Pattern storage in gene-protein networks

Ronald L. Westra

Department of Mathematics, Maastricht University and Transnational University of Limburg, Maastricht, The Netherlands

E-mail to: westra@math.unimaas.nl

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 u* there exists an optimal state of the system 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-out 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.

Keywords: gene protein interaction networks, genetic memory, linear time-invariant dynamical systems, sparse hierarchical networks, network reconstruction.

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. The network architecture has to react adequately and appropriately fast to a given input. In real biological information processing networks this has resulted in a sparse and hierarchical connectivity matrix, and the so-called 'small-world' property. These architectural characteristics pose strict design criteria to the network model.
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. 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 this consists 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.
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 [de Jong et al. 2002; de Jong et al., 2004;

D'haeseleer et al. 2002; Guthke et al. 2002; Hasty et al. 2001; Peeters and Westra 2004; Somogyi et al. 1997; Tegnér et al. 2003; Verdult and Verhaegen 2004; Yeung et al. 2002]. Especially relevant to our approach is the work of Mestl, Plahte and Omholt [Mestl et al. 1995; Plahte et al. 1998; Omholt et al. 1998; Omholt et al. 2000;], Somogyi [Somogyi et al. 1997;], Yeung and Tegnér [Yeung et al. 2002; Tegnér et al. 2003], and Peeters and Westra [Peeters and Westra 2004; Westra 2006a; Westra 2006b]. For a thorough overview consult [Westra 2007a] 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 [Spieth et al. 2004; Mazza 1997; Makoto et al. 1997; Masato 1996; Morita 1996]. For a good survey consult standard books as [Bishop 1995; Duda 2001; Wasserman 1993].
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 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 x, will develop according to the n*x*n gene-protein interaction matrix A and the n*x*p interaction matrix B with the p external outputs, defined by the p-vector u, as :

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

Where $\xi(t)$ represents white Gaussian noise with mean zero and standard deviation 1.
Such simple linear models allow for a limited set of behaviours, 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 function of the number of genes/proteins n, the number of external outputs p, and the sparsity in the interaction matrices A and B.

3. Storage of dynamical patterns and its relation with network reconstruction

There is a important similarity between reconstructing the best fitting matrices A and B from empirical sets data state vectors X and 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 [Peeters and Westra 2004].

We define dynamical patterns as the desired dynamical behaviour of the state vector x(t) for a given input vector u(t), such as a finite impulse or a harmonic signal. In this way, the desired outputs x*(t) and $\dot{x}^*(t)$ for a given input u*(t) can be compared with the observed outputs x(t) and $\dot{x}(t)$. Suppose we have sets of m desired input-output triples as $\{(u_l, x_l, \dot{x}_l) \mid l = 1..m\}$, which can be used to define three data matrices: $X = (x_1 \ldots x_m)$, $\dot{X} = (\dot{x}_1 \ldots \dot{x}_m)$, and $U = (u_1 \ldots u_m)$. In this setting, one could attempt to minimize the error ||x*(t) - x(t)|| as function of the free parameters in A and B. Or, alternatively, minimize the error in the time derivatives: $\| \dot{x}^*(t) - \dot{x}(t) \|_k = \| \dot{x}^*(t) - Ax^*(t) - Bu^*(t) \|_k$. Where $\|*\|_k$ represents a suitable L_k-norm.
Given X, $\dot{X}$, and U one could compute the of 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:

$$\max_{A,B} \| A \|_0 + \| B \|_0 \quad \text{subject to} \quad \dot{X} = AX + BU \qquad (2)$$

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.

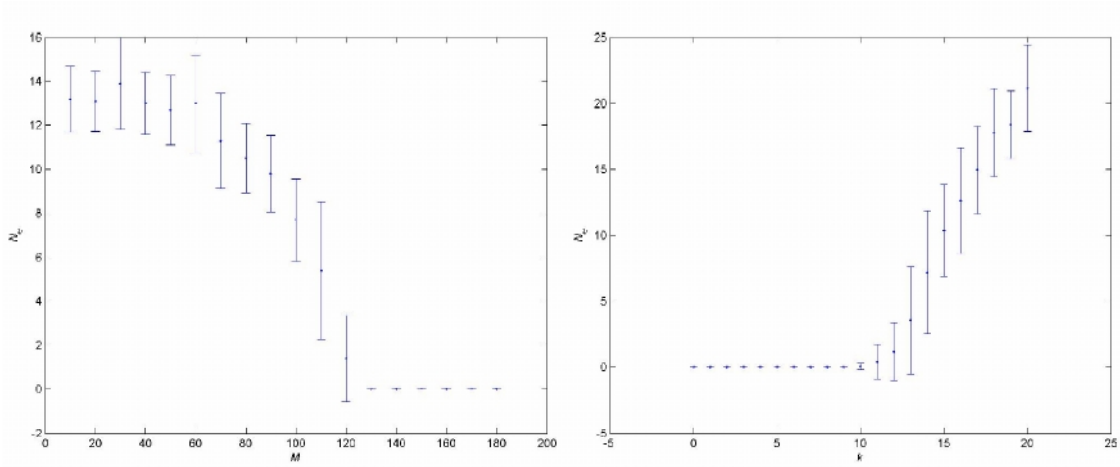

Figure 1, Left: the number of learning errors N_e versus the number of learning examples m, right: N_e versus the number of non-zero elements in A. Images reproduced with permission from Peeters, Westra 2004.

This problem is well described in a number of publications, e.g. Yeung et al. 2002, Peeters&Westra 2004, and Westra et al. 2007b. For a relevant survey see Westra et al 2007a and the references therein.
We here merely note that this problem exhibits a first-order phase transition as evident from Figure 1 below. The number of learning errors N_e suddenly drops to zero as the number of learning examples 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 the same critical value m_C.

