

Convergent Evolutionary Paths in Biological and Technological Networks

Ricard V. Solé · Sergi Valverde ·
Carlos Rodriguez-Caso

Published online: 22 July 2011
© Springer Science+Business Media, LLC 2011

Abstract Technology seems to follow a different type of evolutionary dynamic when compared with biological systems. As pointed out by Francois Jacob, evolution takes place by means of extensive tinkering and does not foresee the future. Engineers will typically have a well-defined purpose and are not—in principle—constrained by the available technological constraints. However, the truth is that technological change shares much more than we might suspect with the patterns and processes displayed by evolution. Using case studies from both protein maps and large-scale software networks, we show that several key traits, such as scale-free structure and modularity, are shared by both man-made and biological evolving systems. Surprisingly, we find convergent evolution in several key features of software systems, indicating that strong constraints are at work. Such constraints force engineers to extensively reuse already constructed parts, thus de facto tinkering with their designs in a way similar to the duplication–diversification mechanism driving genome growth. The evolution of these systems reveals that well-defined patterns are obtained “for free.” Some

of them can be properly interpreted as technological spandrels.

Keywords Evolution · Tinkering · Evolvability · Complex networks · Spandrels

Introduction

At the end of his book *On the Origin of Species*, Charles Darwin used the following famous quote regarding the complexity of life:

There is grandeur in this view of life, with its several powers, having been originally breathed into a few forms or into one; and that, whilst this planet has gone cycling on according to the fixed law of gravity, from so simple a beginning endless forms most beautiful and most wonderful have been, and are being, formed.

Darwin’s fascination for life forms and how they evolve remains as fresh as when *Origin* was written (Barton et al. 2008). The diversity of designs that we can find in the natural world is indeed astonishing. How life has been able to cope with environmental challenges to find the appropriate solutions (on multiple scales) is one of the most interesting problems faced by current evolutionary theory. Such diversity and its evolutionary patterns are also present within the context of technology (Basalla 1988; Arthur 2010). Technological change implies the emergence of new forms of solving given problems, very often by means of improvements in previous designs and sometimes by combining available solutions in novel ways. As it occurs with the evolution of species, new technologies emerge and explode in

R. V. Solé (✉) · S. Valverde · C. Rodriguez-Caso
Complex Systems Lab (ICREA-UPF), Parc de Recerca
Biomedica de Barcelona, Dr Aiguader 88,
08003 Barcelona, Spain
e-mail: ricard.sole@upf.edu

R. V. Solé
Santa Fe Institute, 1399 Hyde Park Road,
Santa Fe, NM 87501, USA

R. V. Solé · S. Valverde · C. Rodriguez-Caso
Institut de Biologia Evolutiva, UPF-CSIC, Passeig de la
Barceloneta 37-49, 08003 Barcelona, Spain

diversity while others become obsolete and go extinct. Not surprisingly, the similarities between technological and biological evolution have generated a considerable literature and inspired evolutionary biologists to look into their parallels (Eldredge 2001).

Darwin's words refer mostly to the external patterns displayed by individuals within species. Such complexity and diversity only increased as scientists were able to look deep inside organisms and cells. Here again endless forms (and molecules) most beautiful were (and are being) found. But together with the view of nature as a diverse array of forms, there is the no less outstanding discreteness and order displayed by nature. As noted by the late Pere Alberch, "we are so bewildered by the diversity of nature that we often forget that the world could have been of a very different shape" (Alberch 1989). Such discreteness is tied to the problem of evolutionary convergence. Many examples illustrate the observation that some particular solutions (from eyes to e-sociality) seem to be re-discovered once and once again by evolution (Conway Morris 2003). Additionally, one view within complex systems science suggests that there are deep constraints to what is possible and thus that the discreteness of living is shaped (to some extent) by fundamental laws of mathematical nature. Some of these constraints are related to the structure of fitness landscapes (Kauffman 1993; Niklas 1994; McGhee 1999).

What general regularities can be found? Below the surface of diversity, there is one seemingly universal property of living structures. From the protein organization of functional/structural domains at the subcellular level to the organization of body plans, we observe modular patterns on different scales. Modularity is by far one of the most important and characteristic features of complex adaptive systems, and it pervades biological complexity (Raff 1996; Hartwell et al. 1999; Wagner et al. 2007; Kepes 2007; Pereira-Leal et al. 2006). Many cell functions are carried out by subsets of units that define functionally meaningful entities. Well-known examples include developmental modules (von Dassow et al. 2000; Solé et al. 2001; Solé et al. 2002a, b). Modularity allows efficient improvement of the adaptive potential of different functions with a small amount of interference from others (Wagner 1995; Calabretta et al. 2000; Lipson et al. 2002). The standard view of the evolution of modular architectures suggests two possible scenarios: either parcellation or integration. The first involves the differential elimination of cross-interactions involving different parts of the system. This is the strategy operating in eukaryotic organization. Membrane parcellation permits the existence of enzymatic processes that cannot occur at the same place,

as well as the generation of electrochemical potentials by selective exchange of ions across membranes. But it also plays key roles at higher scales, such as in brain organization (Sporns 2010) or even ecosystem structure (Olesen et al. 2007). It thus provides a powerful theoretical framework to address the problem of evolutionary hierarchies (Eldredge 1985).

The second instead proposes an alternative scenario where an initial system of interacting components formed by independent parts ends up displaying modularity because differential integration of those independent characters takes place. This scenario seems appropriate in describing the evolution of neural networks (Striedter 2005). In this case, optimization is a major driving force. Wiring and cost minimization, first proposed by Ramón y Cajal more than 100 years ago (Ramón y Cajal 1899), is clearly at work (Chen et al. 2006; Perez-Escudero and de Polavieja 2007). Similarly, the evolution of branching systems (West et al. 1997; Brown and West 2000; see also Durand and Weaire 2004; Durand 2006) associated with transport of fluids and gases in living organisms has been subject to optimization. However, as we will illustrate in this work, optimization is not necessarily the driving force shaping the topology of complex networks.

The origins of modularity in cellular systems can be explored by using complex networks (Albert et al. 2000; Albert and Barabasi 2002; Boccaletti et al. 2006; Bornholdt and Schuster 2003; Dorogovtsev and Mendes 2003). Within cells, interactions between metabolites, proteins, or genes define a web of molecular relations which have well-defined organization. Such patterns can be fully characterized and pose strong constraints on the predictions made by any theoretical model trying to explain the architecture of cellular maps. In this context, recent theoretical studies have shown that many of the topological properties displayed by these maps (including modularity) can be explained in terms of very simple models. These models lack an explicit consideration of functionality, thus suggesting that reuse of preexistent elements might be the driving force shaping the organization of these networks (Solé et al. 2002b; Wagner 2003; Lynch 2007). These results provide evidence for evolutionary constraints as key to explaining the observed patterns (Alberch 1982; Goodwin 1994; Gould 2002). These models incorporate an essential ingredient of evolution: tinkering (Jacob 1977; Solé et al. 2002a; Wilkins 2007). More precisely, as pointed out by Francois Jacob, one important source of divergence between technology and evolution is that natural selection does not work as an engineer, but rather as a tinkerer.

Tinkering implies reuse of available structures and designs, whereas technological changes are (in principle) the result of intentional minds. An intuitive implication of such difference is that designed objects are free from tinkering: They can be obtained without reusing previous components.

