Static and dynamic models of biological networks

Ashish Bhan

Institute of Genomics and Bioinformatics

University of California, Irvine

Irvine, CA 92697

Phone: (949)-824-5693

email: abhan@uci.edu

Fax: (949)-824-4056

Eric Mjolsness

Institute of Genomics and Bioinformatics and

Dept. of Information and Computer Science

UCI ICS TR# 05-15

November 14, 2005

1 Abstract

We consider static and dynamic approaches to the specification of probability distributions on graphs, consistent with desired statistical properties such as degree distributions, for use in modeling biological networks. In the static approach we develop analytical approximations to the Hamiltonian and partition functions. In the dynamic approach, we use a stochastic parameterized grammar to construct an evolutionary tree in which the nodes represent elements such as genes or cells and the links represent inheritance relations between the nodes. The grammar then constructs a network based on the feature vectors of the nodes in the tree.

2 Introduction: Static or equilibrium models

There has been a great deal of interest in the structure of different kinds of networks over the past few years. Some of these networks like the Internet, the World Wide Web, social networks and scientific citation networks are man-made while others like metabolic networks, protein interaction networks and gene regulatory networks are the products of evolution [1]. Despite their different origins, many of these networks share interesting structural features. If $N(k)$ is the number of nodes of degree k i.e. with k immediate neighbors and N is the number of nodes in the network, then $P(k) = \frac{N(k)}{N}$ represents the fraction of nodes of degree k and is the probability distribution corresponding to the degree distribution of the network.

For many real networks, the degree distribution scales as a power law or $P(k) \sim k^{-\gamma}$ where $2 < \gamma < 3$. There has been a lot of research related to developing models of network growth that produce such power-law networks [2] [3] [4] [5] [6] [7].

There have also been some attempts to use the methods of statistical mechanics to study the properties of real and artificially generated networks. In [8], the authors use the ideas of Gibbs free energy, graph Hamiltonians and partition functions to develop an alternative model of network analysis. This powerful approach allows one to estimate various parameters of the network (the average degree, for example) by taking appropriate derivatives of the partition function. It also allows one to reformulate the basic properties of networks in a seemingly unrelated setting that promises to offer further insights into the structure of real networks. In Sections 3-5 we will use some tools from statistical mechanics to propose an alternative approach to constructing networks with desired degree distributions. The basic idea here is to use a generalized energy function to apply constraints on the structure of a network. Instead of proposing a specific dynamic model of network formation, we use the energy formalism to estimate a static set of fugacity (a measure of attractiveness or popularity of a node in the network) values which can be used to generate the network using a simple sampling procedure. The fugacity of a node is directly proportional to the likelihood that an edge in the network is connected to the node. In Sections 6-9 we use stochastic parametrized grammars to construct evolutionary trees based on grammar rules that offer a very general and flexible approach to specifying tree structure. We then construct

graphs or networks based on the evolutionary trees based on the features of the nodes and some similarity measure. The feature vectors and similarity measures here can be chosen in a variety of ways and this could provide a useful approach to modeling complex biological networks.

3 Statistical Mechanics

Using the idea of a Boltzmann distribution defined on the adjacency matrix of a network G, we can define an energy function

$$E_{degree}(G) = \sum_{ij} \mu_i G_{ij} + \sum_i f(\sum_j G_{ij}) \tag{1}$$

where G is a symmetric adjacency matrix. The values of μ_i and the functional form of f are chosen to generate networks with desired properties associated with the degree distribution represented in the adjacency matrix G. The partition function associated with this energy is

$$Z(\mu) = \sum_G e^{-\sum_i \mu_i (\sum_j G_{ij}) - \sum_i f(\sum_j G_{ij})} \tag{2}$$

Eqn(2) can be rewritten as an integral in the form

$$Z(\mu) = \sum_G \int d\lambda_1 \dots d\lambda_N \{\prod_i \delta(\lambda_i - \sum_j G_{ij}) e^{-\sum_i \mu_i \lambda_i} e^{-\sum_i f(\lambda_i)}\} \tag{3}$$

which is equivalent to

$$Z(\mu) = \sum_G \int d\lambda_1 \dots d\lambda_N \{\prod_i \delta(\lambda_i - \sum_j G_{ij}) e^{-\sum_i \mu_i \lambda_i}\} \prod_i e^{-f(\lambda_i)} \tag{4}$$