4. Memory storage of static 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 x* and that $\dot{x} = 0$. This means that to a given input u* there belongs an optimal value x* for the gene expressions and protein densities in the network such that Ax* + Bu* = 0. In our context a (static) input-output pattern is defined as such an input-output pair (u*,x*). This pattern itself can be considered as the outcome of a long evolutionary learning process, where the best combination of gene/protein activations x are selected relative to the faced combination of external inputs (e.g. toxic agents, virus infections) 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 patterns, i.e. external inputs as a p×*m* matrix $U = \begin{pmatrix} u_1 & \dots & u_m \end{pmatrix}$, and the associated (evolutionary) learned states as a n×*m* matrix $X = \begin{pmatrix} x_1 & \dots & x_m \end{pmatrix}$. For each input-pattern couple $\{x_j, u_j\}$ equation (1) implies that: $Ax_j + Bu_j = 0$. 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: $AX + BU = 0_{nm}$. This equation can be rewritten as: $X^T A^T + U^T B^T = O_{mn}$, or:

$\begin{pmatrix} X^T & U^T \end{pmatrix} \begin{pmatrix} A^T \\ B^T \end{pmatrix} = O_{mn}$. This matrix equation can be decoupled in n independent linear equations: $\begin{pmatrix} X^T & U^T \end{pmatrix} \begin{pmatrix} r_A(i)^T \\ r_B(i)^T \end{pmatrix} = 0$, where $\begin{pmatrix} r_A(i)^T \\ r_B(i)^T \end{pmatrix}$ is composed of the i-th rows of A and B.

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

$$Dx = 0 \qquad (3)$$

A practical problem with this reformulation is that, as this equation holds for each combined row x of A and B, its solution also holds for all rows, and therefore A and B consist of dependent columns in the form:

$$v \otimes x = \begin{pmatrix} v_1 x_1 & v_1 x_2 & \cdots & v_1 x_n \\ \vdots & \vdots & \vdots & \vdots \\ v_n x_1 & v_n x_2 & \cdots & v_n x_n \end{pmatrix} \qquad (4)$$

for some non-zero vector $\mathbf{v} = \begin{pmatrix} v_1 & \cdots & v_n \end{pmatrix}^T$. This can be avoided by involving available biological knowledge, such as the hierarchical position of a certain gene/protein-coupling in the actual network.

The objective now is to find sparse A and B that satisfy this condition. As, however, the sparse matrices A = B = 0 pose a trivial but valid solution to (2), an additional constraint is required to exclude this solution. As there is a scaling invariance in the solution – if x is a solution of Dx = 0 then λx is also a solution for all $\lambda \in R$ – the exclusion of A=B=0 can be obtained by fixing the B-part of 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:

$$q^T \boldsymbol{x} = 1 \qquad (5)$$

where the first n components (the "A-part") of q are zero, and the p remaining components q_i, (i = n+1..n+p), (the "B-part") satisfy: $|q_i| = 1$.

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 Dx = 0 has been studied extensively in literature. In [Fuchs 2003] and [Fuchs 2004] and some of the references therein, results are given which state conditions under which optimal sparse solutions can be obtained by the technique of L_1-minimization. This technique aims to compute a vector x of minimal L_1-norm within the solution space S of the linear system Ax = 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 x (associated to corresponding rows in A and B) to the problem of finding the minimum value of the L_1-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 y_1 denote the first n components of x (the "A-part"), and y_2 the other p components (the "B-part"), or: $\mathbf{y}_1 = \mathbf{S}_1\mathbf{x} = \begin{pmatrix}\mathbf{I_n} & 0_{\mathbf{np}}\end{pmatrix}\mathbf{x}$, and $\mathbf{y}_2 = \mathbf{S}_2\mathbf{x} = \begin{pmatrix}0_{\mathbf{pn}} & \mathbf{I_p}\end{pmatrix}\mathbf{x}$. With all these definitions, the regularized minimum value of the L_1-norm of x for the given data of m patterns for n genes/proteins, and p external inputs follows from the following linear programming formulation:

$$\text{LP1:} \quad x^* = \arg\min_{x \in R^{n+p}} \| S_1 x \|_1 + \varepsilon \| S_2 x \|_1 \qquad (6)$$

subject to:

$$Dx = 0$$
$$q^T x = 1$$

This allows one to interpret this problem as a regularized L_1-minimization problem over the affine subspace of vectors 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. Results from numerical experiments for static pattern storage

5.1 Rescaled representation

The formulation defined above allows for numerical experiments with different settings of the system parameters. Here some of these experiments are described. During these experiments, it became clear that it is useful to rescale the number of patterns m to a new value.
The number of sparse patterns m ranges between 1 and $m_{max} = n+p-1$, and it was found to exhibit different behaviour left and right of the value $m_C = p/2$. For this reason, a rescaled version of m is proposed as:

$$m_s = m/m_C, \text{ if } m < m_C = p/2, \qquad (7)$$
$$m_s = (m+ n-1)/(2n+p-2), \text{ if } m >= m_C.$$

This scaling ensures that m_s is between 0 and 1. This bi-scaled variable is useful in comparing situations with different n and p.

In all experiments described here the regularization parameter ε from equation (6) is set to 1. Other values of ε gave quantitative different results that, however, could again be scaled such that they were qualitatively similar. The patterns X and U were drawn uniform randomly from the interval [-1,1].

