**molecular
systems
biology**

## PERSPECTIVE

# Algorithms in nature: the convergence of systems biology and computational thinking

**Saket Navlakha and Ziv Bar-Joseph***

Lane Center for Computational Biology and Machine Learning Department, School of Computer Science, Carnegie Mellon University, Pittsburgh, PA, USA
* Corresponding author. Lane Center for Computational Biology and Machine Learning Department, School of Computer Science, Carnegie Mellon University, 5000 Forbes Avenue, Pittsburgh, PA 15213, USA. Tel.: + 1 412 268 5527; Fax: + 1 412 268 3431; E-mail: zivbj@cs.cmu.edu

Computer science and biology have enjoyed a long and fruitful relationship for decades. Biologists rely on computational methods to analyze and integrate large data sets, while several computational methods were inspired by the high-level design principles of biological systems. Recently, these two directions have been converging. In this review, we argue that thinking computationally about biological processes may lead to more accurate models, which in turn can be used to improve the design of algorithms. We discuss the similar mechanisms and requirements shared by computational and biological processes and then present several recent studies that apply this joint analysis strategy to problems related to coordination, network analysis, and tracking and vision. We also discuss additional biological processes that can be studied in a similar manner and link them to potential computational problems. With the rapid accumulation of data detailing the inner workings of biological systems, we expect this direction of coupling biological and computational studies to greatly expand in the future.
*Molecular Systems Biology* **7**: 546; published online 8 November 2011; doi:10.1038/msb.2011.78
*Subject Categories:* bioinformatics; computational methods
*Keywords:* algorithms; biological coordination; interaction networks; models

## Introduction

Biologists have been increasingly relying on sophisticated computational methods, especially over the last two decades as molecular data have rapidly accumulated. Computational tools for searching large databases, including BLAST (Altschul *et al*, 1990), are now routinely used by experimentalists. Genome sequencing and assembly rely heavily on algorithms to speed up data accumulation and analysis (Gusfield, 1997; Trapnell and Salzberg, 2009; Schatz *et al*, 2010). Computational methods have also been developed for integrating various types of functional genomics data and using them to create models of regulatory networks and other interactions in the cell (Alon, 2006; Huttenhower *et al*, 2009; Myers *et al*, 2009). Indeed, several computational biology departments

have been established over the last few years that are focused on developing additional computational methods to aid in solving life science's greatest mysteries.

Computer scientists have also relied on biological systems for inspiration, especially when developing optimization methods (Table I). Early work on artificial intelligence in the 1960s leveraged ideas related to the activity of neurons in the brain to develop a class of computational methods known as neural networks (Hebb, 1949; Hopfield, 1982; Bishop, 1996), which have been used in many machine learning applications ranging from image segmentation to computing missile trajectories (Bishop, 1996). Other optimization techniques, such as genetic algorithms (Goldberg, 1989), were inspired by common operations in DNA sequence evolution and have been widely applied over the last 20 years. Social insects (Dorigo *et al*, 2006) and particle swarms (Kennedy and Eberhart, 2002) have also motivated the study of how self-organization emerges from local interactions (Abelson *et al*, 2000; Committee on Frontiers at the Interface of Computing, Biology, and National Research Council, 2005). These ideas have been applied extensively to multi-agent system optimization (Deneubourg *et al*, 1990; Ferber, 1999). A number of additional methods, including non-negative matrix factorization (Lee and Seung, 1999) and population protocols (Aspnes and Ruppert, 2009; Chazelle, 2009) have also capitalized on biological insights to derive new computing paradigms.

While all of these methods have led to successful applications, they only relied on a high-level (and sometimes flawed) understanding of the biological processes they were based on, and thus they usually did not directly lead to new biological insights. Similarly, though novel computational methods have been developed to help researchers learn new biology, the application of these methods to the biological problem (i.e., the biological system itself) rarely fed back to help computer scientists design better algorithms. Thus, the two directions—relying on biological ideas to develop computational methods and using computational methods to study biology—remained largely separated (Figure 1).

## Shared principles of computational and biological systems

There are many parallel requirements of computational and biological systems, which suggest that one can learn from the other (Figure 2). First, like virtually all large-scale computing platforms, biological systems are mostly distributed consisting of molecules, cells, or organisms that interact, coordinate, and make decisions without central control (Seeley, 2002; Babaoglu *et al*, 2006; Figure 2A). Second, biological processes need to be able to successfully handle failures and attacks to thrive (Jeong *et al*, 2000; Kitano, 2004). Robustness is also a key property algorithm engineers covet when designing

computing systems that persist in noisy environments. Third, networks serve as an important medium through which interactions occur and information propagates (Alon, 2006; Figure 2B). In both settings, the structure of these networks is

**Table I** Examples of biological systems that have inspired computational algorithms

| Biology | Computer science | References |
| --- | --- | --- |
| Neural networks | Machine learning | Hebb (1949), Hopfield (1982), Bishop (1996) |
| Sequence evolution | Genetic algorithms | Goldberg (1989) |
| Sequence evolution | DNA computing | Adleman (1994), Benenson et al (2004), Istrail et al (2007), Păun et al (2004) |
| Ant colonies, swarms | Search optimization | Kennedy and Eberhart (2002), Committee on Frontiers at the Interface of Computing, Biology, and National Research Council (2005), Dorigo et al (2006) |
| Social insects, flocks | Network design | Deneubourg et al (1990), Beckers et al (1994), Theraulaz and Bonabeau (1995), Coore et al (1997), Ferber (1999), Abelson et al (2000), Fewell (2003), Werfel and Nagpal (2006) |
| Immune system | Modeling and optimization | Forrest et al (1994), Kephart et al (1995), Garrett (2005), Greensmith et al (2005) |
| Visual system | Dimensionality reduction | Lee and Seung (1999) |

Most of these examples were based on a high-level understanding of the system and therefore usually did not directly lead to new biological insights.
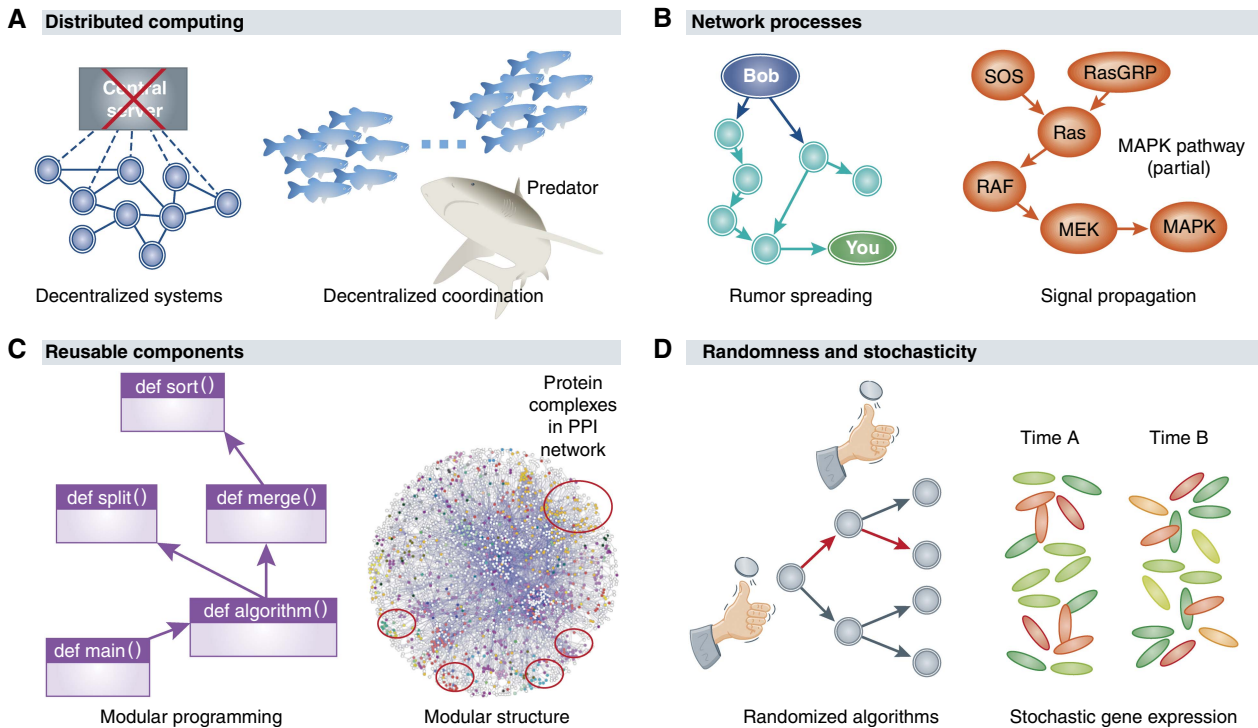
often directly linked to the system's function. Fourth, biological systems are often modular; that is, they reuse certain components in multiple, and sometimes very different, applications. This is a key design principle in many programming languages and in large complex networks (Kashtan and Alon, 2005; Fortunato, 2010; Figure 2C). Fifth, biological processes are often stochastic (Kaern et al, 2005) resembling randomized algorithms whose power has been well documented for the design of approximation algorithms and whose use ranges from single process systems to large distributed systems (Motwani and Raghavan, 1996; Vazirani, 2004; Figure 2D).

