Integrated Information Increases with Fitness in the Evolution of Animats

Jeffrey A. Edlund[1], Nicolas Chaumont[2,3], Arend Hintze[2,3,4], Christof Koch[1,5,6], Giulio Tononi[7], Christoph Adami[2,3,8]*

1 Computation & Neural Systems, California Institute of Technology, Pasadena, California, United States of America, 2 Keck Graduate Institute of Applied Life Sciences, Claremont, Pasadena, California, United States of America, 3 BEACON Center for the Study of Evolution in Action, Michigan State University, East Lansing, Michigan, United States of America, 4 Computer Science and Engineering, Michigan State University, East Lansing, Michigan, United States of America, 5 Department of Brain and Cognitive Engineering, Korea University, Seoul, Korea, 6 Allen Institute for Brain Sciences, Seattle, Washington, United States of America, 7 Department of Psychiatry, University of Wisconsin, Madison, Wisconsin, United States of America, 8 Microbiology & Molecular Genetics, Michigan State University, East Lansing, Michigan, United States of America

Abstract

One of the hallmarks of biological organisms is their ability to integrate disparate information sources to optimize their behavior in complex environments. How this capability can be quantified and related to the functional complexity of an organism remains a challenging problem, in particular since organismal functional complexity is not well-defined. We present here several candidate measures that quantify information and integration, and study their dependence on fitness as an artificial agent ("animat") evolves over thousands of generations to solve a navigation task in a simple, simulated environment. We compare the ability of these measures to predict high fitness with more conventional information-theoretic processing measures. As the animat adapts by increasing its "fit" to the world, information integration and processing increase commensurately along the evolutionary line of descent. We suggest that the correlation of fitness with information integration and with processing measures implies that high fitness requires both information processing as well as integration, but that information integration may be a better measure when the task requires memory. A correlation of measures of information integration (but also information processing) and fitness strongly suggests that these measures reflect the functional complexity of the animat, and that such measures can be used to quantify functional complexity even in the absence of fitness data.

Introduction

Complexity is visible in most scientific disciplines: mathematicians, physicists, biologists, chemists, engineers and social scientists all developed measures to characterize the complexity that they perceive in their systems, borrowing tools from each other but rarely if ever agreeing on a measure that could be used by all of them. Because the objects that each of these disciplines are most concerned with are so different, ranging from mathematical problems and computer programs over physical, chemical, or biological structures, to systems and networks of interacting agents, a convergence of quantitative measures of complexity is perhaps not likely. However, a universal framework that would be capable of adapting its notion to the specific discipline it is applied to would be a welcome trend. Complexity measures abound, but exhaustive reviews are few. A good introduction to the dynamical systems approach to complexity is Ref. [1], but it does not cover biological applications. The overviews [2–4] focus on the complexity of biological sequences but not on their structure, and mostly ignore the complexity of networks. Neural complexity measures are reviewed in [5].

Among the different measures of complexity, some attempt to quantify the structure [6–13], others the sequence giving rise to that structure [14–19], and others again the function of the sequence or system [20–22]. All these studies attempt to capture "that which increases when self-organizing systems organize themselves" [23] (a non-exhaustive list is presented in Ref. [24]). Increasingly, measures based on information theory are being used to quantify the complexity of living systems, because information provides its owner an obvious fitness advantage compared to those without information by conferring the ability to make predictions about the environment they operate in [25–27]. In particular Rivoire and Leibler [27] study statistical measures based in information theory that maximize the fitness of agents that respond to variable environments, but they do not study evolution. Information-theoretic measures of complexity are reviewed in [28] and applications to graphs in [29].

Here, we study how information-theoretic measures of complexity could be applied to capture the complexity of nervous systems [5,30], or more generally speaking, any structure controlling a perception-action cycle. In the absence of any well accepted definition of complexity, we study the correlation of different

Author Summary

Intelligent behavior encompasses appropriate navigation in complex environments that is achieved through the integration of sensorial information and memory of past events to create purposeful movement. This behavior is often described as "complex", but universal ways to quantify such a notion do not exist. Promising candidates for measures of functional complexity are based on information theory, but fail to take into account the important role that memory plays in complex navigation. Here, we study a different information-theoretic measure called "integrated information", and investigate its ability to reflect the complexity of navigation that uses both sensory data and memory. We suggest that measures based on the integrated-information concept correlate better with fitness than other standard measures when memory evolves as a key element in navigation strategy, but perform as well as more standard information processing measures if the robots navigate using a purely reactive sensor-motor loop. We conclude that the integration of information that emanates from the sensorial data stream with some (short-term) memory of past events is crucial to complex and intelligent behavior and speculate that integrated information–to the extent that it can be measured and computed–might best reflect the complexity of animal behavior, including that of humans.

