The pattern memory of gene-protein networks

Ronald L. Westra[1], Karl Tuyls[1], Goele Hollanders[2], Geert Jan Bex[2], Marc Gyssens[2]

[1] Department of Mathematics and Computer Science,
Maastricht University and Transnational University of Limburg,
Maastricht, the Netherlands
[2] Department of Mathematics, Physics, and Computer Science,
Hasselt University and Transnational University of Limburg,
Hasselt, Belgium

Abstract. In this paper we study the potential of simple linear gene-protein interaction networks to store sparse input-output patterns. This problem bears similarity to the engineering task of reconstructing gene networks from time series of expression data, such as sets of microarrays. This is currently an intensively studied field where some results are directly applicable to our context. The central question in this study concerns the memory capacity of a network of n genes and proteins, which interact according to a simple linear state space model with p external outputs. Here it is assumed that to a certain combination of inputs $\mathbf{u}^*$ there exists an optimal state of the system $\mathbf{x}^*$, i.e., values of the gene expressions and protein levels, that has been attained externally, e.g., through evolutionary learning. Given such a set of m learned optimal input-output patterns, the design question here is to find a sparse and hierarchical network structure for the gene-protein interaction matrix A, and the gene-input coupling matrix B. This problem is formulated as an optimization problem in a linear programming setting. From this formulation it is directly evident that the maximum number of patterns that can be stored is: $m_{\max} = n + p - 1$. Furthermore, it allows for a numerical analysis, which shows that there are clear scale-invariant phase transitions for the sparsity, i.e., the relative number of zero-elements, in the matrices A and B as the number of patterns m increases. The sparsity in A and B exhibits continuous second-order phase transitions as the number of patterns reaches $m_1 = p/2$ and $m_2 = 2p/3$ respectively. These phase transitions divide the system in three regions with different memory characteristics. In the first region, below $m = m_1$, the system stores patterns by directly connecting the inputs to the outputs, without directly involving the genes and proteins. In the second region, between $m = m_1$ and $m = m_2$, information is preferentially stored in matrix A, and in the third region, above $m = m_2$, there is no clear preference for storing information in either A or B, and their sparsities behave increasingly identical. It is possible to formulate simple scaling rules for the behaviour of the sparsity in A and B versus m, though the exact morphology of these relations is not scale-invariant. Finally, numerical experiments are described that show that the patterns are stable within a certain finite range around the patterns.

1 Introduction

All biological processes result from the interactions of genes and proteins with external stimuli. The variety of these stimuli is immense; they can be either harmful or beneficial,

obligatory or supplementary, lethal or lifesaving. They range from (lack of) illumination, (too low or high) temperature, (shortage of) food, and water, to toxic agents and viral or bacterial infections. The efficiency of an organism directly depends on its ability to optimally react to these external inputs. In lethal situations it even spells its basic survival. On the molecular level, the reaction to external stimuli consists of the chemical and physical interactions of the external agent with the gene-protein reaction network. For instance, daylight can cause a chemical reaction in the photosynthetic receptors of a plant that result in the production of several organic and inorganic molecules. In general, these stimuli can cause a threat to the survival of the organism, or a potential boost to its existence. If a specific external stimulus is presented with some regularity, it is greatly beneficial for the organism to learn and remember its best response. Response here is on the microscopic level; the best settings for the expressions of the genes and relative densities of the proteins to process the presented stimulus. In this view, the couple of the external stimulus and the optimal network response define an input-output pattern, which the gene-protein interaction network has to "store" in its "memory". The more patterns the organism can store with its current network architecture, the better the organism is equipped to live in its natural environment.

In this study it is assumed that the set of best responses to a given input is available. In reality this is not accomplished by a single organism, but acquired in the cause of many generations through the process of evolution: those individuals that perform better by having a better response to a given stimulus than others have a higher "fitness", and therefore a higher selective advantage in survival and generating offspring. Those that fail to do so may perish. In this way, the selection pressures every organism has to face, set hard constraints on the design of the network.

Biological observation shows that natural information processing networks exhibit sparse and hierarchical connectivity, and the so-called small-world property, see: Watts and Strogatz [6]. These features are derived evolutionary properties of the organisms internal organization, as in Nature, individuals are selected only to their adequate and appropriately fast response to a large set of different input patterns. We use these architectural characteristics as constraints when designing network models in this paper.

The exact outcome of the interaction between the stimulus and the gene-protein network is governed by the actual microscopic details of the underlying chemical-physical laws of nature [39]. These laws allow for a set of free parameters that can be tuned to implement a certain relation between the presented input and the desired output. In our context we look to the response of n genes and proteins to p external inputs, such as a toxic agent or a specific odor. The interaction amongst the genes and proteins can be modeled by a $n \times n$ matrix A, such that a_{ij} represents the influence from gene/protein j on gene/protein i. Similarly, the interaction with the external environment is represented by the $n \times p$ input-output matrix B, such that b_{ij} represents the influence from input j on gene/protein i. The aforementioned free system parameters therefore consist of the system matrices A and B. Tuning these free parameters means modifying the input-output relation of the system. Thus, the learning of the input-output patterns can be obtained by an optimization in the space of these free parameters. In practice, there are tens of thousands of genes and hundreds of thousands—if not millions—of (possible) protein/RNA/inorganic species. Besides the fact that many of these genes and

most of these protein/RNA/inorganic molecules are currently unknown, and besides the randomizing effect of intrinsic and extrinsic noise, purely from the sheer magnitude of the data it would be completely impossible to implement realistic chemical-physical models and perform computational simulations.

For this reason the study in this paper is based on a straightforward and computational manageable model; a linear and continuous time-independent state space model. This model segregates the interaction between the genes and proteins itself from their response to external outputs. With this simple model it is possible to study the qualitative and quantitative aspects of efficient network memory through extensive numerical experiments. As this network can be considered as a first-order approximation, it is possible to extrapolate the results to more realistic, hence complex, interaction network models.

It is difficult to quantify the exact number of patterns an organism with a given number of genes and proteins can store in order to survive in its habitat. Some clues are provided by the "Minimal Genome Project" [36], that endeavors to find the smallest working set of genes necessary for an organism to live and reproduce successfully. The smallest genome size found thus far is for the parasitic archaebacterium Nanoarchaeum equitans with only 400 nm in diameter, and a genome of 460 Kbp organized in 487 ORFs. Even smaller is the SARS Corona Virus, a retro virus of 100 nm and a genome of 30 Kbp organized in just 5 ORFs. We can compare this to the well known eubacterium Escherichia coli (diameter 2000nm) with a genome of 4639 Kbp organized in 4377 DNA-genes. The regulated environment in a host in the lifestyle of N. equitans and SARS, provides less diversity in inputs patterns, and therefore requires less pattern memory from the organism. Besides this qualitative argument, in the last Section we will use the formalism developed here to provide some rough estimates for the number of stored patterns.

In the context of the linear time-invariant state space systems as studied in this paper, the problem of storing input-output patterns results in an optimization process on the system matrices A and B for mapping the given input to the given output. The derived design criteria as sparsity and hierarchy are directly used as additional constraints in this optimization. We note that this topic is related to two well-studied areas of research: artificial neural networks and "inverse engineering": gene network reconstruction from time series data. Firstly, the problem of designing a proper network to fit a given set of input-output patterns is of course identical to finding the best network parameters for a given time series of synchronous measurements of inputs and outputs. The difference is that the wealth of allowed architectures consistent with a limited set of data is beneficial for an organism in the natural world, and detrimental for the biologist who wants to determine the genetic pathway from a small number of microarray measurements. There is an extensive literature on gene network reconstruction from time series of gene expressions [17, **?**,15]. Especially relevant to our approach is the work of Mestl, Plahte and Omholt [22, 25, 26, 28], Somogyi [29], Yeung and Tegner [43, 35], and Peeters and Westra [27, 40, 42]. For a thorough overview consult [16, 41] and the references therein.

Secondly, the problem of storing patterns in a given network architecture and the required memory size is an intensively studied subject in the area of artificial neural

networks with an extensive literature, e.g., see [18–21, 31], and standard books as [2, 5, 37].

Finally, also the dynamic behavior of gene-protein networks is an important and fertile field of research, see for instance A. Goldbeter [12, 14], Elowitz and Swain [7, 33, 32], and others [34, 24]. Both the practical studies from Elowitz and Swain, and the theoretical work of Goldbeter, Tyson and Novak, and Steuer, show that organisms are in a dynamical equilibrium. In embryonic growth, for instance, these equilibrium states correspond to the so-called 'checkpoints'. The system dynamics can be seen as perturbations around the momentary equilibrium. These equilibria act as local attractors to the system dynamics: the system state vector orbits the local equilibrium in a periodic cycle or in a fractal orbit (a "strange attractor"). An external stimulus can disrupt this equilibrium, after which the system can come in another part of the phase space – i.e. the space of all possible gene expressions and protein densities. This is modeled in the so-called Piecewise Linear Models, which study the dynamics between state changes of the system, see: R.L Westra et al. [42].