These shared principles indicate that thinking computationally about biological systems may lead to better understanding of their function. However, there are also differences between biological and computational systems. While computational systems are often focused on speed, biological systems can adapt dynamically to changing environments and pressures and have a keen ability to reach joint decisions with very limited knowledge of their surroundings. These aspects, while desirable, are still not fully realized by computational systems. By studying how biological systems operate within these constraints and conditions to solve problems, we may be able to infer new directions for addressing these issues within computational systems.

With recent advances in our ability to generate and analyze biological data, new studies have emerged that have linked computational problems to biological problems, including cell-fate determination, slime mold tunneling, fault tolerance in regulatory networks, and bat navigation (Table II). In this review, we discuss a number of these studies focusing on three areas of interest: coordination in large distributed systems, network processes and design, and tracking and vision (Figure 3). For each area, we highlight several studies that demonstrate how computational thinking (Wing, 2006) can
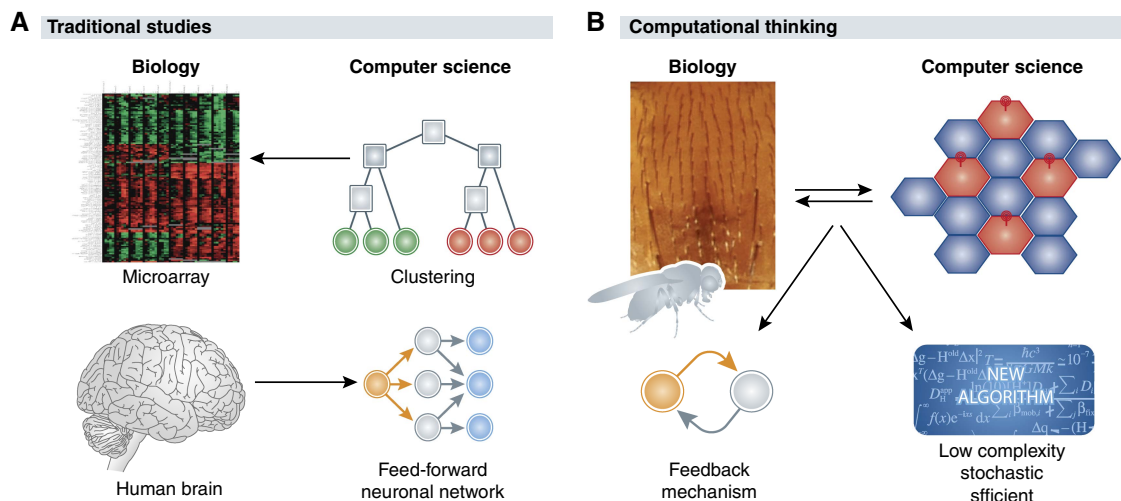


**Figure 1** Traditional studies versus computational thinking. (**A**) Traditionally, biologists leveraged computing power to analyze and process data (e.g., hierarchically clustering gene expression microarrays to predict protein function), and computer scientists used high-level design principles of biological systems to motivate new computational algorithms (e.g., neural networks). Rarely these two directions were coupled and mutually beneficial. (**B**) By thinking computationally about how biological systems process information (Nurse, 2008; Hogeweg, 2011), we can develop improved models and algorithms *and* provide a more coherent explanation of how and why the system operates as it does.

**A** Distributed computing

Decentralized systems   Decentralized coordination

**B** Network processes

Rumor spreading   Signal propagation

**C** Reusable components

Modular programming   Modular structure

**D** Randomness and stochasticity

Randomized algorithms   Stochastic gene expression

**Figure 2** Parallels between computational and biological systems. (**A**) Biological and computational systems often coordinate and maintain functionality without relying on a central controller. (**B**) Networks are often the medium through which processes spread, either on the Web or in the cell. (**C**) Modularity is a common design principle in programming languages and also serves as the basis for increasing complexity and flexibility in evolving systems. (**D**) Stochastic processes in computational and biological settings help to remain robust in constrained, noisy environments.

lead to new biological insights and new algorithms. We also discuss several additional biological processes for which better understanding of how they operate could lead to solutions to long-standing computer science problems.

Note that from the computer science point of view, our focus in this review is on algorithms. Another important area which has both benefited from, and contributed to, the improved understanding of biological systems relates to engineering computational systems. We briefly discuss some examples for this direction including robotics and synthetic biology. However, this direction deserves its own review(s) and is thus out of the scope of this paper.

## Coordination

Coordination is a major challenge for both computational and biological processes. At the molecular level, coordination is required to activate sets of genes that together respond to external conditions (Kaur *et al*, 2010). At the cellular level, cells coordinate to determine cell fate during development (Afek *et al*, 2011) and to synchronize heart beats (Mirollo and Strogatz, 1990). Organisms, including the octopus, need to coordinate the activation of limbs (Sumbre *et al*, 2001), while shoals of fish coordinate to avoid predators (Ward *et al*, 2011). In computational systems, coordination is required for virtually all large-scale computing systems. Examples include search engines that coordinate thousands of back-end servers

to quickly respond to user queries (Brin and Page, 1998), sensor networks that aggregate data when monitoring environments (Mainwaring *et al*, 2002; Werner-Allen *et al*, 2006), and mobile networks that synchronize data and schedules across multiple devices (Bernard *et al*, 2004). Below we discuss a few examples in detail.

### The fly brain and maximal independent set

A recent example of a coordination-focused study that jointly addressed a biological and computational problem is from work on fly brain development. During development of the fly brain, some cells in the pre-neural clusters become sensory organ precursor (SOP) cells. These cells later attach to small bristles on the fly's forehead that are eventually used to sense the environment. It has long been known that the distribution of SOP and non-SOP cells follow a strict spacing pattern (Rendel *et al*, 1965). In particular, cells that become SOPs inhibit their neighbors so that SOPs are surrounded by non-SOPs (Figure 3A, left). However, the exact process by which such global organization is achieved in a short time and by starting with all-equivalent cells remains unknown.