In order to simplify the integral in Eqn(4), we introduce the function ϕ which is related to f as a Laplace transform in exponential form

$$e^{-f(\lambda)} = \int_0^\infty d\nu e^{-\nu\lambda - \phi(\nu)} \tag{5}$$

This allows us to rewrite Eqn(4) as

$$Z(\mu) = \sum_G \int d\lambda_1 \dots d\lambda_N \{\prod_i \delta(\lambda_i - \sum_j G_{ij}) e^{-\sum_i \mu_i \lambda_i}\} \prod_i \int_0^\infty d\nu_i e^{-\nu_i \lambda_i - \phi(\nu_i)} \tag{6}$$

which is equivalent to

$$Z(\mu) = \sum_G \int d\lambda_1 \dots d\lambda_N \{\prod_i \delta(\lambda_i - \sum_j G_{ij}) e^{-\sum_i \mu_i \lambda_i} \int_0^\infty d\nu_i \dots d\nu_i e^{-\nu_i \lambda_i - \phi(\nu_i)}\} \tag{7}$$

Rearranging terms we get

$$Z(\mu) = \sum_G \int_0^\infty d\nu_1 \ldots \int_0^\infty d\nu_N \int d\lambda_1 \ldots \int d\lambda_N \prod_i \delta(\lambda_i - \sum_j G_{ij}) e^{-\sum_i \mu_i \lambda_i - \sum_i \nu_i \lambda_i - \sum_i \phi(\nu_i)} \tag{8}$$

which allows us to change variables back and write Eqn(8) as

$$Z(\mu) = \sum_G \int_0^\infty d\nu_1 \ldots \int_0^\infty d\nu_N e^{-\sum_i \mu_i (\sum_j G_{ij}) - \sum_i \nu_i (\sum_j G_{ij}) - \sum_i \phi(\nu_i)} \tag{9}$$

We now use the change of variables

$$z_i = e^{-\mu_i}, c_i = e^{-\nu_i}, z_0 = c_0 = 1, \tag{10}$$

$$Z(z) = \sum_G \int_0^\infty d\nu_1 \ldots \int_0^\infty d\nu_n \prod_{i=1}^N (c_i z_i)^{\sum_j G_{ij}} e^{-\sum_i \phi(\nu_i)} \tag{11}$$

Using the relation

$$e^{-f(\lambda)} = \int_1^\infty dc \quad c^{-\lambda} \psi(c) \tag{12}$$

Eqn(12) can be rewritten as

$$Z(z) = \sum_G \int_0^1 dc_1 \ldots \int_0^1 dc_N \prod_{i=1}^N (c_i z_i)^{(\sum_j G_{ij})} \prod_{i=1}^N \psi(c_i) \tag{13}$$

$$Z(z) = \sum_G \int_0^1 dc_1 \ldots \int_0^1 dc_N \prod_{i=1}^{N} (c_i z_i)^{G_{ij}} \prod_{1 \le i \le j \le N} (c_i c_j z_i z_j)^{G_{ij}} \prod_{i=1}^{N} \psi(c_i) \tag{14}$$

Note that Eqn(14) allows one to use a direct sampling method to estimate the energy function from the relation

$$P(z) = \int \int dz_1 dz_2 \delta(P - \frac{z_1 z_2}{1 + z_1 z_2}) \psi(z_1) \psi(z_2) \tag{15}$$

4 Simulations: Static models

In this section we try to use the formalism outlined above to generate networks with desired degree distributions. In the case when the desired degree distribution is a power law, we can use the Laplace transform pair

$$L[\nu^{k-1} e^{-a\nu}] = \Gamma(k)(\lambda + a)^{-k} \tag{16}$$

in combination with the relationship in Eqn(5) to give

$$\phi(\nu) = a\nu - (k-1)\log\nu \rightarrow f(\lambda) = k\log(\lambda + a) - \log(\Gamma(k)) \tag{17}$$

