimuli, as described below. For each network, we looked for possible memories by considering all possible choices of nodes to serve as inputs or outputs in a training assay. We exhaustively considered all choices of nodes subject to the requirement that any node can only be a valid UCS if it triggers R prior to training, and any node can serve as a NS if it does not trigger R prior to training. In 3 cases, (Arabidopsis thaliana Cell Cycle, Iron acquisition and oxidative stress response in aspergillus fumigatus and Budding Yeast Cell Cycle 2009), we could not find any combinations matching this feasibility condition and thus considered the amount of “no memory” to be 100%. The set of all feasible stimulus-response combinations is a subset of all possible combinations, the cardinality of which is given by $P(N, 3) = \frac{N!}{(N-3)!}$. We compute a memory profile for each feasible combination by passing it through a series of detection steps (Figure 3). We first let the BN settle on an attractor by initiating it with a state consisting of all “off” and simulating it for 500 time-steps.

Then, we evaluated the memory of each network, given a choice of nodes as CS, UCS, and R via a sequence of steps picked from the following general recipe (the specific steps followed depends on the type of the memory being evaluated): 1) choose a stimulus set; 2) flip the state of the stimuli and fix them in that state, referred to as *clamping* (we did not let other genes to alter the state of UCS and all equations associated with different genes and UCS get the clamped value of UCS); 3) simulate the BN for *M* time-steps; 4) record the state of *R* compared to its state prior to the clamping step; 5) unclamp the stimuli (allow them to update states), referred to as *relaxation*; 6) simulate the BN for *M* time-steps; 7) record the state of *R* compared to its state prior to relaxation; 8) choose a different stimulus set; 9) flip and clamp the stimuli; 10) simulate the BN for *M* time-steps; 11) record the state of *R* compared to its state prior to the clamping step 9; 12) relax the network; and 13) record the state of *R*. We deemed a given stimulus-response combination as having elicited a specific type of memory if it satisfies the associated set of conditions:

i) UCS *Based Memory* (UM): choose the stimulus set consisting of in step 1, verify that R has flipped in step 3, and finally verify that R has *not* flipped in step 7. UM captures the idea that R may permanently remember changes in the activity of UCS.

ii) *Pairing Memory* (PM): choose the stimulus set consisting of {UCS, NS} in step 1, verify that R has flipped in step 3, and finally verify that R has *not* flipped in step 7. PM captures the idea that R may permanently remember changes in the joint activities of UCS and NS. Even though the detection of PM is like AM, there are crucial differences (see AM definition below).

iii) *Transfer Memory* (TM): choose the stimulus set consisting of {UCS} in step 1, verify that R has flipped in step 3, choose the stimulus set consisting of {NS} in step 8, and finally verify that *R* has flipped in step 11. TM captures the possibility that even though NS could not flip R initially, it may be able to do so after activating UCS, effectively transforming NS into CS.

iv) *Associative Memory* (AM): choose the stimulus set consisting of {UCS, NS} in step 1, verify that R has flipped in step 3, choose the stimulus set consisting of {NS} in step 8, and finally verify that *R* has flipped in step 11. AM describes classical conditioning: after successful pairing of UCS and current NS, the NS is conditioned to become CS. This causes the NS to become CS and can be able to trigger R. In other words, we call it an AM if after successful pairing, NS can flip R.

   a. *Long Recall Associative Memory* (LRAM): Following the AM steps, verify that R has *not* flipped in step 13 compared to its state prior to the relaxation step 12. LRAM captures the idea that R may permanently remember changes to the activity of CS.

   b. *Short Recall Associative Memory* (SRAM): Following the AM steps, verify that R has flipped in step 13 compared to its state prior to the relaxation step 12. SRAM captures the idea that R may only transiently remember changes to the activity of CS.

v) *Consolidation Memory* (CM): choose the stimulus set consisting of {UCS, NS} in step 1, verify that R has flipped in step 3, choose the stimulus set consisting of {NS} in step 8, verify that *R* has *not* flipped in step 11, and finally verify that R has flipped compared to its state prior to the clamping step 9. CM captures the idea that even though associative conditioning may not immediately turn NS into CS, it may do so after relaxing the BN.

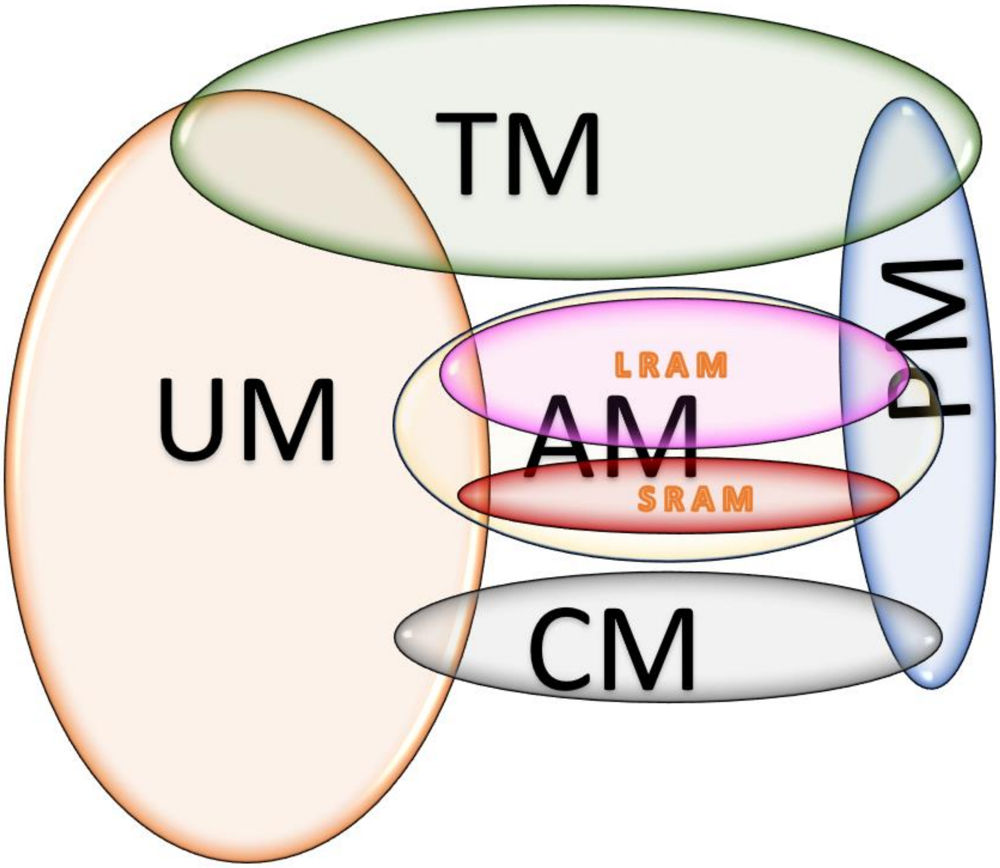
Note that UM and PM are mutually exclusive, as are TM and {AM, CM} (see Figures 2,3 for details).