While studying this problem, Afek *et al* (2011) made a new and interesting observation: the requirements from SOP selection—that each cell was either an SOP or directly connected to an SOP and that no two SOPs were connected—are similar to the formulation of the maximal independent set (MIS) problem in computer science. The MIS problem seeks to

**Table II** Examples of new synergistic relationship between biology and computer science

| Area | Biological system | Computational problem | References | Model | Algorithm |
|---|---|---|---|---|---|
| Coordination | Fly SOP selection | Maximal independent set | Afek *et al* (2011) | ✓ | ✓ |
| | Fireflies flashing | Synchronization | Glass (2001), Lucarelli and Wang (2004), Hong and Scaglione (2005), Werner-Allen *et al* (2005), Babaoglu *et al* (2007), Pagliari and Scaglione (2011) | ✓ | ✓ |
| | *Octopus neural control | Hierarchical computing | Sumbre *et al* (2001) | × | × |
| | *Fish shoals/honeybees | Ensemble methods | Marshall *et al* (2009), Conradt (2011), Ward *et al* (2011) | ✓ | × |
| | *Quorum sensing | Consensus | (Marshall *et al*, 2009) | ✓ | × |
| | *DNA replication | Resource allocation | Farkash-Amar *et al*, (2008) | × | × |
| | *Mate selection | Graph matching | — | × | × |
| Networks | Slime mold tunneling | Network design | Li *et al* (2010), Tero *et al* (2010), Watanabe *et al* (2011) | ✓ | ✓ |
| | Gene regulation | Fault tolerance | Gu *et al* (2003), Balaji *et al* (2006), Gitter *et al* (2009), Pomerance *et al* (2009) | ✓ | ✓ |
| | *Protein localization | Routing | Shapiro *et al* (2009) | × | × |
| | *Brain networks | Network design | Chen *et al* (2006) | × | × |
| | *Cell signaling | Clustering | Wokoma *et al* (2005), Charalambous and Cui (2008) | ✓ | ✓ |
| Tracking and Vision | Visual cortex | Object recognition | Riesenhuber and Poggio (1999), Serre *et al* (2007a, b) | ✓ | ✓ |
| | Echolocation in bats | Localization | Ghose *et al* (2006), Yovel *et al* (2010) | ✓ | ✓ |
| | *Visual cortex | One-shot machine learning | Li *et al* (2006), Serre and Poggio (2010) | × | × |

The biological systems are divided into three areas: coordination, networks, and tracking and vision. For each system, we show the analogous computational problem and whether a model or algorithm was derived by previous works. Rows beginning with a star denote potential future work.
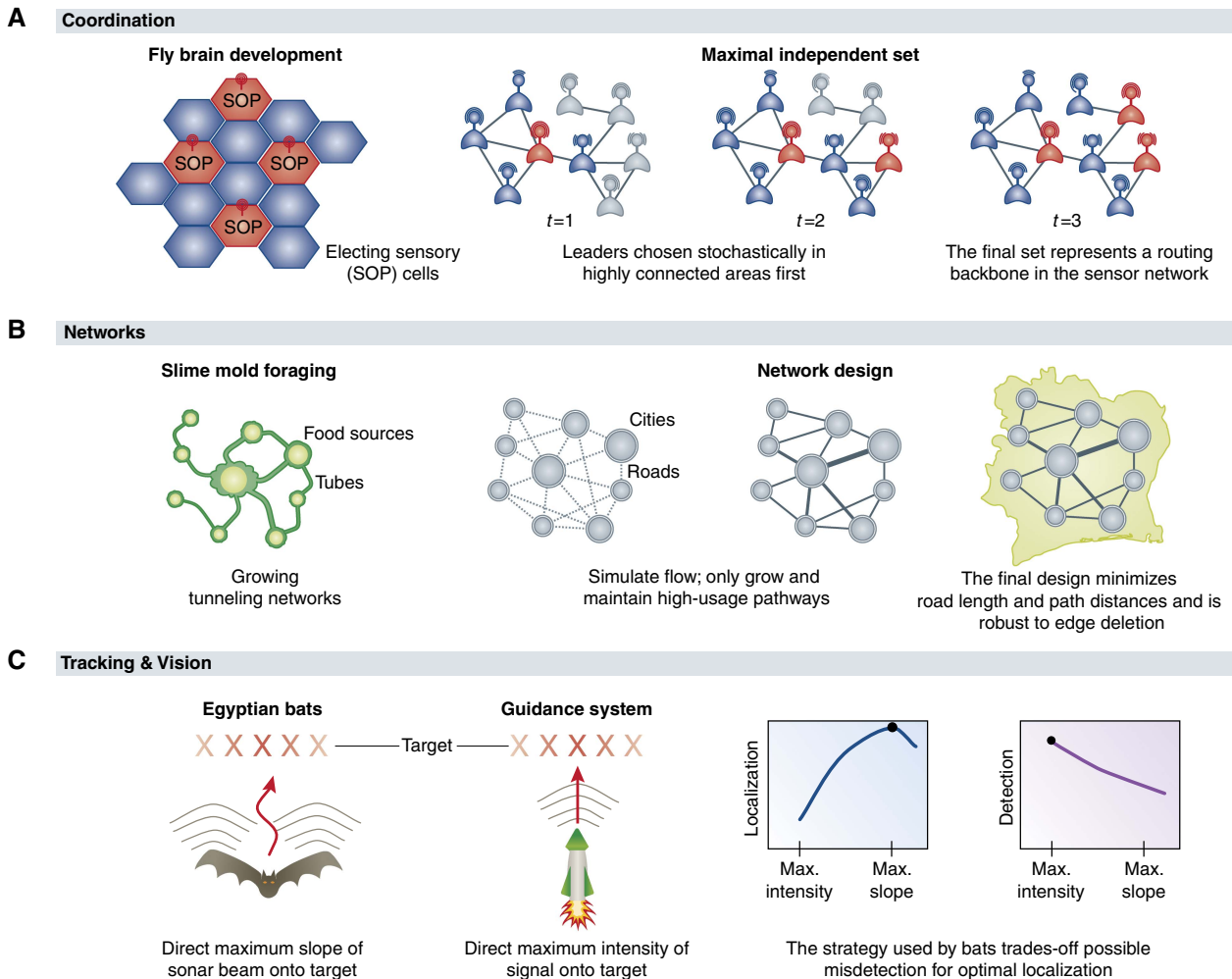
elect a set of local leaders in a graph such that all other nodes in the graph are connected to a member of the MIS and no two MIS members are connected to each other. An MIS in an *ad-hoc* wireless network serves as a routing backbone by which nodes can communicate. It also corresponds to a distributed partitioning of the nodes into clusters that can be used to optimize network bandwidth and resource distribution in the network. Thus, this is an important practical problem that has been studied for more than two decades. However, the solution used by the fly seemed different from all previous algorithms suggested for this task (Luby, 1985; Alon *et al*, 1986). In particular, the fly's solution relied on very short messages and did not use any information regarding the number of neighbors that can still become SOPs.

To study how the fly was still able to solve the SOP selection problem under such limitations, Afek *et al* (2011) used microscopy experiments to follow SOP selection in developing flies. Coupling the results of these experiments with possible molecular processes for the activation of the specific proteins known to be involved in the SOP selection process, they discovered that a stochastic feedback process, in which selection probability increases as a function of time, provides the best match to the experimental results. Based on these findings, they were able to develop a new computational algorithm for MIS selection that is more efficient and more robust than previous methods. Unlike previous MIS selection methods (Luby, 1985; Alon *et al*, 1986), the fly-based method does not use any knowledge about the number of neighbors a node has (its degree). Instead, with probability that increases exponentially over time, each node that has not already connected to an MIS node proposes itself as an MIS node. This leads to a selection pattern in which denser areas in the graph are assigned MIS nodes early on and less populated areas are

assigned at later stages (see Figure 3A). While the original algorithm of Afek *et al* (2011) required that nodes know an upper bound on the total number of nodes in the network, a new version of this algorithm (Afek *et al*, 2011) removes this requirement, leading to the first MIS selection algorithm that does not require any knowledge of the network or its topology. This fly-based method is specifically appropriate for sensor networks in which nodes might be randomly and dynamically distributed in an environment and have no knowledge of global topology. Thus, by studying a very specific cellular developmental system, the authors helped resolve a long-standing computer science problem and improved our understanding of cell-fate determination.