measures to *organismal fitness*, following the intuition that a well-defined measure of control structure complexity should increase during adaptation [20]. Fitness is a quantitative measure that predicts the long-term success of a lineage [31,32], and is given by the *expected number of offspring* of an average representative with the given genotype. Unfortunately, this is only a quantitative measure for the simplest of organisms where the expected number of offspring can be determined from the replication rate, or in direct competition experiments (see, e.g., [33]). For more complex organisms, relative fitness can only be estimated in hindsight, and cannot be used as a proxy for organism complexity. However, if we evolve control structures *in silico* where complete fitness information is available, we can use fitness (within a niche) as an independent arbiter of putative information-based measures of complexity: any measure that does not increase as the organism learns to exploit its environment is unlikely to reflect complex information processing. Because in this type of evolution experiment the number of offspring is directly proportional–on average–to the *performance* of the organism in a task critical to its survival, we here study the correlation of complexity directly with performance or function.

Note that because fitness necessarily refers to the environment (it measures how well the organism "fits" its niche by exploiting the niche's attributes), fitness cannot be used to compare organism complexity across niches (such as attempting to compare an elephant and an ant in terms of their fitness), but it does reveal functional differences between types that are due to efficiencies of exploiting the same environment. For biological organisms that occupy the same niche, that is, "make a living" in the same manner, relative fitness should correlate with relative functional complexity. Is it true that given a constant environment the more complex organism is necessarily more fit? Answering this question in the affirmative clearly biases our notion of complexity: only useful characters are deemed complex, useless ones are not. While such a bias may be restrictive for structural complexity, it is not so for information-theoretic measures of complexity, as information (if it can be used to reduce uncertainty) will *always* be useful: if it were not, it should be called entropy instead [25,34].

Predictive information

Perhaps the best known information-based measure of functional complexity is "predictive information" [35], which quantifies the amount of information that can be extracted from sensorial data in order to select actions that are useful to the organism. In this manner, predictive information is able to separate out those features of the sensorial data that are relevant for behavior, and quantifies the amount of information processed by the organism. Predictive information has also been proposed as a measure of complexity of function [35].

If we describe a control network's input variables ("sensors", or "stimuli") at time t by the random variable S_t and the output variables ("motors", or "response") at that time by R_t, then the shared information (used for prediction) is [35]

$$I_{\text{pred}} = I(S_t : R_{t+1}) = H(R_{t+1}) - H(R_{t+1}|S_t) = \sum_{s,r} p(s_t,r_{t+1}) \log \frac{p(s_t,r_{t+1})}{p(s_t)p(r_{t+1})} , \qquad (1)$$

where $\text{Pr}(S_t = s_t) \equiv p(s_t)$ and $\text{Pr}(R_t = r_t) \equiv p(r_t)$ are the probability distributions of the sensor and response variables at time t, respectively, and $p(s_t,r_{t+1})$ is the joint probability distribution of the sensor and response variables "in the future and the present" [35] (we use the binary logarithm throughout and assume that the network evolves along discrete time steps). I_{pred} characterizes the capacity of the control system to predict the future one time step ahead, using the present sensorial information. Essentially, it quantifies the correlation between inputs and outputs, and can be thought of as the Kullback-Leibler divergence (or relative entropy) between the full probability distribution $p(s_t,r_{t+1})$ and the product of the independent ones, $p(s_t)p(r_{t+1})$.

Note that for Markov processes, the one-step shared entropy (1) is equal to the shared entropy between the entire past and the entire future (see [36], Appendix A.1), while this is not true for processes that can use memory. Predictive information was previously used to characterize the complexity of autonomous robot behavior without memory in Ref. [36] (see also Text S1). If the control structure is not purely reactive and uses information encoded in internal nodes to integrate sensorial information streams, we will need complexity measures that move beyond predictive information [27].

Integrated information

A fundamental and unique design principle of nervous systems is their extraordinary degree of integration among highly-specialized modules [5,37,38]. Functional integration is achieved by an extended network of intra- and inter-areal connections, and is reflected in dynamically shifting patterns of synchronization. A precise way to measure a system's capacity to integrate information was developed recently [39,40], and applied to small, simple example networks. This measure, called Φ and measured in bits, is based on the notion that information integration is achieved by architectural designs that give rise to a single, functionally unified complex (high integration) while ensuring that such a complex has a very large repertoire of discriminable states (high information). Φ captures to what extent, informationally, the whole is more than the sum of its parts, and cannot therefore be reduced to those parts. In this sense, Φ represents the synergy of the system. Before introducing Φ proper, we define a few related quantities.