In the following Section we first define the linear time-invariant continuous state space model and its relation to a gene-protein interaction network. In Section 3, we describe pattern storage of dynamical patterns and its relation to network reconstruction. In Section 4 the storage of static patterns is discussed. Next, in Section 5, we discuss the results of some numerical experiments, and relate these to practical observations regarding the underlying geometry of the problem. In Section 6 we focus on phase transitions in the network. In Section 7 we briefly discuss the dynamics and stability of the stored patterns. We close with a discussion of the obtained results.

2 Modelling dynamic gene-protein networks as linear state space systems

In this study we will concentrate on systems sufficiently near their equilibrium. A more realistic system will exhibit multiple of such equilibria, and given the proper stimulus will switch from one equilibrium to another, and therewith transform the system from one biological state to another. Near a suitable smooth equilibrium the activity of the n genes and proteins, represented by the n-vector $\mathbf{x}$, will develop according to the $n \times n$ gene-protein interaction matrix A and the $n \times p$ interaction matrix B with the p external inputs, defined by the p-vector $\mathbf{u}$, as :

$$\dot{\mathbf{x}} = A\mathbf{x} + B\mathbf{u} + \nu\xi(t) \tag{1}$$

Where $\xi(t)$ represents white Gaussian noise with mean zero and standard deviation 1. Such simple linear models allow for a limited set of behaviors, but here they fully suit our purpose as a simple and transparent metaphor for studying the number m of patterns that the system can store as a function of the number of genes/proteins n, the number of external inputs p, and the sparsity in the interaction matrices A and B.

3 Storage of patterns of time series and its relation to network reconstruction

There is an essential similarity between reconstructing the best fitting matrices A and B from empirical data sets (state vectors X and inputs U), and the task of designing the optimal interaction matrices A and B from optimal patterns X for a given input U. This similarity allows us to utilize the extensive literature on this subject to this field. Therefore, we can be brief on this subject. Here, we follow Peeters and Westra [27].

In this context we define a *pattern* as a time series that represents the desired dynamical behavior of the state vector $\mathbf{x}(t)$ in response to a given input vector $\mathbf{u}^*(t)$, such as a finite impulse or a harmonic signal. In this way, the desired system outputs $\mathbf{x}^*(t)$ and $\dot{\mathbf{x}}^*(t)$ for a given input $\mathbf{u}^*(t)$ can be compared with the observed outputs $\mathbf{x}(t)$ and $\dot{\mathbf{x}}(t)$. Suppose we have sets of m desired input-output triples as $\{(\mathbf{u}_l^*, \mathbf{x}_l^*, \dot{\mathbf{x}}_l^*) \mid l = 1 \dots m\}$, which can be used to define three data matrices: $X = (\mathbf{x}_1^* \dots \mathbf{x}_m^*)$, $\dot{X} = (\dot{\mathbf{x}}_1^* \dots \dot{\mathbf{x}}_m^*)$, and $U = (\mathbf{u}_1^* \dots \mathbf{u}_m^*)$. In this setting, one could attempt to minimize the error $\|\mathbf{x}^*(t) - \mathbf{x}(t)\|$ as function of the free parameters in A and B. Or, alternatively, minimize the error in the time derivatives: $\|\dot{\mathbf{x}}^*(t) - \dot{\mathbf{x}}(t)\|_k = \|\dot{\mathbf{x}}(t) - A\mathbf{x}(t) - B\mathbf{u}(t)\|_k$. Where $\|\cdot\|_k$ represents a suitable L_k-norm. This error expresses the difference between the observed rate of change $\dot{\mathbf{x}}(t)$ and predicted model output $A\mathbf{x}(t) + B\mathbf{u}(t)$.

Given X, $\dot{X}$, and U one could compute the matrices A and B that minimize this error function. This optimization can be formulated as a least squares problem—i.e., setting $k = 2$, and given sufficient data X, $\dot{X}$, and U, in general a solution can be obtained. In case that the number of patterns m is much less than the number of genes/proteins n and inputs p the problem is under-determined. In that case, it is useful to involve the biological constraint that A and B are sparse and hierarchical matrices. In that case, the maximization is performed over the number of zeros in A and B while relation (1) is considered as a constraint:

$$\max_{A,B} \|A\|_0 + \|B\|_0 \text{ subject to: } \dot{X} = AX + BU \tag{2}$$

As indicated above, this optimization problem is exactly equal to the problem of reconstructing the underlying interaction network from a time series of (e.g., micro array) measurements.

This problem is well described in a number of publications, e.g., [43, 27], and Westra et al. [40]. For a relevant survey see Westra et al.[42] and the references therein. We here merely note that this problem exhibits a first-order phase transition as evident from Figure 1.

Let N_e indicate the number of errors found in the reconstruction of matrices A and B, i.e. the number of components of the matrices where the reconstructed value of the matrix differs from its correct value above a given threshold, e.g. the machine precision. In our context this indicates the number of pattern storage errors, where the system found a sub-optimal solution. This number of errors N_e suddenly drops to zero as the number of patterns m passes through a critical value M_C. Similarly, N_e suddenly increases linearly from strictly zero as the number of non-zero elements in A (and likewise B) passes through a critical value K_C.

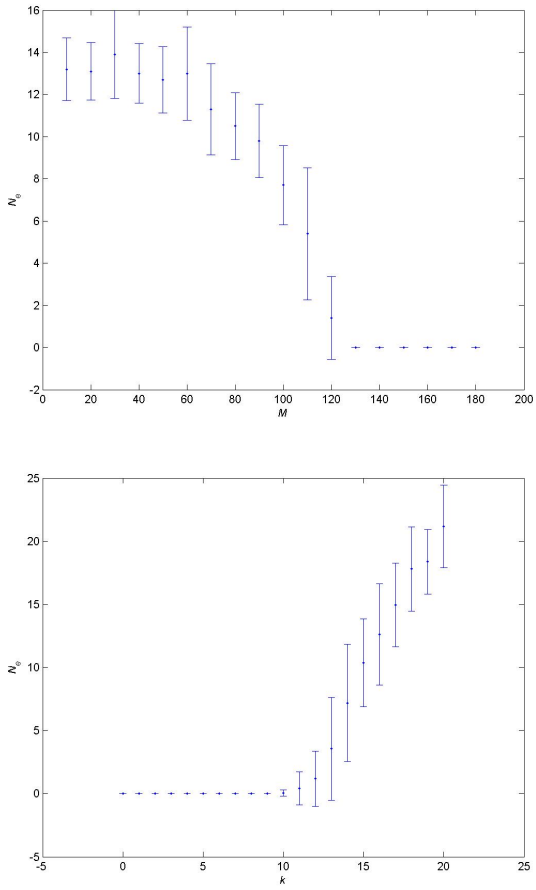

Fig. 1. Above: the number of memory storage errors N_e versus the number m of stored patterns. Below: number of memory storage errors N_e versus the sparsity (i.e. number of non-zero connections) of the gene-interaction network.

4 Memory storage of equilibrium state patterns in linear networks

4.1 Linear state space formulation

The problem studied in this Section concerns the storage of static input-output patterns in a gene-protein network represented by a linear time-invariant dynamical system of the form as in equation 1 with matrices A and B, which exhibit biological relevant properties such as sparsity, hierarchy, high local clustering, and the "small-world" property. In the static context, we assume that the system has converged to an equilibrium state $\mathbf{x}^*$ and that $\dot{\mathbf{x}} = 0$. This means that to a given input $\mathbf{u}^*$ there belongs an optimal value $\mathbf{x}^*$ for the gene expressions and protein densities in the network such that $A\mathbf{x}^* + B\mathbf{u}^* = \mathbf{0}$. In

our context a (static) input-output pattern is defined as such an input-output pair $(\mathbf{u}^*, \mathbf{x}^*)$. This pattern itself can be considered as the outcome of a long evolutionary learning process, where the best combination of gene/protein activations $\mathbf{x}$ are selected relative to the faced combination of external inputs (e.g. toxic agents, virus infections) $\mathbf{u}$.

Here we are interested in the maximum number m_{max} of linearly independent patterns that can be stored in a network of n genes or proteins, and p external inputs. Suppose that there exists a set of m linearly independent patterns, i.e. external inputs as a $p \times m$ matrix $U = \{\mathbf{u}[1], \ldots, \mathbf{u}[m]\}$, and the associated (evolutionary) learned states as a $n \times m$ matrix $X = \{\mathbf{x}[1], \ldots, \mathbf{x}[m]\}$. For each input-pattern couple $\{\mathbf{x}[j], \mathbf{u}[j]\}$ equation 1 implies that: $A\mathbf{x}[j] + B\mathbf{u}[j] = \mathbf{0}$. For the entire set of patterns X and U this means that the unknown matrices A and B feature in a linear way in the resulting matrix equation: $AX + BU = 0_{mn}$. This equation can be rewritten as: $\begin{pmatrix} X^T & U^T \end{pmatrix} \begin{pmatrix} A^T \\ B^T \end{pmatrix} = O_{mn}$.