As we will see in the next section, simple growth models based on duplication and diversification can explain a large part of the structures found in protein networks. Duplication events leave a fingerprint in terms of modular organization in the topological structure, thus suggesting that (against our intuition) modularity could emerge through non-adaptive processes. Moreover, a similar phenomenon is also found in some technological networks, thus suggesting that some type of universality is at work. Such universality would actually be responsible for the convergent evolution of structural patterns on different scales.

Patterns of Network Organization

In order to illustrate our ideas, we use the protein–protein interaction map (or proteome) as our case study. Some of the structures found in these networks

will reappear when we consider technological webs. In this context, proteome maps will help us understand the origins of a deep similarity between networks generated through evolution and their man-made counterparts. An example of protein network is shown in Fig. 1, where we display the human transcription factor (TF) network (Rodriguez-Caso et al. 2005). Here nodes represent TFs and edges linking two nodes indicate a physical interaction between them. TFs are proteins that directly interact with DNA, and to some extent what we display here is the map of the cell “hardware” that drives DNA dynamics. The links between different proteins indicate that two given TFs interact physically at some point, somewhere within the cell. They thus provide a picture of the cell’s complex machinery. In this context, the so-called *degree* k of a given protein is the number of proteins with which it interacts.

The pattern displayed by this web is fairly typical and is shared by many other biological, social, and technological graphs (Buchanan 2003). This particular network contains $N = 230$ elements (nodes) and $L = 850$ interactions (links), thus corresponding to an average connectivity (average number of links emanating from a node) $\langle k \rangle = 2L/N = 3.70$. This is a low value, indicating that the network is sparse: the

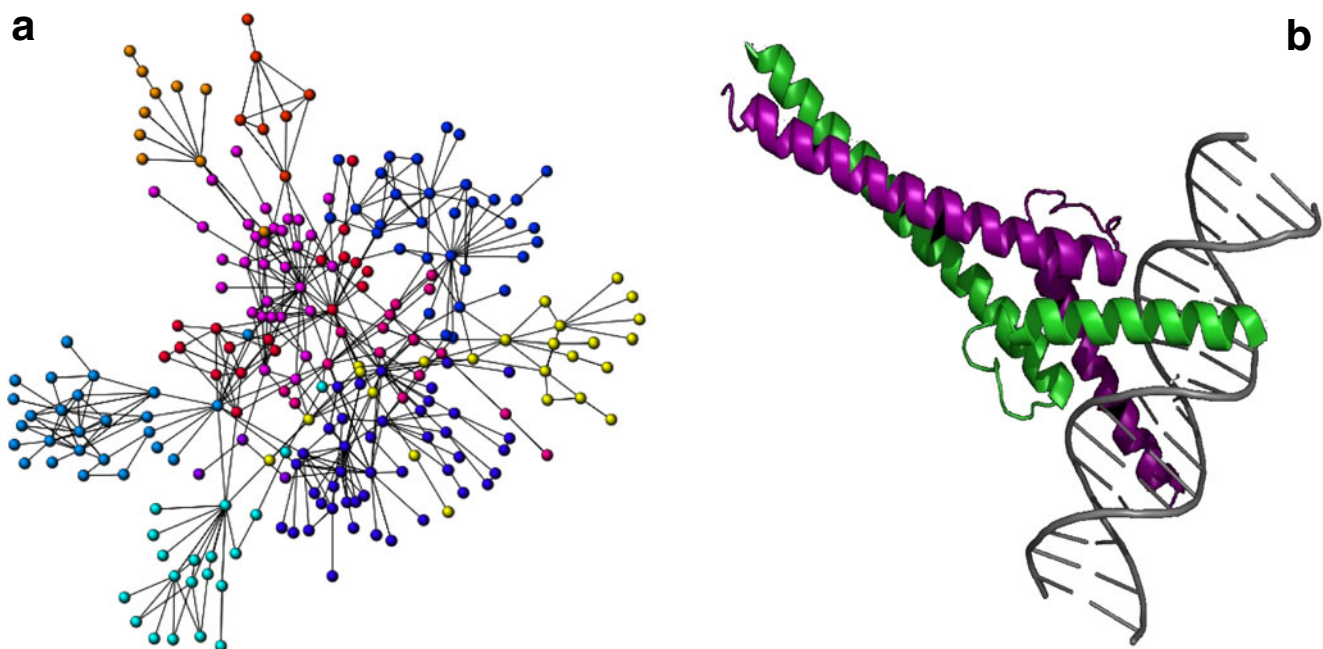


Fig. 1 Cellular networks are heterogeneous. In **a**, we display the human transcription factor (TF) network (Rodriguez-Caso et al. 2005) where each node is a TF and links indicate the presence of protein–protein interactions. Different modules are indicated by means of different colors (see text). These modules are typically

organized around hubs. Each TF is a protein that binds to DNA (an example is shown in **b**), and thus, the TF network would be a map of the cell hardware that “reads” the information stored in the DNA sequence (which would be the software)

average number of interactions is much smaller than the maximum possible ($k_{\max} = N - 1$). But from Fig. 1, we can clearly appreciate that the number of links of different proteins is very diverse: Many TFs have just one or two links, whereas a few—the so-called hubs—display many connections. The network also displays correlated subsets of proteins (Fig. 1a) indicated with different colors associated with the presence of modules (see below). Hubs and their connections with

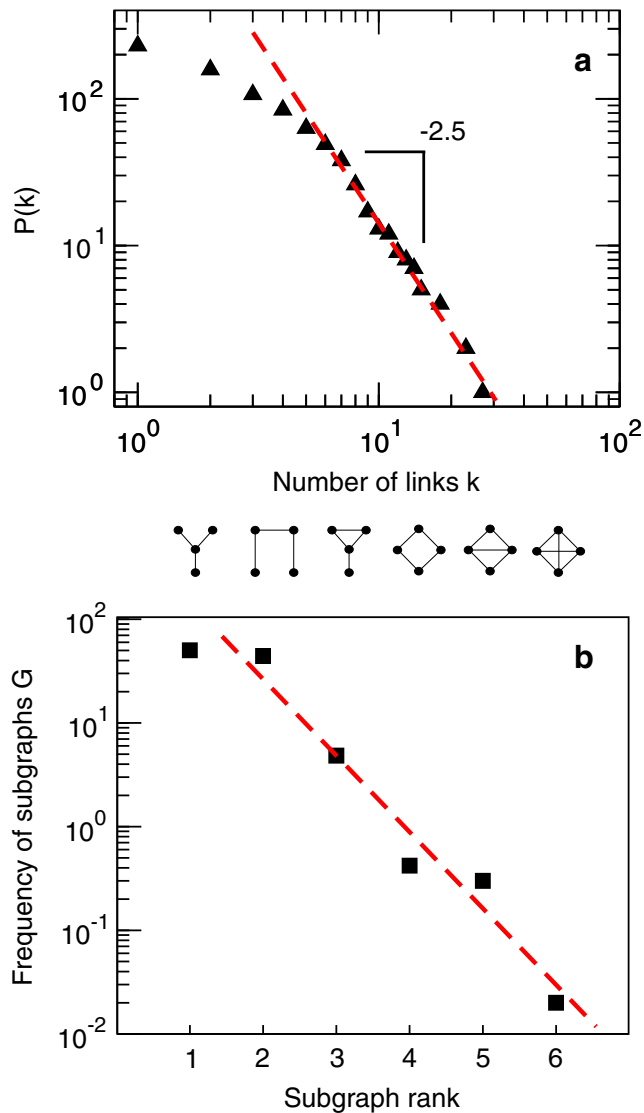


Fig. 2 Statistical patterns of organization in the human transcription factor network (Rodríguez-Caso et al. 2005). In **a**, we display the degree distribution $P(k)$ associated with this network on a log–log scale. A heterogeneous distribution is observed, with a predominance of elements having one or two links but also with a few hubs (compare with Fig. 1). The statistics of network subgraphs shown in **b** reveal that they follow a decreasing exponential decay, with subgraphs ranked from the most frequent to the least