In order to study information integration, we have to define the information processed by the entire network, not just the sensors and motors as in Eq. (1). Let us represent the system as a joint

random variable $X = X^{(1)}X^{(2)}\cdots X^{(n)}$, where the $X^{(i)}$ represent the *elements* of the system (the nodes of a control structure, such as a neuronal network). The random variable X evolves as the system progresses forward in time, i.e., $X(t=0) \equiv X_0 \to X_1 \to X_2 \to \cdots X_t$, and each variable X_t is described by a probability distribution $p(x_t)$ to be found in states x_t (here, we will restrict ourselves to binary random variables). At the same time, each node i of the system has a time progression $X_0^{(i)} \to X_t^{(i)}$, and each variable $X_t^{(i)}$ is described by a probability distribution $p(x_t^{(i)})$. In the following, we formally define measures of information integration through t time steps (from $0 \to t$), but later focus on the computationally more accessible integration through a single average step from $t \to t+1$.

The amount of information that is processed by the *entire system* through t time steps is given by

$$I(X_0 : X_t) = \sum_{x_0, x_t} p(x_0, x_t) \log \frac{p(x_0, x_t)}{p(x_0)p(x_t)} \ . \qquad (2)$$

where $p(x_0)$ and $p(x_t)$ are the probability distributions of the system at time $t=0$ and t respectively, and $p(x_0, x_t)$ is their joint distribution. This measure reduces to the predictive information Eq. (1) for Markov processes connecting only sensor and response nodes, that is, if there are no internal (or hidden) variables.

One way to measure information integration is to ask how much information is processed by the system above and beyond what is processed by the individual nodes or groups of nodes (modules). To do this, we introduce a *partition* of the network into k parts, $P = \{P^{(1)}, P^{(2)}, \cdots, P^{(k)}\}$, where each $P^{(i)}$ is a part of the network: a non-empty set of nodes with no overlap between parts that completely tile the network. We can then define a quantity that measures how much the information processed by the entire network is more than the information processed by all the parts in this particular partition as follows.

Let $I(P_0^{(i)} : P_t^{(i)})$ be the information processed by the ith part as the system progresses from time 0 to time t. Then, the synergistic information SI processed by the network X given a partition P quantifies the extent to which the entire processed information is a sum of the information processed by the system's parts, and is calculated as:

$$SI(X_0 \to X_t | P) = I(X_0 : X_t) - \sum_{i=1}^{k} I(P_0^{(i)} : P_t^{(i)}) \ . \qquad (3)$$

From an information-theoretic point of view, the synergistic information measures the excess amount of information that can be encoded in a "multiple access" channel with correlated sources and a joint decoder [41] over and above what each of the individual channels (the parts of the partition $P^{(i)}$) can encode separately. A measure related to the synergistic information is the "effective information" EI:

$$EI(X_0 \to X_t | P) = \sum_{i=1}^{k} H(P_0^{(i)} | P_t^{(i)}) - H(X_0 | X_t) \ . \qquad (4)$$

Here, $H(P_0^{(i)} | P_t^{(i)})$ is the conditional entropy of partition $P_0^{(i)}$ given the state of that partition t time steps later, and $H(X_0 | X_t)$ is the conditional entropy of the entire system X at time step $t=0$ given the state of that system t steps later (see also Text S3). The quantity (4) is the average over network states at time t (states x_t) of the quantity called the "effective information across a partition P" in Ref.

[40]. If the probability distribution governing X_0 is uniform (maximum entropy), the two measures agree: $SI(X_0 \to X_t | P) = EI(X_0 \to X_t | P)$, but they are different in general (see Text S3). Below, we will mostly use Eq. (4).

In order to determine how a network integrates information, we should look for a partition that *minimizes* (4), because it is easy to find a high value of EI by assigning different parts to nodes that are strongly correlated. In essence, looking for the partition that minimizes EI is tantamount to searching for the groups of nodes that are separated from other groups of nodes by a weak informational link [40]. To find this partition, expression (4) needs to be normalized because otherwise the partition that minimizes (4) will almost always be the one that divides a network of N parts into one with $N-1$ parts and a single other node [40]. We define the "Minimum Information Partition" (or 'MIP') as that partition that minimizes a *normalized EI*:

$$\mathrm{MIP}_0 = \arg\min_P \frac{EI(X_0 \to X_t | P)}{(k-1)\min_i \left[H_{\max}(P_0^{(i)}) \right]} \ , \qquad (5)$$

where $H_{\max}(P_0^{(i)})$ is the maximum entropy of the ith partition $P_0^{(i)}$. If the neurons are binary, then $H_{\max}(P_0^{(i)})$ is just the number of neurons in partition i. Armed with this definition of the MIP, our measure of information integration is:

$$\Phi_0 = EI(X_0 \to X_t | P = \mathrm{MIP}_0) \ . \qquad (6)$$

Note that Φ_0 represents the average (over all possible final states of the network) of the state-dependent quantity $\Phi(x_t)$ defined previously [40], and the subscript 0 reminds us that the integration is measured from an initial probability distribution at time $t=0$ that is uniform.