Further, using the relation in Eq(12), we obtain

$$\psi(c) = c^{a-1}(-\log c)^{k-1} \tag{18}$$

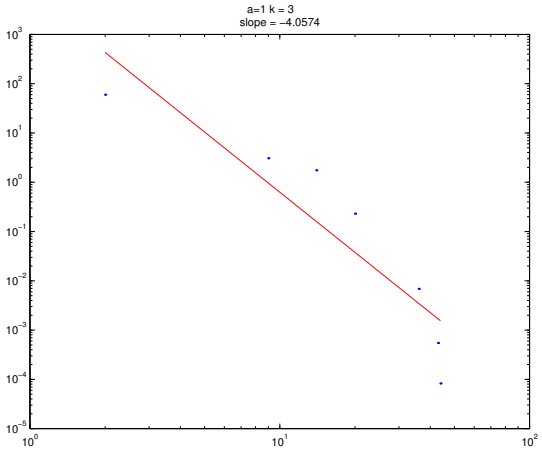

Figure 1: Fix k=3, a = 1

We now use the relation in Eqn(15) to estimate $P(z)$ for various values of a and k of interest from the study of real world networks. For many real networks $P(k) \sim k^{-\gamma}$ where $2 < \gamma < 3$. We use the function $\psi(c)$ with $k = 3, a = 1$ and the formula in Eqn(15) to generate a set of $P(z)$ values which we plot on the log-log scale in Fig.1. Note that the function is linear on the log-log scale and represents a power-law of the form predicted by the parameter choices above. The power-law relation is exhibited over a limited range of parameter values and the decaying tail may be to due to finite-size effects.

In Fig.2 below, we fix $a = 0$ and vary k to obtain four different simulations for $P(z)$ when $N = 100$. The resulting power-law distributions (plotted on the log-log scale) have exponentially decaying tails and this may be due to finite-size effects. Note that the scaling exponent in the power law matches the predicted value very well.

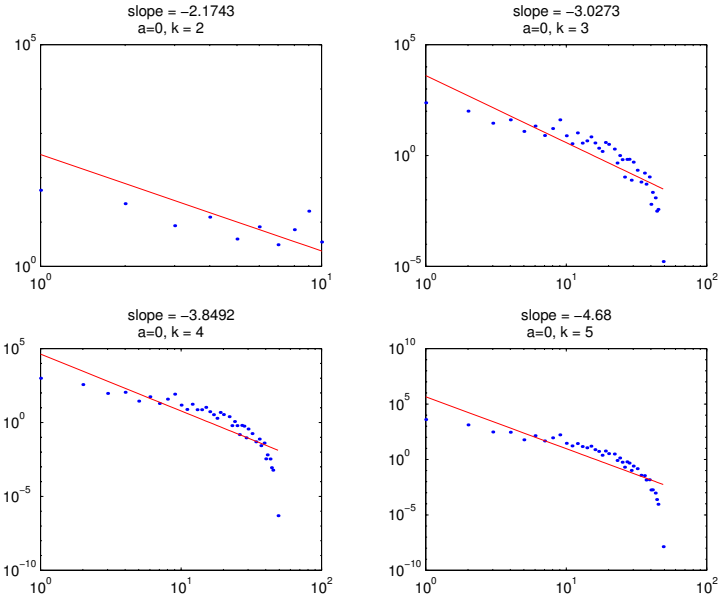

Figure 2: Fix k, vary a

5 Discussion: Static models

In [8], Newman suggested a class of Boltzmann distributions including higher order correlations between the degrees of related nodes. There has been more recent interest in approaches of this sort in [9]. Quite a variety of algebraic forms have been suggested for such additional terms, with highly variable cost of simulation on fixed architecture machinery as a function of graph sparsity and other statistics. On the other hand we have shown how one family of objective function terms (Eqn 1) can be represented in terms of another family that requires less graph communication (Eqn 9), at the cost of introducing new variables at each node i, in order to enforce a desired degree distribution. This observation raises the question: what is a minimal set of terms in the energy function E, that can represent other energy

functions which depend on any desired set of statistics of the network topology i.e. into which such terms can be mapped by adding or removing node variables? This would be a form of *universality* for such energy functions on graphs. Our suggestion for such a universal energy function is the "Equilibrium Graph Automaton", with node label vectors $\vec{x_i}$ of the form

$$E(\vec{x}, G) = C_0 \sum_i F_1(\vec{x_i}) + C_1 \sum_{ij} G_{ij} F_2(\vec{x_i}, \vec{x_j}) + C_2 \sum_{ij} G_{ij} G_{ji} + C_3 \sum_{ijk} G_{ij} G_{jk} G_{ki} \qquad (19)$$