This matrix equation can be treated in a row-by-row fashion. Denoting the (unknown)i-th row of A by the row vector α_i^T, the (unknown) i-th row of B by the row vector β_i^T, this yields the following decoupled set of N linear systems of equations of size $m \times (n + p)$:

$$\begin{pmatrix} X^T & U^T \end{pmatrix} \begin{pmatrix} \alpha_i^T \\ \beta_i^T \end{pmatrix} = 0, \qquad (i = 1, 2, \ldots, M). \tag{3}$$

With abuse of notation define the $m \times (n + p)$ matrix $D = \begin{pmatrix} X^T & U^T \end{pmatrix}$, and the $(n + p)$ vector: $x = \begin{pmatrix} \alpha_i^T \\ \beta_i^T \end{pmatrix}$, this can be written as:

$$D\mathbf{x} = \mathbf{0} \tag{4}$$

The objective now is to find sparse A and B that satisfy this condition. However, there are two practical problems with this formulation. First, as this equation holds for each combined row $\mathbf{x}$ of A and B, its solution also holds for all rows, and therefore A and B consist of dependent columns in the form: $A_{i,j} = v_i.x_j$ for some non-zero vector $\mathbf{v}$. Second and more seriously, the sparse matrices $A = B = 0$ pose a trivial but valid solution to equation 4.

These problems can be solved by introducing additional constraints that express biological knowledge, such as the known influence of a certain agent on a certain gene/protein, Autocatalyzation or natural degradation of proteins. The latter reads that each protein autocatalyses or decays with a specific rate $\lambda > 0$, and therefore the associated component of the gene/protein interaction matrix A has a value: $\pm\lambda$. The known non-zero influence of an input j on a certain gene/protein i indicates that the $(i, j)-$component of matrix B is non-zero. Thus, in general, certain specified components of A and B have non-zero values.

Moreover, there is a scaling invariance in the solution – if $\mathbf{x}$ is a solution of $D\mathbf{x} = \mathbf{0}$, then $\lambda\mathbf{x}$ is also a valid solution for all $\lambda \in R$. The exclusion of $A = B = 0$ can thus be obtained by fixing the B-part of $\mathbf{x}$ to a specific value, and thus express the value of A relative to B. This can be obtained by using constraints like $\|B\|_1 = constant$, or $\sum_{ij} b_{ij} = constant$. In this study, this constraint is realized as:

$$\mathbf{q}^T\mathbf{x} = 1 \tag{5}$$

where the first n components (the "A-part") of $\mathbf{q}$ are zero, and the remaining p components q_i, $(i = n+1..n+p)$, (the "B-part") satisfy: $|q_i| \in [\frac{1}{2}, 2]$. This vector was generated with a uniform distribution on $[-2, -\frac{1}{2}] \cup [\frac{1}{2}, 2]$

4.2 Robust estimation as an approximation to the L_0-norm of A

The problem of computing a sparse solution to a consistent underdetermined linear system of equations $Ax = b$ has been studied extensively in literature. J.J. Fuchs in [9] and [10], and some of the references therein, described state conditions under which optimal sparse solutions can be obtained by the technique of L_1-minimization. This technique aims to compute a vector $\mathbf{x}$ of minimal L_1-norm within the solution space S of the linear system $A\mathbf{x} = \mathbf{b}$. It is well known that this problem can be reformulated as a linear programming (LP) problem. Thus, the difficult combinatorial problem of finding a vector in S having as many zero entries as possible is avoided and replaced by the much easier to solve problem of finding a vector in S for which the L_1-norm $\|x\|_1$ is minimal. In our case, we can thus reformulate the original quest for a sparse vector $\mathbf{x}$ (associated to corresponding rows in A and B) to the problem of finding the minimum value of the L1-norm of x, i.e. $\|x\|_1$, under the given constraints. As we may weigh the importance of sparsity in A and B different, we introduce a regularization parameter ε. Furthermore, let $\mathbf{y}_A$ denote the first n components of $\mathbf{x}$ (the "A-part"), and $\mathbf{y}_B$ the other p components (the "B-part"). With all these definitions, the regularized minimum value of the L_1-norm of $\mathbf{x}$ for the given data of m patterns for n genes/proteins, and p external inputs follows from the following linear programming formulation:

$$
\begin{aligned}
LP1: \quad & \mathbf{x}^* = \arg\min_{x \in R^{n+p}} \|\mathbf{y}_A\|_1 + \varepsilon \|\mathbf{y}_B\|_1 \\
& \text{subject to:} \\
& \quad \mathbf{x} = \mathbf{y}_A + \mathbf{y}_B \\
& \quad D\mathbf{x} = \mathbf{0} \\
& \quad \mathbf{q}^T \mathbf{x} = 1 \\
& \quad x_i = r_i, i = 1, \ldots, C
\end{aligned}
\tag{6}
$$

where r_i represents a random nonzero-constant.

This defines LP1 as a regularized L_1-minimization problem over the affine subspace of vectors $\mathbf{x}$ satisfying the given constraints. This is a primal LP-problem but not in the standard form. This LP problem can be reformulated to its dual LP formulation which is computationally more efficient.

5 Numerical experiments for storing equilibrium state patterns

5.1 Scale-free transitions in the relation between the model sparsity and the number of patterns

The formulation in equation 6 allows for numerical experiments with different settings of the model parameters X, U, m, n, and p. The value of the number of patterns m varies between 0 and $n + p - 1 - c$, as will be shown later in this Section. Here c is the number of extra constraints in LP1. The experiments were performed with a specific interest in

partial sparse solutions for A and B. Here we introduce k_A as the number of non-zero elements in matrix A, and similarly k_B for B. Figure 2 shows two results from such numerical experiments. The patterns X and U were drawn uniform randomly from the interval [-1,1]. As this Figure clearly shows, it was found that – for fixed regularization parameter $\varepsilon = 1$ – the sparsity in A and B exhibits different behavior as the number of patterns crossed the value $m_{C1} = p/2$. Thus, a scale-free representation for studying the sparsity is obtained by defining:

$$\begin{aligned} &* \text{ the relative sparsity in } A : p_A = k_A/n; \\ &* \text{ the relative sparsity in } B : p_B = k_B/p; \\ &* \text{ the scaled number of patterns: } \mu : \\ &\quad \text{if: } m < m_{C1} \text{ then: } \mu = m/m_{C1} \\ &\quad \text{if: } m \geq m_{C1} \text{ then: } \mu = (m + n - 1)/(2n + p - 2) \end{aligned} \tag{7}$$

This scaling ensures that μ, p_A, and p_B are between 0 and 1. The scaled variables are useful in comparing situations with different genes/proteins n and inputs p. Other values of regularization parameter ε gave quantitative different results that, however, could again be scaled such that they were qualitatively similar.

In these plots the sparsity in the matrix is plotted versus the scaled number of patterns μ. The lower curve gives the relative non-zeros p_A in matrix A. This exhibits a clear transition at $\mu = 1/2$. The upper curve gives the relative non-zeros in matrix B. In the scaled representation these plots are almost identical for different values of p and n, though the morphology somewhat depends on $n + p$.

5.2 Considerations from the underlying geometry

The observed relations between the parameters k_A, k_B, m, n, and p can be understood from an analysis of the underlying geometric constraints of the optimization process in R^{n+p}, for an overview of the employed matrix algebra consult [13]. The solution $\mathbf{x} \in R^{n+p}$ to the problem lies on the intersection between the two major constraints of the problem. The first constraint is: $D\mathbf{x} = 0$, where D is the data matrix $\left(X^T\ U^T\right)$, and the first n components of $\mathbf{x}$ correspond to a row of matrix A, and the last p components to the corresponding row of matrix B. Thus, the solution $\mathbf{x}$lies in the $(n + p - m)$-dimensional linear null-subspace V_1 of D. Second, the solution $\mathbf{x}$ is bound by the $(n + p - 1)$-dimensional affine space V_2 defined by $\mathbf{q}^T\mathbf{x} = 1$, where the first n components of $\mathbf{q}$ are zero, and the p remaining components $q_i, (i = n + 1, .., n + p)$, satisfy: $|q_i| \in [\frac{1}{2}, 2]$. The intersection $V_3 = V_1 \cap V_2$ contains the solution $\mathbf{x}$ and, except for special cases of Lebesque measure zero when V_1 and V_2 are parallel, V_3 is a $(n + p - m - 1)$-dimensional affine space. Therefore, the short answer to the maximum number m_{max} of patterns that can be stored is: $m_{max} = n + p - 1$. Adding $(c - 1)$ extra constraints of the form: $x_i = constant$ effectively lower the dimension, but do not alter the basic topology, and leads to:

$$m_{max} = n + p - c \tag{8}$$

The situation becomes more complex when we analyze the number k of exact zeros in $\mathbf{x}$, as the optimization is aimed at increasing this amount. k_A denotes the number