other proteins are highlighted in Fig. 1a. Hubs include the basal transcription initiator, several tumor suppressors (p53, P300) and proto-oncogenes (c-jun, c-myc, or c-fos). Their topological importance is consistent with an important role in terms of essential cellular functions. This heterogeneity is captured by the degree distribution $P(k)$ (Fig. 2a) which gives the probability that a given TF has k links (Amaral et al. 2000).

These networks are also small worlds (SW). This interesting behavior relates two apparently antagonistic properties at the local and global scale in sparse networks. At the local level, it is shown that nodes in a SW graph are connected to a small number of neighbors (on average) and they tend to be connected among them too (i.e., many triangles are observed). On the other hand, most nodes can be reached from all others by a small number of hops, also known as the number of degrees of separation (d). In other words, the local structure is compatible with very efficient communication (Watts and Strogatz 1998).

Modules and Motifs

Protein networks display modularity. One way of detecting this is to look at the topology of interactions among proteins. A module in this context would be a subset of elements having more connections among them than with other parts of the web (Wagner et al. 2007). Modularity can be detected and measured in different ways (Kepes 2007) and allows the detection of communities of preferentially related nodes. An example is shown in Fig. 1a, where proteins belonging to the same module share a common color. In many of these networks, modules appear to be organized around hubs, as can be seen by comparing a and b of Fig. 1.

Beyond modular patterns, a second level of analysis considers the frequency of subgraphs of a given size, also known as subgraph census (see Wasserman and Faust 1994 and references therein). In Fig. 2b, we display the subgraph census for $n = 4$ subgraphs for the human TF web. In this plot, we show the frequency of subgraphs ordered by rank, from the most common to the least. As we can see, their abundance decays quickly as the graphs become more dense (Solé and Valverde 2007).

These subgraphs have received considerable attention in relation to the so-called *network motifs* (Milo et al. 2002, 2004; Wolf and Arkin 2003). These are patterns of interconnections occurring in complex networks at numbers that are significantly higher than those expected from a randomized graph having the

same number of nodes and links (Milo et al. 2002, 2004; Valverde and Solé 2005a, b). Some subgraphs (motifs) would be much more common while others (anti-motifs) would be much less common than expected. The analysis of their statistical distribution reveals that each class of natural and artificial network seems to display a common patterns of motif abundances.

The statistical pattern has been interpreted as functionally meaningful: the higher relevance of a motif would be tied to a potentially important functional role. In this perspective, motifs have been proposed as the minimal building blocks of network complexity (Milo et al. 2002, 2004). It seems thus reasonable to expect their abundances to reflect some type of adaptive trait (Kahstan and Alon 2005). But other analyses do not support this view (Solé and Valverde 2007). In this context, a recent study (Mazurie et al. 2005) shows that network motifs are not subject to any particular evolutionary pressure to be preserved. The reason is that most motifs are not found in isolation and are part of larger aggregates. This is consistent with other studies indicating that a more appropriate, functionally meaningful approach requires considering higher-order interconnection patterns (Dobrin et al. 2004). Actually, improved understanding of cellular webs has led to the identification of much larger superstructures, some of them dubbed with new labels, such as *themes* and *thematic maps* (Zhang et al. 2005).

These studies show that reducing biological complexity to small subsystems of fixed size might not be possible. On the other hand, cellular networks are not generated through random processes, and we can ask how much of the deviations from random expectations are associated with the rules of network growth. As shown below, non-adaptive processes might pervade the origins of both modules and motifs.

Modularity from Tinkering?

How can we incorporate tinkered evolution into a model of proteome evolution? The simplest approach is using a duplication–divergence (DD) model (Ohno 1970; Patthy 1999; Wagner 2001; Hogeweg 2002) where growth takes place through random duplications followed by divergence in the redundant genes. Instead of taking into account many of the underlying complexities of the process, we will restrict ourselves to a graphic theoretic description of protein–protein interactions, as previously followed by several authors (Solé et al. 2002b; Dokholyan et al. 2006; Vázquez et al. 2003; Pastor-Satorras et al. 2003; Teichmann and Babu 2004; van Noort et al. 2004; Colizza et al. 2005; Ispolatov et al.

2005; Cordero and Hogeweg 2006; Foster et al. 2006; see also Koonin et al. 2006)

These models involve tinkering, since duplication is nothing but reuse of previous parts. In our context, we also tinker with connections, since every duplication event implies that previous links are also inherited. As will be shown below, a very similar mechanism is at work in technology design. We will use one of the simplest DD models of protein network evolution (Vázquez et al. 2003), which involves the following set of rules, to be applied a given number of times, until N nodes are present. Assuming that we have a graph of size n , we iterate the following rules:

1. Duplication: Choose a node $v_i \in V$ at random and duplicate it, thus generating a new node v_{n+1} .
2. Link deletion: The new node shares a set of neighboring nodes $\{v_j\}$ with its predecessor. For each common pair of common links, i.e., $e_{i,j}$ and $e_{n+1,j}$, we choose one of them and delete it with probability δ . This rule thus removes (probabilistically) redundant relations among proteins.
3. Link addition: A link is added among nodes v_i and v_{n+1} with probability α . This is a small number and allows new functionalities to emerge by linking the twin proteins.

This model (and its variants) has been shown to successfully capture most statistical features displayed by real protein networks (Vázquez et al. 2003; Rice et al. 2005; Maere et al. 2005). The model has two parameters which can be tuned. Moreover, it has been shown (Wagner 2001) that the rates of link deletion are large, while new links have a much smaller probability. An important point to be made is that for large deletion rates, the system can become disconnected, whereas for small ones it becomes fully and densely connected. At intermediate values (consistent with estimations), sparse networks with most nodes forming a single large graph are observed. These networks are actually very similar to those found in their real counterparts (see Fig. 3a, b). They are small worlds, exhibit the same heterogeneity, and display hubs. These are not just qualitative observations: the simulated nets fit measurable properties found in protein nets very well.