The $\vec{x_i}$ node state vectors can be used to transmit fixed-length messages (C_1 term) and to implement other sparse graph interactions of sufficiently low cost. The C_0 and C_1 terms are of sufficient generality to encompass the degree and sparseness constraint of Eqn(9), for example. The final C_3 term of the sum above involving $\sum_{ijk} G_{ij} G_{jk} G_{ki}$ makes a contribution to the energy from any loop that begins at i and passes through j and k to end at i again. By analogy with email, one can imagine the information being passed along this loop as the email addresses in a forwarded email from i to j, and enables k to send information back to both j (under term C_2) and k (under term C_3), but no further in one interaction. More generally, changes to the $F1$ and $F2$ functions, and to the dimension of the parameter vector $\vec{x_i}$, in the very local interaction expression above may allow one to emulate the effect of a

wide variety of other energy function terms such as

$$\sum_{\{i_1 \ldots i_n\}} \prod_{ab} G_{i_a i_b}^{g_{ab}} \tag{20}$$

which favors or disfavors particular network motifs with 0/1 valued adjacency matrix g_{ab}. But emulating (20) in terms of (19) is an open question.

As a step towards Equilibrium Graph Automata implementations of message-passing protocols, consider the "waterfall" energy function

$$\begin{aligned} E_W(x, y, G) &= \frac{1}{2} \sum_{ia} \left(x_i^{(a0)} - \xi_i \right)^2 + \frac{\epsilon}{2} \sum_{ij} G_{ij} \sum_{\tau=0}^{T} \\ &\quad \left[\epsilon^{2\tau} \sum_{ab} W_\tau^{(ba)} (x_i^{(a\tau)} - y_j^{(b\tau)})^2 + \epsilon^{2\tau+1} \sum_{ab} \hat{W}_\tau^{(ab)} (x_i^{(a\tau+1)} - y_j^{(b\tau)})^2 \right] \end{aligned} \tag{21}$$

Clearly this falls within the form of the C_0 and C_1terms of the EGA energy. We assume other $O(\epsilon^0)$ EGA terms (such as those studied above) ensure a roughly constant fanin $\sum_j G_{ij}$ and fanout $\sum_j G_{ji}$ for G at each site i. For small ϵ and very low temperatures, probability is concentrated on low-energy states which satisfy the following T-round protocol of communications between sites i and j:

$$\partial E_W/\partial x_i^{(a0)} \approx 0 \quad \Rightarrow \quad x_i^{(a0)} \approx \xi_i$$

$$\partial E_W/\partial y_j^{(b\tau)} \approx 0 \quad \Rightarrow \quad y_j^{(b\tau)} \approx \frac{\sum\limits_{jb} G_{ij} W_\tau^{(ba)} x_i^{(a\tau)}}{\left(\sum\limits_j G_{ij}\right)\left(\sum\limits_b W_\tau^{ab}\right)}, \quad 0 \le \tau \le T$$

$$\partial E_W/\partial y_j^{(b\tau)} \approx 0 \quad \Rightarrow \quad x_i^{(a\tau+1)} \approx \frac{\sum\limits_{ia} G_{ij} \hat{W}_\tau^{(ab)} y_j^{(b\tau)}}{\left(\sum\limits_i G_{ij}\right)\left(\sum\limits_a \hat{W}_\tau^{(ab)}\right)}, \quad 0 \le \tau \le T$$

in which the weight matrices W_τ^{ba} and $\hat{W}_\tau^{ab}$ act as codes for linearly superimposed messages sent back and forth at each step, starting with the fixed initial messages ξ_i. Although the graph G influences the messages x_i and y_i, the reverse is not true for E_W because of the small parameter ϵ . Finally if we replace G by $G^{(\tau)}$ in the expression for E_W, then we find the "clocked objective function" [10]