The measure can be adapted to characterize the information integration across a single time step simply by defining

$$\Phi = EI(X_t \to X_{t+1} | P = \mathrm{MIP}) \qquad (7)$$

with a commensurately defined MIP:

$$\mathrm{MIP} = \arg\min_P \frac{EI(X_t \to X_{t+1} | P)}{(k-1)\min_i \left[H_{\max}(P_t^{(i)}) \right]} \ , \qquad (8)$$

where $H_{\max}(P_t^{(i)})$ is the maximum entropy of the ith partition at time step t. Note that we have omitted an index t to Φ as defined in Eq. (7) as we assume that for large t Φ becomes stationary: $\Phi_t \to \Phi$ ($t \to \infty$). This MIP, just as the one defined by Balduzzi and Tononi [40], divides the network into disjoint parts that are maximally informationally disparate–those parts that are most independent. As defined here, Φ is equivalent to the recently defined $\tilde{\Phi}_E$ [42], because $EI(X_t \to X_{t+1} | P = \mathrm{MIP})$ is based on the reduction (at time step $t+1$) in the Shannon entropy based on the empirical entropy at time step t, not on the reduction from the maximum entropy at time step 0 as in [40]. Thus, our Eq. (7) is equivalent to Eq. (29) in Ref. [42] (with $\gamma\theta a\varrho 81$ replaced with $\gamma\theta \tilde{a}\varrho 81$), except that we search all partitions rather than just bi-partitions, and the normalization factor of Barrett and Seth uses the largest of the actual entropies of the parts. Because we will measure information integration for time series generated by a moving animat, we will use Eq. (7) to quantify the animat's complexity in what follows.

If networks are small, it is possible to find the MIP by brute-force testing all possible partitions. The number of partitions of n nodes is the nth Bell number, B_n [43]. Searching across all partitions is exceedingly expensive and scales faster than exponential. For example, $B_3 = 5$, $B_{10} = 115,975$, and $B_{16} \approx 1.05 \times 10^{10}$. For networks of realistic size, search heuristics will be the only way to find the MIP: for the nematode *C. elegans*, for example, $n = 302$ [30], and the number of partitions of this network is the absurdly large number $B_{302} \approx 4.8 \times 10^{457}$. Here, the largest networks we analyze have 12 nodes, but we have been able to calculate Φ for networks with up to 18 nodes using a fast exact algorithm that does not store all the partitions.

Main complex

A system composed of a large network together with a single disconnected unit will always have $\Phi = 0$, because minimizing over all partitions finds the informational disconnect between the network and the disconnected node, and the minimum effective information between these parts is zero [40]. A measure that captures information processing that is synergistic without being trivial can be obtained by defining the network's computational *proper complex* [44] as a subset S of (joint) random variables within the system X ($S \in X$) that maximizes Φ over all subsets *and* supersets, that is:

If $\Phi(S)$ is defined as the Φ of subset S, then

$$S \subseteq X \text{ is a proper complex if } \left(\begin{array}{l} \Phi(S) \geq \Phi(R) \ \forall \ R \subset S \\ \Phi(S) > \Phi(T) \ \forall \ T \supset S. \end{array} \right) \quad (9)$$

Each network can have several (proper) complexes, with smaller complexes of higher Φ embedded within larger complexes of lower Φ. We define the *proper main complex* as the subset associated with largest Φ values over all subsets of the entire system. We denote the information integration in the proper main complex as Φ_{MC}. A simple network with MIP and main complex identified is shown in Fig. 1.

Other integration measures

Among all possible partitions, the "atomic partition" that partitions the network into its individual nodes, plays an important role. For example, we can define the information processed by the network above and beyond the information processed by the individual nodes as

$$SI_{atom} = I(X_t : X_{t+1}) - \sum_{i=1}^{n} I(X_t^{(i)} : X_{t+1}^{(i)}), \quad (10)$$

where the first term is the total processed information I_{total}, defined as

$$I_{total} = I(X_t : X_{t+1}) = H(X_t) - H(X_t | X_{t+1}), \quad (11)$$

The negative Eq. (10) has previously been used to quantify the redundancy of information processing of a neural network [45,46], see also [47]. Incidentally, Barlow has long argued that reducing redundancy (and thus compressing the sensorial information stream maximally) is the main purpose of the structure of the sensorial information-processing system [48], and we would then, if I_{total} is fixed, expect a maximization of fitness to go hand-in-hand with a minimization of SI_{atom} and therefore a maximization of redundancy.