After confirmation of each case of Transfer, Associative and Consolidation memory, we checked whether the change in the property of NS in inducing R is permanent. We deactivated the CS to check if R is also deactivated; again we activated CS to see if R is triggered back, and continued the activation/deactivation process 20 times to see if causality between NS and R is stable. If stable, we called it second order memory.

Mathematically, in an N node GRN, there may be P_N^3 such combinations. Here, we considered the current node as R if the R is stable over a certain period called *Constancy Length* during the relaxation phase of the network (see Supplement 16). We coded the methodology in MATLAB 2019a.

Supplements

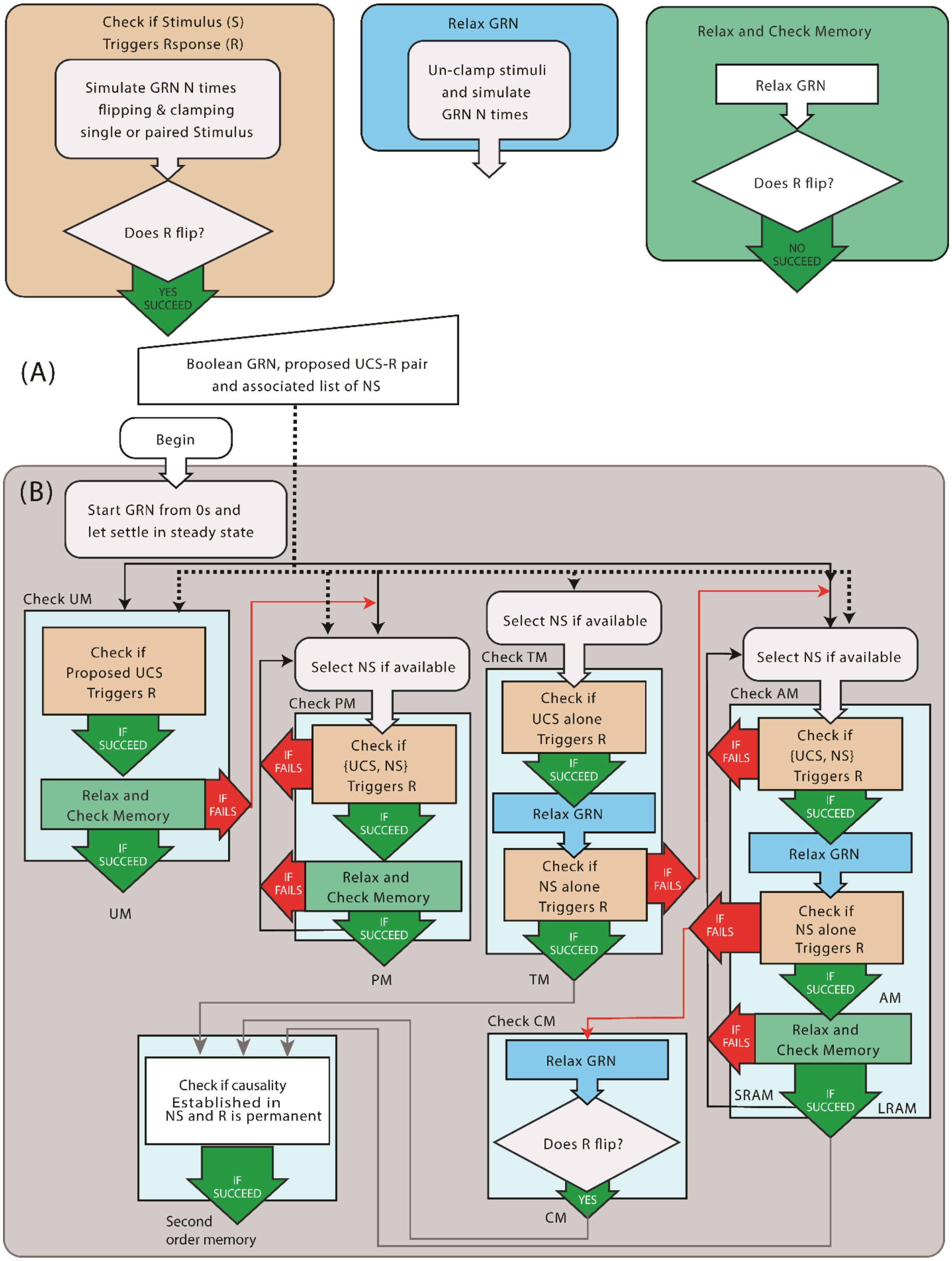
Supplemental Figure 1: Definition of and functional relationship among the different memory types [This item relates to Figure 2]



UM: UCS Based Memory
PM: Pairing Memory
TM: Transfer Memory
AM: Associative Memory
LRAM: Long Recall AM
SRAM: Short Recall AM
CM: Consolidation Memory
NM: No Memory

Legend: The definition and abbreviations of the defined memory types are as follows. UCS Based Memory UM: R retains the activation by UCS after UCS deactivated. Pairing Memory (PM): R retains the repetitive activation by {UCS, NS} pair even after their deactivation. Transfer Memory (TM): activation by UCS alone (not pairing) converts NS to CS. Associative Memory (AM): paired activation of {UCS, NS}, converts NS to CS. Long Recall AM (LRAM): this conversion of NS to CS is permanent. Short Recall AM (SRAM): the conversion is temporary (the association is lost). Consolidation Memory (CM): the pairing of {UCS, NS} does not immediately turn NS into CS but eventually does so after an elapsed time. The overlap/hierarchy of the ovals represents the relationship between the different types and subtypes of memory.

Supplemental Figure 2: Flowchart of Memory Evaluation [This item relates to Figure 2]