$$\begin{aligned} E_{3-\text{phase}}(x,y,z,G) \;=\; & \\ & \frac{1}{2}\sum_{ia}(x_i^{(a0)} - \epsilon_i)^2 + \epsilon \sum_{\tau=0}^{T} [\epsilon^{3\tau} E_{\text{EGA}}(z^{(\tau)}, G^{(\tau)}) + \epsilon^{3\tau} \sum_{ij} g(G_{ij}^{(\tau)}, x_i^{(\tau)}, x_j^{(\tau)}) \\ & + \frac{1}{2}\sum_{ij} G_{ij}^{(\tau)} [\epsilon^{3\tau+1} \sum_{ab} W_\tau^{(ba)} (x_i^{(a\tau)} - y_j^{(b\tau)})^2 + \\ & \epsilon^{3\tau+2} \sum_{ab} \hat{W}_\tau^{(ab)} (x_i^{(a\tau+1)} - y_j^{(b\tau)})^2]] \end{aligned} \tag{22}$$

This permits two-way interaction between the coded messages x_i^τ and the graph G, again at

low temperature. However, this formulation is *not* itself in the form of an Equilibrium Graph Automaton because of the extra index on G. Instead it is equivalent to a non-equilibrium dynamical system with τ playing the role of time.

6 Introduction: Dynamic models

There is a great deal of interest in formulating probabilistic models of complex dynamical systems. There is a basic distinction in the description of these time-varying systems between those with static and dynamic structure. From the graph theory point of view, the distinction corresponds to that between dynamics of variables related by a fixed graph, and dynamics that governs a varying graph connectivity structure as well as the values of variables defined on its nodes. Examples in scientific modeling include the production of particular molecules as a result of chemical reactions, and their dynamic association into loosely bound multimolecular complexes or their enclosure within compartments. An example from cell biology is the birth and death of cells [11], and their spatially contingent mechanical and signaling relations within a developing organism (morphogenesis) [12]. In these systems, there is a need to dynamically evolve the topology of the network (cells are born and they grow and divide and modify their neighborhood connections as a result of division) and to update the features of the nodes (states of cells or genes) [11]. Dynamic-structure systems are more difficult to model and infer than are systems with static structure. The existence

of domain objects, and their relationships, vary over time; typically this means that mathematical equations expressing such models change over time or take a more elaborate form than do static-structure systems. In addition, variable-structure systems can grow infinitely large, though intrinsic resource constraints can be included to prevent this from occurring. In this paper, we consider a simple *stochastic parametrized grammar* (SPG) as a dynamic approach to reconstructing features of evolving, biological networks. The formal semantics of SPG's in general, and their application to graph grammars in particular, are specified in terms of stochastic processes in [13].

7 Lineage Tree and Graph Grammar Models

In this approach we consider a simple lineage tree of nodes bearing feature vectors. The tree has a fixed, arbitrary distribution $q(n)$ on the number of children at each node. The tree consists of feature vector nodes with only local dependencies between the features of child and parent feature vectors. This model incorporates a birth-and-death process [14] for the tree nodes and has applications to hierarchical clustering in $\Re^d$, cell lineage trees with a state vector for each cell, and evolutionary phylogeny trees with a genotype (e.g. a discrete sequence string in Z_2^d or Z_4^d) for each species. It can also serve as a scaffold structure (e.g. a cell lineage tree) for many other more complex dynamic-structure systems. A tree of feature vectors is "context-free" in our terminology if it can be generated by a context-free stochastic grammar, i.e. one in which each rule has only one term on the left hand side. In

the grammar below, this means that each node in the tree has a feature vector that is conditionally independent of all others except for the feature vector of its parent node, and the number of child nodes is conditionally independent of everything else except the existence of the parent node within the tree. The grammar that we outline below operates on four kinds of objects: "node" objects (the final output tree nodes), "nodeset" objects (a single one of which is the input nodeset object), and intermediate "child" objects that eventually become nodesets that give rise to nodes and more children and "OIDgen" objects that label the nodes as they are created. The OID's are only needed for the context-sensitive *graphgen* grammar that follows; the features themselves can be generated in a context-free manner. The grammar that constructs the tree is given below