I_{total} measures the shared entropy between the system at adjacent time points, and is a useful measure to determine whether
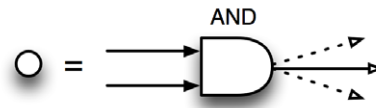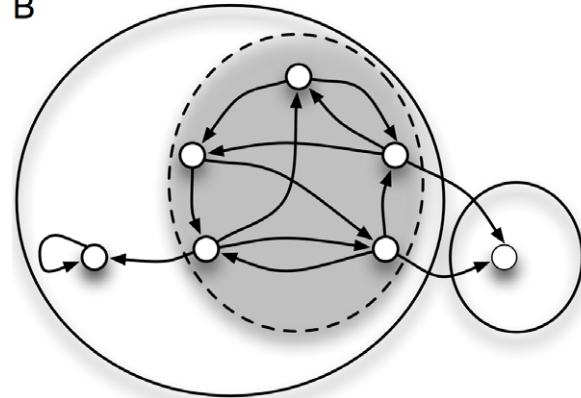
Figure 1. Exemplar MIP and main complex. A: The logical units are AND gates with multiple outputs (each output is the AND of the two inputs). **B**: A network of seven such units (877 distinct possible partitions). The MIP for the entire system (solid lines) is a bi-partition, and the main complex (dashed line, shaded area) consists of five units. We compute $\Phi = 0.269$ bits for the entire network, while $\Phi_{MC} = 1.327$ bits.
doi:10.1371/journal.pcbi.1002236.g001

an increase in Φ is due solely to increased information processing by the entire network (resulting in an increased I_{total}) rather than the effective integration of that information. Writing $I(X_t^{(i)} : X_{t+1}^{(i)}) = H(X_t^{(i)}) - H(X_t^{(i)} | X_{t+1}^{(i)})$ for each node i, we see that

$$SI_{atom} = -H(X_t | X_{t+1}) + \sum_{i=0}^{n} H(X_t^{(i)} | X_{t+1}^{(i)}) - \mathcal{I}, \quad (12)$$

where n is the number of individual nodes in the network and where

$$\mathcal{I} = \sum_{i=1}^{n} H(X_t^{(i)}) - H(X_t). \quad (13)$$

This quantity has been called "multi-information" [47,49]), and was used as a measure of brain complexity called "integration" in [50–52], where the sum was over the components of a network rather than the nodes. Thus, $\mathcal{I}$ is an "atomic" form of the Tononi-Sporns-Edelman (TSE)-complexity [50]. Note that none of the measures discussed in this section should depend on t if t is large enough because we assume that at large times the probability distribution $p(X_t)$ becomes stationary.

The first part in Eq. (12) is nothing but the effective information EI (4), but for the "atomic partition", that is, the partition where each part is given by the individual nodes in the entire network and for $t \rightarrow t + 1$. Thus,

$$SI_{atom} = \Phi_{atom} - \mathcal{I}, \quad (14)$$

where

$$\Phi_{\text{atom}} = EI(X_t \to X_{t+1} | P = P_{\text{atom}}) =$$

$$\sum_{i=0}^{n} H(X_t^{(i)} | X_{t+1}^{(i)}) - H(X_t | X_{t+1}). \tag{15}$$

Eq. (15) may be a particularly useful measure to approximate Φ when a search for MIPs is computationally infeasible. It has previously been introduced under the name "stochastic information" by Ay [53–55]. However, it is neither an upper nor a lower bound on Φ. Because of its construction ($\Phi_{\text{atom}} = SI_{\text{atom}} + I$), it incorporates elements of information processing (the excess information processed, in time, by the system above and beyond the information processed by each of the nodes) as well as integration. In other words, Φ_{atom} encompasses both temporal and spatial synergies of the network.

Results

In order to test how different measures of functional complexity change as a system adapts to function in its world, we evolve controllers for animats [56] that have to solve a task that requires sensory-motor coordination as well as memory. Ay and coworkers tested predictive information Eq. (1) as a measure of system complexity when evolving a simulated autonomous robot to solve a simple maze, and found that I_{pred} reflects the performance of the robot [36]. Lungarella and coworkers used information-based complexity measures to understand how appropriate motor action of embodied agents shapes the signal structure perceived by the agent's sensors [51], and studied the information flow through the control structures [52]. Klyubin and coworkers used mutual information between an agent's starting position and a representation of this information in the agent's memory to evolve sensorimotor control structures, and used measures of synergy to study whether the positional information could be factorized within the sensors [57].

Description of evolutionary system

Our animats are embodied controllers with six binary sensors and two (binary) actuators, as well as four internal bits that can be used for memory or processing (Fig. 2 and Methods).

The controllers are stochastic Markov networks (see, e.g., [58]), that is, networks of random variables with the Markov property, where edges between nodes encode arbitrary fuzzy logic gates. As such, the edges could represent simple binary logic gates or more complex computational units. Because these networks actually encode *decisions*, strictly speaking they are encoding discrete-time stochastic Markov decision processes (MDPs). Fundamentally, our Markov networks are related to the hierarchical temporal memory (HTM) model of neocortical function [59–61] and the HMAX algorithm [62], except that the organization of our stochastic Markov networks need not be strictly hierarchical because it is determined via genetic evolution rather than top-down design (see Methods).