## Additional coordination problems

Several additional coordination problems have been studied, separately, in biology and computer science. The octopus presents a particularly interesting case with its eight arms. From what we know (Mather *et al*, 2010), the octopus employs a hierarchy of coordination elements. At the top level, the brain is in charge of what arm to use when, for example, seeking prey; however, at the lowest level each arm appears to act independently, adjusting its grip, size, and rotation without consulting the brain (Sumbre *et al*, 2001). It is believed that a large number of neurons in the octopus are outside the brain and in the arms which is highly unusual for vertebrates (Mather *et al*, 2010). Further, Mather *et al* (2010) claim that there is a third intermediate layer that allows coordination and communication between neighboring arms without going through the brain. Hierarchical distributed systems have also been studied as a means to improve load balancing, scheduling, and fault tolerance (Feitelson, 1990). Though no

## A    Coordination

**Fly brain development**



Electing sensory
(SOP) cells

**Maximal independent set**



*t*=1

Leaders chosen stochastically in
highly connected areas first

*t*=2

*t*=3

The final set represents a routing
backbone in the sensor network

## B    Networks

**Slime mold foraging**



Food sources

Tubes

Growing
tunneling networks

Cities

Roads

Simulate flow; only grow and
maintain high-usage pathways

**Network design**

The final design minimizes
road length and path distances and is
robust to edge deletion

## C    Tracking & Vision

**Egyptian bats**

X X X X X ——Target—— X X X X X

**Guidance system**

Direct maximum slope of
sonar beam onto target

Direct maximum intensity of
signal onto target

Localization

Max.
intensity

Max.
slope

Detection

Max.
intensity

Max.
slope

The strategy used by bats trades-off possible
misdetection for optimal localization

**Figure 3**    Examples of biological systems and their analog computational problems. Bidirectional studies have led to insights into how biological systems operate, while at the same time advancing computational methods. These studies have explored coordination issues (**A**), network design principles (**B**), and vision problems (**C**). See text for complete details.

works that we are aware of has attempted to integrate these studies, lessons learned from the octopus can potentially be used to improve these systems, while also helping to better understand how different layers of control collectively coordinate actions.

Another example of a problem pertinent to biological and computational systems is distributed consensus (Simeone *et al*, 2008). Consensus is a building block in several computational procedures and is known to be an extremely hard problem for massive, asynchronous, and failure-prone systems (Lynch, 1996). In biology, consensus has an important role in coordinating populations. Fish, for example, are constantly under the threat of predation and must be able to quickly sense the environment and make decisions. The formation of groups can aid in collective vigilance because surveillance can be distributed among a set of fish. In addition, the group is likely to avoid individual errors by pooling sensory data together. However, the group also bears additional communication costs of propagating information to other members of the group, which delays decision making

(Ward *et al*, 2011). Ward *et al* (2011) have empirically shown that fish make faster and more accurate decisions under predation threat in large groups (shoals) compared with smaller groups. The self-organization of shoals, the authors argue, naturally emerges from quick reaction to local neighbors that spreads quickly through the shoal. Honeybees have also been shown to optimally minimize the decision time for a desired level of accuracy (under certain models) when deciding between candidate new homes (Marshall *et al*, 2009). Here, honeybees also attempt to balance between accuracy (choosing the best home) and speed (time to make a decision).

Finally, synchronization is another form of coordination that many biological systems rely on, including pacemaker cells in the heart (Kuramoto and Yamagishi, 1990) and fireflies that flash in unison (Winfree, 1967; Mirollo and Strogatz, 1990). By developing models and methods based on studying how fireflies synchronize (Mirollo and Strogatz, 1990), the theory of pulse-coupled oscillators has flourished and has been used in wireless sensor networks (Lucarelli and Wang, 2004; Hong and Scaglione, 2005; Werner-Allen *et al*, 2005; Pagliari

and Scaglione, 2011) and in overlay networks (Babaoglu *et al*, 2007) for clock management purposes and to integrate time-sensitive data. Algorithms based on these systems have resulted in faster convergence times and improved our understanding of biological self-stabilization (Daliot *et al*, 2003, 2004). In all cases, as modeling efforts continue, it is possible that better distributed ensemble protocols can be developed, which may lead to new testable biological hypotheses.

## Networks

Networks underlie many computational systems. Understanding the structure and dynamics of networks such as the Internet, social networks, and mobile networks has been the focus of several recent studies (Easley and Kleinberg, 2010). Key questions studied include how to best design such networks, how to determine what are the important nodes, how to route messages efficiently from one node to another, and how to maintain functionality despite errors or noise. These issues have also recently been the focus of several biological studies that have investigated how entities in molecular (signaling, regulatory, and metabolic) networks, cellular networks in the brain, and organism-level networks collectively interact and self-organize to efficiently accomplish tasks. We highlight a few recent examples below.

### Slime mold tunneling for network design

When the slime mold *Physarum polycephalum* forages in an environment, it grows to become a tubular structure that links all the resources it finds together via direct connections or intermediate junctions (Figure 3B, left). Evolution has designed the network such that sources are connected via efficient paths, while also ensuring minimal production cost and robustness to disconnection. To better understand the design principles balancing these requirements, Tero *et al* (2010) created a map of 36 food sources ('cities') placed on a map of the Tokyo area. Remarkably, they found that the slime-mold designed network and the actual Tokyo rail system were very similar. In particular, both networks had comparable cost (rail length), efficiency (short paths), and resiliency (redundant paths)—all under various definitions and models of these factors. What is interesting, however, is that the slime mold designed its tunneling system in a completely distributive manner without any global coordination. Clearly, the same cannot be said of the rail system.

Inspired by the characteristics of this process, Tero *et al* (2010) developed an algorithm for adaptive network construction based on driving flow through the network and dynamically selecting for preferential tubes (Figure 3C, right). They show that their model can produce networks that closely resemble the actual networks grown by the slime mold. Such a model—which mathematically captures how the slime mold self-organizes, self-optimizes, and self-repairs itself (Marwan, 2010)—can be used to quantify the trade-off between various desired network properties. These principles can be applied to designing mobile communication networks, in which various nodes or devices need to maintain communication pathways under a changing and dynamic environment. Indeed, follow-up work has shown that a model inspired by the slime mold foraging strategy can be used to design path formation protocols in wireless sensor networks (Li *et al*, 2010) and optimize traffic distribution in transportation networks (Watanabe *et al*, 2011) . On the other hand, mapping the slime mold tunneling process to a network design problem has also led to a greater understanding of how the slime mold balances conditions and constraints when foraging.

### Gene backup and fault tolerant systems

Errors, failures, and attacks commonly occur in molecular processes. Examples includes DNA replication errors, protein failures due to misfolding or mutations affecting transcription, and pathogens that enter the cell and interfere with normal function. DNA repair, backup, and innate immunity are all cellular processes that have been remarkably designed to identify and overcome these issues. Gaining a better understanding of how these systems operate is important for many health-related studies.

Fault tolerance is also a critical design principle of computational systems. Consider a popular website backed by thousands of servers. At any time, one or more of these servers may fail due to error or malicious attack. Overcoming these issues and making them transparent to the user (by, e.g., redirecting queries to active servers that also act as backup mechanisms; Guerraoui and Schiper, 1996) is an important algorithm design problem. Improving these techniques is key in creating better and more usable distributed systems.

So, how do molecular systems deal with failures? Recent work suggests that molecular systems do not in fact use dedicated failure detection mechanisms. Rather, they seamlessly integrate backup resources into their design. Using single gene-deletion mutants in *S. cerevisiae*, Gu *et al* (2003) found that knocking out duplicate genes (genes with paralogs) resulted in a higher probability of functional compensation (i.e., higher fitness) than when knocking out non-paralogous genes.

Gitter *et al* (2009) reported a similar result in gene regulatory networks. These networks encode the relationships between transcription factors (TFs) and the genes they regulate. When knocking out TFs with a strong paralog, it was observed that the expression level of its regulated genes did not significantly change, due to compensation by the TF's paralog. On the other hand, when knocking out a singleton TF, there was a bigger change in expression of direct targets due to the lack of a backup mechanism. Balaji *et al* (2006) propose a fault tolerant computational model based on the design of regulatory networks. To create a network that is fault tolerant to hub deletion they propose to take the original network and do a transformation (multiplying $A'A$ where $A$ is adjacency matrix for the network). In the resulting network, two nodes (or TFs) will be connected if they share many common neighbors in the network. This common-neighbors structure arises naturally in biological systems from the duplication-divergence model discussed below.