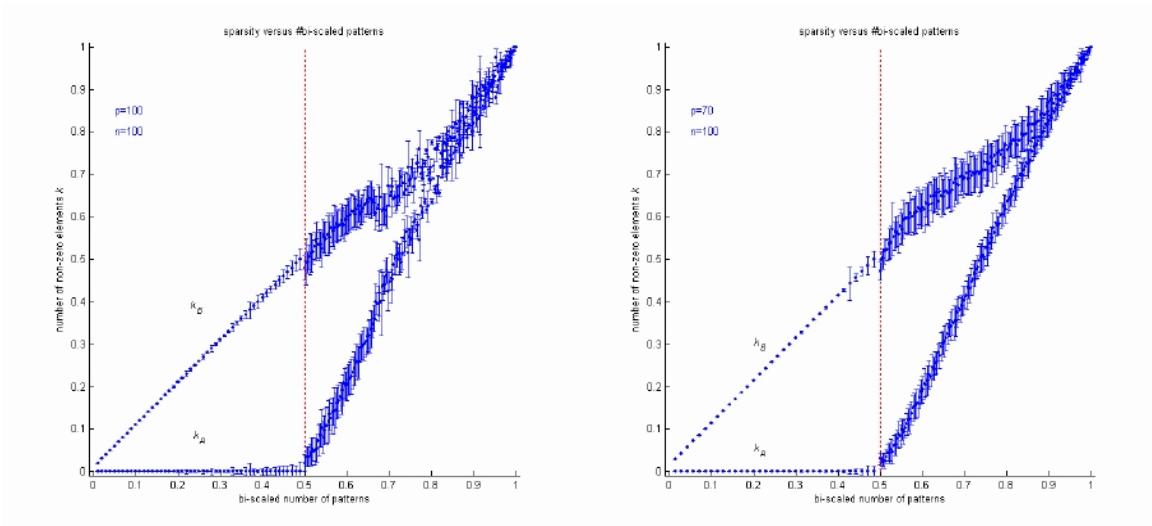

Figure 2. Solutions of the LP-approach to memory storage in linear networks. Left from this value the relations are slightly but statistically significant curved. Left a plot for n=100 and p=100, right a plot for n=70 and p=100.

5.2 The morphology of the relation between sparsity and number of patterns

The value of m varies between 0 and n+p-1. More specifically, we are interested in partial sparse solutions for B and A. Figure 2 shows two results from such experiments. In these plots the sparsity in the matrix is plotted versus the bi-scaled number of patterns m_S. The ordinate represents the relative number of non-zeros in the matrix. This value is scaled to its maximum value so that it ranges from zero to one. Here we introduce k_A as the number of non-zero elements in matrix A, and similarly k_B for B. Therefore, the relative values are respectively k_A/n and k_B/p. The lower curve gives the relative non-zeros in matrix A. This exhibits a clear transition at $m_S = ½$.. The upper curve gives the relative non-zeros in matrix B.

In the rescaled representation these plots are almost identical, though the morphology somewhat depends on n+p.

5.3 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}. 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: Ax = 0, where A is the m**x**(n+p) data matrix (X^T U^T), and the first n components of x correspond to a row of the original matrix A, and the last p components to the corresponding row of the original matrix B. Thus, the solution $x \in R^{n+p}$ lies in the (n + p – m)-dimensional linear subspace V_1 defined by Ax = 0. Second, the solution x is bound by the (n + p – 1)-dimensional affine space V_2 defined by $q^T \boldsymbol{x} = 1$, where the first n components of q are zero, and the p remaining components q_i , (i = n+1..n+p), satisfy: $|q_i| = 1 \| q_i \| \in [½,2]$. The intersection $V_3 = V_1 \cap V_2$ contains the solution 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:

$$m_{max} = n + p - 1 \qquad (8)$$

The situation becomes more complex when we analyze the number k of exact zeros in x, as the optimization is aimed at increasing this amount. k_A denotes the number of non-zeros in the first n components of 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 $\left(X^T \; U^T\right)\begin{pmatrix} r_A^T \\ r_B^T \end{pmatrix} = 0$ the condition $k_A = 0$ means that the row of A $r_A^T = 0_{n1}$, therefore: $U^T r_B^T = 0$, meaning that r_B^T lies in the null space of U^T. The condition $k_A = 0$ thus implies that: $x_1 = x_2 = \ldots = x_n = 0$. This 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 until $V_3 \cap V_4 \neq \varnothing$. $V_3 \cap V_4$ is defined by (m+1+n) equations with (n + p) variables that implies an upper bound m^A_{up} for $k_A = 0$, namely: $m^A_{up} = p - 1$.

However, as apparent from Figure 2, k_A already abruptly starts deviating from zero at $m = m_C \equiv p/2 < m^A_{up}$. Though in that case $V_3 \cap V_4$ is not-empty, the L_1-optimization process favours another solution with a smaller value for k_A. Finding a solution with $k_A = 0$ results in solving the linear equation: Zx = z, with:

$$Z = \begin{pmatrix} A \\ q^T \\ I_n \quad 0_{n,p} \end{pmatrix}, \quad z = \begin{pmatrix} 0_{m,1} \\ 1 \\ 0_{n,1} \end{pmatrix}. \qquad (9)$$