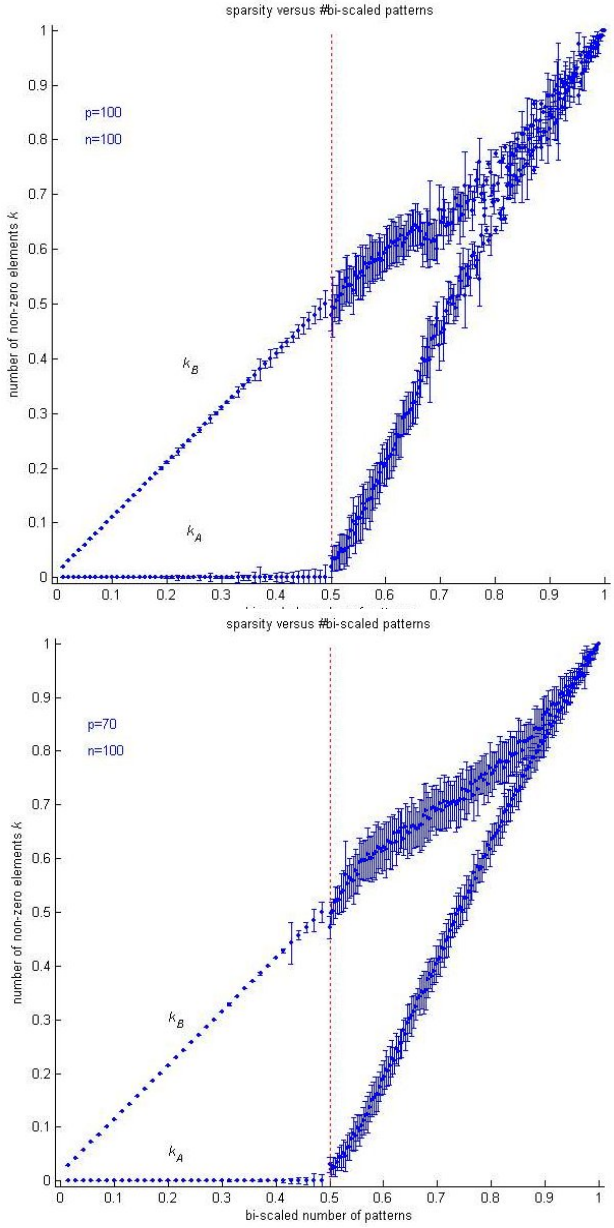

Fig. 2. Plot of the scaled matrix sparsity versus the scaled number of patterns. The solutions of the LP-approach to memory storage in linear networks, exhibit a clear phase transition at $p/2$ patterns. Above a plot for n=100 and p=100, below a plot for n=70 and p=100.

of non-zeros in the first n components of $\mathbf{x}$, and k_B in its last p components. In this formalism the maximum sparsity in A is obtained when $k_A = 0$. Here, we step aside the biological complication that this represents a situation without gene-protein-couplings, as the system output is entirely defined by the couplings to the external inputs. For the original equation, $\begin{pmatrix} X^T & U^T \end{pmatrix}\begin{pmatrix} \alpha_i^T \\ \beta_i^T \end{pmatrix} = \mathbf{0}$, the condition $k_A = 0$ means that the associated row of A is zero: $\alpha_i = 0_{1,n}$, therefore: $U^T\beta_i^T = \mathbf{0}$, meaning that β_i^T lies in the null-space of U^T. The condition $k_A = 0$ implies that: $x_1 = x_2 = \ldots = x_n = 0$, which defines a p-dimensional linear subspace V_4 in R^{n+p}. The intersection $V_3 \cap V_4$ contains all fully sparse A-solutions that satisfy the two constraints. So, the condition $k_A = 0$ can be reached whilst $V_3 \cap V_4 \neq \emptyset$. $V_3 \cap V_4$ is defined by $(m + 1 + n)$ equations with $(n + p)$ variables that imply an upper bound m_A^{up} for $k_A = 0$, namely: $m_A^{up} = p - 1$. However, as apparent from Figure 3, k_A already abruptly starts deviating from zero at $m = m_C = p/2 < m_A^{up}$. Though in that case $V_3 \cap V_4$ is not-empty, the L_1-optimization process favors another solution with a *smaller* value of the L_1-norm of matrix A – but an unsolicited *larger* value for the number of non-zeros k_A. Finding a solution with $k_A = 0$ results in solving the linear equation: $Z\mathbf{x} = \mathbf{z}$, with:

$$Z = \begin{pmatrix} D \\ \mathbf{q}^T \\ I_n \;\; 0_{n,p} \end{pmatrix}, \mathbf{z} = \begin{pmatrix} 0_{m,1} \\ 1 \\ 0_{n,1} \end{pmatrix} \tag{9}$$

This system has a consistent solution when $\mathbf{z}$ is a vector in the column space of Z. If this is not the case, the affine solution space S can be described implicitly as the solution of the consistent linear system of equations $Z\mathbf{x} = \mathbf{z}_{proj}$, where the vector $\mathbf{z}_{proj}$ denotes the orthogonal projection of $\mathbf{z}$ onto the column space of Z. An explicit description of S follows from computing the solution space of the system $Z\mathbf{x} = \mathbf{z}_{proj}$, which can be achieved in a numerically reliable way by employing SVD or QR-decomposition. Thus, S can be formally written as: $\hat{\mathbf{x}} = N(Z)\xi + Z^+\mathbf{z}$, where $N(Z)$ is the $n \times \dim(\text{null}(Z))$ matrix whose columns consist of the of the basis of the $\dim(\text{null}(Z))$-dimensional null-space of Z, and Z^+ is the pseudo-inverse of Z. ξ is a $\dim(\text{null}(Z))$-dimensional vector that parameterizes S. The support vector $Z^+\mathbf{z}$ may indeed have a lower k_A than the solution $\mathbf{x}^*$ of the LP, as visible in Figure 3. This is an artifact of the L_1-minimization process that can favor a smaller value of the absolute value of a non-zero component rather than a smaller value of k_A. Hence, in general: $criterion(\mathbf{x}^*) = \|\alpha^*\|_1 + \varepsilon\|\beta^*\|_1 < \|\hat{\alpha}\|_1 + \varepsilon\|\hat{\beta}\|_1 = criterion(\hat{\mathbf{x}})$.

In the general case, when $k_A > 0$, the subset in V_3 that contains exactly k_A non-zeros in the first n and k_B non-zeros in the last p coordinates of $\mathbf{x}$, is defined by V_3 , the $(n + p - k_A)$-dimensional linear subspace with k_A non-zeros in the first n coordinates of $\mathbf{x}$, and the $(n + p - k_B)$-dimensional linear subspace with k_B non-zeros in the last p coordinates of $\mathbf{x}$. There are respectively $\binom{n}{n - k_A}$ and $\binom{p}{p - k_B}$ possible combinations of the form $x_i = 0$ to realize such equations. This defines $(m + 1 + n - k_A + p - k_B)$ equations with $(n + p)$ variables that, thus, in general this has a solution if:

$$k_A + k_B \geq m + 1 \tag{10}$$

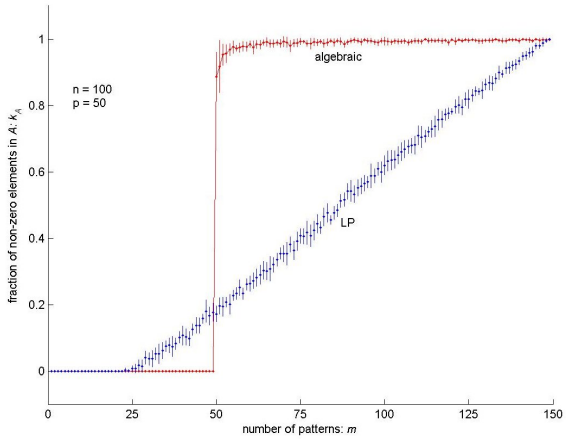

Fig. 3. Comparison of LP-solution (tilted line) and the algebraic solution (step function). m varies between 1 and n+p-1 = 149. At m = p - 1 = 49 the algebraic solution suddenly jumps from 100% to 0% sparsity of A, while the LP-solution exhibits a 1st order phase transition at m = p/2 = 25.

This relation is indeed found empirically as shown in Figure 4 where k_B/m is plot against k_A/m. The location of the critical value of the phase transition $m_c/p = 1/2$ is a function of the regularization parameter ε.