Together, these studies show how backup is naturally built into decentralized molecular systems. Evolutionarily, it is possible that backup naturally emerged from the process of duplication and divergence. In this model, a gene is duplicated to yield a functional equivalent, which over time diverges to specialize in a similar subtask. Thus, the gene and its duplicate

share similar sequence and interaction partners and either can be used as a backup gene for the other. The challenge, from the design standpoint, is to determine which local nodes (genes) should be replicated in order to globally minimize the effect of errors. Recent work on robustness of networks to node and edge failures (Ciliberti *et al*, 2007; Zhong *et al*, 2009) may provide further information on how such ideas can be applied to computational systems.

## Additional network problems

Effective routing of proteins to their designated subcellular locations is another problem that may benefit from joint computational and biological modeling. It is known that routing occurs via signals encoded in the amino-acid sequence of the protein (Shapiro *et al*, 2009); however, we are not aware of any model of distributed routing inspired by protein localization. Such a model, as with most other biological systems, might have several desired properties, such as efficiency and fault tolerance to specific conditions or environments.

Another network that holds much promise for future work is the network of neurons that are active in the brain. As mentioned previously, neural networks represent a computational optimization framework that is based on these networks. However, with our increased ability to study brain activity (Sporns, 2010), we can start addressing other questions about the design and architecture of real neural networks. For example, some models have been proposed to generate networks with brain-like connectivity (Kaiser and Hilgetag, 2004; Koene *et al*, 2009) while minimizing wiring and path lengths (Chen *et al*, 2006). What local rules do these models employ and what do they tell us about why the brain is organized the way it is?

## Vision and tracking

The visual system in the mammalian brain has an exceptional ability to process imagery and build a coherent representation of the world. It can seamlessly detect and track objects, reconstruct scenes, and restore images (Shapiro and Stockman, 2001), all in real time and under various amounts of noise, lighting conditions, and transformations. No computer algorithm today comes close to matching its performance. Understanding how this system works and replicating it in the computer has been a long sought goal of biology and computer scientists, respectively. By teaching a computer how to 'see', automated systems can be designed for face detection and authentication, 3D model building, medical image processing, tracking, and surveillance, among many more (Szeliski, 2010).

### Bat echolocation and tracking

Bats are notoriously well versed at tracking objects at night. Using their highly evolved sonar system, bats rely on echoes to forage and navigate efficiently at night. In biology, pursuit is a basic task to all prey-seeking animals. In computer science, localization is an important problem with many applications to defense and guidance (Figure 3C, left) systems. A common strategy for localization is to keep the stimulus of interest constantly centered in the field of view. However, recent results from bats led to the discovery that this common perception is not optimal.

Yovel *et al* (2010) studied how Egyptian fruit bats use echolocation to home-in on a target (Figure 3C, left). They measured the direction of the bats' sonar clicks in darkness and found that, instead of centering the sonar beam onto the target, they alternated the beam to the left and right of the target and thereby pointed the maximum slope of the beam onto the target. As a result, any subsequent motion of the object (relative to the bat) will maximally change the echo intensity and the sign of the change will indicate the direction of motion. They show that this strategy results in a less jittery path (i.e., a smoother trajectory) to the target that is easier and quicker to follow. In fact, the authors proved that this strategy is optimal with respect to localization, though it does pay some cost of possible misdetection (since less sound is emitted by not pointing the beam directly at the target; see Figure 3C, right).

This study shows that even high-level biological processes can reveal interesting and computationally relevant models that can help clarify the logic of biological actions.

## The visual cortex and computer vision

Visual processing in the brain has long been modeled by hierarchical models. By encoding objects at varying levels of sophistication, hierarchies are believed to be capable of recognizing objects under various scales, perspectives, and natural conditions. Different levels of the hierarchy respond to stimuli that become increasingly complex; neurons in the lower levels respond to simple bars and shapes, whereas neurons in the upper levels respond to more complex combinations and whole objects (Serre and Poggio, 2010). Recently, Riesenhuber and Poggio (1999) and others (Serre *et al*, 2007a, b) 'opened up' the brain to see how well such hierarchical models fared to actual neurophysiological data of neuron activation patterns in the inferotemporal cortex in the macaque brain. From these data, they proposed a new hierarchical feed-forward model for object recognition that adheres better to anatomical and physiological constraints and that better matched the corresponding neural recordings. The model is based on a novel way to pool or integrate responses from lower level neurons by identifying the strongest, most active signal present. This makes recognition robust to random noise and fluctuation. By representing features that are invariant to transformations, the model also illustrates how the visual system is able to recognize objects across different scales given only a single view at one scale. From the computational standpoint, the model has been shown to perform well for robust object recognition, action recognition, and face processing compared with many previous (non-biologically inspired) techniques (Jhuang *et al*, 2007; Serre *et al*, 2005, 2007a; Meyers and Wolf, 2008).

## Additional tracking and vision problems

Research in computational neuroscience holds much promise for inspiring new models of visual processing (Serre and Poggio, 2010). While hierarchical feed-forward models were shown to closely replicate neural recordings for immediate

recognition (before eye movement and high-level processing occurs), feed-back projections are also extremely prevalent in the cortex and must be incorporated to model visual perception beyond immediate recognition (Serre and Poggio, 2010). Better understanding of how feed-forward and feed-back connections interface with one another will likely lead to better algorithms and better understanding of how more advanced processing (e.g., memory and learning) occurs in the brain.

Another area where neuroscience may be helpful is in learning. Humans have a remarkable ability to recognize and detect objects very quickly (Thorpe et al, 1996) and using only a few training examples. Most computational classifiers today, however, typically require hundreds or even thousands of training examples to robustly learn discriminative features. It may be possible to develop additional one-shot machine learning algorithms (Li et al, 2006) by studying how visual processing circuits in the brain generalize concepts. There has also been work on drawing parallels between human and machine semisupervised learning (Zhu et al, 2007). In return, these computational models may lead to better theories of human learning that quantify its limitations and biases.

## Engineering-based studies

While our focus in this review is on algorithmic aspects of bidirectional studies, another area that has proliferated in recent years involves the engineering of various systems by tightly linking computational thinking and biology. We briefly mention a few examples here, though as mentioned above these deserve a separate review. One example of such an approach is recent studies in robotics in which researchers aim to mimic several aspects of living organisms for tasks that include flying (Franceschini et al, 2007), crawling (Hatton and Choset, 2010) and distributed control of arm movements (Laschi et al, 2009). These studies often involve careful analysis of how these tasks are performed by organisms ranging from birds to snakes to octopuses. In some cases, such an analysis can also lead to new insights about biological systems. For example, Franceschini et al (2007) recently developed a robotic helicopter that demonstrated how flying insects manage to avoid the ground by using an optic-flow regulator, which shed new light on unexplained findings regarding insects' visually guided performances.

Synthetic biology and DNA computing are another biocomputational area that has received tremendous interest recently. Here, the goal is either to engineer novel systems using biological components (synthetic biology) or to use molecules to perform computations (DNA computing). In many cases, these biological constructs utilize ideas that were previously used when designing computational systems. One of the first studies that demonstrated the computational power of biology was done by Adleman who showed how DNA can be manipulated to solve complex graph-theoretic optimization problems (Adleman, 1994; Păun et al, 2004). More recent studies have focused on the logical control of gene expression (Benenson et al, 2004), the construction of synthetic genomes and cells (Tian et al, 2004; Gibson et al, 2010), and the replication of neural systems using DNA computing (Qian et al, 2011).

## Discussion and conclusions

The mutual relationship between biology and computer science goes back a long time. However, until recently, the application of computational methods to study biological processes and the influence of biological systems on the design of algorithms remained mostly separate. The former involved developing computational methods to solve specific biological problems, while the latter primarily viewed biological systems as motivation to derive new optimization methods.