In what follows, the edges connecting the random variables are implemented as *Hidden Markov Gates* (HMGs). Each such gate is a probabilistic finite state machine defined by its input/output structure and state transition probabilities (see Fig. 3A). For example, if '100' was applied to the input state of the HMG in Fig. 3A, '11' is the output with probability p_{43} [$P(100 \to 11) = p_{43}$], while an input '111' generates '01' with probability p_{71} [$P(111 \to 01) = p_{71}$], and so forth. Such a gate can also be represented as its dual graph, where the signal lines become the nodes of the Markov network, and the
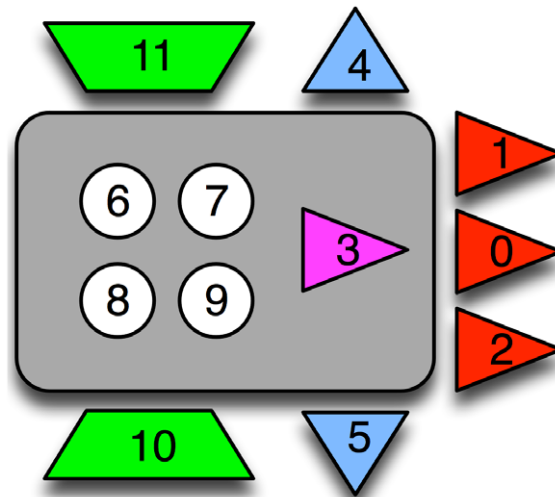

Figure 2. Embodied virtual agent (animat) with six sensors, two actuators, and four internal nodes. The complete animat is described by 12 bits: three front sensors (red triangles; # 0,1 & 2), two lateral collision detectors (blue triangles; # 4 & 5), and a single "door" sensor (magenta, #3) that relays the direction of the next opening in the maze (but only while standing in the door). The actuators (trapezoids; # 10 & 11) encode the actions "move left, move right, move forward, do nothing". The internal nodes (circles; # 6–9) can potentially store states used for internal processing.
doi:10.1371/journal.pcbi.1002236.g002

edges between them represent the computation performed by the HMG (Fig. 3B). In this representation, arrows indicate causal influence via an HMG, so in Fig. 3B for example, variable 4 is influenced by variables 1,2, and 3 (as is variable 3), while variables 1 and 2 only have outgoing arrows: they only influence variables 3 and 4 but are not affected by any other variable.

The $2^n \times 2^m$ probabilities of an n-input and m-output state transition table, as well as how each HMG is connected to other gates, is encoded within a genome that, when read by an interpreter, creates the network (see Methods, Text S2 and Figure S1, as well as Ref. [63] for a similar structure). Populations of genomes are evolved using a standard Genetic Algorithm (but without crossover, see Methods). To calculate the fitness of each genome, the controller generated from the sequence is transplanted into the animat shown in Fig. 2 and tested on its ability to traverse a maze that consists of repeated vertical walls at varying distance to each other, with a single door placed at random locations within the wall. Within each door, a "beacon" indicates the direction to follow for the shortest path to the next door, but this information is erased the moment the animat emerges from the door. Thus, in order to use this information, it has to be stored in memory for later usage. The actual maze has at least 26 walls to traverse before the maze repeats. A section of a typical maze along with an adapted animat's trajectory as well as the states of the memory and motor bits are shown in Fig. 4. Videos S1 to S3 show several movies that depict the motion of the animat, at different evolutionary stages, traveling through the maze.