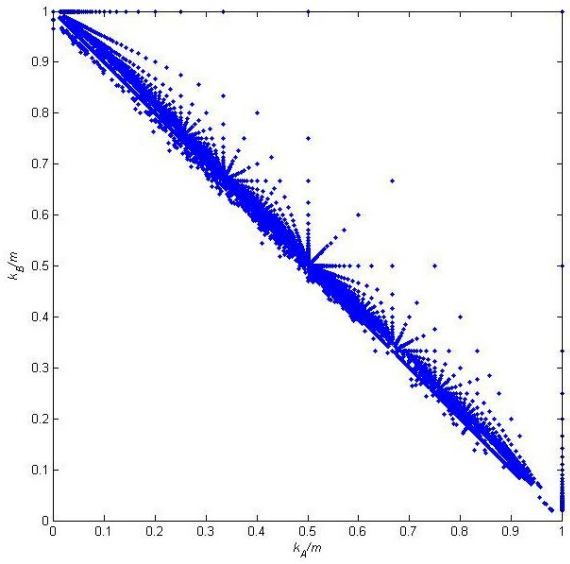

Fig. 4. The plot of k_B/m versus k_A/m exhibits the relation: $k_B/m + k_A/m = 1$.

6 Phase transitions in the matrix sparsity define distinct learning strategies

Using the scale-free definitions 7, different experimental settings can be straightforwardly compared. In Figure 6 the smoothed plot for the relative sparsities $p_A = k_A/n$ and $p_B = k_B/p$ versus the scaled number of patterns μ is depicted for $n = 100$ genes/proteins, and $p = 100$ inputs, averaged over 10116 measurements. Again, all results are shown for fixed regularization parameter $\varepsilon = 1$. This plot exhibits the typical characteristics representative for all the networks in our context. Firstly, the behavior of p_A for $\mu_{C1} = 0.50 \pm 0.01 \approx 1/2$ changes abruptly from constant zero to a near linear increase. Secondly, the characteristic plateau visible in p_B for $\mu_{C2} = 0.67 \pm 0.05 \approx 2/3$ is not an artefact due to noisy measurements or processing deficiencies, but is present in all such graphs. Next, inspection of Figure 6 and numerical analysis show that both graphs converge to the line $p_{A,B} = (1.51 \pm 0.14)\mu - (0.51 \pm 0.13) \approx (3\mu - 1)/2$. Finally, for $\mu > 1$ the system of (independent) inequalities becomes inconsistent and therefore has no solution.

This graph therefore shows three different types of storing behavior separated by phase transitions. The presence of phase transitions in the storage of information in linear random sparse networks is reminiscent of the situation for the propagation of information through these networks.

It is well known that sparse networks under certain conditions exhibit phase transitions. Starting from a sparse regular network, the gradual addition of random links reduces the direct path between any pair of vertices in the network from being very long to being very short. This change is achieved abruptly as a phase transition to a "small-world" network, characterized by short overall path lengths, small overall connectivity, high information processing time, and high local clustering, see: Watts and Strogatz [6], Barabasi et al. [1], Newman [23], and Schäfer [30]. This phase transition occurs at a critical threshold p_C for the probability of rewiring a given connection in the network. The system then changes abruptly from slow to fast information processing. The numbers k_A and k_B of non-zero elements in our random networks can be related to this rewiring probability, as the probability that a connection between two genes/proteins exists equals $p_A = k_A/n$, and the probability that a connection between a gene/protein and a certain input exists equals $p_B = k_B/p$. There are combinatorial many possibilities to realize this condition, that peak for $k_A = n/2$, $k_B = p/2$.

6.1 Entropy

The wiring-probabilities p_A and p_B quantify the information that is stored in the system matrices A and B. A convenient measure for this information is the information entropy S. The entropy S_{tot} measures the total number Ω_{AB} of networks $\{A, B\}$ consistent with k_A non-zeros in A and k_B non-zeros in B as: $S_{tot} = \ln \Omega_{AB}$. The entropy therefore relates to the probability of finding the specific network $\{A, B\}$ by pure chance alone: a *high* entropy S means a *low* probability for finding this specific network, that therefore represents a *high* amount of information. In the context of the network rewiring probability p, the entropy is expressed as:

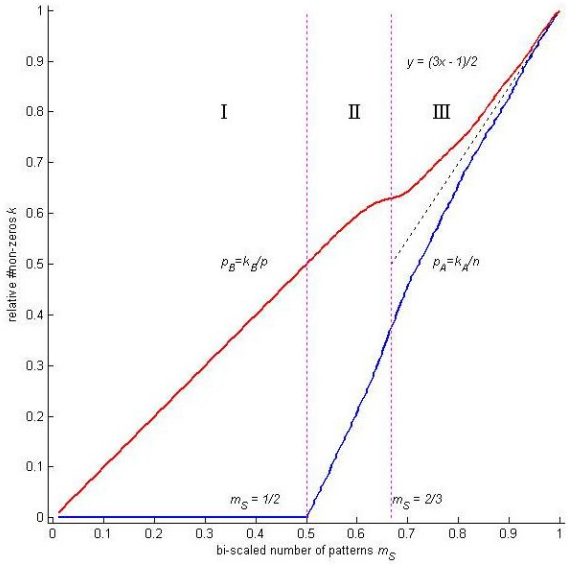

Fig. 5. The connectivity probabilities $p_A = k_A/n$ and $p_B = k_B/p$ for $n = 100$, $p = 100$, and $C = 0$, averaged over 10116 measurements, versus the scaled number of patterns μ. This indicates three major regions I, II, and III, where the network memory behaves different. The transitions occur for the scaled $\mu = 1/2$ for the transition I to II, and $\mu \approx 2/3$ for II to III. In the final phase of region III both probabilities converge to $p = (3\mu - 1)/2$.

$$S(p) = -p \ln p - (1 - p) \ln(1 - p) \tag{11}$$

For network $\{A, B\}$ therefore: $S_{tot} = S(p_A) + S(p_B)$. Relevant to understanding the maximum memory capacity of a network $\{A, B\}$ is the gain of information when *one* pattern is add to the network. This is expressed by the entropy increase $\partial S/\partial m$:

$$\frac{\partial S}{\partial m} = -\left(\frac{\partial p}{\partial m}\right) \ln\left(\frac{p}{1 - p}\right) \tag{12}$$

This quantity depends on the rate of change of the connection probabilities $p_{A,B}(m)$ as function of the number of patterns m stored in the network, and their fluxes $\partial p_A/\partial m$ and $\partial p_B/\partial m$ – all relating to Figure 6. In Figure 6.1 these quantities are shown for $n = 100$, $p = 30$, based on 1180 observations. The total entropy S_{tot} increases moderately for small m until a sudden change at $\mu = 1/2$, subsequently exhibits an optimum around $\mu = 2/3$, and finally decreases monotonously to zero. Consequently, the entropy increase $\partial S/\partial m$ increases slightly for small μ, peaks at $\mu = 1/2$, and then slowly but constantly decreases while crossing zero at $\mu \approx 2/3$. It is remarkable that these changes happen around the same values for μ as those evident in Figure 5. With the similarities in these Figures it is thus possible to classify the network pattern memory

in three distinctive regions separated by phase transitions. Each region is characterized by its typical learning strategy.

Region I is defined as $m < p/2$, so for a relative small numbers of patterns. In this region all information is stored exclusively in the input-output matrix B and none in A. Thus, the system has 'solved' the learning problem by directly wiring the input to the output. Consequently, there are no direct interactions required between the genes and proteins to act to this stimulus. In this region the relations of k_A and k_B to m are essentially linear.

At the first critical point $\mu_{C1} = 1/2$, there is an explicit second order continuous phase transition to region II, clearly visible in the behavior of k_A and k_B. Here, it is evident that the derivatives of k_A and k_B to μ are discontinuous. In region II the information is stored in both matrices, however increasingly in matrix A as μ increases. Matrix B levels off to a plateau around $\mu = 2/3$. In this region, the relations of k_A and k_B to m are essentially not linear, as most notably in k_B.

With increasing m, there is an order transition to region III at the second critical point $\mu_{C2} = 2/3$, that is best visible in the plots of k_B and $\partial S/\partial m$ versus μ. This transition is located at the plateau in the plot of k_B, and at the zero-intersection in the plot of $\partial S/\partial m$. In region III there is no clear preference for storing information in either A or B as matrix B moves away from its plateau. Both matrices are now almost entirely full - i.e. without any zeros. In this region the plots of p_A and p_B versus μ become increasingly identical, so: $\partial p_B/\partial p_A \rightarrow 1$, as evident in Figure 6. It is difficult to determine whether the derivatives $\partial k_A/\partial m$ and $\partial k_B/\partial m$ are truly discontinuous at this transition, so to determine whether this is a true second order phase transition. This is because of the effect of finite size that tends to make all relations appear continuous.

Finally, the maximum memory storage is reached at $\mu = 1$, or $m = n + p - c$. At this instant the system of inequalities becomes inconsistent and there are no solutions to this over-determined system. In this sense, this could be named the fourth region, where the organism can not learn new patterns, and is therefore forced to extend its networks A and/or B by creating new genes and/or proteins that respond to the new input pattern.