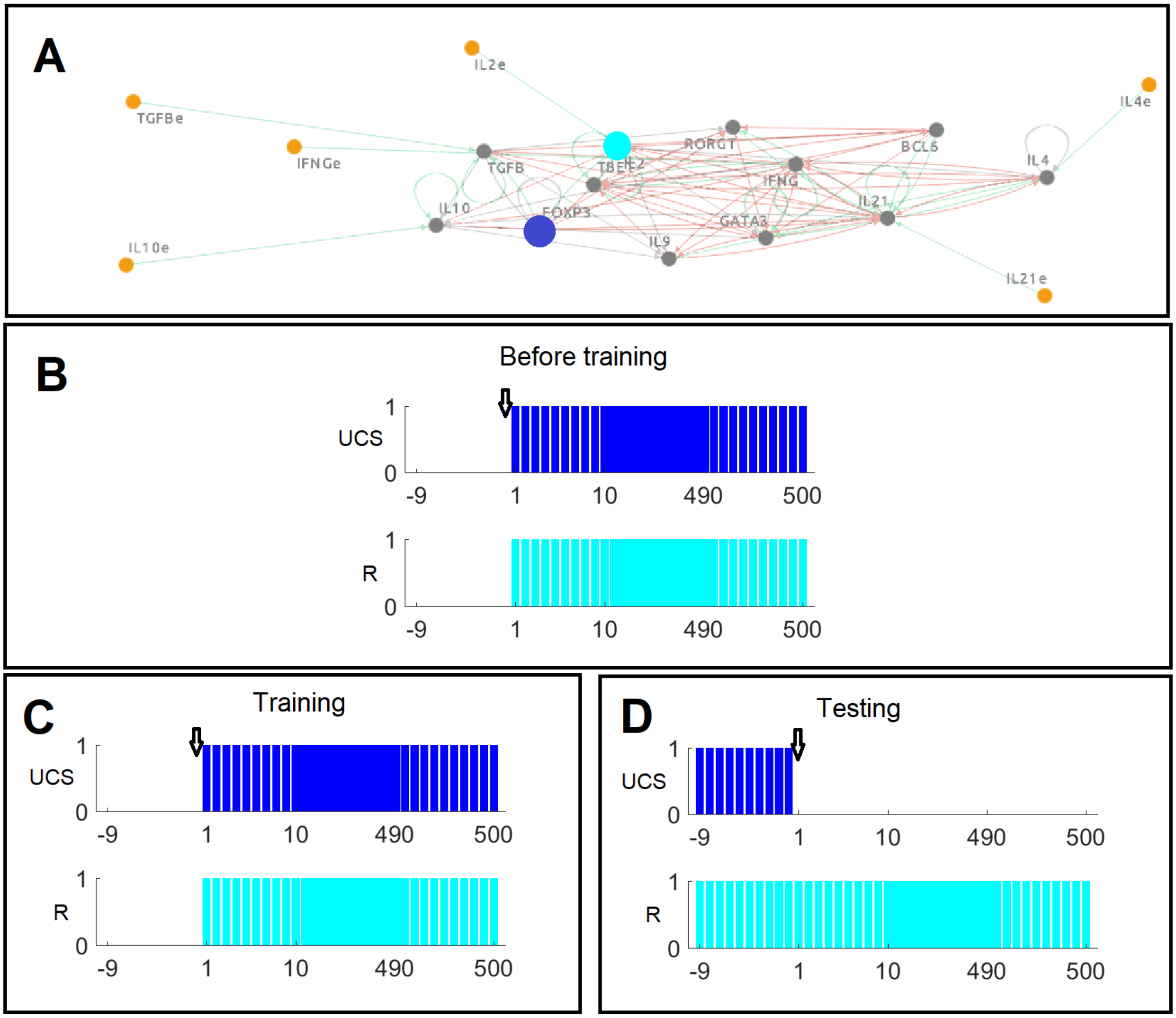
Legend: The computational procedures for our evaluation of five kinds of memories are shown here, namely, UM, PM, TM, AM and CM. We consider each of the two subcategories of AM, LRAM and SRAM, as individual memory types. (A) Input of a GRN with a R-UCS pair and a probable list of NS. (B) The memory detection process. At the top of the figure we define the different modules frequently used in the section B. The process works as follows. 1) choose a stimulus set; 2) flip the state of the stimuli and fix them in that state, referred to as clamping; 3) simulate the BN for M time-steps; 4) record the state of R compared to its state prior to the clamping step; 5) unclamp the stimuli (allow them to update states), referred to as relaxation; 6) simulate the BN for M time-steps; 7) record the state of R compared to its state prior to relaxation; 8) choose a different stimulus set; 9) flip and clamp the stimuli; 10) simulate the BN for M time-steps; 11) record the state of R compared to its state prior to the clamping step 9; 12) relax the network; and 13) record the state of R. We deem a given stimulus-response combination as having elicited a specific type of memory if it satisfies a number of specific conditions described fully in the Methods.

Supplemental Table 1: GRNs analyzed from cell collective [This item relates to Figure 3]

IDs	GRNs
#1	Arabidopsis thaliana Cell Cycle
#2	Aurora Kinase A in Neuroblastoma
#3	B cell differentiation
#4	BT474 Breast Cell Line Long-term ErbB Network
#5	BT474 Breast Cell Line Short-term ErbB Network
#6	Body Segmentation in Drosophila 2013
#7	Budding Yeast Cell Cycle
#8	Budding Yeast Cell Cycle 2009
#9	CD4+ T Cell Differentiation and Plasticity
#10	Cardiac development
#11	Cell Cycle Transcription by Coupled CDK and Network Oscillators
#12	Cortical Area Development
#13	FGF pathway of Drosophila Signaling Pathways
#14	Fanconi anemia and checkpoint recovery
#15	HCC1954 Breast Cell Line Long-term ErbB Network
#16	HCC1954 Breast Cell Line Short-term ErbB Network
#17	HH Pathway of Drosophila Signaling Pathways
#18	Human Gonadal Sex Determination
#19	Iron acquisition and oxidative stress response in aspergillus fumigatus
#20	Lac Operon
#21	Mammalian Cell Cycle
#22	Mammalian Cell Cycle 2006
#23	Metabolic Interactions in the Gut Microbiome
#24	Neurotransmitter Signaling Pathway
#25	Oxidative Stress Pathway
#26	Predicting Variabilities in Cardiac Gene
#27	Processing of Spz Network from the Drosophila Signaling Pathway
#28	Regulation of the L-arabinose operon of Escherichia coli
#29	SKBR3 Breast Cell Line Long-term ErbB Network
#30	SKBR3 Breast Cell Line Short-term ErbB Network
#31	T cell differentiation
#32	T-LGL Survival Network 2011 Reduced Network
#33	TOL Regulatory Network
#34	Toll Pathway of Drosophila Signaling Pathway
#35	VEGF Pathway of Drosophila Signaling Pathway