With the improvement in our ability to study detailed aspects of molecular, cellular, and organism-level biological systems, it is now possible to unite these two directions. This direction involves thinking about biological processes as algorithms that nature has designed to solve difficult real-world problems under a variety of constraints and conditions. By viewing these systems as information processing units, we can better understand their biological properties and their survival mechanisms, and can use the gained knowledge to design and improve computational systems.

Over the last few years, this strategy has been successfully used in a number of different application areas including coordination, networks, and tracking and vision. The list of studies in Table II is not meant to be comprehensive but rather is meant to point out various systems that have been studied this way and additional systems that are ripe for analysis. Table III lists the shared principles that were identified as the basis for some of these studies. Such studies usually begin by identifying a problem addressed by both computational and biological systems. For example, methods to cluster networks have been widely studied in computer science (Schaeffer, 2007; Fortunato, 2010); a number of biological processes, including lateral induction during intercell signaling (Charalambous and Cui, 2008) and decision-making processes by

**Table III** The five primary studies highlighted in this review (rows) each annotated with the principles it shares with computational systems (columns)

| | Distributed | Robust | Networks | Modular | Stochastic | Adaptive |
|---|---|---|---|---|---|---|
| Fly SOP selection | ✔ | ✔ | ✔ | ✔ | ✔ | ✕ |
| Slime mold tunneling | ✔ | ✔ | ✔ | ✕ | ✔ | ✔ |
| Gene regulation | ✔ | ✔ | ✔ | ✔ | ✕ | ✔ |
| Bat localization | ✕ | ✔ | ✕ | ✕ | ✕ | ✔ |
| Brain processing | ✔ | ✔ | ✔ | ✔ | ✕ | ✔ |

Most biological systems operate distributedly and seek a design that is robust and adaptive to changing environments. Networks often serve as a basis for carrying forth interactions and propagating information. These similarities provide a deep basis for the shared analysis of biological processes and computational algorithms.

quorum-sensing bacteria (Wokoma *et al*, 2005; Ren and Meng, 2006), perform a very similar task. The next step, before designing experiments, is to think about the biological system's computational requirements. For example, is the system distributed? What is known about how communication occurs within the system? What qualities (e.g., robustness and efficiency) may the system be trying to optimize? Does randomness or stochasticity have a role? Such questions can help pinpoint particular biological questions regarding the function of the process and may provide a unique point of view for even well-studied processes. Next, a model is often constructed based on the experimental results and this model is used to design an improved solution to the computational problem. In the clustering example, perhaps the model can lead to improved clustering algorithms that operate in noisy and distributive environments.

In this review, we argued that by adopting a 'computational thinking' approach to studying biological processes, we can improve our understanding of biological processes and at the same time improve the design of algorithms.

# Acknowledgements

# Conflict of interest

The authors declare that they have no conflict of interest.

# References

Abelson H, Allen D, Coore D, Hanson C, Homsy G, Knight Jr TF, Nagpal R, Rauch E, Sussman GJ, Weiss R (2000) Amorphous computing. *Commun ACM* **43:** 74–82

Adleman LM (1994) Molecular computation of solutions to combinatorial problems. *Science* **266:** 1021–1024

Afek Y, Alon N, Barad O, Hornstein E, Barkai N, Bar-Joseph Z (2011) A biological solution to a fundamental distributed computing problem. *Science* **331:** 183–185

Afek Y, Alon N, Bar-Joseph Z, Cornejo A, Haeupler B, Kuhn F (2011) Beeping a maximal independent set. *Proceedings of the International Symposium on Distributed Computing (DISC)*. Lecture Notes in Computer Science, Rome, Italy

Alon N, Babai L, Itai A (1986) A fast and simple randomized parallel algorithm for the maximal independent set problem. *J Algorithms* **7:** 567–583

Alon U (2006) *An Introduction to Systems Biology: Design Principles of Biological Circuits*, 1st edn. Boca Raton, FL: Chapman & Hall/CRC

Altschul SF, Gish W, Miller W, Myers EW, Lipman DJ (1990) Basic local alignment search tool. *J Mol Biol* **215:** 403–410

Aspnes J, Ruppert E (2009) An introduction to population protocols. In: B Garbinato, H Miranda, and L Rodrigues (eds), *Middleware for Network Eccentric and Mobile Applications*, pp 97–120. Berlin, Heidelberg: Springer-Verlag

Babaoglu O, Binci T, Jelasity M, Montresor A (2007) Firefly-inspired heartbeat synchronization in overlay networks. *Proceedings of the International IEEE Conference on Self-Adaptive and Self-Organizing Systems (SASO)*, pp 77–86. Washington, DC: IEEE Computer Society

Babaoglu O, Canright G, Deutsch A, Di Caro GA, Ducatelle F, Gambardella LM, Ganguly N, Jelasity M, Montemanni R, Montresor A, Urnes T (2006) Design patterns from biology for distributed computing. *ACM Trans Auton Adapt Syst* **1:** 26–66

Balaji S, Iyer LM, Aravind L, Babu MM (2006) Uncovering a hidden distributed architecture behind scale-free transcriptional regulatory networks. *J Mol Biol* **360:** 204–212

Beckers R, Holland O, Deneubourg JL (1994) From local actions to global tasks: stigmergy and collective robots. *Proceedings of the Workshop on Artificial Life*, pp 181–189. Cambridge, MA: MIT Press

Benenson Y, Gil B, Ben-Dor U, Adar R, Shapiro E (2004) An autonomous molecular computer for logical control of gene expression. *Nature* **429:** 423–429

Bernard G, Ben-othman J, Bouganim L, Canals G, Chabridon S, Defude B, Ferrié J, Gançarski S, Guerraoui R, Molli P, Pucheral P, Roncancio C, Serrano-Alvarado P, Valduriez P (2004) Mobile databases: a selection of open issues and research directions. *ACM SIGMOD Rec* **33:** 78–83

Bishop CM (1996) *Neural Networks for Pattern Recognition*, 1st edn. USA: Oxford University Press

Brin S, Page L (1998) The anatomy of a large-scale hypertextual web search engine. *Comput Netw ISDN Syst* **30:** 107–117

Charalambous C, Cui S (2008) A bio-inspired distributed clustering algorithm for wireless sensor networks. *Proceedings of the International Conference on Wireless Internet*, pp 6:1–6:8. Brussels, Belgium: ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering)

Chazelle B (2009) Natural algorithms. *Proceedings of the Annual ACM Symposium on Discrete Algorithms (SODA)*, pp 422–431. Philadelphia, PA: Society for Industrial and Applied Mathematics

Chen BL, Hall DH, Chklovskii DB (2006) Wiring optimization can relate neuronal structure and function. *Proc Natl Acad Sci USA* **103:** 4723–4728

Ciliberti S, Martin OC, Wagner A (2007) Robustness can evolve gradually in complex regulatory gene networks with varying topology. *PLoS Comput Biol* **3:** e15

Committee on Frontiers at the Interface of Computing, Biology, and National Research Council (2005) *Catalyzing Inquiry at the Interface of Computing and Biology*, 1st edn. Washington, DC: National Academies Press

Conradt L (2011) Collective behaviour: when it pays to share decisions. *Nature* **471:** 40–41

Coore D, Nagpal R, Weiss R (1997) *Paradigms for structure in an amorphous computer*. Technical report. Massachusetts Institute of Technology: Cambridge, MA, USA.