The surprising similarity between non-functional toy models and the real protein maps is remarkable. This supports the idea that the “shapes” of protein maps result from non-adaptive processes. Such a view is fully confirmed by the analysis of modularity: Although modular organization is a desirable and functionally key property of cellular maps, it can arise as a by-product of the rules driving network growth. Modularity would be obtained “for free” without a small-scale

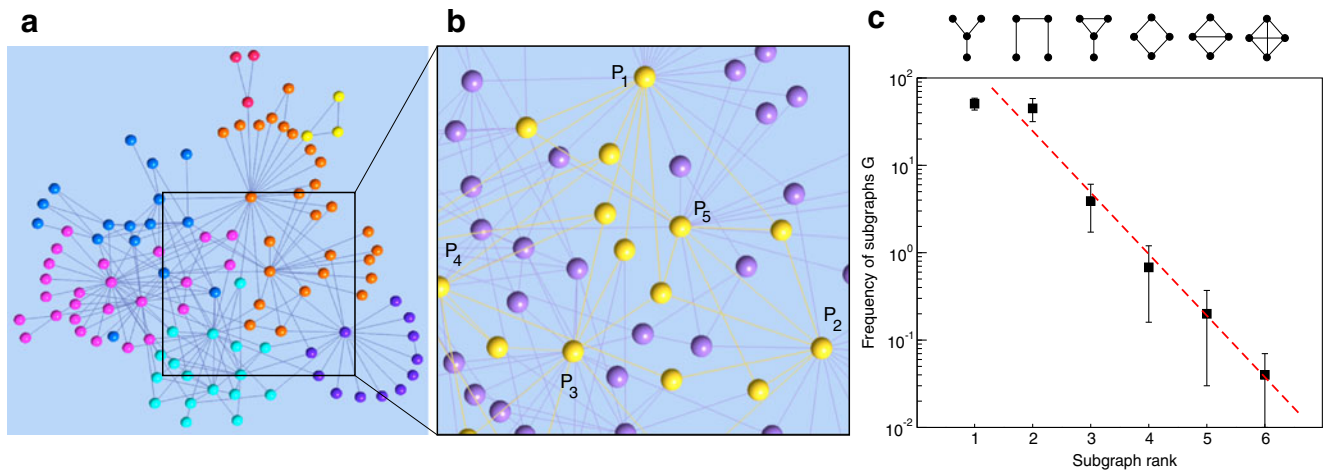


Fig. 3 Modeling tinkered evolution in complex cellular networks. Using the rules described in the text, an *in silico* proteome is obtained (**a**) displaying all features observed in real protein interaction networks, including modularity (modules are indicated by *different colors*). Hubs emerge as a consequence of duplication–divergence rules, and they are also organized as in

tuning of protein–protein interactions. These results are confirmed by the analysis of motif abundances.

Are Motifs Spandrels?

The previous results indicate that the topological patterns exhibited by protein maps could be a by-product of the network construction process. When looking at the network organization, we surely will recognize some noticeable forms and shapes that can easily be interpreted as resulting from selective pressures. But sometimes ordered structures have no adaptive meaning. They are actually examples of what Stephen Jay Gould and Richard Lewontin called *spandrels* (Gould and Lewontin 1979). The term *spandrel*, borrowed from the vocabulary of architecture by these authors, defines the space between two arches or between an arch and a rectangular enclosure. In evolutionary biology, a *spandrel* is a phenotypic characteristic that evolved as a side effect of a true adaptation. We can summarize the features of evolutionary *spandrels* as follows:

- They are the by-product of building rules.
- They have intrinsic, well-defined, non-random features.
- Their structure reveals some of the underlying rules of the system’s construction.

It has been shown (Solé and Valverde 2006, 2007) that the distribution of subgraphs is also a consequence

of tinkering (see also Banzhaf and Kuo 2004; Kuo et al. 2006; Knabe et al. 2008). Specifically, if we count the number of different subgraphs using our previous model, it can be shown that the same census plot can be obtained (Fig. 3c). For most parameter combinations, the shape of this census is the same: an exponential function with less dense subgraphs being more common than denser ones. Interestingly, it was observed that a very good match for four-node subgraphs appears to occur at parameter values where proteome network distributions are recovered (Solé and Valverde 2007). Thus, provided that we tune deletion and addition rates so that a heterogeneous net is obtained, subgraph abundances are also fitted.

From the previous definition, motif abundances might well be the *spandrels* of network complexity. Why? They follow the previous list of requirements:

- Their abundance is matched by *in silico* models lacking real functionality and are thus a by-product of the network building rules.
- They exhibit highly non-random features at several scales, and these are particularly obvious when considering how motifs form clusters (Fig. 1b).
- The aggregates strongly indicate that duplication–rewiring processes, which generate the whole structure, are also responsible for their presence and specific regularities.

However, this pattern does not rule out an active role of selection at the lower scale: links need to be introduced and removed at appropriate rates so that the

network integrity is maintained. In other words, although modules, motifs, and robustness might be, to some extent, a by-product of the duplication–divergence scenario, network connectivity needs to be properly tuned.

The results reported here with a non-directed model have been shown to be robust when more detailed implementations are used. When regulatory interactions are considered using protein interactions matching binding sites, it has been shown that the abundance of some well-known motifs (such as feed-forward loops) is largely a consequence of duplication and divergence (Cordero and Hogeweg 2006). These *in silico* results are consistent with data from transcription networks.

The Engineer as a Tinkerer

Let us go back to our comparison between design and evolution. Apparently, technological artifacts are expected to result from a tinkering-free process. It is actually interesting to see that engineering has been a source of inspiration for some system biologists who see biological designs as closely related to engineered structures (see Lazebnik 2002). Although all of them agree that there is no intentional design, there is some belief that structures closely relate to optimal (or nearly optimal) functions. If the proteome resembles the Internet (for example), the implication would be that they share common optimality principles of organization. Some examples support this view. A recent study by Moses et al. (2008) on the structure of microprocessors suggests that they follow similar scaling laws relating energy dissipation and spatial organization. But if we seek insight from technology, we should first ask ourselves if technology is really free from constraints and tinkered evolution. In this context, it is worth mentioning that electronic circuits have been shown to display small-world and scale-free patterns of organization (Ferrer-Cancho et al. 2001). This is a surprising result, given the well-known wiring cost problem affecting circuit designs and the almost two-dimensional packing of components on a surface. In spite that the rules of construction are different, the common pattern of topological organization suggests (once again) that strong constraints canalize network architecture. As will be shown below, this seems the case in large-scale technological networks (Fig. 4).

Computational models of technological innovation offer a promising approach to the evolution of artifacts. An example of this approach is the work of Brian Arthur and Wolfgang Polak. These authors used a simple evolutionary algorithm to evolve complex elec-

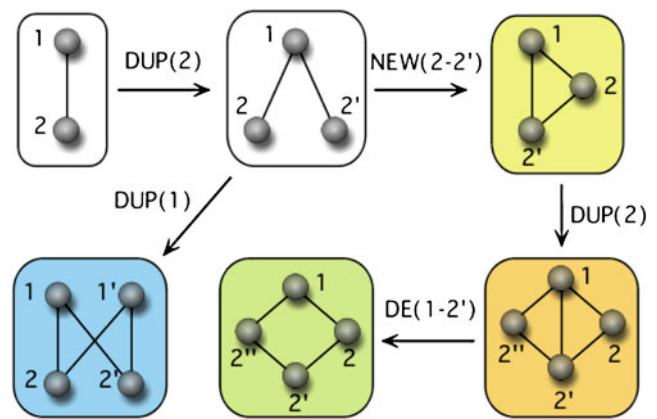


Fig. 4 Network motifs from tinkering. Motifs are identified as small groups of interacting elements with a well-defined arrangement of links. These links can correspond to physical interactions among proteins or to regulatory links among genes. In this figure, gray balls and links indicate units and their interactions, respectively. Starting from the simplest graph (a), we can generate different motifs by gene duplication (DUP) or link addition (ADD). Three common motifs found in cellular networks are highlighted by the colored boxes. Here DUP, NEW, and DE indicate duplication events, introduction of new links, or their deletion, respectively

tronic circuits by combination (Arthur and Polak 2006). Specifically, this work starts using a minimal logic gate, the so-called NAND gate (Fig. 5a) as the elementary building block. A previous list of *goals* is given, defined as desired computations to be performed. Such goals allow the definition of a fitness function and selection of some given designs. The gates are randomly wired and the resulting circuits tested, so that those that work better are selected. Selected circuits are then preserved as new building blocks (Fig. 5b, c), and thus the evolutionary process is carried out by combining more and more complex subsystems.