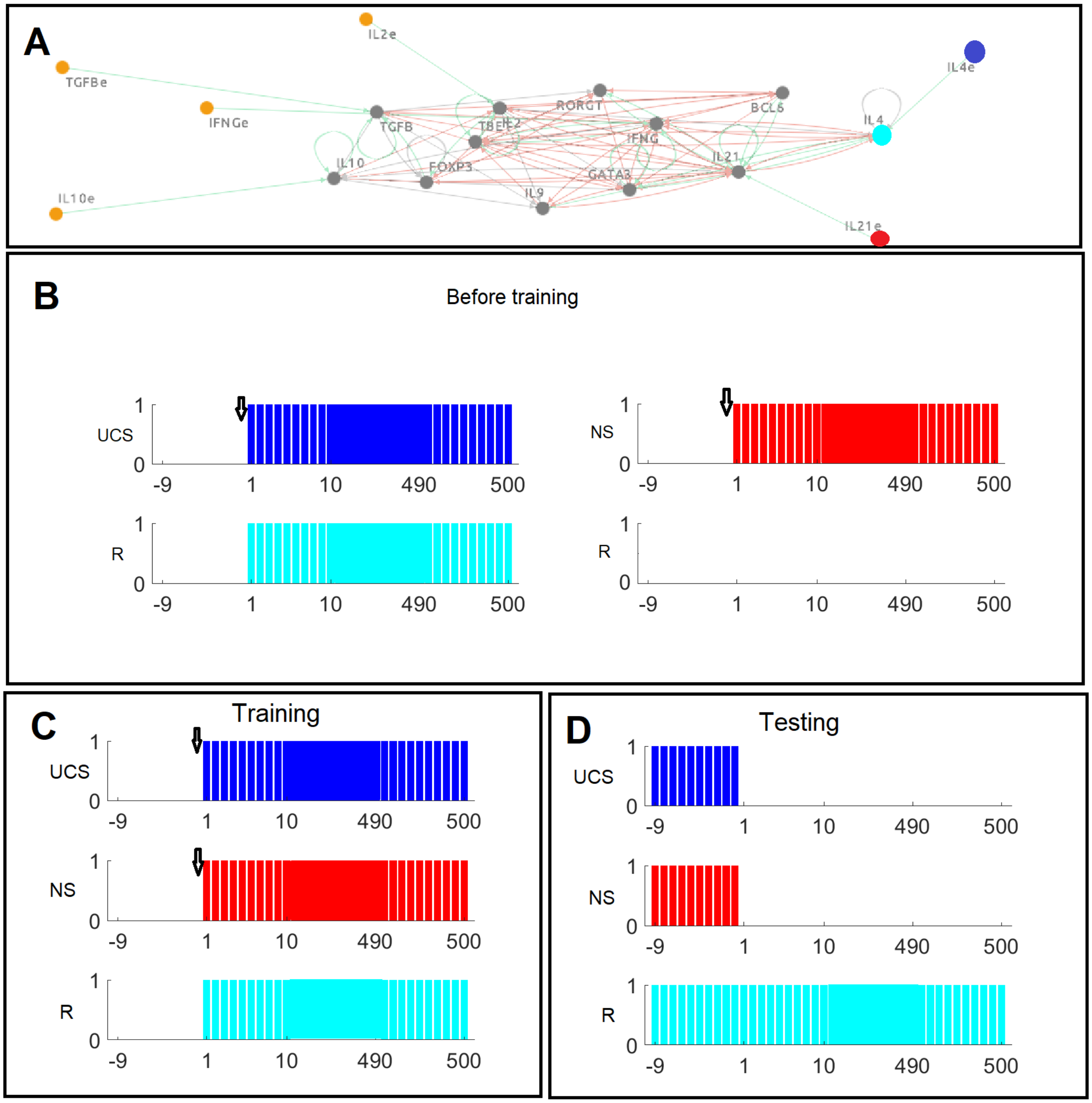
Legend: GRNs from Cell Collective that were analyzed, all having 25 of fewer nodes.

Supplemental Figure 3: Time series data from the evaluation of a sample UCS Based Memory [This item relates to Figure 4]



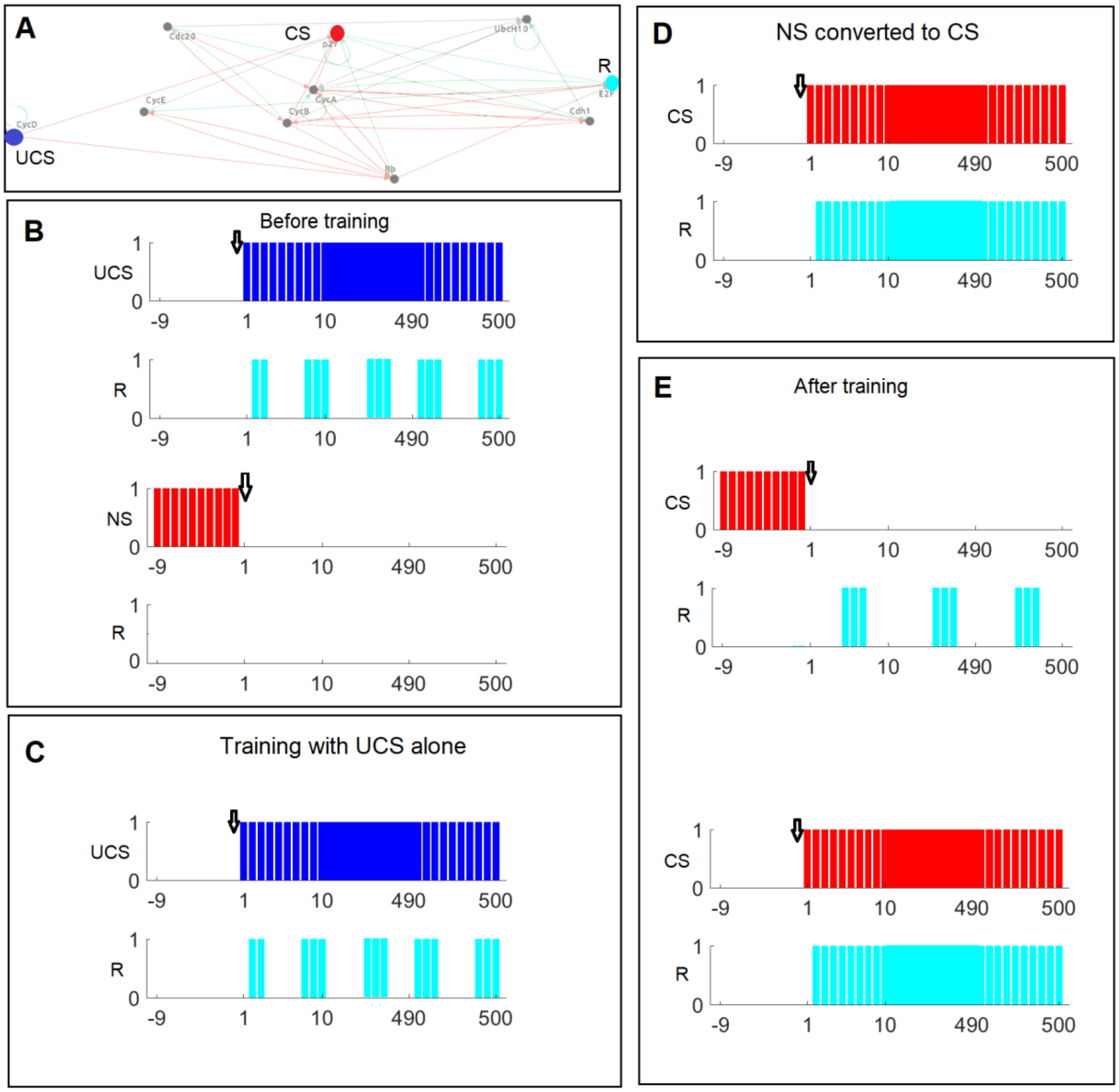
Legend: These node traces show the timeseries data for an example of UCS Based memory evaluation in the CD4+ T Cell Differentiation and Plasticity GRN. Here, the memory is established between FOXP3 gene as UCS and IL2 gene as response. A) The network with specified stimulus-response combination. B) The pre-requisite before learning that UCS has the capability of triggering R. C) The training of R by inducing UCS repeatedly. D) Testing of R making UCS off.