The transitions between these regions can in each case be interpreted as an order-disorder transition. As in region I all information is stored in B, this represents the highest 'order'. In region II the information is preferably stored in A. In region III there is no clear preference for A or B, so this represents the highest 'disorder'.

A typical second order phase transition exhibits a well-defined peak in the fluctuation of the energy near the critical point. The most obvious candidate for energy in our context is the criterion function in LP1: $E_1 = \|A\|_1 + \varepsilon\|B\|_1$. This quantity behaves very flat over the scope of m, only exponentially growing near the end, i.e. as μ approaches 1. It does not show any peaks related to the phase transitions. But as we originally are interested in the number of zeros in these matrices, the function $E_0 = \|A\|_0 + \varepsilon\|B\|_0 = k_A + \varepsilon k_B$ seems like a good approximation, where $\|.\|_0$ denotes the number of exact zeros. As visible in Figure 6.1b, this quantity indeed shows a strong and broad peak in its fluctuation near $\mu = 1/2$ that relates to the main phase transition from area I to II. It also exhibits a very strong but also very narrow peak at the transition II/III that is significant above the noise level. However, it also exhibits other strong and broad peaks, such as near

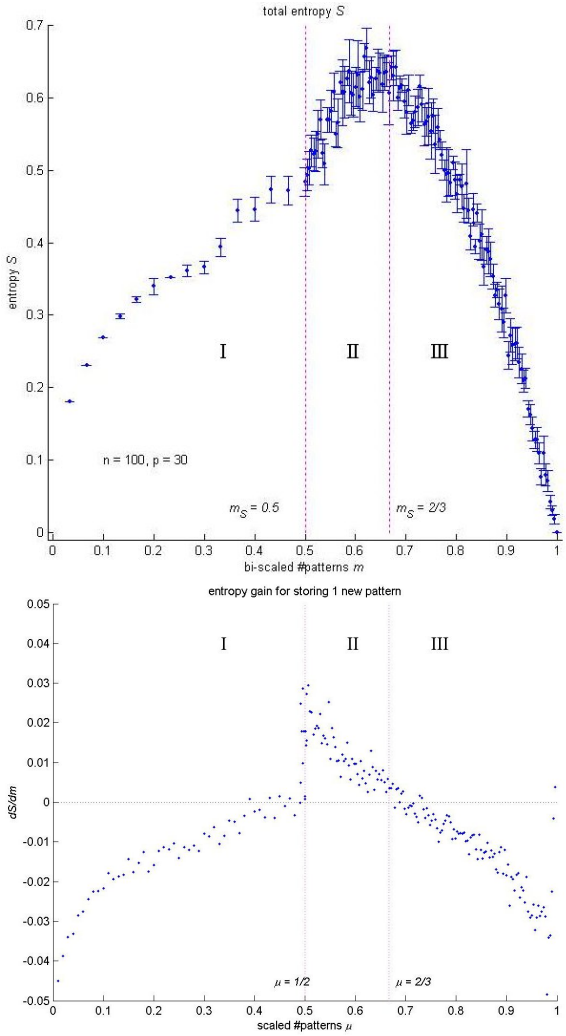

Fig. 6. Above: the entropy S versus the number of patterns m for $n = 100$, $p = 30$, $C = 0$ based on 1180 observations Below: the gain in entropy for the same data set.

$\mu \approx 0.8$ and $\mu \approx 0.9$. Therefore, it is not clear which entity serves as energy in these phase transitions.

Figure 6.1a shows that for very small p_A the fraction $\partial p_B/\partial p_A$ becomes infinite. This represents the situation in region I where p_A is fixed to (almost) zero. For values near $p_A = 1$ the fraction $\partial p_B/\partial p_A$ equals 1. This represents region III, so here $p_B \approx p_A$. The central plateau in this graph fits very well with $p_B \approx (p_B + 1)/2$, and this is representative for region II.

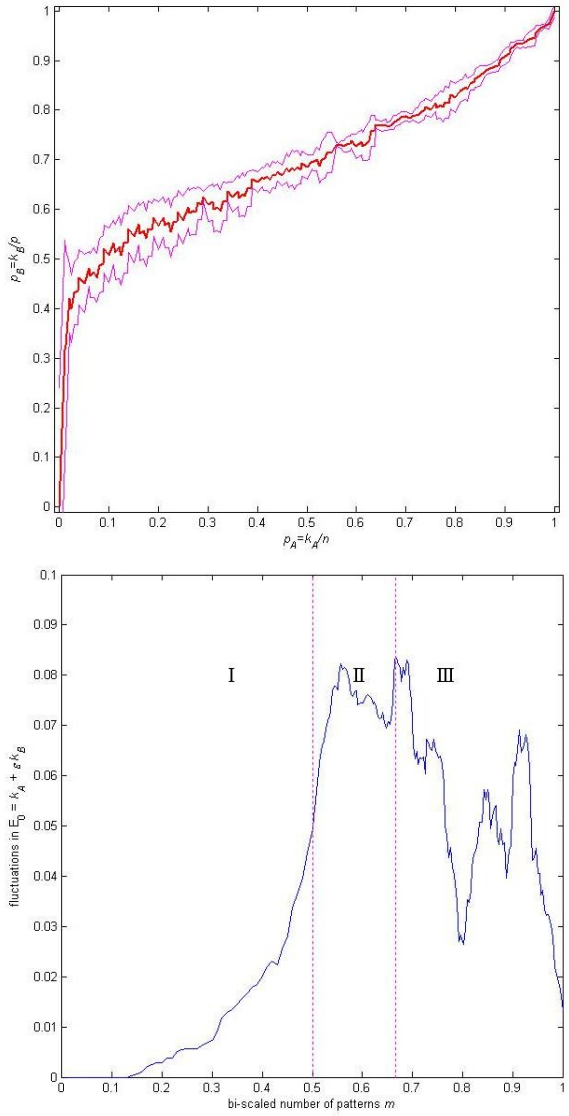

Fig. 7. Above: the relation of the relative sparsity $p_A = k_A/n$ in the gene-gene network and the relative sparsity in the input-gene connections $p_B = k_B/p$. This graph is averaged over 10K measurements. Below: fluctuations in the 'energy' indicate phase transitions. Here the number of zeros in the system matrices: $E_0 = k_A + \varepsilon * k_B$ acts as energy function, plotted versus the number of patterns m.

These conclusions are valid for the entire range of n/p stretching from 0.05 to 20, as investigated in this study. Furthermore, the cut-off point μ depends strongly on the

regularization parameter ε. As indicated before, in all experiments presented here the value $\varepsilon = 1$ was used, resulting in the critical point for the I/II transition $\mu_{C1} = 1/2$.

6.2 Influence of the extra constraints

In LP1, i.e. equation 6, explicit constraints in the form: $x_i = constant(i), i = 1, 2, \ldots, C$ were add to avoid the matrices A and B from becoming equal to zero. This addition, however, has little impact on the behavior of the system. With an increasing number C of such constraints the phase transition at m_c disappears, until at $C = n^2+np$ all values of A and B are specified by the constraints. Also, with an increasing number of constraints C the morphology to the right of m_c gradually flattens, causing the disappearance of the transition II/III, as evident in Figure 6.2, containing $C = 10$ constraints.

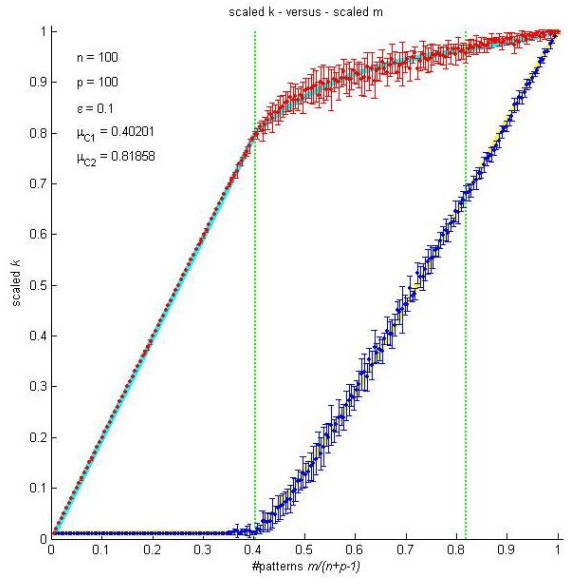

Fig. 8. The connectivity probabilities k_A/n and k_B/p for n=100 and p=100, with $C = 10$ extra constraints in the form: $x_i = constant$. This also exhibits the strong phase transition at m_c. Right to this value, the morphology however is flatter than for less extra constraints.