In each of 64 independent evolution experiments, a population of 300 *initially random* genomes (encoding random controllers, see Text S2) was evolved for 50,000 generations each. We calculate fitness (f) and control fitness f_{ctrl} both for the highest fitness animats at every generation and for genomes on the line of descent (LOD) of the last common ancestor of the population that existed at generation 50,000 (see Methods). The control fitness f_{ctrl} tests the performance of the controller on ten randomly generated
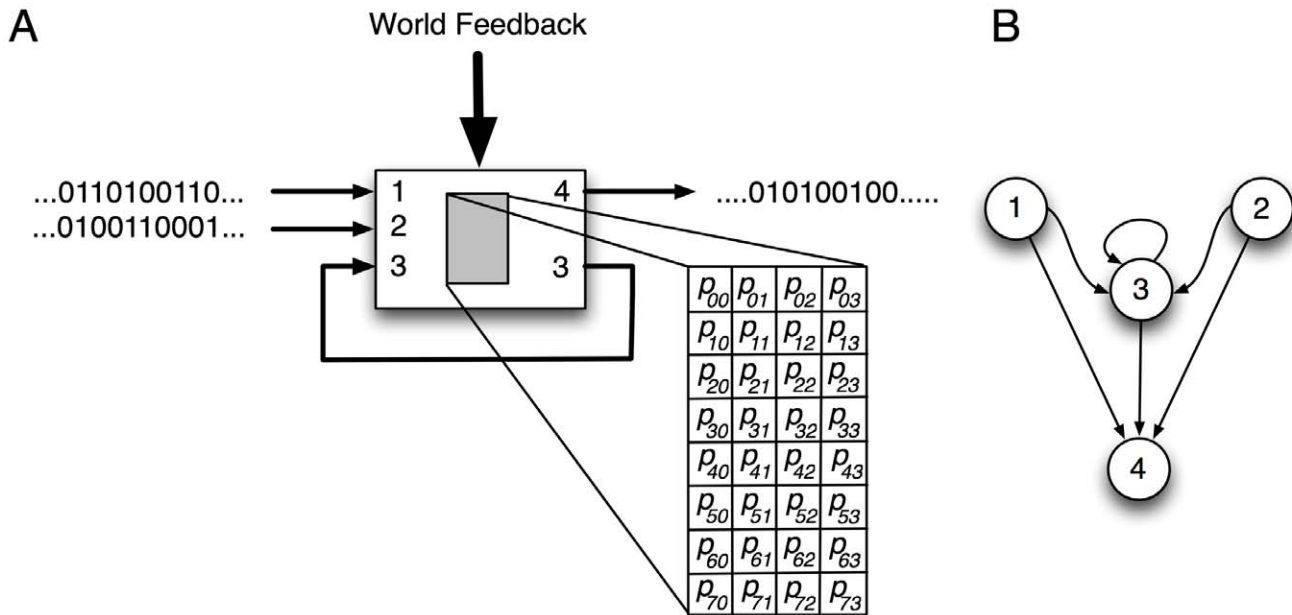
Figure 3. Hidden Markov Gate representation. A: An HMG with three binary input and two output Markov variables, where one of the outputs is fed back into the HMG (a hidden variable). The state transition table has $2^3 \times 2^2$ entries that are determined by genetic evolution (see Methods and Text S2). In the gate shown, bit three is a hidden state and can be used to implement a one-bit memory. In principle, the probabilities in the HMG transition table can also be tuned via reinforcement learning using a signal from the environment ("World Feedback"). However, this capacity is not utilized in the present work. **B**: The "dual" representation of this gate, where the Markov variables are nodes, and the gate connects these via edges. This network is obtained by drawing a directed edge between bits that affect each other causally via the logic gate. Because bit 3 feeds back to itself, for example, it is given the same identifier and there is a directed arrow from bit 3 to itself as well as bit 3 to bit 4. See Text S2 and Fig. S1 for details on the genetic encoding and network visualization of HMGs.
doi:10.1371/journal.pcbi.1002236.g003

mazes that the animat has never before encountered (see Methods), in order to test whether the animat evolved the navigation principles or simply adapted to a particular instance of the problem.