Daliot A, Dolev D, Parnas H (2003) Self-stabilizing pulse synchronization inspired by biological pacemaker networks. *Proceedings of the International Conference on Self-Stabilizing Systems (SSS 2003)*, pp 32–48. Berlin, Heidelberg: Springer-Verlag

Daliot A, Dolev D, Parnas H (2004) Brief announcement: linear time byzantine self-stabilizing clock synchronization. *Proceedings of the ACM Symposium on Principles of Distributed Computing (PODC)*, pp 379–379. New York, NY: ACM

Deneubourg JL, Goss S, Franks N, Sendova-Franks A, Detrain C, Chrétien L (1990) The dynamics of collective sorting robot-like ants and ant-like robots. From animals to animats. *Proceedings of the International Conference on Simulation of Adaptive Behavior*, pp 356–363. Cambridge, MA: MIT Press

Dorigo M, Birattari M, Stützle T (2006) Ant colony optimization—artificial ants as a computational intelligence technique. *IEEE Comput Intell Mag* **1:** 28–39

Easley DA, Kleinberg JM (2010) *Networks, Crowds, and Markets—Reasoning About a Highly Connected World*. New York, NY: Cambridge University Press

Farkash-Amar S, Lipson D, Polten A, Goren A, Helmstetter C, Yakhini Z, Simon I (2008) Global organization of replication time zones of the mouse genome. *Genome Res* **18:** 1562–1570

Feitelson DG (1990) Distributed hierarchical control for parallel processing. *IEEE Computer* **23:** 65–77

Ferber J (1999) *Multi-Agent Systems: An Introduction to Distributed Artificial Intelligence*. Harlow, UK: Addison-Wesley Professional

Fewell JH (2003) Social insect networks. *Science* **301:** 1867–1870

Forrest S, Perelson AS, Allen L, Cherukuri R (1994) Self-nonself discrimination in a computer. *Proceedings of the International Symposium on Security and Privacy*, pp 202–212. Washington, DC: IEEE Computer Society

Fortunato S (2010) Community detection in graphs. *Phys Rep* **486:** 75–174

Franceschini N, Ruffier F, Serres J (2007) A bio-inspired flying robot sheds light on insect piloting abilities. *Curr Biol* **17:** 329–335

Garrett SM (2005) How do we evaluate artificial immune systems? *Evol Comput* **13:** 145–177

Ghose K, Horiuchi TK, Krishnaprasad PS, Moss CF (2006) Echolocating bats use a nearly time-optimal strategy to intercept prey. *PLoS Biol* **4:** e108

Gibson DG, Glass JI, Lartigue C, Noskov VN, Chuang RY, Algire MA, Benders GA, Montague MG, Ma L, Moodie MM, Merryman C, Vashee S, Krishnakumar R, Assad-Garcia N, Andrews-Pfannkoch C, Denisova EA, Young L, Qi ZQ, Segall-Shapiro TH, Calvey CH *et al* (2010) Creation of a bacterial cell controlled by a chemically synthesized genome. *Science* **329:** 52–56

Gitter A, Siegfried Z, Klutstein M, Fornes O, Oliva B, Simon I, Bar-Joseph Z (2009) Backup in gene regulatory networks explains differences between binding and knockout results. *Mol Syst Biol* **5:** 276

Glass L (2001) Synchronization and rhythmic processes in physiology. *Nature* **410:** 277–284

Goldberg DE (1989) *Genetic Algorithms in Search, Optimization and Machine Learning*, 1st edn. Boston, MA: Addison-Wesley Longman Publishing Co., Inc.

Greensmith J, Aickelin U, Cayzer S (2005) Introducing dendritic cells as a novel immune-inspired algorithm for anomaly detection. In *Artificial Immune Systems* **3627**, pp 153–167. Berlin/Heidelberg: Springer

Gu Z, Steinmetz LM, Gu X, Scharfe C, Davis RW, Li WH (2003) Role of duplicate genes in genetic robustness against null mutations. *Nature* **421:** 63–66

Guerraoui R, Schiper A (1996) Fault-tolerance by replication in distributed systems. *Proceedings of the Ada-Europe International Conference on Reliable Software Technologies*, pp 38–57. London, UK: Springer-Verlag

Gusfield D (1997) *Algorithms on Strings, Trees, and Sequences: Computer Science and Computational Biology*. New York, NY: Cambridge University Press

Hatton RL, Choset H (2010) Generating gaits for snake robots: annealed chain fitting and keyframe wave extraction. *Auton Robots* **28:** 271–281

Hebb DO (1949) *The Organization of Behavior: A Neuropsychological Theory*. New York, NY: Wiley

Hogeweg P (2011) The roots of bioinformatics in theoretical biology. *PLoS Comput Biol* **7:** e1002021

Hong Y-W, Scaglione A (2005) A scalable synchronization protocol for large scale sensor networks and its applications. *IEEE J Selected Areas Commun* **23:** 1085–1099

Hopfield JJ (1982) Neural networks and physical systems with emergent collective computational abilities. *Proc Natl Acad Sci USA* **79:** 2554–2558

Huttenhower C, Mutungu KT, Indik N, Yang W, Schroeder M, Forman JJ, Troyanskaya OG, Coller HA (2009) Detailing regulatory networks through large scale data integration. *Bioinformatics* **25:** 3267–3274

Istrail S, De-Leon SB, Davidson EH (2007) The regulatory genome and the computer. *Dev Biol* **310:** 187–195

Jeong H, Tombor B, Albert R, Oltvai ZN, Barabasi AL (2000) The large-scale organization of metabolic networks. *Nature* **407:** 651–654

Jhuang H, Serre T, Wolf L, Poggio T (2007) A biologically inspired system for action recognition. *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pp 1–8. Los Alamitos, CA: IEEE Computer Society

Kaern M, Elston TC, Blake WJ, Collins JJ (2005) Stochasticity in gene expression: from theories to phenotypes. *Nat Rev Genet* **6:** 451–464

Kaiser M, Hilgetag CC (2004) Spatial growth of real-world networks. *Phys Rev E Stat Nonlin Soft Matter Phys* **69**(3 Pt 2): 036103

Kashtan N, Alon U (2005) Spontaneous evolution of modularity and network motifs. *Proc Natl Acad Sci USA* **102:** 13773–13778

Kaur A, Van PT, Busch CR, Robinson CK, Pan M, Pang WL, Reiss DJ, DiRuggiero J, Baliga NS (2010) Coordination of frontline defense mechanisms under severe oxidative stress. *Mol Syst Biol* **6:** 393

Kennedy J, Eberhart R (2002) Particle swarm optimization. *Proceedings of the IEEE International Conference on Neural Networks*, Vol. 4, pp 1942–1948, Piscataway, NJ: IEEE Press

Kephart JO, Sorkin GB, Arnold WC, Chess DM, Tesauro GJ, White SR (1995) Biologically inspired defenses against computer viruses. *Proceedings of the International Conference on Artificial Intelligence*, pp 985–996. San Francisco, CA: Morgan Kaufmann

Kitano H (2004) Biological robustness. *Nat Rev Genet* **5:** 826–837

Koene RA, Tijms B, van Hees P, Postma F, de Ridder A, Ramakers GJ, van Pelt J, van Ooyen A (2009) Netmorph: a framework for the stochastic generation of large scale neuronal networks with realistic neuron morphologies. *Neuroinformatics* **7:** 195–210

Kuramoto T, Yamagishi H (1990) Physiological anatomy, burst formation, and burst frequency of the cardiac ganglion of crustaceans. *Physiol Zool* **63:** 102–116

Laschi C, Mazzolai B, Mattoli V, Cianchetti M, Dario P (2009) Design of a biomimetic robotic octopus arm. *Bioinspir Biomim* **4:** 015006

Lee DD, Seung HS (1999) Learning the parts of objects by non-negative matrix factorization. *Nature* **401:** 788–791

Li F-F, Fergus R, Perona P (2006) One-shot learning of object categories. *IEEE Trans Pattern Anal Mach Intell* **28:** 594–611

Li K, Thomas K, Torres C, Rossi L, Shen C-C (2010) Slime mold inspired path formation protocol for wireless sensor networks. *Proceedings of the International Conference on Swarm intelligence (ANTS 2010)*, pp 299–311. Berlin, Heidelberg: Springer-Verlag

Luby M (1985) A simple parallel algorithm for the maximal independent set problem. *Proceedings of the Annual ACM Symposium on Theory of Computing (STOC)*, pp 1–10. New York, NY: ACM

Lucarelli D, Wang I-J (2004) Decentralized synchronization protocols with nearest neighbor communication. *Proceedings of the International ACM Conference on Embedded Networked Sensor Systems (SenSys)*, pp 62–68. New York, NY: ACM

Lynch NA (1996) *Distributed Algorithms*. San Francisco, CA: Morgan Kaufmann Publishers Inc.

