Available online at www.sciencedirect.com
ScienceDirect
Journal of Theoretical Biology 247 (2007) 331–336
Journal of Theoretical Biology
www.elsevier.com/locate/yjtbi

Do scale-free regulatory networks allow more expression than random ones?

Miguel A. Fortuna[a,*], Carlos J. Melián[a,b]

[a]Integrative Ecology Group, Estación Biológica de Doñana, CSIC, Avda. Ma Luisa s/n, 41013, E-41080 Sevilla, Spain
[b]National Center for Ecological Analysis and Synthesis, University of California, 735 State St., Suite 300, Santa Barbara, CA 93101, USA

Received 23 October 2006; received in revised form 13 March 2007; accepted 13 March 2007
Available online 21 March 2007

Abstract

In this paper, we compile the network of software packages with regulatory interactions (dependences and conflicts) from Debian GNU/Linux operating system and use it as an analogy for a gene regulatory network. Using a trace-back algorithm we assemble networks from the pool of packages with both scale-free (real data) and exponential (null model) topologies. We record the maximum number of packages that can be functionally installed in the system (i.e., the active network size). We show that scale-free regulatory networks allow a larger active network size than random ones. This result might have implications for the number of expressed genes at steady state. Small genomes with scale-free regulatory topologies could allow much more expression than large genomes with exponential topologies. This may have implications for the dynamics, robustness and evolution of genomes.

Keywords: Complex networks; Gene networks; Network assembly; Regulatory interactions; Transcriptional regulatory networks

1. Introduction

In the last years an increasing number of systems have been described as networks (i.e., a set of nodes connected between them by links) and represented as graphs (e.g., Strogatz, 2001; Albert and Barabási, 2002; Newman, 2003). Physical and social systems such as the World Wide Web (Albert et al., 1999; Huberman and Adamic, 1999), the Internet (Doyle et al., 2005), the worldwide air transportation network (Guimerá and Amaral, 2004; Guimerá et al., 2005), networks of acquaintance or other connections between individuals (Newman et al., 2002; Liben-Nowell et al., 2005), scientific collaboration networks (Newman, 2001; Barabási et al., 2002), and the network of human sexual contacts (Liljeros et al., 2001) are all examples of different systems studied under the network approach.

In addition, biological systems such as food webs (Paine, 1966; Cohen, 1978; Pimm, 1982), plant–animal mutualistic networks (Bascompte et al., 2003; Jordano et al., 2003), metabolic networks (Jeong et al., 2000; Ravasz et al., 2002), protein networks (Jeong et al., 2001; Giot et al., 2003; LaCount et al., 2005), and gene regulatory networks (Davidson et al., 2002; Luscombe et al., 2004), have also been explored using graph-theory methods. Perhaps the most challenging of such biological networks is that governing gene expression in a cell.

In a genome, thousands of genes direct the formation of proteins, including transcription factors that can activate or inhibit the transcription of genes to give mRNAs. Since these transcription factors are themselves products of genes, the ultimate effect is that genes regulates each other's expression as a part of gene regulatory networks (Davidson, 2001; Guelzim et al., 2002; Lee et al., 2002; Albert, 2005). The patterns of regulatory interactions at genomic scale (in which genes can affect each other's expression) are becoming increasingly resolved (Davidson et al., 2002; Guelzim et al., 2002; Lee et al., 2002; Stuart et al., 2003; Luscombe et al., 2004).

Recent evidence from whole-genome sequence suggests that organismal complexity arises much more from the elaborate regulation of gene expression than by the genome size itself (Knight, 2002; Levine and Tjian, 2003). In this context, previous results on small subsets of genes (Albert

*Corresponding author. Tel.: +34 954 23 23 40; fax: +34 954 62 11 25.
E-mail address: fortuna@ebd.csic.es (M.A. Fortuna).

0022-5193/$ - see front matter © 2007 Elsevier Ltd. All rights reserved.
doi:10.1016/j.jtbi.2007.03.017

and Othmer, 2003) have shown that the robustness of the network is depending on the topology (i.e., the distribution of the number of interactions a gene participates in) and the signature of regulatory interactions (i.e., whether the interaction activates or inhibits a gene). The effects of the topology of regulatory interactions on gene expression in large networks are, however, difficult to asses because the interaction signature is only known for a small subset of genes (Davidson et al., 2002; Guelzim et al., 2002; Lee et al., 2002; Albert and Othmer, 2003; Luscombe et al., 2004, see, however, Madan Babu et al., 2006).