This system has a consistent solution when 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 $Zx = z_{proj}$, where the vector z_{proj} denotes the orthogonal projection of z onto the column space of Z. An explicit description of S follows from computing the solution space of the system $Zx = z_{proj}$, which can be achieved in a numerically reliable way by employing SVD or QR-decomposition. Thus, S can be formally written as: $x_{alg} = N(Z).\xi + Z^{+}.z$, where N(Z) is the $n \times d_{null}$ matrix whose columns consist of the of the basis of the d_{nul}–dimensional null-space of Z, and Z^{+} is the pseudo-inverse of Z. ξ is a d_{nul}–dimensional vector that parameterizes S. The support vector $Z^{+}.z$ may indeed have a lower k_A than the solution x* of the LP, as shown in Figure 3. This is an artefact of the L_1-minimization process that can favour a smaller value of the absolute value of a non-zero component rather than a smaller value of k_A. Hence, in general:

$$\begin{aligned} &\text{criterion}(x^*) = \\ &\| x^*(1:n) \|_1 + \varepsilon_1 \| x^*(n+1:n+p) \|_1 < \| x_W(1:n) \|_1 + \varepsilon_1 \| x_W(n+1:n+p) \|_1 \\ &= \text{criterion}(x_w) \end{aligned} \qquad (10)$$

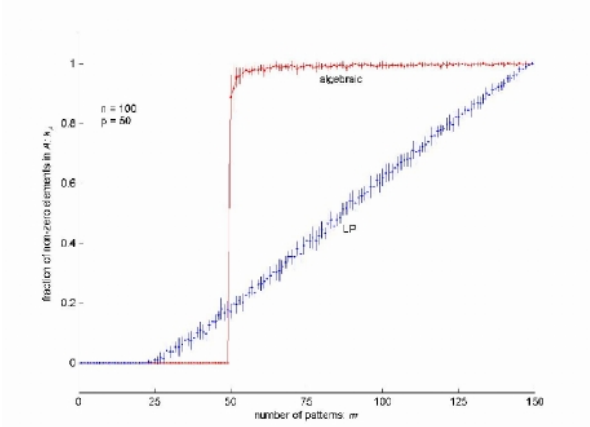

Figure 3 Comparison of LP-solution (tilted line) and the algebraic solution (step function). m varies between 1 and n+p–1 = 149. At m = p – 1 = 49 the algebraic solution suddenly jumps from 100% to 0% sparsity of A, while the LP-solution exhibits a 1st order phase transition at m = p/2 = 25.

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 x, is defined by V_3 , the $(n + p – k_A)$-dimensional linear subspace with k_A of the first n coordinates of x equal to zeros, and the $(n + p – k_B)$-dimensional linear subspace with k_B of the last p coordinates of x equal to zeros. 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:

$$k_A + k_B \geq m + 1 \qquad (11)$$

This relation is indeed found empirically as shown in Figure 4 where k_B/m is plot against k_A/m.

That the phase transition for the LP1 should occur at m/p = ½ is not evident from these considerations. For other values of the regularization parameter ε quantitative different values from ½ were found, so that this critical value is a function of ε .

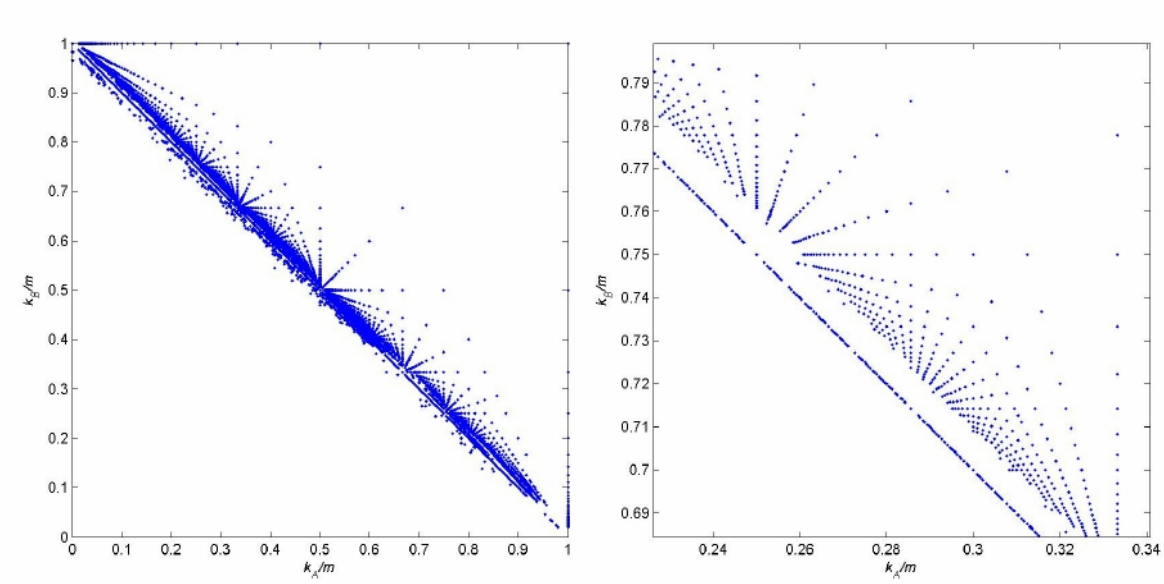

Figure 4 Left: the relation between k_B/m and k_A/m, right: detail that exhibits rich substructures of lines radiating from certain foci on $k_B/m + k_A/m = 1$.

6. Phase transitions in the matrix sparsity define distinct learning strategies.

The advantage of the rescaled representation of the relative numbers of zeros k_A/n and k_B/p with the bi-scaled number of patterns m_S is that thus different system settings can be straightforwardly compared. In the Figure below, the smoothed plot for k_A/n and k_B/p versus the bi-scaled m_S is depicted for n=100 and p=100, averaged over 10116 measurements.

This plot exhibits the typical characteristics representative for all the networks in our context. Firstly, the behaviour of k_A/n for $m_S = ½$ changes abruptly from constant zero tot an almost linear increase. Secondly, the characteristic plateau visible in k_B/p for $m_S = 2/3$ is not an artefact due to noisy measurements or processing deficiencies, but is present in all such graphs. Next, both graphs converge to the line $k_{rel} = 1.5m_S – 0.5$. Finally, for $m_S > 1$ the system of (independent) inequalities becomes inconsistent and therefore has no solution.