Mainwaring A, Culler D, Polastre J, Szewczyk R, Anderson J (2002) Wireless sensor networks for habitat monitoring. *Proceedings of the ACM International Workshop on Wireless Sensor Networks and Applications (WSNA)*, pp 88–97, New York, NY: ACM

Marshall JA, Bogacz R, Dornhaus A, Planque R, Kovacs T, Franks NR (2009) On optimal decision-making in brains and social insect colonies. *J R Soc Interface* **6:** 1065–1074

Marwan W (2010) Amoeba-inspired network design. *Science* **327:** 419–420

Mather JA, Anderson RC, Wood JB (2010) *Octopus: The Ocean's Intelligent Invertebrate*, 1st edn. Portland, OR: Timber Press

Meyers E, Wolf L (2008) Using biologically inspired features for face processing. *Int J Comput Vision* **76:** 93–104

Mirollo RE, Strogatz SH (1990) Synchronization of pulse-coupled biological oscillators. *SIAM J Appl Math* **50:** 1645–1662

Motwani R, Raghavan P (1996) Randomized algorithms. *ACM Comput Surv* **28:** 33–37

Myers CL, Chiriac C, Troyanskaya OG (2009) Discovering biological networks from diverse functional genomic data. *Methods Mol Biol* **563:** 157–175

Nurse P (2008) Life, logic and information. *Nature* **454:** 424–426

Pagliari R, Scaglione A (2011) Scalable network synchronization with pulse-coupled oscillators. *IEEE Trans Mobile Comput* **10:** 392–405

Păun G, Rozenberg G, Salomaa A (2004) *DNA Computing—New Computing Paradigms*. Berlin, Heidelberg: Springer-Verlag

Pomerance A, Ott E, Girvan M, Losert W (2009) The effect of network topology on the stability of discrete state models of genetic control. *Proc Natl Acad Sci USA* **106:** 8209–8214

Qian L, Winfree E, Bruck J (2011) Neural network computation with DNA strand displacement cascades. *Nature* **475:** 368–372

Rendel JM, Sheldon BL, Finlay DE (1965) Canalisation of development of scutellar bristles in drosophila by control of the scute locus. *Genetics* **52:** 1137–1151

Ren H, Meng MMQ-H (2006) Biologically inspired approaches for wireless sensor networks. *Proceedings of the IEEE International Conference on Mechatronics and Automation*, pp 762–768. Washington, DC: IEEE

Riesenhuber M, Poggio T (1999) Hierarchical models of object recognition in cortex. *Nat Neurosci* **2:** 1019–1025

Schaeffer SE (2007) Graph clustering. *Comput Sci Rev* **1:** 27–64

Schatz MC, Delcher AL, Salzberg SL (2010) Assembly of large genomes using second-generation sequencing. *Genome Res* **20:** 1165–1173

Seeley TD (2002) When is self-organization used in biological systems? *Biol Bull* **202:** 314–318

Serre T, Kreiman G, Kouh M, Cadieu C, Knoblich U, Poggio T (2007a) A quantitative theory of immediate visual recognition. *Prog Brain Res* **165:** 33–56

Serre T, Poggio T (2010) A neuromorphic approach to computer vision. *Commun ACM* **53:** 54–61

Serre T, Wolf L, Bileschi S, Riesenhuber M, Poggio T (2007b) Robust object recognition with cortex-like mechanisms. *IEEE Trans Pattern Anal Mach Intell* **29:** 411–426

Serre T, Wolf L, Poggio T (2005) Object recognition with features inspired by visual cortex. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp 994–1000. Washington, DC: IEEE Computer Society

Shapiro L, McAdams HH, Losick R (2009) Why and how bacteria localize proteins. *Science* **326:** 1225–1228

Shapiro LG, Stockman GC (2001) *Computer Vision*. Upper Saddle River, NJ: Prentice Hall

Simeone O, Spagnolini U, Bar-Ness Y, Strogatz S (2008) Distributed synchronization in wireless networks. *IEEE Signal Process Mag* **25:** 81–97

Sporns O (2010) *Networks of the Brain*. Cambridge, MA: MIT Press

Sumbre G, Gutfreund Y, Fiorito G, Flash T, Hochner B (2001) Control of octopus arm extension by a peripheral motor program. *Science* **293:** 1845–1848

Szeliski R (2010) Computer vision: algorithms and applications. *Online* **5:** 979

Tero A, Takagi S, Saigusa T, Ito K, Bebber DP, Fricker MD, Yumiki K, Kobayashi R, Nakagaki T (2010) Rules for biologically inspired adaptive network design. *Science* **327:** 439–442

Theraulaz G, Bonabeau E (1995) Coordination in distributed building. *Science* **269:** 686–688

Thorpe S, Fize D, Marlot C (1996) Speed of processing in the human visual system. *Nature* **381:** 520–522

Tian J, Gong H, Sheng N, Zhou X, Gulari E, Gao X, Church G (2004) Accurate multiplex gene synthesis from programmable DNA microchips. *Nature* **432:** 1050–1054

Trapnell C, Salzberg SL (2009) How to map billions of short reads onto genomes. *Nat Biotechnol* **27:** 455–457

Vazirani V (2004) *Approximation Algorithms*. Berlin, Heidelberg: Springer

Ward AJ, Herbert-Read JE, Sumpter DJ, Krause J (2011) Fast and accurate decisions through collective vigilance in fish shoals. *Proc Natl Acad Sci USA* **108:** 2312–2315

Watanabe S, Tero A, Takamatsu A, Nakagaki T (2011) Traffic optimization in railroad networks using an algorithm mimicking an amoeba-like organism, physarum plasmodium. *Biosystems* **105:** 225–232

Werfel J, Nagpal R (2006) Extended stigmergy in collective construction. *IEEE Intell Syst* **21:** 20–28

Werner-Allen G, Lorincz K, Welsh M, Marcillo O, Johnson J, Ruiz M, Lees J (2006) Deploying a wireless sensor network on an active volcano. *IEEE Internet Comput* **10:** 18–25

Werner-Allen G, Tewari G, Patel A, Welsh M, Nagpal R (2005) Firefly-inspired sensor network synchronicity with realistic radio effects. *Proceedings of the International ACM Conference on Embedded Networked Sensor Systems (SenSys)*, pp 142–153. New York, NY: ACM

Winfree AT (1967) Biological rhythms and the behavior of populations of coupled oscillators. *J Theor Biol* **16:** 15–42

Wing JM (2006) Computational thinking. *Commun ACM* **49:** 33–35

Wokoma I, Shum LL, Sacks L, Marshall I (2005) A biologically-inspired clustering algorithm dependent on spatial data in sensor networks. *Proceedings of the European Workshop on Wireless Sensor Networks*, pp 386–390. Washington, DC: IEEE

Yovel Y, Falk B, Moss CF, Ulanovsky N (2010) Optimal localization by pointing off axis. *Science* **327:** 701–704

Zhong Q, Simonis N, Li QR, Charloteaux B, Heuze F, Klitgord N, Tam S, Yu H, Venkatesan K, Mou D, Swearingen V, Yildirim MA, Yan H, Dricot A, Szeto D, Lin C, Hao T, Fan C, Milstein S, Dupuy D *et al* (2009) Edgetic perturbation models of human inherited disorders. *Mol Syst Biol* **5:** 321

Zhu X, Rogers T, Qian R, Kalish C (2007) Humans perform semi-supervised classification too. *Proceedings of the Conference on Artificial Intelligence (AAAI)*, pp 864–869. Menlo Park, CA: AAAI Press