grammar (discrete-time) *nodegen* (start(x) $\rightarrow$ node(k, x)}) {

start(x) $\rightarrow$ nodeset(x), OIDgen(1)

nodeset(x), OIDgen(k) $\rightarrow$ node(k, x), {child}, OIDgen($k+1$)

child(y) $\rightarrow$ nodeset($x, l+1$) **with** $\phi(x|y)$

}

In the syntax above, "**grammar**" is a keyword that declares what follows to be a grammar; "*nodegen*" is the name of the grammar; start(x) $\rightarrow$ { node(k, x)} specifies that the input of the grammar is a single start object and its output is a set of node objects each with

a value for a random variable x_i (arbitrarily numbered). Thus the grammar implements a single rule which could be invoked recursively by this or any other grammar. This grammar implements a branching stochastic process which starts with a "node" with feature vector x, and generates a set of terminal "fnode" terms with probability 1. In the simplest case we consider, $q(n)$ is a geometric distribution of the form p^n. Each child changes its real-valued feature vector x according to the conditional distribution ϕ. We consider the simple case where the feature vector takes the form $x = (\mu, \sigma)$ and $y = (\hat{\mu}, \hat{\sigma})$ and the function ϕ is of the form $\phi = G(|x - y||\hat{\sigma})\delta(\sigma, \hat{\sigma}/\sigma_r)$. With *clustergen*, we have thus defined the context-free feature tree family. This would imply that the child of a particular node would be assigned a feature vector from a Gaussian distribution centered at the feature vector of its parent node. The variance of the Gaussian distribution depends on the level of the parent node and decreases by a constant factor σ_r at each level. In our implementation of this grammar, a node can give rise to n children with probability $q(n) = p^n$ where p is some fixed constant between 0 and 1. Once a tree has been created by the grammar above, we use the grammar *graphgen* below to construct a graph based on the evolutionary information in the tree.

grammar (discrete-time) *graphgen* { start $\rightarrow$ { link$(k_i, k_j$ } } {

start $\rightarrow$ { node } **via** *nodegen*

node(k_1, x_1), node $(k_2, x_2) \rightarrow$ node(k_1, x_1), node (k_2, x_2), link(k_1, k_2) **with** $f(|x_1 - x_2|)$

node$(i, x) \rightarrow$ fnode (i, x)

}

The grammar above assigns a "link" object connecting the nodes k_1 and k_2 based on the value of $f(|x_1 - x_2|)$. As a simple example, if the value of $|x_1 - x_2|$ is less than a threshold, a link could be formed between nodes k_1 and k_2. We then analyze the properties of the resulting network and try to adjust the parameters of q, ϕ and f to construct networks that display some topological features of biological networks like protein-interaction networks or metabolic networks.

Biological motivations for this kind of graph grammar include: (1) mechanical interactions between cells are influenced by similarity in their position (adjacency) [11] and complementary adhesion and regulatory interactions between vectors of membrane-bound molecules; (2) molecular recognition of targets in axonal guidance and synapse formation, which generate neuronal networks; (3) the empirical description of subgraph frequency in protein-protein interaction data in terms of embedding in low dimension spaces [15]; and (4) models of complementarity in "shape space" for molecular binding interactions [16].

8 Simulations: Dynamic models

In this section we run simulations of some of the grammars described above to try and construct networks that display some topological features of biological networks. We use the

geometric probability distribution $q(n) = p^n$ where $p = 0.7$. The root node is centered at the point $(0.5, 0.5)$ in $2-$dimensional Euclidean space. The position of the children of a node come from a Gaussian distribution with mean at the position of the parent node and with variance set at 0.1 for the first level and decreasing by a fixed factor at every subsequent level. The variance for the Gaussian distribution representing the position of the children of a node decreases at each level since we assume the variation decreases with each generation. The tree is grown till there are hundred nodes and two nodes are connected by an edge if the distance between them is less than a threshold $t = 0.002$. As Fig.3 indicates, increasing the factor by which the variance is reduced causes the network to become more clustered. We also plot the clustering coefficient as a function of the factor by which the variance is reduced in Fig.4 to make the relationship between the two quantities more evident.