Supplemental Figure 4: Time series data from the evaluation of a sample Pairing Memory [This item relates to Figure 4]



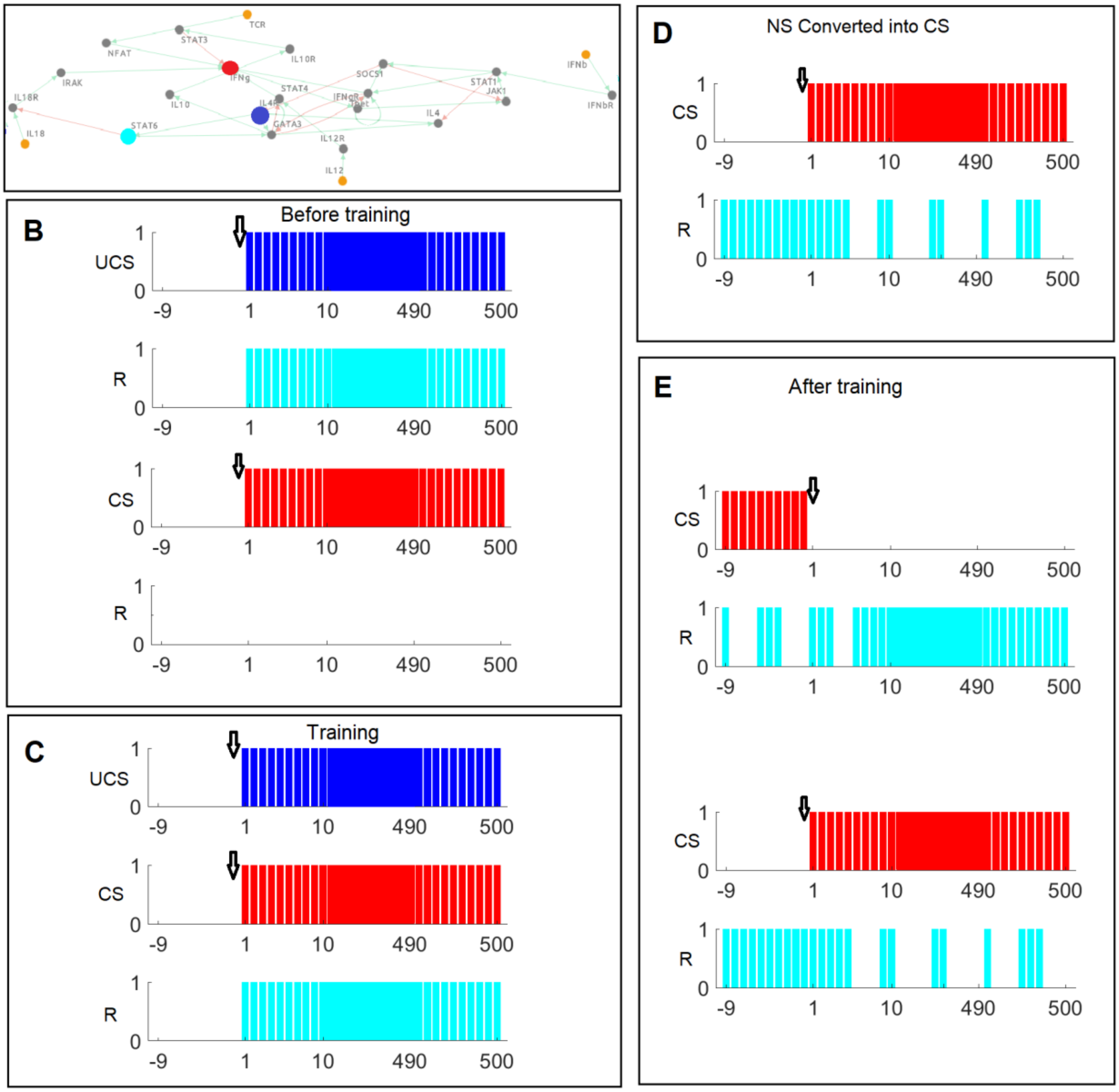
Legend: These node traces show the timeseries data for an example of Pairing memory evaluation in the CD4+ T Cell Differentiation and Plasticity GRN. Here, the memory is established among IL4e gene as UCS, IL2e as NS and IL4 gene as response. A) The network with specified stimulus-response combination. B) The pre-requisite before learning that UCS has the capability of triggering R and that NS should not trigger R. C) The training by inducing {UCS, NS} together repeatedly. D) testing of R making the stimuli off.

Supplemental Figure 5: Time series data from an evaluation of a sample Transfer Memory [This item relates to Figure 4]



Legend: These node traces show the timeseries data for an example the evaluation of Transfer memory in Mammalian Cell Cycle 2006 GRN. Here, the memory is established among CycD gene as UCS, P27 as NS/CS and E2F gene as response. A) The network with specified stimulus-response combination. B) The pre-requisite before learning that UCS has the capability of triggering R and that NS should not trigger R. C) The training by inducing {UCS, NS} together repeatedly. D) Testing of R to check if NS has converted to CS through training. Here, when the CS was turned off after learning, the R is not totally off but exhibits some ripples; during the on phase of CS response of R is consistent. E) As further confirmation of stable causality established between CS and R by training, we first deactivated CS, to see R get deactivated, and then reactivated the CS to ensure that it can activate R again.

Supplemental Figure 6: Time series data from the evaluation of a sample Consolidation Memory [This item relates to Figure 4]



Legend: These node traces show the timeseries data for a Consolidation memory in T cell differentiation GRN. Here, the memory is established among IL4R gene as UCS, IFNG as NS/CS and STAT6 gene as response. A) The network with specified stimulus-response combination. B) The pre-requisite of learning, stated as before learning shows that UCS has the capability of triggering R and that NS should not trigger R. C) Sows The training by inducing {UCS, NS} together repeatedly and D) testing that NS is converted to CS, i.e. it alone can induce R. Here, CS has an inverse relation with R, i.e. when CS is on makes R off and when CS goes off triggers R. The CS being on does not make R totally off but exhibits some ripples in R. But when CS is off, R lately becomes fully ON.

E) As further confirmation of stable causality established between CS and R by training, we first deactivate CS, to see R get deactivated, and then reactivated the CS to ensure that it can activate R again.

Data S1: Set of Boolean Expressions for each of the GRNs. [This item relates to Figure 3]

Legend: See file Expressions.zip – This file provides Boolean equations required to simulate each of the 35 GRNs used here from Cell Collective database (https://cellcollective.org/). The equation associated with a gene of a GRN comprises its regulators related by Boolean operators like AND, OR and NOT. We evaluate the equation during GRN simulation and assigns the result as the state of the current gene. We assign 0 to an external component during simulation.

Supplemental Table 2: Memory Evaluation of GRNs [This item relates to Figure 7]

UM	PM	TM	AM	LRAM	SRAM	CM	NM
0.00	0.00	0.00	0.00	0.00	0.00	0.00	100.00
29.79	0.51	6.98	2.64	2.47	0.17	0.00	60.09
46.82	0.74	11.12	0.11	0.11	0.00	0.00	41.21
95.25	0.00	4.75	0.00	0.00	0.00	0.00	0.00
96.02	0.00	3.98	0.00	0.00	0.00	0.00	0.00
0.00	0.00	0.00	0.00	0.00	0.00	0.00	100.00
0.00	0.00	0.00	0.00	0.00	0.00	0.00	100.00
0.00	0.00	0.00	0.00	0.00	0.00	0.00	100.00
59.94	2.21	29.02	0.95	0.95	0.00	0.00	7.89
47.58	0.00	3.86	0.00	0.00	0.00	0.00	48.55
0.00	0.00	0.00	2.33	2.33	0.00	0.00	97.67
42.11	0.00	21.05	0.00	0.00	0.00	0.00	36.84
0.00	0.00	0.00	0.00	0.00	0.00	0.00	100.00
0.00	0.00	0.00	0.00	0.00	0.00	2.84	97.16
95.22	0.00	4.78	0.00	0.00	0.00	0.00	0.00
90.76	0.00	9.24	0.00	0.00	0.00	0.00	0.00
0.00	0.00	0.00	0.00	0.00	0.00	0.00	100.00
81.19	0.30	16.45	0.00	0.00	0.00	0.00	2.06
0.00	0.00	0.00	0.00	0.00	0.00	0.00	100.00
0.00	0.00	0.00	0.00	0.00	0.00	0.00	100.00
76.63	0.00	10.02	0.00	0.00	0.00	0.00	13.36
12.04	0.00	4.63	2.78	0.93	1.85	0.00	80.56