This graph therefore shows three different types of storing behaviour 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 random sparse networks under certain conditions exhibit 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 [Barabasi and Reka Albert 1999; Newman, 2005; Watts and Strogatz, 1998].
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 p, 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 $k_A = n/2$, $k_B = p/2$.

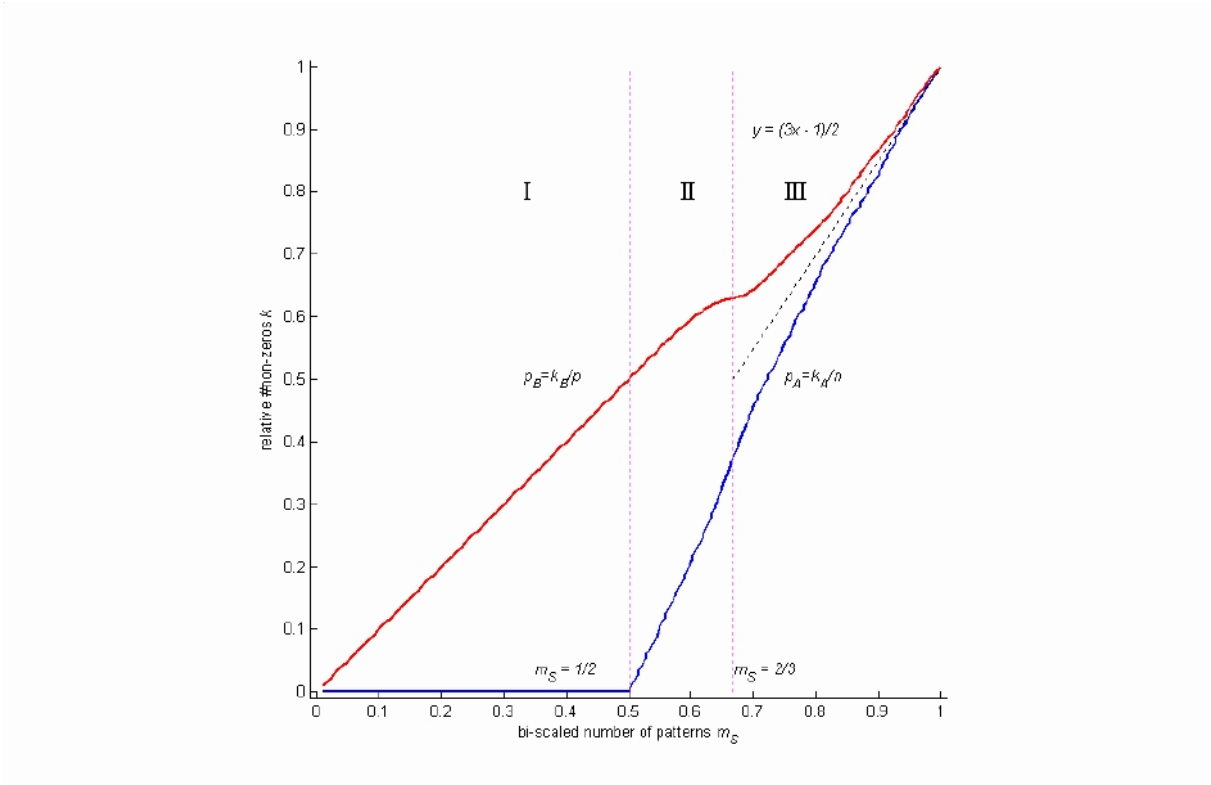

Figure 5 The connectivity probabilities k_A/n and k_B/p for n=100 and p=100, averaged over 10116 measurements, versus the bi-scaled number of patterns m_S. This indicates three major regions I, II, and III, where the network memory behaves different. The transitions occur for the bi-scaled m_S = ½ for the transition I to II, and m_S = 2/3 for II to III. In the final phase of region III both probabilities converge to $p = (3m_S - 1)/2$.

The wiring-probabilities p_A and p_B quantify the information that is stored in the matrices A and B. A convenient measure for this information is the information entropy, that in this case equals:

$$S = -p_A \ln p_A - (1-p_A)\ln(1-p_A) - p_B \ln p_B - (1-p_B)\ln(1-p_B) \qquad (13)$$

The entropy increase $\partial S/\partial m$ indicates the gain of information when one pattern is added to the network. This depends on the rate of change of the connection probabilities as function of the number of patterns m stored in the network, $\partial p_A/\partial m$ and $\partial p_B/\partial m$. In Figure 6 the entropy change $\partial S/\partial m$ for adding one more pattern is shown for n=100, p=30, based on 1180 observations. The total entropy increases moderately for small m until a sudden change at m_S = ½, subsequently exhibits an optimum around m_S =2/3, and finally decreases monotonously to zero. Consequently, the entropy increase $\partial S/\partial m$ is almost constant for small m_S, sharply peaks at m_S = ½, and then slowly but constantly decreases while crossing zero at $m_S \sim 2/3$.

It is remarkable that these changes happen around the same values for m_S 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.