The model is successful in generating all desired computations, although the exact hardware that implements them is path-dependent: different runs generate different solutions. In spite of the enormous combinatorial space to be—in principle—explored, it is remarkable that complex functions (such as eight-bit adders) are obtained. Moreover, the final circuits involve some amount of junk pieces resembling some of the non-functional structures found in biological systems.

Further, the length of time designs take to evolve (how fitness changes) indicates that punctuated equilibrium seems to be an essential ingredient of change. It is also worth noting that this combinatorial origination of technological complexity leads to a timeline of design evolution that cannot be collapsed into a true evolutionary tree. If we try to draw such a thing, we end up defining a phylogenetic network, similar to the

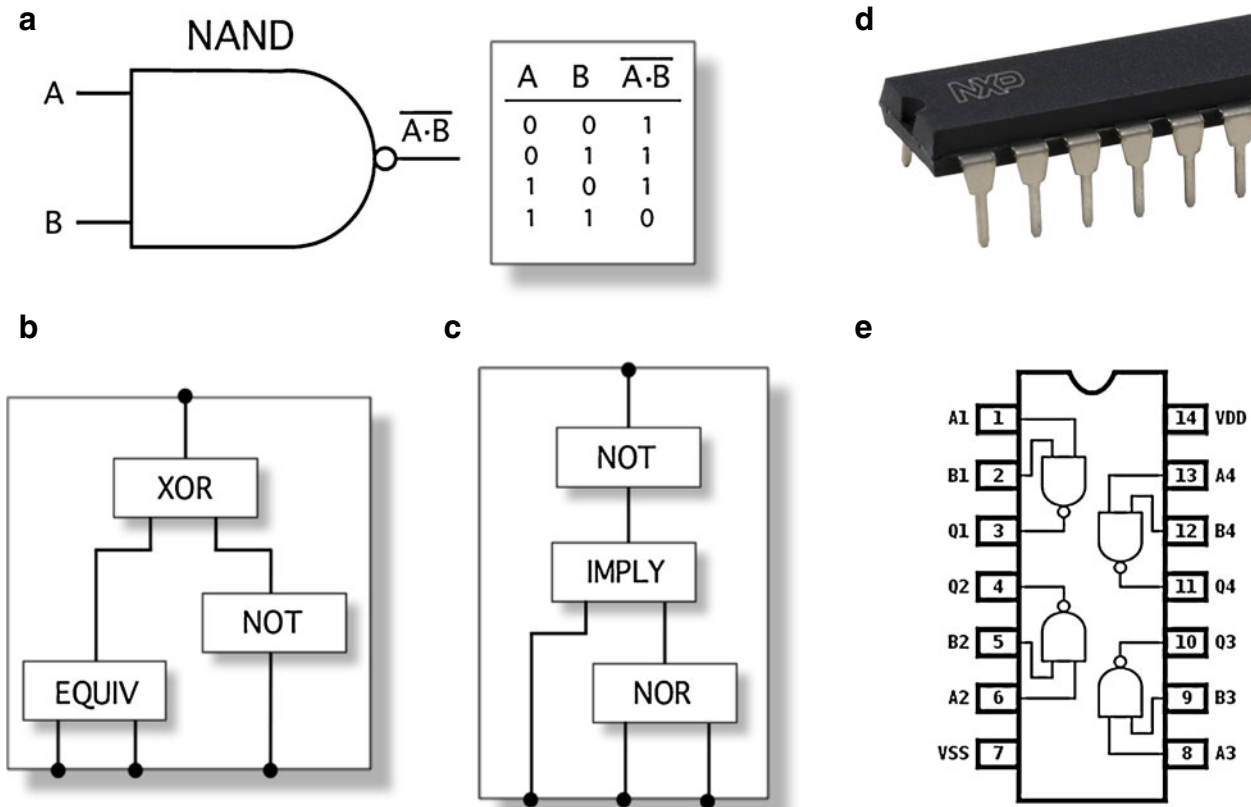


Fig. 5 Starting from a basic logic block, such as a NAND gate (a), a combination algorithm allows generating more complex gates which can themselves be used to build further gates. The two circuits shown in b and c are examples of these evolved circuits (redrawn from Arthur and Polak 2006). The basic result of this combinatorial evolution is that more and more complex

computational blocks (indicated as XOR, EQUIV, IMPLY, etc.) are generated, thus defining a variety of modules. Such modular organization is a characteristic feature of electronic designs, where integrated circuits (such as the one shown in d–e) are fixed combinations of simpler gates

ones associated with prokaryotic evolution (Dagan and Martin 2009). Due to the widespread role of horizontal gene transfer, branches in these trees are inevitably connected through genetic exchange events. In a sentence: if something can be combined, it gets combined.

The previous model illustrates one approach to evolved systems that is based in simulation. Using a given set of basic rules and an appropriate definition of fitness, we can generate complex structures. Brian Arthur's view of technological evolution strongly advocates for combination as the key source of change (Arthur 2010). In this view, technological designs would be closer to chemistry than to selection-based evolutionary forces. The latter would be involved as soon as key ideas are born and applied. And yet, as we turn our attention to the most important piece of technology used in our daily lives, the fingerprints of evolutionary dynamics and in particular tinkering reveal unexpected similarities between engineering and biology.

Tinkering with Software Designs

The Arthur–Polak model deals with a simplified view of technological change. But it certainly provides proof of our concept picture. The model generates new structures by combining of preexisting components, somehow as technology creates new artifacts by means of using existing ones. But we can also analyze real designs by looking at the network structure of some designs and its evolution through time.

Most previous studies on cultural and technological evolution and their similarities have concentrated mostly on old forms of technology. This includes writing, steam engines, computers, and other artifacts. In some cases, it is possible to follow not only how structures change over time but also how they are linked through phylogenetic relations (Temkin and Eldredge 2007). However, the current, invisible but most widely used technological advance is software. Soon after the first computers were designed, software also emerges

and, after a delay, starts becoming the key actor for the computer revolution along with smaller machines. Software programs perform controlling tasks, which are executed by means of the underlying hardware. Thus, software defines the set of logical constraints acting on a given hardware, seemingly as what happens between proteins and DNA. In this sense, software might be a better approach to biological networks and pathways than hardware (Cardelli 2007). Actually, it is interesting to see that software and programming languages have evolved through time-experiencing life-like patterns of change. These include both gradual changes and horizontal evolution but also sharp innovations (Sammet 1969). Interestingly, one of the greatest changes was the transition toward modular structures, the so-called object-oriented language. In such a language, the building blocks involve well-defined, reusable units of programming logic. In this way, it became possible to build programs composed of self-sufficient modules (objects), each containing all the information needed to manipulate its own data structure.