80.41	0.00	9.28	0.00	0.00	0.00	0.00	10.31
0.00	1.16	0.58	1.16	1.16	0.00	0.00	97.09
0.00	0.00	0.00	0.00	0.00	0.00	0.00	100.00
47.58	0.00	3.86	0.00	0.00	0.00	0.00	48.55
0.00	0.00	0.00	0.00	0.00	0.00	0.00	100.00
0.00	0.00	0.00	0.00	0.00	0.00	0.00	100.00
93.20	0.00	6.80	0.00	0.00	0.00	0.00	0.00
97.60	0.00	2.40	0.00	0.00	0.00	0.00	0.00
64.48	0.00	15.51	0.00	0.00	0.00	0.07	19.94
93.71	0.00	0.00	0.00	0.00	0.00	0.00	6.29
0.00	0.00	0.00	0.00	0.00	0.00	0.00	100.00
0.00	0.00	0.00	0.00	0.00	0.00	0.00	100.00
0.00	0.00	0.00	0.00	0.00	0.00	0.00	100.00

Legend: For each GRN, the proportion of each memory type out of the total memory (within the available feasible combinations) has been calculated and put in the table. For those networks where no feasible combinations of stimulus/response were available, the proportion of no memory (NM) was considered as 100%.

Supplemental Table 3: p-value Test: comparing incidence of memories within GRNs vs. RBNs. [This item relates to Figure 8]

GRNs	UM	PM	TM	AM	LRAM	SRAM	CM	NM
#1	0.85	0.99	0.96	1	1	1	0.99	0.85
#2	0.1	0.15	0.08	0.03	0.03	0.09	0.7	0.08
#3	0.02	0.12	0.06	0.76	0.81	0.9	0.7	0.03
#4	0	0.8	0.15	0.73	0.81	0.86	0.58	0
#5	0	0.86	0.11	0.84	0.89	0.93	0.88	0
#6	0.62	0.86	0.74	0.84	0.89	0.93	0.88	0.52
#7	0.57	0.81	0.58	0.76	0.81	0.9	0.7	0.33
#8	0.62	0.86	0.74	0.84	0.89	0.93	0.88	0.52
#9	0.02	0.03	0	0.1	0.05	0.93	0.88	0
#10	0.05	0.86	0.11	0.84	0.89	0.93	0.88	0.06
#11	0.92	1	0.98	0	0	1	1	0.92
#12	0.06	1	0.02	1	1	1	1	0.06
#13	0.57	0.81	0.58	0.76	0.81	0.9	0.7	0.33
#14	0.62	0.86	0.74	0.84	0.89	0.93	0.03	0.58
#15	0	0.8	0.15	0.73	0.81	0.86	0.58	0
#16	0	0.86	0.02	0.84	0.89	0.93	0.88	0
#17	0.57	0.81	0.58	0.76	0.81	0.9	0.7	0.33
#18	0	0.14	0.01	0.84	0.89	0.93	0.88	0
#19	0.57	0.81	0.58	0.76	0.81	0.9	0.7	0.33
#20	0.85	0.99	0.96	1	1	1	0.99	0.85
#21	0	0.81	0.07	0.76	0.81	0.9	0.7	0.01
#22	0.13	0.99	0	0	0	0	0.99	0.1
#23	0	0.99	0	1	1	1	0.99	0
#24	0.62	0.04	0.74	0.07	0.04	0.93	0.88	0.58
#25	0.62	0.86	0.74	0.84	0.89	0.93	0.88	0.52
#26	0.05	0.86	0.11	0.84	0.89	0.93	0.88	0.06
#27	0.57	0.81	0.58	0.76	0.81	0.9	0.7	0.33
#28	0.85	0.99	0.96	1	1	1	0.99	0.85
#29	0	0.8	0.1	0.73	0.81	0.86	0.58	0
#30	0	0.86	0.2	0.84	0.89	0.93	0.88	0
#31	0	0.81	0.03	0.76	0.81	0.9	0.7	0.01
#32	0	0.86	0.74	0.84	0.89	0.93	0.88	0
#33	0.57	0.81	0.58	0.76	0.81	0.9	0.7	0.33
#34	0.85	0.99	0.96	1	1	1	0.99	0.85
#35	0.62	0.86	0.74	0.84	0.89	0.93	0.88	0.52
(%)	37.142	5.714	20	5.571	11.428	2.857	2.857	37.142

Legend: For each GRN, we calculated a p-value for the position of the incidence of each type of memory within the probability distribution of 100 similar size RBNs. If the p-value is less than or equal to 0.05, the statistical test rejects the null hypothesis that the incidence of that type of memory in the GRN is not an outlier amongst similar sized RBN memories.

Supplemental Table 4: Outlier Test: comparing incidence of memories within GRNs vs. RBNs. [This item relates to Figure 8]

GRNs	UM	PM	TM	AM	LRAM	SRAM	CM	NM
#1	0	0	0	0	0	0	0	0
#2	0	0	0	1	1	0	0	0
#3	1	0	0	0	0	0	0	1
#4	1	0	0	0	0	0	0	1
#5	1	0	0	0	0	0	0	1
#6	0	0	0	0	0	0	0	0
#7	0	0	0	0	0	0	0	0
#8	0	0	0	0	0	0	0	0
#9	1	1	1	0	0	0	0	1
#10	0	0	0	0	0	0	0	0
#11	0	0	0	1	1	0	0	0
#12	0	0	1	0	0	0	0	0
#13	0	0	0	0	0	0	0	0
#14	0	0	0	0	0	0	1	0
#15	1	0	0	0	0	0	0	1
#16	1	0	1	0	0	0	0	1
#17	0	0	0	0	0	0	0	0
#18	1	0	1	0	0	0	0	1
#19	0	0	0	0	0	0	0	0
#20	0	0	0	0	0	0	0	0
#21	1	0	0	0	0	0	0	1
#22	0	0	1	1	1	1	0	0
#23	1	0	1	0	0	0	0	1
#24	0	1	0	0	1	0	0	0
#25	0	0	0	0	0	0	0	0
#26	0	0	0	0	0	0	0	0
#27	0	0	0	0	0	0	0	0
#28	0	0	0	0	0	0	0	0
#29	1	0	0	0	0	0	0	1
#30	1	0	0	0	0	0	0	1
#31	1	0	1	0	0	0	0	1
#32	1	0	0	0	0	0	0	1
#33	0	0	0	0	0	0	0	0
#34	0	0	0	0	0	0	0	0
#35	0	0	0	0	0	0	0	0
# Outlier	13	2	7	3	4	1	1	13
(%)	37.142	5.714	20	5.571	11.428	2.857	2.857	37.142

Legend: We tested whether the incidence of each type of memory of a GRN is an outlier in the pool of similar size of 100 RBNs: ‘1’ if outlier; ‘0’ otherwise. The last two rows show total number and percentage of outliers in each memory type.