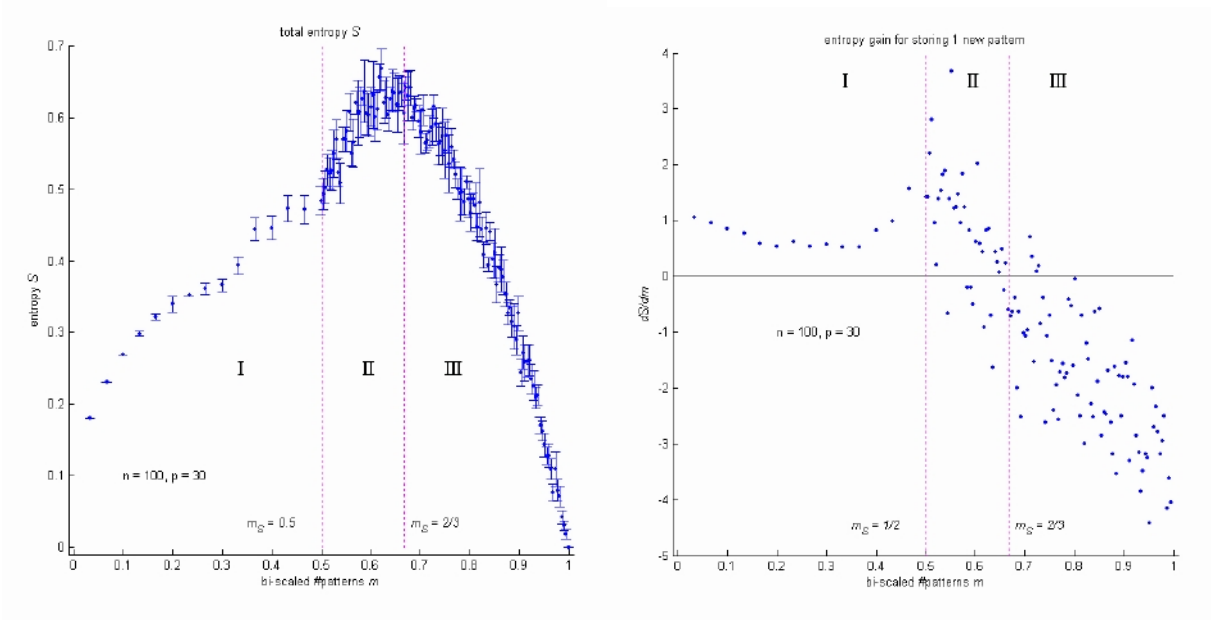

Figure 6 Left: the entropy S versus for n=100, p=30, based on 1180 observations Right: the gain in entropy for the same data set.

At m = p/2, there is an explicit second order continuous phase transition to region II, clearly visible in the behaviour of k_A and k_B. Here, it is evident that the derivatives of k_A and k_B to m_S are discontinuous. In region II the information is stored in both matrices, however increasingly in matrix A as m_S increases. Matrix B levels off to a plateau around m_S =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 m_S =2/3, that is best visible in the plots of k_B and $\partial S / \partial m$ versus m_S. 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 (the bi-scaled) m_S become increasingly identical, so: $\partial p_B / \partial p_A \to 1$, as evident in Figure 5. 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 $m_S = 1$, or $m = n + p - 1$. At this time 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.

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 $E = \| A \|_1 + \varepsilon \| B \|_1$. This quantity behaves very flat over the scope of m, only exponentially growing near the end, i.e. as m_S 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. This quantity indeed shows a strong and broad peak in its fluctuation near m_S = ½ 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 $m_S \sim 0.8$ and $m_S \sim 0.9$. Therefore, it is not clear which entity serves as energy in these phase transitions.

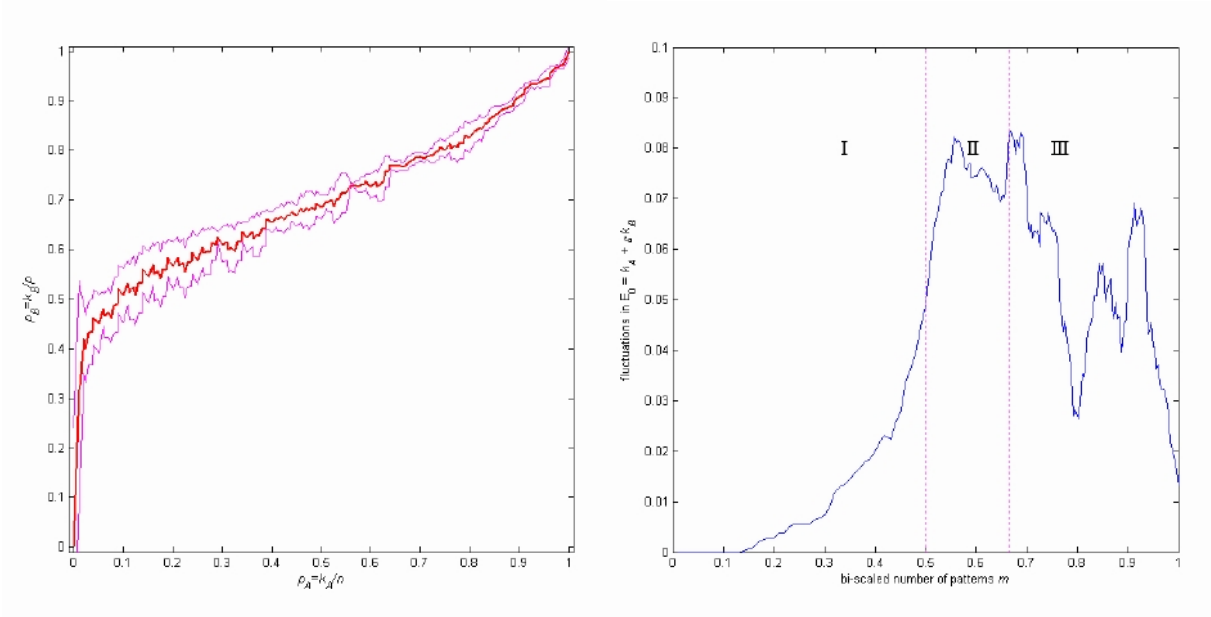

Figure 7 Left: the relation between $p_A = k_A/n$ and $p_B = k_B/p$ averaged for 10116 measurements. Right: fluctuations in the weighted number of zeros: $E_0 = k_A + \varepsilon k_B$.