As it happens with cellular networks, the topology of software maps is both scale-free and modular (Fig. 6a) despite the fact that it emerges as the outcome of de-

signed evolution (Myers 2003; Valverde and Solé 2006). Obviously, there must be a basic building plan toward a final function or set of functions. The software engineer foresees the outcome of his work, although big software systems are in fact created by teams of developers. Each software engineer has been trained with some specific design principles (Pressman 2005) that should allow these systems to be of high quality, easily maintainable, faster, and evolvable. But there are a number of constraints affecting the software building process. Modularization is present and is the fingerprint of task allocation within the system: different blocks will be involved in more specific subfunctions. On the other hand, increased complexity leads to conflicts between different subparts. This is reflected for example at the topological level: small software maps tend to display a clean tree-like structure, whereas larger systems typically display much more complex patterns (Valverde and Sole 2006). But textbooks on software design principles make no mention of small worlds or scale-freeness. Actually, some universal patterns shown by these systems are in conflict with the basic rules of thumb used in software engineering. The reason for these patterns and why they seem inevitable stems from an unexpected actor: tinkering.

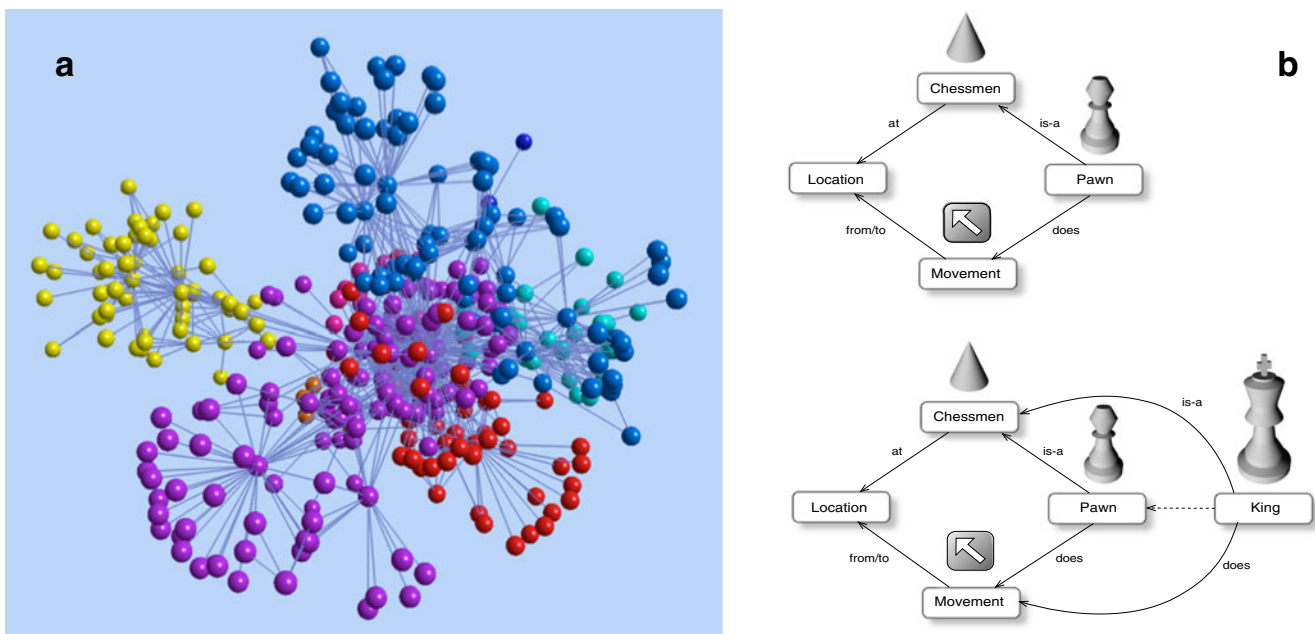


Fig. 6 Even technological, design-driven change can be strongly constrained by tinkered evolution. Software engineering illustrates this point very clearly. Software systems can be seen as complex networks of interacting blocks (**a**) describing given structures or functions. These networks are scale free and also display modularity (indicated by *different colors*). In **b** (upper plot), we show a small piece of a computer program for playing

chess. Basic nodes include the location of each chessman, its definition and identity, and the movement. Although each chessman is a different component, the logic of relations is conserved when a code duplication is performed (**b**, lower) to generate a new element from a previous one. These duplication events (followed by rewire) are widespread

Duplications of software parts is much more common than one might think. They occur naturally at the level of code writing, when a given item to be created shares a number of common features with other previously created items. They are generally used at different scales, including large blocks involving millions of code lines. These blocks typically deal with some given tasks and define themselves as special subsets within software graphs. Since they are highly elaborated structures, once a given block has been generated, a large part of its structure might be reused, and it will. As the system becomes more and more complex, programmers are more prone to this reuse (known as cloning) since they know that a given structure already behaves well. New structures are thus developed within an increasingly complex and integrated structure where interdependencies force the reuse of previous parts. Moreover, empirical analysis of software evolution reveals that duplicated code is more stable than non-duplicated code (Krinke 2008).

An example of small-scale tinkering is illustrated in Fig. 6b. The nodes and their links correspond to part of a chessplayer program. Each element on the chessboard is characterized by its location in space, its pattern of movement, and its identity both as a chessman and a special class of it. All these generic features are required to define each piece and its behavior. If we have already written the code describing a *Pawn*, generating the new one for *King* is relatively easy. First, we clone the *Pawn* module and afterward changes are made so that the new chessman is created. As we would expect, the previous links to the general definition of chessmen and the movement object are also cloned. In other words, in order to create a new class of piece, we can take advantage of the already defined elements (Valverde and Solé 2005a). But in many other situations, reuse has to do with the intrinsic complexity of the system. It has been shown that the abundance of subgraphs in software networks can be explained as a result of the rules of network growth underlying their evolution (Valverde and Solé 2005a, b). These models are very similar to those described above in the context of proteome evolution.

If we take two different software projects and follow in time their evolution, some inevitable regularities will be found. They always evolve toward small-world, scale-free structures. On the other hand, the pattern of change displays universal trends, including an increase with time in the number of avalanches of modifications that need to be introduced, which follow a punctuated equilibrium pattern (Gorshenev and Pismak 2004). More importantly, the final outcomes of these projects are large-scale networks displaying very similar quanti-

tative traits, including motif abundances (Valverde and Solé 2005a, b) and degree distributions. No matter the different natures of the function performed by each system, their structural pattern of organization converges toward the same, life-like architecture.

Discussion

In this paper, we have presented evidence for non-adaptive processes prevailing the evolution of both cellular and technological systems. Under tinkered evolution, growing networks spontaneously develop three key desirable properties: (a) small world organization, (b) heterogeneity, and (c) modularity. All of them are a consequence of the fluctuations associated with duplication dynamics (Solé et al. 2001), although modularity seems an even more general property of growing random networks (Guimerà et al. 2002). Protomodules are thus an inevitable outcome of the growth rules, suggesting that even modular structures might have emerged *for free* (Solé and Valverde 2007). Given the great advantages provided by modular patterns, such spontaneous generation of modules would have been rapidly exploited by natural selection.