9 Discussion: Dynamic models

The dynamic models proposed above are similar to Equilibrium Graph Automata, but in a non-equilibrium setting and without the two-way interaction between labels $\vec{x_i}$ and connections in G_{ij}. Many other recursive/stochastic patterns are possible with Stochastic Parametrized Grammars. For example, an alternative kind of graph grammar also uses a subgrammar to specify the cell lineage tree, but also uses recursive substitution rules to substitute subgraphs from a library for individual nodes and links, as illustrated by the following grammar "graph-recursion". The a indices are arbitrary "colors" in addition to the

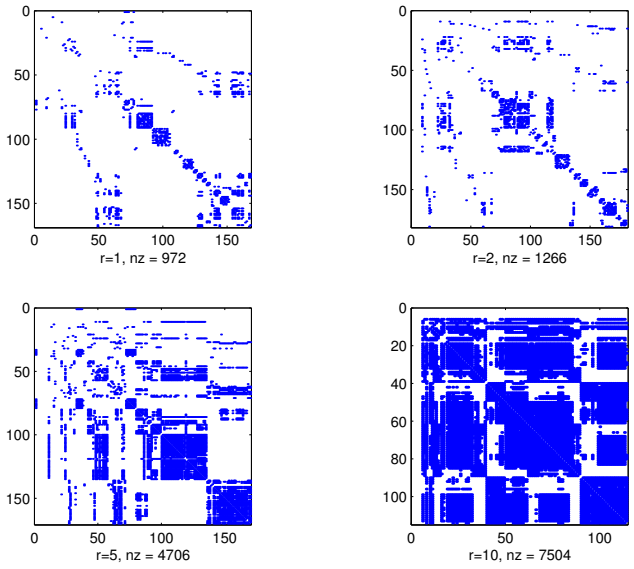

Figure 3: Effect of reducing the variance on the structure of networks

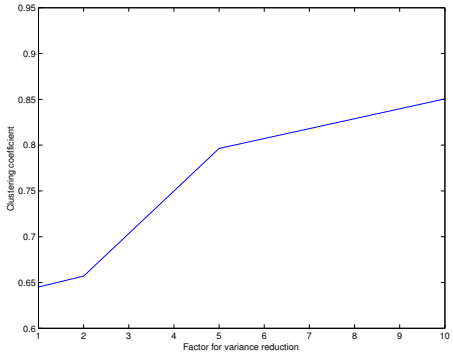

Figure 4: Clustering coefficient vs. Factor of variance reduction

node index sequences $\mathbf{i}$.

grammar (discrete-time) *graph-recursion* (start $\rightarrow$ {node($\mathbf{i}$), G-connection($a, \mathbf{i}, \mathbf{j}$) }) {

start $\rightarrow$ node((0)), $G - connection(1, (0), (0))$

N=node($\mathbf{i}$) $\rightarrow$ N=node($\mathbf{i}$), { node($(\mathbf{i}, i_n)$)$|A_{(\mathbf{i},i_n)} = 1^i_n < i_{max}$ }

under $E = \mu \sum_{(\mathbf{i},i_n)} A_{(\mathbf{i},i_n)}$

G-connection($a, \mathbf{i}, \mathbf{j}$), N=node($(\mathbf{i}, i_n)$), M=node($(\mathbf{j}, j_n)$)

$\rightarrow$ { G-connection($b, (\mathbf{i}, i_n), (\mathbf{j}, j_n)$)$|G^{ab}_{i_n j_n} = 1$ }, N, M

}

In this grammar, boldface indices $\mathbf{i}$ etc. correspond to (e.g. binary) index tuples (i_1, i_2, i_{n-1}) and $(\mathbf{i}, i_n)$ denotes the tuple extension to $(i_1, i_2, ...i_n)$. Also $A_{(\mathbf{i},i_n)}$ is a set of 0-1 valued indicator variables representing the existence of particular indexed nodes as determined for example by *nodegen* .