Figure 7a shows that for very small p_A the fraction $\partial p_B / \partial p_A$ is infinite. This represents the situation in region I where p_A is fixed to 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 $p_B \approx (p_A + 1)/2$, and this is representative for region II.

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 m_S depends strongly on the regularization parameter ε. As indicated before, in all experiments presented here the value ε =1 was used, resulting in the critical point for the I/II transition m_S = ½ .

7. Stability of the stored patterns

As the LP-based approach to constructing the matrices A and B in principle generates identical rows, and therefore matrices of rank 1, it is desirable to add extra (preferably biologically inspired) constraints. In our context the extra condition concerns the stability of the system, and states that the diagonal elements of A should be negative, i.e. less than a given negative constant: $a_{ij} \le \varepsilon_2$. In our numerical experiments a good value turned out to be: $\varepsilon_2 = -1/3$. The revised linear programming definition, LP2, for the j-th row of A and B therefore states:

$$\text{LP2:} \quad x^* = \arg\min_{x \in R^{n+p}} \| S_1 x \|_1 + \varepsilon \| S_2 x \|_1 \qquad (13)$$

subject to:

$$Dx = 0$$
$$q^T x = 1$$
$$x_j \le \varepsilon_2$$

using the notation and definitions proposed earlier.
To study the stability of the patterns, a set of 50 linearly independent patterns was created, so m=64. With n = 100 nodes ("genes/proteins"), and p = 6 inputs, the vector x contains 106 elements. The associated input u consists of the 6-bit coding of the number.

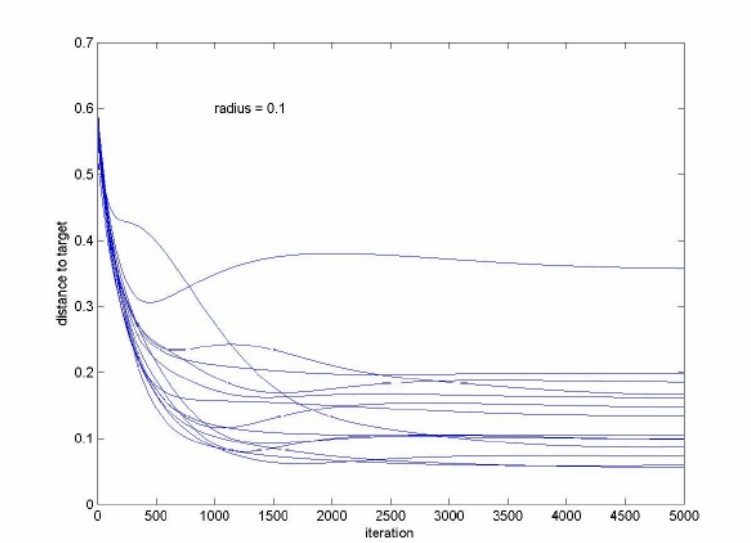

Figure 8 Evolution of the distance to pattern 1 if a random pattern is created within a distance 0.1 from there. In one case this apparently converges to a mixed pattern between pattern 1 and pattern 2.

Numerical experiments were performed such that a given input u_i – with i in {1..64} – was offered to a random pattern x_0 and its subsequent convergence relative to pattern x_i as function of the distance between x_0 and x_i. This is performed by studying the evolution of x(t) using: $\dot{x} = Ax + Bu_i$, and $x(0) = x_0$. Numerical analysis shows that A only has negative eigen-values. However, the largest eigen-value is just below zero: max(eigen-values(A)) ~ 10^{-11}. Consequently, the state vector x will converge, but not necessarily to the pattern x_i associated to the input u_i. If the distance between x_0 and 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 8 indicates that the end result when offered input 2 to a random pattern 1 may differ from the target pattern 2. Obviously it has converged to a mixed state somewhere in-between pattern 1 and pattern 2.

Figure 8 shows the evolution of the distance to pattern 1 if a random pattern is created within a distance r = 0.1 from pattern 1. This Figure shows that most random patterns converge to a state near pattern 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 behaviours 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 $m_1 = p/2$, and $m_2 = n/3 + 2p/3 – 1/3$ respectively, or in rescaled coordinates respectively for $m_{S,1} = ½$, and $m_{S,2} = 2/3$. These transitions divide the learning in three distinct regions. In the first region where $m < m_1$ 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, $m_1 < m < m_2$ the information is stored increasingly in the gene-protein interaction matrix A. Finally, for $m > m_2$ the information is stored evenly in both matrices.
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.