The seemingly universal class of patterns generated by network growth dynamics is fully supported by the architecture of software networks. Although human minds are involved (instead of the blind watchmaker) and software systems are not biological entities (Nehaniv et al. 2006), tinkering seems to be largely responsible for the shaping of these large-scale technologies. Moreover, common terms in software design include fault-tolerance, reliability, and extensibility, the last two closely related to the internal evolvability of the software system. The fact that a simple duplication–divergence model is able to account for most properties of software maps (Valverde and Solé 2005a, b) provides an unexpected twist to Jacob's views: even the engineer, who certainly foresees the future, is eventually forced to tinker.

Acknowledgements We thank the members of the Complex Systems Lab for useful discussions on networks and evolution. This work has been supported by a James McDonnell Grant. The authors also thank the Santa Fe Institute, where an important part of this work has been done.

References

- Alberch P. Developmental constraints in evolutionary processes. In: Bonner JT, editor. *Evolution and development*. Berlin: Springer; 1982. p. 313–32.

- Alberch P. The logic of monsters: evidence for internal constraint in development and evolution. *Geobios*. 1989;12:21–57.
- Albert R, Barabási A-L. Statistical mechanics of complex networks. *Rev Mod Phys*. 2002;74:47–97.
- Albert R, Jeong H, Barabási A-L. Error and attack tolerance of complex networks. *Nature*. 2000;406:378–82.
- Amaral LAN, Scala A, Barthelemy M, Stanley HE. Classes of small-world networks. *Proc Natl Acad Sci USA*. 2000;97:11149–52
- Arthur B. The nature of technology. New York: Simon and Schuster; 2010.
- Arthur B, Polak W. The evolution of technology within a simple computer model. *Complexity*. 2006;11:23–31.
- Banzhaf W, Kuo PD. Network motifs in natural and artificial transcriptional regulatory networks. *J Biol Phys Chem*. 2004;4:85–92.
- Barton NH, Briggs DEG, Eisen JA, Goldstein DB, Patel NH. *Evolution*. Cold Spring Harbor: Cold Spring Harbor Laboratory; 2008.
- Basalla G. The evolution of technology. New York: Cambridge University Press; 1988.
- Boccaletti S, Latora V, Moreno Y, Chavez M, Hwang DU. Complex networks: structure and dynamics. *Phys Rep*. 2006;424:175–308.
- Bornholdt S, Schuster HG. *Handbook of graphs and networks*. Berlin: Wiley; 2003.
- Brown JH, West GB. *Scaling in biology*. New York: Oxford University Press; 2000.
- Buchanan M. NEXUS: small worlds and the groundbreaking theory of networks. New York: Norton and Company; 2003.
- Calabretta R, Nolfi S, Parisi D, Wagner GP. Duplication of modules facilitates the evolution of functional specialization. *Artif Life*. 2000;6:69–84.
- Cardelli L. Can a systems biologist fix a Tamagotchi? Position paper for the Gilles Kahn Colloquium, Paris; 2007.
- Chen BL, Hall DH, Chklovskii DB. Wiring optimization can relate neuronal structure and function. *Proc Natl Acad Sci USA*. 2006;103:4723–8.
- Colizza V, Flammini A, Maritan A, Vespignani A. Characterization and modeling of protein–protein interaction networks. *Physica A*. 2005;352:1–27.
- Conway Morris S. *Life's solution: inevitable humans in a lonely universe*. Cambridge: Cambridge University Press; 2003.
- Cordero OX, Hogeweg P. Feed-forward loop circuits as a side effect of genome evolution. *Mol Biol Evol*. 2006;23:1931–6.
- Dagan T, Martin W. Getting a better picture of microbial evolution en route to a network of genomes. *Philos Trans R Soc Lond B*. 2009;364:2187–96.
- Dobrin R, Beg Q, Barabási A-L, Oltvai Z. Aggregation of topological motifs in the *Escherichia coli* transcriptional regulatory network. *BMC Bioinformatics*. 2004;5:10.
- Dokholyan NV, Shakhnovich B, Shakhnovich EI. Expanding protein universe and its origin from biological Big Bang. *Proc Natl Acad Sci USA*. 2006;99:14132–6.
- Dorogovtsev SN, Mendes JFF. *Evolution of networks: from biological nets to the Internet and WWW*. New York: Oxford University Press; 2003.
- Durand M. Architecture of optimal transport networks. *Phys Rev E*. 2006;73:016116.
- Durand M, Weaire D. Optimizing transport in a homogeneous network. *Phys Rev E*. 2004;70:046125.
- Eldredge N. Unfinished synthesis: biological hierarchies and modern evolutionary thought. New York: Oxford University Press; 1985
- Eldredge N. Biological and material cultural evolution: are there any true parallels? *Perspect Ethol*. 2001;13:113–53.
- Ferrer-Cancho R, Janssen C, Solé RV. The topology of technology graphs: small world pattern in electronic circuits. *Phys Rev E*. 2001;64:32767.
- Foster DV, Kauffman SA, Socolar ES. Network growth models and genetic regulatory networks. *Phys Rev E*. 2006;73:031912.
- Goodwin BC. *How the leopard changed its spots: the evolution of complexity*. New York: Charles Scribner's Sons; 1994.
- Gorshenev AA, Pismak YuM. Punctuated equilibrium in software evolution. *Phys Rev E*. 2004;70:067103.
- Gould SJ. *The structure of evolutionary theory*. Cambridge: Harvard University Press; 2002.
- Gould SJ, Lewontin RC. The spandrels of San Marco and the Panglossian paradigm: a critique of the adaptationist programme. *Proc R Soc Lond B*. 1979;205:581–98.
- Guimerà R, Sales-Pardo M, Amaral L. Modularity from fluctuations in random graphs and complex networks. *Phys Rev E*. 2002;66:056120.
- Hartwell LH, Hopfield JJ, Leibler S, Murray AW. From molecular to modular cell Biology. *Nature*. 1999;402 Suppl:C47–C52.
- Hogeweg P. Multilevel evolution: the fate of duplicated genes. *Z Phys Chem*. 2002;216:77–90.
- Ispolatov I, Krapivsky PL, Yuryev A. Duplication–divergence model of protein interaction network. *Phys Rev E*. 2005;71:061911.
- Jacob F. Evolution as tinkering. *Science*. 1977;196:1161–6.
- Kashtan N, Alon U. Spontaneous evolution of modularity and network motifs. *Proc Natl Acad Sci USA*. 2005;102:13773–8.
- Kauffman SA. *Origins of order*. New York: Oxford University Press; 1993.
- Kepes F (ed). *Biological networks*. Singapore: World Scientific; 2007.
- Kauffman SA. *The origins of order*. New York: Oxford University Press; 1993.
- Knabe JF, Nehaniv CL, Schilstra MJ. Do motifs reflect evolved function? No convergent evolution of genetic regulatory network subgraph topologies. *BioSystems*. 2008;94:68–74.
- Koonin EV, Wolf YI, Karev GP (eds). *Power laws, scale-free networks and genome biology*. New York: Springer; 2006.
- Kuo PD, Banzhaf W, Leier A. Network topology and the evolution of dynamics in an artificial regulatory network model created by whole genome duplication and divergence. *Biosystems*. 2006;85:177–200.
- Krinke J. Is cloned code more stable than non-cloned code? Eighth IEEE international working conference on source code analysis and manipulation; 2008. p. 57–66.
- Lazebnik V. Can a biologist fix a radio? -or, what I learned while studying apoptosis. *Cancer Cell*. 2002;2:179–82.
- Lipson H, Pollack JB, Suh NP. On the origin of modular variation. *Evolution*. 2002;56:1549–56.
- Lynch M. The evolution of genetic networks by non-adaptive processes. *Nature Rev Gen*. 2007;8:803–13.
- Maere S, et al. Modeling gene and genome duplications in eukaryotes. *Proc Natl Acad Sci USA*. 2005;102:5454–9.
- Mazurie A, Bottani S, Vergassola M. An evolutionary and functional assessment of regulatory network motifs. *Genome Biol*. 2005;6:R35.
- McGehee GR. *Theoretical morphology*. New York: Columbia University Press; 1999.