For $C = 0$ and μ below μ_{C1}, the solution to LP1 yields the matrix $A = 0$ as solution. Thus, besides the desired patterns $(\mathbf{x}^*, \mathbf{u}^*)$ there are many other unsolicited equilibria, as all $(\mathbf{y}, \mathbf{u}^*)$ are also valid solutions because $A\mathbf{x}^* = A\mathbf{y} = \mathbf{0}$. This condition allows all possible values for the genes and proteins, because the input $\mathbf{u}^*$ is directly coupled the desired equilibrium $\mathbf{x}^*$. The genes are simply not involved, and the system is a chemical buffer that has evolutionary learned to counter the important stimulus 'impulsively'.

However, right to the critical point μ_{C1} an increasing number of components of A and B become non-zero. Therefore it is *not* necessary to impose the extra constraints

on LP1: it does not change the behavior of p_A and p_B below μ_{C1} (except a vertical translation), and above μ_{C1} there are sufficient non-zeros in both schemes anyway. Biologically, this means that here the genes and proteins are involved in selecting the equilibrium. The network processes the input information, and then 'decides' – computes – the best system response $\mathbf{x}^*$.

7 Stability of the stored patterns

To study the stability of the patterns, a set of $m = 64$ linearly independent patterns was created, and offered to a system with $n = 100$ nodes – i.e. genes/proteins – and $p = 6$ inputs, so the vector $\mathbf{x}$ contains 106 elements. $C = 1$ extra constraint was used: $x_1 \leq -1/3$. The associated input $\mathbf{u}_i$ consists of the 6-bit coding of the index-number $i \in \{1, \ldots, 64 = 2^6\}$ of the pattern $\mathbf{m}_i$. Numerical experiments were per-

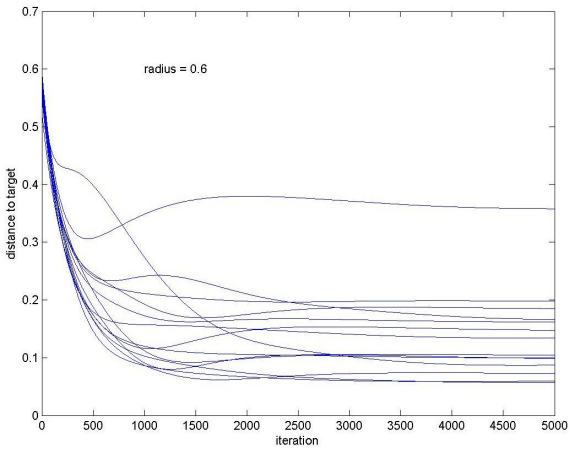

Fig. 9. Dynamical behavior of a network which has 'learned' to respond with a system state $\mathbf{x}_1$ to an input $\mathbf{u}_1$. This plot shows the evolution of the distance of the actual system state $\mathbf{x}(t)$ to the target system state $\mathbf{x}_1$, when an input $\mathbf{u}$ is created a distance r away from the input $\mathbf{u}_1$. In one case the system apparently converges to a mixed pattern between the target patterns.

formed such that a given input $\mathbf{u}_i$ was offered to the system in a random equilibrium state $\mathbf{x}_0$. The subsequent convergence relative to the pattern $\mathbf{x}_i$ is studied as function of the distance between $\mathbf{x}_0$ and $\mathbf{x}_i$. This is performed by computing the evolution of $\mathbf{x}(t)$ using: $\dot{\mathbf{x}} = A\mathbf{x} + B\mathbf{u}_i$, and $\mathbf{x}(0) = \mathbf{x}_0$. Numerical analysis shows that A only has negative eigen-values. However, the largest eigen-value is just below zero: $\max(\text{eigenvalues}(A)) \approx 10^{-11}$. Consequently, the state vector $\mathbf{x}(t)$ will converge, but not necessarily to the pattern $\mathbf{x}_i$ associated to the input $\mathbf{u}_i$. If the distance between $\mathbf{x}_0$ and $\mathbf{x}_i$ becomes too large, namely in the order of the distance between the patterns themselves, it is likely that the state vector converges to a mixed state somewhere between

the patterns. Figure 9 indicates that the end result when offered input $\mathbf{u}_2$ to a system in equilibrium $\mathbf{x} = \mathbf{x}_1$ may converge to a state other than the target pattern $\mathbf{x}_2$. Obviously it has converged to a mixed state somewhere in-between pattern $\mathbf{x}_1$ and pattern $\mathbf{x}_2$. Figure 9 shows the evolution of the distance to pattern $\mathbf{x}_1$ if a random pattern is created within an Euclidean distance $r = 0.6$ from pattern $\mathbf{x}_1$. This Figure shows that most random patterns converge to a state near pattern $\mathbf{x}_1$, so close that it is practically identical.

8 Discussion and conclusions

In this paper we discussed the characteristics of the genetic/proteomic memory of simple linear and sparse random networks. It is found that a linear time-invariant state space formulation provides a comprehensible and transparent model that is accessible to numerical and mathematical analysis. Even this simple model exhibits a range of complex behaviors for storing input-output patterns in a sparse representation. Most important is the fact that the storage of both dynamical and static patterns in sparse networks exhibit distinct phase transitions. In the case of static input-output memory there are two second-order continuous phase transitions. These transitions occur for a number of patterns that equals $m_{C1} = p/2$, and $m_{C2} = n/3 + 2p/3 - 1/3$ respectively, or in scaled coordinates respectively for $\mu_{C1} = 1/2$, and $\mu_{C2} = 2/3$. These transitions divide the learning in three distinct regions. In the first region where $m < m_{C1}$ all information is exclusively stored in the input-output matrix B, so that this state represents a high degree of order. In the second region, $m_{C1} < m < m_{C2}$ the information is stored increasingly in the gene-protein interaction matrix A. Finally, for $m > m_{C2}$ the information is stored evenly in both matrices.

The parameters n, p, and ε influence the behavior of the system to different extent. The model is symmetrical in $\mathbf{x}$ and $\mathbf{u}$, as $A\mathbf{x}+B\mathbf{u} = \mathbf{0}$, and the criterion function is: $\|A\|_1 + \|B\|_1$, as the regularization parameter ε was set to 1. However, n and p have basically different influences on the behavior. Using the scale-transformations 7, it was possible to obtain an almost scale-free representation of the graphs of resulting sparsity versus number of stored patterns. Below the first critical point m_{C1} the relation is essentially scale-free. However, above this value the morphology of the graph dependents on n and p, as visible in Figure 2. Especially the plateau near the second critical point μ_{C2} becomes better visible for higher p. Moreover, the first critical point itself is found to occur at $p/2$, and is therefore independent of n.

In this study, we considered only linearly independent patterns. However, real gene-protein networks are often considered to be modular. In this case, different sub-patterns may combine to give the global pattern. Therefore the global patterns are not necessarily linearly independent, but can be constructed from a set of linearly independent sub-patterns.

As an example of a practical application we consider the olfactory system, see: [44]. In mammals, the odorant receptors are transmembrane molecules, synthesized by the largest gene family with approximately 1000 gene members, capable of recognizing some 10,000 different odors. In this case, we might assign the number of possible patterns m to about 10K. The number of inputs p is here about 1000, as there are about 1000 functional odor receptors. The model parameter n here is the number of genes, RNAs,

and proteins involved in the odor-receptor signalling pathway. From a database search in EMBL/GenBank for the ontology of this receptor protein signaling pathway indicates that there at least 2000 of such proteins known. So, n here is at least: 1000 genes + 1000 tRNAs + 2000 signal molecules + 1000 receptor molecules, so at least several thousand, say $n \approx 5000$. So, for the odor receptor pathway, $n \approx 5000$, $p \approx 1000$, and $m \approx 10,000$. The minimum number $(n + p)_{min}$ of genes/proteins and inputs necessary to 'learn' m patterns in a sparse setting equals: $(n + p)_{min} \approx \log m$, see [43, 27]. Estimating the constant in this relation roughly based on this work, we find: $(n + p)_{min} = 50 * \log m = 400$. So: $n + p = 6000 \gg (n + p)_{min}$. In a non-sparse setting, there are m equations with about $n^2 + np$ free parameters, so non-sparse $m_{max} = n^2 + np \approx 3.10^7$, which is considerable larger than the 10,000 odor patterns. In both settings, there seems to be an enormous overkill of pattern memory in the odor receptor network. These are very rough estimates.

As a final point, the stability of the thus stored patterns was examined. It was found that the patterns are stable, each pattern having a well-defined basin of attraction. However, the size of these basins vary with the number of stored patterns, which is intuitively clear as the available space remains the same, so the space-per-patterns decreases. However, even close to the patterns there are paths to mixed states in which some patterns are "confused".

These simple experiments hint to some characteristics to be expected for memory storage in more realistic gene-protein networks. Firstly, memory storage can be understood as fixing the free model parameters such that a presented input matches with the desired output. In the natural world this learning process is performed by the actions of biological evolution. Secondly, also in real biological information networks it is possible to directly couple the input to the output without altering the gene-protein interactions. This creates an "instinctive" reaction to the presented input, that only indirectly alters the states of the genes and proteins. This also includes epigenetic memory storage, as it can be understood as a specific state vector of the proteome without regard to the states of the genome. Thirdly, realistic networks potentially may exhibit phase transitions as the number of stored patterns grows. These transitions can divide the storage process in distinct regions with its own learning strategy. Finally, even in real biological networks this has lead to predominantly stable patterns, just as in the context of the simple networks studied here.