References

1. Barabasi Albert-Laszlo and Reka Albert. Emergence of scaling in random networks. Science, 286:509–512, 1999.
2. Bishop, C.M. (1995) Neural Networks for Pattern Recognition, Oxford: Oxford University Press. ISBN 0-19-853849-9 (hardback) or ISBN 0-19-853864-2 (paperback)

3. Bower J.M., Bolouri H.(Editors), Computational Modelling 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. Davidson E.H. (1999), A View from the Genome: Spatial Control of Transcription in Sea Urchin Development, Current Opinions in Genetics and Development, 9, pp. 530 – 541.
6. Duda, R.O., Hart, P.E., Stork, D.G. (2001) Pattern classification (2nd edition), Wiley, ISBN 0-471-05669-3
7. Duncan J. Watts and Steven H. Strogatz. Collective dynamics of small-world networks. Nature, 393:440–442, 06 1998.
8. 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.
9. Endy, D, Brent, R. (2001) Modelling Cellular Behavior, Nature 2001 Jan 18; 409(6818):391-5.
10. 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.
11. Fuchs J.J. (2004), On sparse representations in arbitrary redundant bases, IEEE Trans. on IT, June 2004.
12. 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
13. Goldbeter A (2002) Computational approaches to cellular rhythms. Nature 420, 238-45
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. Guthke R., M¨oller U., Hoffmann M., Thies F., T¨opfer S., 2004, Dynamic network reconstruction from gene expression data applied to immune response, Bioinformatics, 2004, pp 2261
16. 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.
17. Jong de H., Gouze J.L., Hernandez C., Page M., Sari T., Geiselmann J., Qualitative simulation of genetic regulatory networks using piecewise-linear models, Bull Math Biol. 2004 Mar;66(2): pp 301–40.
18. Jong de H., Modelling and Simulation of Genetic Regulatory Systems: A Literature Review, Journal of Computational Biology, 2002, Volume 9, Number 1, pp. 67–103
19. Kampen van N. G. (1992), Stochastic Processes in Physics and Chemistry, Elsevier ScienceB. V., Amsterdam, (1992).
20. Makoto Hirahara, Natsuki Oka and Toshiki Kindo (1997), Associative memory with a sparse encoding mechanism for storing correlated patterns, Neural Networks, Volume 10, Issue 9, December 1997, Pages 1627-1636

21. Masahiko Morita (1996), Memory and Learning of Sequential Patterns by Nonmonotone Neural Networks, Neural Networks, Volume 9, Issue 8, November 1996, Pages 1477-1489

22. Masato Okada (1996), Notions of Associative Memory and Sparse Coding, Neural Networks, Volume 9, Issue 8, November 1996, Pages 1429-1458

23. Mazza Christian (1997), On the Storage Capacity of Nonlinear Neural Networks, Neural Networks, Volume 10, Issue 4, June 1997, Pages 593-597 .

24. Mestl T, Plahte E, Omholt SW. A mathematical framework for describing and analysing gene regulatory networks. J Theor Biol. 1995 Sep 21;176(2):291-300.

25. Newman M. E. J.. Power laws, pareto distributions and zipf 's law. Contemporary Physics, 46:323, 2005.

26. Novak B, Tyson JJ (1997) Modelling the control of DNA replication in fission yeast, PNAS, USA, Vol. 94, pp. 9147-9152, August 1997.

27. Omholt SW, Kefang X, Andersen O, Plahte E. Description and analysis of switchlike regulatory networks exemplified by a model of cellular iron homeostasis. J Theor Biol. 1998 Dec 7;195(3):339-50.

28. Omholt SW, Plahte E, Oyehaug L, Xiang K. Gene regulatory networks generating the phenomena of additivity, dominance and epistasis. Genetics. 2000 Jun;155(2):969-80.

29. 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

30. Plahte E, Mestl T, Omholt SW. A methodological basis for description and analysis of systems with complex switch-like interactions. J Math Biol. 1998 Mar;36(4):321-48.

31. Rosenfeld N, Young JW, Alon U, Swain PS, Elowitz MB, Gene regulation at the single-cell level, Science 307 (2005) pp 1962.

32. Schuppen van J.H. (2004), System theory of rational positive systems for cell reaction networks, CWI Report MAS-E0421, December 2004, ISSN 1386-3703

33. 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.

34. Spieth Christian, Streichert Felix, Speer Nora and Zell Andreas (2004), Iteratively Inferring Gene Regulatory Networks with Virtual Knockout Experiments, Lecture Notes in Computer Volume 3005, Science, Springer Publisher Berlin / Heidelberg, ISSN 0302-9743, pp 104-112.

35. 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.

36. Swain P.S., Efficient attenuation of stochasticity in gene expression through posttranscriptional control, J Mol Biol 344 (2004) pp 965.

37. Swain P.S., Elowitz MB, Siggia ED, Intrinsic and extrinsic contributions to stochasticity in gene expression, PNAS 99 (2002) pp 12795.

38. Tegn´er J., Yeung M.K.S., Hasty J., Collins J.J., Reverse engineering gene networks: Integrating genetic perturbations with dynamical modelling, Proc. Nat. Acad. Science, vol. 100, no. 10, 2003, pp. 5944–5949.

39. Verdult V., Verhaegen M., Subspace Identification of Piecewise Linear Systems, In Proc. 43rd IEEE Conference on Decision and Control (CDC), pp 3838–3843, Atlantis, Paradise Island, Bahamas, December 2004.

40. Wasserman, Philip (1993) Advanced Methods in Neural Computing, Van Nostrand Reinhold, ISBN 0-442-00461-3

41. Westra R.L., G. Hollanders, R.L.M. Peeters (2006a), 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

42. Westra R.L., G. Hollanders, R.L.M. Peeters, K. Tuyls (2006b), Piecewise Linear Modeling of Gene-Protein Interaction Networks, KDECB'06: May 10th 2006, Ghent, Belgium

43. Westra R.L., G. Hollanders, G-J. Becx, M. Gyssens, K. Tuyls (2007a), Knowledge discovery and emergent complexity in bio-informatics, Springer Lecture Notes in Bioinformatics (LNBI) Volume Number 4366, January 2007

44. Westra R.L., G. Hollanders, G-J. Becx, M. Gyssens, K. Tuyls (2007b), The Identification of Dynamic Gene-Protein Networks, Springer Lecture Notes in Bioinformatics (LNBI) Volume Number 4366, January 2007

45. Yeung M.K.S., Tegn´er 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.