Supplemental Table 5: p-value Test: comparing memories in GRNs vs. in Configuration models [This item relates to Figure 9]

GRNs	UM	PM	TM	AM	LRAM	SRAM	CM	NM
#1	0.7	0.92	0.74	0.89	0.94	0.92	0.79	0.55
#2	0.05	0.05	0.03	0.01	0.01	0.04	0.89	0.03
#3	0.03	0.02	0.01	0.04	0.04	1	0.93	0.02
#4	0	0.92	0.05	0.91	0.93	0.98	0.93	0
#5	0	0.95	0.11	0.97	0.97	1	0.96	0
#6	0.76	0.9	0.78	0.91	0.92	0.99	0.96	0.68
#7	0.76	0.9	0.74	0.92	0.93	0.99	0.94	0.68
#8	0.62	0.81	0.6	0.86	0.89	0.95	0.84	0.46
#9	0	0	0	0	0	1	0.94	0
#10	0.11	0.83	0.2	0.85	0.87	0.97	0.84	0.13
#11	0.7	0.9	0.71	0.08	0.07	0.98	0.94	0.6
#12	0.19	0.98	0.1	0.98	0.98	1	0.99	0.19
#13	0.85	0.96	0.93	0.97	0.97	1	1	0.83
#14	0.73	0.89	0.7	0.92	0.93	0.99	0.04	0.64
#15	0	0.92	0.08	0.92	0.93	0.98	0.91	0
#16	0	0.95	0.06	0.95	0.95	1	0.95	0
#17	0.94	0.99	0.94	0.99	1	0.99	1	0.93
#18	0	0.13	0.03	0.81	0.85	0.93	0.72	0
#19	0.64	0.87	0.62	0.86	0.88	0.97	0.93	0.52
#20	0.75	0.96	0.78	0.93	0.95	0.98	0.98	0.68
#21	0	0.87	0.08	0.85	0.89	0.94	0.82	0
#22	0.25	0.9	0.25	0.07	0.1	0.01	0.89	0.24
#23	0	0.94	0.1	0.94	0.94	1	0.96	0
#24	0.6	0.11	0.68	0.17	0.12	0.92	0.85	0.45
#25	0.67	0.92	0.71	0.91	0.93	0.98	0.87	0.54
#26	0.05	0.89	0.16	0.88	0.9	0.96	0.9	0.06
#27	0.74	0.97	0.85	0.94	0.95	0.99	0.96	0.69
#28	0.83	0.97	0.83	0.97	0.98	0.99	0.97	0.79
#29	0	0.89	0.03	0.9	0.9	0.97	0.89	0
#30	0	0.96	0.11	0.96	0.96	1	0.99	0
#31	0	0.88	0	0.87	0.88	0.98	0.06	0
#32	0	0.81	0.6	0.78	0.8	0.93	0.7	0.01
#33	0.82	0.98	0.87	0.98	0.98	1	0.97	0.78
#34	0.69	0.96	0.88	0.97	0.97	1	0.98	0.64
#35	0.86	0.98	0.91	0.98	0.98	1	0.98	0.83
(%)	37.143	5.714	17.14	8.571	8.571	5.714	2.857	40

Legend: For each GRN, the stated p-value represents the incidence of each memory type fit into the probability distribution of its random ensemble. If the p-value is less than or equal to 0.05, the statistical test rejected the null hypothesis that the amount of the memory in the GRN is not an outlier in its random ensemble.

Supplemental Table 6: Outlier Test: comparing memory incidence in GRNs vs. Configuration models [This item relates to Figure 9]

GRNs	UM	PM	TM	AM	LRAM	SRAM	CM	NM
#1	0	0	0	0	0	0	0	0
#2	0	0	1	1	1	1	0	1
#3	1	1	1	1	1	0	0	1
#4	1	0	0	0	0	0	0	1
#5	1	0	0	0	0	0	0	1
#6	0	0	0	0	0	0	0	0
#7	0	0	0	0	0	0	0	0
#8	0	0	0	0	0	0	0	0
#9	1	1	1	1	1	0	0	1
#10	0	0	0	0	0	0	0	0
#11	0	0	0	0	0	0	0	0
#12	0	0	0	0	0	0	0	0
#13	0	0	0	0	0	0	0	0
#14	0	0	0	0	0	0	1	0
#15	1	0	0	0	0	0	0	1
#16	1	0	0	0	0	0	0	1
#17	0	0	0	0	0	0	0	0
#18	1	0	1	0	0	0	0	1
#19	0	0	0	0	0	0	0	0
#20	0	0	0	0	0	0	0	0
#21	1	0	0	0	0	0	0	1
#22	0	0	0	0	0	1	0	0
#23	1	0	0	0	0	0	0	1
#24	0	0	0	0	0	0	0	0
#25	0	0	0	0	0	0	0	0
#26	0	0	0	0	0	0	0	0
#27	0	0	0	0	0	0	0	0
#28	0	0	0	0	0	0	0	0
#29	1	0	1	0	0	0	0	1
#30	1	0	0	0	0	0	0	1
#31	1	0	1	0	0	0	0	1
#32	1	0	0	0	0	0	0	1
#33	0	0	0	0	0	0	0	0
#34	0	0	0	0	0	0	0	0
#35	0	0	0	0	0	0	0	0
#Outlier	13	2	6	3	3	2	1	14
(%)	37.143	5.714	17.14	8.571	8.571	5.714	2.857	40

Legend: We tested whether the incidence of a certain memory type of a GRN is an outlier in its random ensemble (corresponding values of 100 configuration models of the GRN). ‘1’ if outlier; ‘0’ otherwise. The last two rows show the total number and percentage of outlier in each memory type.

Supplemental Table 7: The distribution of the transient length of RBNs. [This item relates to Figure 8]

N	**Median**	**Mean**	**99%**	**Max**
10	3	3.414	12.01	29
15	11	13.648	53.01	104
20	14	19.302	86.02	142
25	77	113.639	494.18	1180
30	122	192.661	1070.54	2572
35	1605	2558.796	14593.67	36478
40	3918	6755.941	42507.43	78016

Legend: The distribution of the transient length of RBNs. This table shows transient length data obtained from simulations of 1000 RBNs for each N (number of nodes), with each RBN run 1000 times starting from a different initial condition each time. Here, transient length indicates the number of time steps taken by a Boolean network to reach an attractor from a given initial state.