In this way, $G^{ab}_{i_n j_n}$ acts as a reusable wiring pattern or "cable" or, when $\mathbf{i} = \mathbf{j}$, as a reusable subgraph. The corresponding graph adjacency matrix recursion relation is

$$G^{a_0}_{(i_1 \ldots i_L)(j_1 \ldots j_L)} = \sum_{\{a_l\}} \prod_{l=1}^{L} (G^{(a_{l-1} a_l)}_{(i_l j_l)})^{A_{(i_1 \ldots i_L)} A_{(j_1 \ldots j_L)}} \tag{23}$$

Such recursion relations are very powerful ways to specify network structure and have been used within a successful machine learning search for functional graph architectures [17]. Whether and how simply they can also be translated into the more biological feature-

similarity style of graph grammars such as *graphgen* (in a manner analogous to our translation of degree distribution Hamiltonian terms into computations on node-level fugacity labels) is an open question.

References

[1] Albert, R. and Barabasi, A-L. Statistical mechanics of complex networks. Reviews of Modern Physics 2002, 74, pp 47-97.

[2] Barabasi, A-L. and Albert, R. Emergence of Scaling in Random Networks. Science 1999, 286, pp 509-512.

[3] Bhan,A.; Dewey, T. G. and Galas, D. A Duplication Growth Model of Gene Expression Networks. Bioinformatics 2002.

[4] Dorogovtsev, S.N. and Mendes, J.F.F. Scaling properties of scale-free evolving networks: Continuous approach. Physical Review E 2001, 63, 056125, pp 1-18.

[5] Dorogovtsev, S.N.; Mendes, J.F.F and Samukhin, A.N. Structure of Growing Networks: Exact Solution of the Barabasi-Albert's Model. Phys. Rev. Lett. 2000, 85, 4633.

[6] Krapivsky, P.L.; Rodgers, G.J. and Redner, S. Degree Distributions of Growing Networks. Phys. Rev. Lett. 2001, 86, 5401.

[7] Ravasz, E., Somera; A.L., Mongru; D.A., Oltvai, Z.N. and Barabasi, A-L. Hierarchical Organization of Modularity in Metabolic Networks. Science 2002, 297, 1551.

[8] Newman, M.E.J and Park, J. The statistical mechanics of networks. URL: http://aps.arxiv.org/abs/cond-mat/0405566/.

[9] Ziv, E.; Koytcheff, R. and Wiggins, C. Novel systematic discovery of statistically significant network features. Phys. Rev. E, In Press.

[10] Miranker, W, L. and Mjolsness, E. A Lagrangian formulation of neural networks II: Clocked objective functions and applications. Neural, Parallel and Scientific Computations 1998, 6 (3), pp 337-372.

[11] Fracchia, F. D. Integrating Lineage and Interaction for the Visualization of Cellular Structures. In: Graph Grammars and Their Application to Computer Science, Springer Verlag, Berlin, 1996, pp 521-535.

[12] Lück, J. and Lück, H.B. Cellworks with Cell Rewriting and Cell Packing for Plant Morphogenesis. In: Graph Grammars and Their Application to Computer Science, Springer Verlag, Berlin, 1996, pp 536-549.

[13] Mjolsness, E. Stochastic Process Semantics for Parameterized Grammar Syntax. Technical Report UCI ICS TR# 05-14, October 2005. URL: http://computableplant.ics.uci.edu/papers/#multiscalemodeling

[14] Athreya, K.B. and Ney, P.E. Branching Processes, Springer Verlag; Dover.

[15] Przulj, N.; Corneil, D. G. and Jurisica, I. Modeling Interactome: Scale-Free or Geometric?. Bioinformatics, 2004, 20(18), pp 3508-1515.

[16] Perelson, A. and Oster, G. Theoretical Studies of Clonal Selection: Minimal Antibody Repertoire Size and Reliability of Self-NonSelf Discrimination. J. Theor. Biol., 1979, 81, pp 645-67.

[17] Mjolsness, E.; Sharp, D.H., and Alpert, B.K. Scaling, machine learning, and genetic neural nets. Advances in Applied Mathematics, 1989.

Captions

Figure 1: Fix k=3, a = 1

Figure 2: Fix k, vary a

Figure 3: Effect of reducing the variance on the structure of networks

Figure 4: Clustering coefficient vs. Factor of variance reduction