2. Methods and results

In the present study we compiled the network of software packages of Debian GNU/Linux operating system along with their dependence and conflict interactions with the aim of shedding some light on the effect of the regulatory network structure on the number of active transcriptors. The interactions between software packages we consider to be regulatory interactions in the sense that they may or may not allow the installation of packages in the system. On the one hand, the package i depends (k^{dep}) of the package j when j has to be installed for i work (i.e., j activates i because i needs j to work). On the other hand, the package i has a conflict (k^{con}) with the package j when i does not work if j is installed in the system (i.e., j inhibits i). It does not necessarily mean that the package j also has a conflict with the package i (sometimes the package j is an improved version of the package i in a way that if i is already installed in the system then j improves it, but if j is installed then it already contains i and the later cannot be installed). Because links are directed we can find packages with ingoing and outgoing links (k_{in} and k_{out}, respectively). In a detailed picture of the network, we can identify all node types as a function of their k_{in} and k_{out} interactions. On the one hand, packages with just $k_{in}^{dep}>0$, packages with just $k_{in}^{con}>0$, and packages with both $k_{in}^{dep}>0$ and $k_{in}^{con}>0$, if they depend or have a conflict with other packages, or both, respectively. On the other hand, packages with $k_{out}^{dep}>0$, packages with $k_{out}^{con}>0$, and packages with $k_{out}^{dep}>0$ and $k_{out}^{con}>0$, if other packages depend or enter into conflict with them, or both, respectively.

To clarify the relationship between a regulatory gene network and the dependence network of software packages we must simplify the former. A gene network has two types of nodes, which correspond to transcription factors and the genes encoding them, and two types of directed links, which correspond to transcriptional regulation and translation (Lee et al., 2002). For simplicity, transcription factors are often combined with the genes encoding them (thus all nodes correspond to genes), and transcription and translation are condensed to one link (the assumption being if any of both processes happens, the other occurs too; see Albert, 2005). The nodes representing target genes that do not encode transcription factors become sinks (the above described packages with $k_{out}=0$) while non-transcriptionally regulated transcription factors correspond to sources ($k_{in}=0$). If the gene i encodes a transcriptional factor that activates the transcription of the mRNA of the gene j it will be said that the gene i activates the gene j, and if the gene i encodes a transcriptional factor that inhibits the transcription of the mRNA of the gene j it will be said that the gene i inhibits the gene j. These types of regulatory interactions are quite analogous to dependences k^{dep} and conflicts k^{con} in the network of software packages. Hence, if a gene i has $k_{in}^{dep}>0$ interactions it means that a k number of genes are needed to activate it. In the same way, if a gene i has $k_{out}^{dep}>0$ interactions it means that the gene i encodes a k number of transcriptional factors that activate other genes. Similarly, inhibition is analogous to conflict, k^{con}.

Let us now assume that the rules governing the transcription of a gene are determined by a Boolean function of the state of its transcriptional activators and inhibitors (Kauffman, 1969; Albert and Othmer, 2003). Transcription will only begin if the activators are expressed and the inhibitors are not (Kauffman, 1969). The effect of transcriptional activators and inhibitors is never additive, but rather inhibitors are dominant. The states of the nodes evolve in discrete time steps under several rules to a steady state in all nodes (Albert and Othmer, 2003). Each steady state or fixed point has a specific number of active and inactive transcriptors. The total number of active genes in each steady state represents the active network size. After n replicates of the network, the frequency of each steady state represents the distribution of the active network size (see Li et al., 2004, table 1).

Although we have defined the similarities between transcriptional and dependence networks, we should point out that there are some particularities of gene networks that preclude a full comparison of the two types of networks. Specifically, the self-degradation processes, the complex dynamics of activator and repressor, and the feedback circuits in which some genes are embedded make a perfect comparison difficult. In the Boolean network model, and in real gene networks, in addition to fixed points, cyclic attractors may also exist (Kauffman, 1969). This is not the case for the dependence network of software packages, in which a steady state of installed packages is reached once no more packages can be installed without entering into conflict with the previously installed packages. Another important difference is that in the Boolean network model the set of genes that are expressed in the attractors may be very different from the set of genes that were originally expressed in the initial condition. In contrast, in the dependence network of software packages all the installed packages (expressed genes) are retained throughout time, so that at the end all the packages that were originally installed remain installed. The analogy we can obtain, however, is the similarity of the final states in both types of networks. The total number of active genes in gene networks or