- Milo R, Shen-Orr S, Itzkovitz S, Kashtan N, Chklovskii D, Alon U. Network motifs: simple building blocks of complex networks. *Science*. 2002;298:824–7.
- Milo R, Itzkovitz S, Kashtan N, Levitt R, Shen-Orr S, Ayzenshtat I, Sheffer M, Alon U. Superfamilies of designed and evolved networks. *Science*. 2004;303:1538–42.
- Moses ME, Forrest S, Davis AL, Brown JH. Scaling theory for information networks. *J Roy Soc Interface*. 2008;5:1469–80.
- Myers CR. Software systems as complex networks: structure, function, and evolvability of software collaboration graphs. *Phys Rev E*. 2003;68:046116.
- Nehaniv CL, Hewitt J, Christianson B, Wernick P. What software evolution and biological evolution don't have in common. Second international IEEE workshop on software evolvability. Philadelphia: IEEE Computer Society; 2006. p. 58–65.
- Niklas KJ. Morphological evolution through complex domains of fitness. *Proc Natl Acad Sci USA*. 1994;91:6772–6779.
- Ohno S. Evolution by gene duplication. Berlin: Springer; 1970.
- Olesen JM, Bascompte J, Dupont JL, Jordano P. The modularity of pollination networks. *Proc Natl Acad Sci USA*. 2007;104:19891–19896.
- Pastor-Satorras R, Smith E, Solé RV. Evolving protein interaction networks from gene duplication. *J Theor Biol*. 2003;222:199–210.
- Patthy L. Protein evolution. Oxford: Blackwell; 1999.
- Pereira-Leal JB, Levy ED, Teichmann SA. The origin of functional modules: lessons from protein complexes. *Phil Trans R Soc B*. 2006;361:507–17.
- Perez-Escudero A, de Polavieja GG. Optimally wired subnetwork determines neuroanatomy of *Caenorhabditis elegans*. *Proc Natl Acad Sci USA*. 2007;104:17180–5.
- Pressman RS. Software engineering: a practitioner's approach. 6th ed. Boston: McGraw-Hill; 2005.
- Raff RA. The shape of life. Chicago: Chicago U. Press; 1996.
- Ramón y Cajal S. Textura del Sistema Nervioso del Hombre y de los Vertebrados. Madrid: Nicolas Moya; 1899.
- Rice JJ, Kershbaum A, Stolovitzky G. Lasting impressions: motifs in protein–protein maps may provide footprints of evolutionary events. *Proc Natl Acad Sci USA*. 2005;102:3173–4.
- Rodríguez-Caso C, Medina MA, Solé RV. Topology, tinkering and evolution of the human transcription factor network. *FEBS J*. 2005;272:6423–34.
- Sammet JE. Computer languages: history and fundamentals. New York: Prentice-Hall; 1969.
- Solé RV, Valverde S. Are network motifs the spandrels of cellular complexity? *Trends Ecol Evol*. 2006;21:419–22.
- Solé RV, Valverde S. Spontaneous emergence of modularity in cellular networks. *J Royal Soc Interface*. 2007;5:129–133.
- Solé RV, Ferrer i Cancho R, Montoya JM, Valverde S. Selection, tinkering, and emergence in complex networks. *Complexity*. 2002a;8:20–33.
- Solé RV, Pastor-Satorras R, Smith E, Kepler TS. A model of large-scale proteome evolution. *Adv Complex Syst*. 2002b;5:43–54.
- Solé RV, Salazar-Ciudad I, García-Fernández J. Common pattern formation, modularity and phase transitions in a gene network model of morphogenesis. *Physica A*. 2001;305:640–7.
- Sporns O. Networks of the brain. Cambridge: MIT; 2010.
- Striedter GF. Principles of brain evolution. New York: Sinauer; 2005.
- Teichmann SA, Babu MM. Gene regulatory network growth by duplication. *Nature Genet*. 2004;36:492–6.
- Temkin I, Eldredge N. Phylogenetics and material cultural evolution. *Curr Anthropol* 2007;48:146–53.
- Valverde S, Solé RV. Network motifs in computational networks: a case study in software architecture. *Phys Rev E*. 2005a;72:026107.
- Valverde S, Solé RV. Logarithmic growth dynamics in software networks. *Europhys Lett*. 2005b;72:858–64.
- Valverde S, Solé RV. On the nature of design. In: Minai A, Braha D, Bar-Yam Y, editors. Complex engineered systems. New York: Springer; 2006.
- van Noort V, Snel B, Huynen M. The yeast coexpression network has a small-world, scale-free architecture and can be explained by a simple model. *EMBO Reports*. 2004;5:280–4.
- Vázquez A, Flammini A, Maritan A, Vespignani A. Modeling of protein interaction networks. *ComplexUs*. 2003;1:38–44.
- von Dassow G, Meir E, Munro EM, Odell GM. The segment polarity network is a robust developmental module. *Nature* 2000;406:188–92.
- Wagner G. Adaptation and the modular design of organisms. *Lec Notes Comp Sci*. 1995;929:315–28.
- Wagner A. The yeast protein interaction network evolves rapidly and contains few redundant duplicate genes. *Mol Biol Evol*. 2001;18:1283–92.
- Wagner A. Does natural selection mold molecular networks? *Science STKE*; 2003. p. 41–3.
- Wagner GP, Pavlicev M, Cheverud JM. The road to modularity. *Nature Rev Genet*. 2007;8:921–31.
- Wasserman S, Faust K. Social network analysis. UK: Cambridge University Press; 1994.
- Watts DJ, Strogatz SH. Collective dynamics in 'small-world' networks. *Nature (Lond)*. 1998;393:440–2.
- West GB, Brown JH, Enquist BJ. A general model for the origin of allometric scaling laws in biology. *Science*. 1997;276:122–126.
- Wilkins AS. Between “designs” and “bricolage”: genetic networks adaptive evolution. *Proc Natl Acad Sci USA*. 2007;104:8590–96.
- Wolf DM, Arkin AP. Motifs, modules and games in bacteria. *Curr Opin Microbiol*. 2003;6:125–34.
- Zhang LV, et al. Motifs, themes and thematic maps of an integrated *Saccharomyces cerevisiae* interaction network. *J Biol*. 2005;4:6.