The authors wish to thank the anonymous reviewers for their thorough and inspired review of the text. This research was partly supported by the EU-project FP6 project 013569 NiSIS (Nature-inspired Smart Information Systems).

References

1. Barabasi A.L., Reka A., Emergence of scaling in random networks, *Science*, 286:509-512, 1999.
2. Bishop C.M., Neural Networks for Pattern Recognition, Oxford University Press, 1995.
3. Bower J.M., Bolouri H.(Editors), Computational Modeling of Genetic and Biochemical Networks, *MIT Press*, 2001.
4. D'haeseleer P., Liang S., Somogyi R., Genetic Network Inference: From Co-Expression Clustering to Reverse Engineering, *Bioinformatics*, vol. **16**, no. 8, 2000, pp. 707–726.

5. Duda, R.O., Hart, P.E., Stork, D.G. Pattern classification, Wiley, 2001.
6. Duncan J.W, Strogatz S.H.,Collective dynamics of small-world networks. *Nature*, 393:440-442, 06 1998.
7. Elowitz M.B., Levine A.J., Siggia E.D., Swain P.S., Stochastic gene expression in a single cell, *Science*, vol.**297**, August 16, 2002, pp.1183–1186.
8. Endy, D, Brent, R. (2001) Modeling Cellular Behavior, *Nature* 2001 Jan 18; 409(6818), pp.391-5.
9. Fuchs J.J. (2003), More on sparse representations in arbitrary bases, in: *Proc. 13th IFAC Symp. on System Identification*, Sysid 2003, Rotterdam, The Netherlands, August 27-29, 2003, pp. 1357–1362.
10. Fuchs J.J. (2004), On sparse representations in arbitrary redundant bases, *IEEE Trans. on IT*, June 2004.
11. Glass L., Kauffman S.A. (1973), The Logical Analysis of Continuous Non-linear Biochemical Control Networks, *J.Theor.Biol.*, 1973 Vol. 39(1), pp. 103–129
12. Goldbeter A (2002) Computational approaches to cellular rhythms. *Nature* 420, 238-45
13. G.H. Golub, C.F. van Loan, *Matrix Computations* (2nd ed.). Baltimore: The Johns Hopkins University Press, 1989.
14. Gonze D, Halloy J, and Goldbeter A (2004) Stochastic models for circadian oscillations : Emergence of a biological rhythm. *Int J Quantum Chem* **98**, pp 228–238.
15. Hasty J., McMillen D., Isaacs F., Collins J. J., (2001), Computational studies of gene regulatory networks: in numero molecular biology,*Nature Reviews Genetics*, vol. 2, no. 4, pp. 268–279, 2001.
16. de Jong H., Modeling and Simulation of Genetic RegulatorySystems: A Literature Review, *Journal of Computational Biology*, 2002, Volume 9, Number 1, pp. 67–103
17. de Jong H., Gouze J.L., Hernandez C., Page M., Sari T., Geiselmann J., Qualitative simulation of genetic regulatory networks usingpiecewise-linear models, *Bull. Math Biol.* 2004 Mar;66(2): pp 301–40.
18. Makoto Hirahara, Natsuki Oka, Toshiki Kindo, Associative memory with a sparse encoding mechanism for storing correlated patterns, *Neural Networks*, Volume 10, Issue 9, December 1997, pp. 1627-1636
19. Masahiko Morita, Memory and Learning of Sequential Patterns by Nonmonotone Neural Networks, *Neural Networks*, Volume 9, Issue 8, November 1996, pp. 1477-1489
20. Masato Okada, Notions of Associative Memory and Sparse Coding, *Neural Networks*, Volume 9, Issue 8, November 1996, pp. 1429-1458
21. Mazza Christian, On the Storage Capacity of Nonlinear Neural Networks, *Neural Networks*, Volume 10, Issue 4, June 1997, pp. 593-597
22. Mestl T, Plahte E, Omholt SW., A mathematical framework for describing and analysing gene regulatory networks. *J Theor Biol.* 1995 Sep 21;176(2), pp. 291-300
23. Newman M. E. J., Power laws, pareto distributions and zipf 's law, *Contemporary Physics*, 2005, pp. 46:323
24. Novak B, Tyson JJ , Modeling the control of DNA replication in fission yeast, *PNAS*, USA, Vol. 94, August 1997, pp. 9147-9152
25. Omholt SW, Kefang X, Andersen O, Plahte E., Description and analysis of switch-like regulatory networks exemplified by a model of cellular iron homeostasis, *J Theor Biol.*, 1998 Dec 7;195(3), pp.339-50
26. Omholt SW, Plahte E, Oyehaug L, Xiang K., Gene regulatory networks generating the phenomena of additivity, dominance and epistasis. *Genetics*, June 2000; 155(2), pp. 969-80
27. Peeters R.L.M., Westra R.L., On the identification of sparse gene regulatory networks, *Proc. of the 16th Intern. Symp. on Mathematical Theory of Networks and Systems* (MTNS2004) Leuven, Belgium July 5-9, 2004

28. Plahte E, Mestl T, Omholt SW, A methodological basis for description and analysis of systems with complex switch-like interactions, *Journal of Mathematical Biology*, 36, 321-348, 1998.
29. Somogyi R., Fuhrman S., Askenazi M., Wuensche A. (1997). The Gene Expression Matrix: Towards the Extraction of Genetic Network Architectures. Nonlinear Analysis, *Proc. of Second World Cong. of Nonlinear Analysis* (WCNA96) 30(3) pp 1815–1824.
30. Schfer J., Strimmer K., An empirical Bayes approach to inferring large-scale gene association networks, *Bioinformatics*, No. 21, 2005, pp. 754-764.
31. Spieth Ch., Streichert F., Speer N., Zell A., Iteratively Inferring Gene Regulatory Networks with Virtual Knockout Experiments, *Lecture Notes in Computer Science*, Volume 3005, 2004, Springer Publishers, pp 104-112.
32. Swain P.S., Efficient attenuation of stochasticity in gene expression through post-transcriptional control, *J Mol Biol* 344 (2004) pp 965.
33. Swain P.S., Elowitz MB, Siggia ED, Intrinsic and extrinsic contributions to stochasticity in gene expression, *Proc. Nat. Acad. Science* 99 (2002) pp 12795.
34. Steuer R. (2004), Effects of stochasticity in models of the cell cycle:from quantized cycle times to noise-induced oscillations, *Journal of Theoretical Biology* 228 (2004) 293-301.
35. Tegnér J., Yeung M.K.S., Hasty J., Collins J.J., Reverse engineering gene networks: Integrating genetic perturbations with dynamical modeling, *Proc. Nat. Acad. Science*, vol. **100**, no. 10, 2003, pp. 5944–5949.
36. Venter J.C. et al., Essential genes of a minimal bacterium, *Proc. Nat. Acad. Science*, vol. **103**, no. 10, 2006, pp. 425-430.
37. Wasserman, Ph., Advanced Methods in Neural Computing, Van Nostrand Reinhold Publishers, 1993
38. Westra R.L., Modelling and identification of dynamical gene interactions, Workshop Gene Expression Based Individualized Medicine, 14th May 2004, Jena/Germany
39. van Kampen N. G. (1992), Stochastic Processes in Physics and Chemistry, Elsevier Science Publishers, Amsterdam, (1992).
40. Westra R.L., Hollanders G. , Peeters R.L.M., Reconstruction of Flexible Gene-Protein Interaction Networks using Piecewise Linear Modeling and Robust Regression, AISB'06: Adaptation in Artificial and Biological Systems, April 3rd-6th 2006, University of Bristol, Bristol, England
41. Westra R.L., Hollanders G., Bex G-J., Gyssens M., Tuyls K., Knowledge discovery and emergent complexity in bio-informatics, *Lecture Notes in Bioinformatics* (LNBI), Volume Number 4366, January 2007, Springer Publishers
42. Westra R.L., G. Hollanders, G-J. Becx, M. Gyssens, K. Tuyls (2007b), The Identification of Dynamic Gene-Protein Networks, *Lecture Notes in Bioinformatics* (LNBI), Volume Number 4366, January 2007, Springer Publishers
43. Yeung M.K.S., Tegnér J., Collins J.J., Reverse engineering gene networks using singular value decomposition and robust regression, *Proc. Nat. Acad. Science*, Vol. **99**, no. 9, 2002, pp. 6163–6168.
44. Zozuluya S.F., Echeverri F., Nguyen T., The human olfactory receptor receptoire, *Genome Biology*, No.2, 2001.