Supplemental Table 8: Edge densities of biological GRNs. [This item relates to Figure 7]

GRNs	Number of Nodes	Number of Edges	Edge Density
Arabidopsis thaliana Cell Cycle	14	66	33.673
Aurora Kinase A in Neuroblastoma	23	43	8.129
B cell differentiation	22	39	8.058
BT474 Breast Cell Line Long-term ErbB Network	25	70	11.200
BT474 Breast Cell Line Short-term ErbB Network	16	46	17.969
Body Segmentation in Drosophila 2013	17	29	10.035
Budding Yeast Cell Cycle	20	42	10.500
Budding Yeast Cell Cycle 2009	18	59	18.210

CD4+ T Cell Differentiation and Plasticity	18	78	24.074
Cardiac development	15	38	16.889
Cell Cycle Transcription by Coupled CDK and Network Oscillators	9	19	23.457
Cortical Area Development	5	14	56.000
FGF pathway of Drosophila Signalling Pathways	23	24	4.537
Fanconi anemia and checkpoint recovery	15	66	29.333
HCC1954 Breast Cell Line Long-term ErbB Network	25	70	11.200
HCC1954 Breast Cell Line Short-term ErbB Network	16	46	17.969
HH Pathway of Drosophila Signaling Pathways	24	32	5.556
Human Gonadal Sex Determination	19	79	21.884
Iron acquisition and oxidative stress response in aspergillus fumigatus	22	38	7.851
Lac Operon	13	22	13.018
Mammalian Cell Cycle	20	51	12.750
Mammalian Cell Cycle 2006	10	35	35.000
Metabolic Interactions in the Gut Microbiome	12	30	20.833
Neurotransmitter Signaling Pathway	16	22	8.594
Oxidative Stress Pathway	19	32	8.864
Predicting Variabilities in Cardiac Gene	15	38	16.889
Processing of Spz Network from the Drosophila Signaling Pathway	24	28	4.861
Regulation of the L-arabinose operon of Escherichia coli	13	17	10.059
SKBR3 Breast Cell Line Long-term ErbB Network	25	81	12.960
SKBR3 Breast Cell Line Short-term ErbB Network	16	41	16.016
T cell differentiation	23	34	6.427
T-LGL Survival Network 2011 Reduced Network	18	43	13.272
TOL Regulatory Network	24	48	8.333
Toll Pathway of Drosophila Signaling Pathway	11	11	9.091
VEGF Pathway of Drosophila Signaling Pathway	18	18	5.556
Average edge density (%)			**15.401**

Legend: For each biological GRN that we analyzed, we show here the number of edges they contain, the edge density (calculated as the proportion of the number of actual edges with respect to the number of possible edges which is simply $(Number\ of\ Nodes^2)$. The average edge density of around 13% was used as a basis for the choice of the edge density (10%) for the RBNs that we simulated.

Supplemental Table 9. The distribution of the constancy length of RBNs. [This item relates to Figure 8]

N	Median	Mean	95%	99%	Max
10	1	1.709313	4	8	31
15	2	2.553708	7	13	183
20	2	3.056888	8	18	94
25	3	4.872672	13	32	489
30	4	6.44254	18	51	1195
35	8	17.08245	29	167.46	8598
40	10	31.6757	36	397	27908

Legend: This table shows constancy length data obtained from simulations of 1000 RBNs for each N (number of nodes), with each RBN run 1000 times starting from a different initial condition each time. Here, constancy length indicates the maximum number of contiguous steps during which a node preserves its state (0 or 1) in an attractor, taken as the maximum over all nodes. For example, consider a network with N>2 nodes, of which node 'X' goes through the following states in an attractor cycle (period length of 13 steps) in the same order: 0110001001111, and node 'Y' goes through the following states in the same order: 0110000001111. Here, the constancy length of 'X' is 4 and that of 'Y' is 6. The constancy length of the network would be the maximum of the constancy lengths of the individual nodes.

Data S2: See file Violins.zip – plots comparing distribution of memories for GRNs to their randomized configuration models. [This item relates to Figure 9]

Legend: This supplement provides the violin plots of the set of all 35 GRNs (Plot 1-35) from the Cell Collective database (https://cellcollective.org/) compared (in terms of memories) to their configuration models. We show the mean (black line), median (red line), 5th percentile (teal line) and 95th percentile (pink line). The actual frequency of memory of the real GRN is represented as a red star. We calculated the conditional entropy among the different types of memories of GRNs and Configuration models, normalized these conditional entropies, applied Gaussian smoothing and visualized the results obtained.

References Cited

Graudenzi, A., Serra, R., Villani, M., Colacci, A., and Kauffman, S.A. (2011). Robustness analysis of a Boolean model of gene regulatory network with memory. Journal of Computational Biology *18*, 559-577.

Groß, A., Kracher, B., Kraus, J.M., Kühlwein, S.D., Pfister, A.S., Wiese, S., Luckert, K., Pötz, O., Joos, T., and Van Daele, D. (2019). Representing dynamic biological networks with multi-scale probabilistic models. Communications biology *2*, 1-12.

Helikar, T., Kowal, B., McClenathan, S., Bruckner, M., Rowley, T., Madrahimov, A., Wicks, B., Shrestha, M., Limbu, K., and Rogers, J.A. (2012). The cell collective: toward an open and collaborative approach to systems biology. BMC systems biology *6*, 96.

Herrmann, F., Gross, A., Zhou, D., Kestler, H.A., and Kuhl, M. (2012). A boolean model of the cardiac gene regulatory network determining first and second heart field identity. PLoS One *7*, e46798.

Kauffman, S., Peterson, C., Samuelsson, B.r., and Troein, C. (2003). Random Boolean network models and the yeast transcriptional network. PNAS *100*.

Mochizuki, A., Fiedler, B., Kurosawa, G., and Saito, D. (2013). Dynamics and control at feedback vertex sets. II: A faithful monitor to determine the diversity of molecular activities in regulatory networks. Journal of theoretical biology *335*, 130-146.

Naldi, A., Hernandez, C., Levy, N., Stoll, G., Monteiro, P.T., Chaouiya, C., Helikar, T., Zinovyev, A., Calzone, L., and Cohen-Boulakia, S. (2018). The CoLoMoTo interactive notebook: accessible and reproducible computational analyses for qualitative biological networks. Frontiers in physiology *9*, 680.

Serraa, R., d, M.V., Damiania, C., Graudenzia, A., Colaccib, A., and Kauffmanc, S.A. (2007). Interacting Random Boolean Networks. Proceedings of ECCS07: European Conference on Complex Systems.

Shmulevich, I., and Dougherty, E.R. (2010). Probabilistic Boolean networks: the modeling and control of gene regulatory networks (SIAM).

Shmulevich, I., and Kauffman, S.A. (2004). Activities and sensitivities in boolean network models. Phys Rev Lett *93*, 048701.

Veliz-Cuba, A., Aguilar, B., Hinkelmann, F., and Laubenbacher, R. (2014). Steady state analysis of Boolean molecular network models via model reduction and computational algebra. BMC Bioinformatics.

Xiao, Y. (2009). A Tutorial on Analysis and Simulation of Boolean Gene Regulatory Network Models. Current Genomics *10*, 511-525.

Zhai, X., Zhou, W., Fei, G., Liu, W., Xu, Z., Jiao, C., Lu, C., and Hu, G. (2018). Null model and community structure in multiplex networks. Scientific reports *8*, 1-13.