The LOD recapitulates the evolutionary history of the population, and allows a reconstruction of the path taken mutation by mutation. Fig. 5 shows the evolution of fitness and control fitness for one of 64 experiments [panel (A) shows the fitness on the
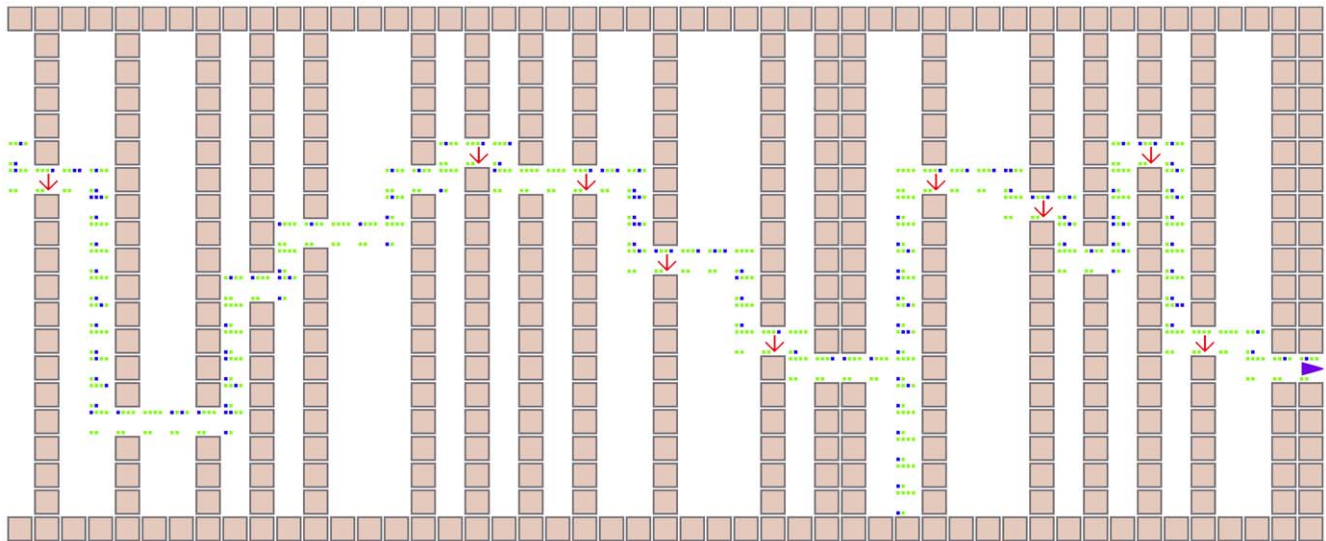
Figure 4. Maze structure and animat trajectory. Part of one of the test mazes, along with the trajectory of an adapted animat as well as a view of the animat's brain (the four internal nodes 6–9, top four pixels in each animat location) and the motor outputs (bottom two pixels). A bit set to '1' is indicated in green, while blue indicates a bit set to '0'. The value of the sensory bits can be inferred from the animat's location. The downward pointing arrow inside a door reminds us that the animat would perceive a '1' on its door sensor at that location (indicating that the next door will be found to the right of the animat's position). If the door is straight ahead or to the left, the door sensor will be set to '0'. The animat's goal is to move as far across the maze as possible (see Methods). Note that this representation does not show when the animat is stationary (waits) or retraces its path.
doi:10.1371/journal.pcbi.1002236.g004
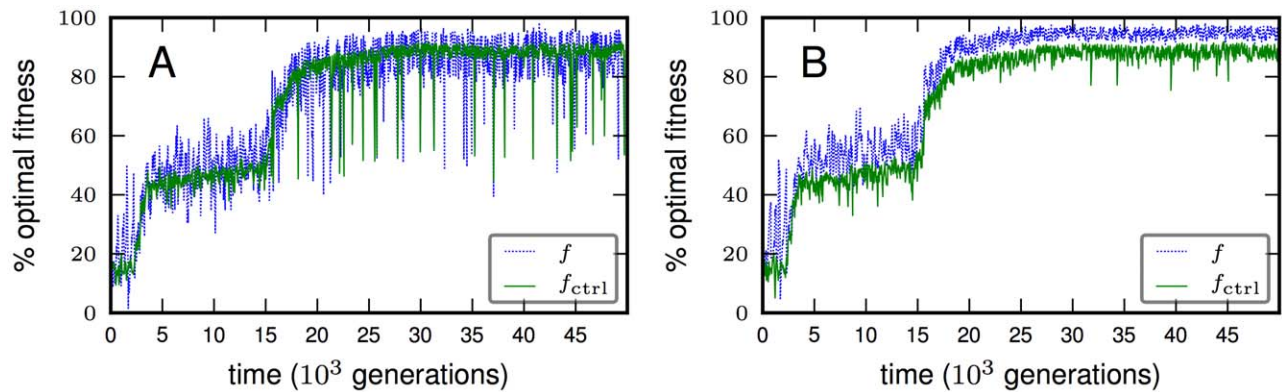
Figure 5. Fitness evolution on the line of descent and in the population. A: Fitness (blue line) and control fitness (green line) for genotypes on the LOD. **B**: Fitness and control fitness for the same run as shown in (A), but for the fittest individual in the population in each generation. Colors as in (A).
doi:10.1371/journal.pcbi.1002236.g005

LOD, while panel (B) shows the corresponding fitness of the best in the population of 300 individuals]. Notice that in Fig. 5B the fitness of the fittest individual is almost always larger than the control fitness for the same individual, while the fitness on the LOD instead scatters around the control fitness, as seen in Fig. 5A. There is a good reason for this difference: in any population, an animat can be fit by chance through having correctly "guessed" the next door position repeatedly. The control fitness removes this element of chance by testing the individual on ten randomly generated mazes. The individuals on the LOD on the other hand are there for a reason and not by chance: their genes have proven themselves in later generations. In the run depicted in Fig. 5 (see also the movies Video S1, S2, S3), the animat evolved a sophisticated (but not perfect) algorithm to navigate the maze, including the use of memory around generation 15,000 to store the doorway bit until the animat reaches the next wall. The wiring diagram of the animat at generation 49,000 is depicted in Fig. 6A. The animat uses only internal node 9 as memory, whose permanency is ensured using auto-feedback. The other nodes

are connected but have no fitness impact whatsoever at this time, as determined by a knock-out analysis (see Methods, data not shown), but may have been useful earlier on. The controller contains a total of 17 HMGs, but only nine HMGs (including two pairs of redundant HMGs) are responsible for this wiring. Of the nine useful HMGs, five have three inputs and one output, the other four HMGs are NOT gates. Note that if more than one HMG output serves as input for another HMG, their values are combined using an OR gate. The animat uses the information from the 3-bit retina, the lateral sensors, as well as the conditional information from the door beacon (sensor 3 in Fig. 6A) effectively by integrating this information within the decision machinery for navigation. The central hub is the network's memory: internal bit 9 is set to 0 if the door beacon is detected in the "on" state (b3 = 1) in a doorway, and to 1 if not. The value of bit 9 is maintained until the animat reaches the next wall. (The value of the door bit itself is erased from the sensor after the animat passes through the door, and therefore cannot be accessed by simply re-reading that value.) At that point the value of bit 9 determines if the animat goes left
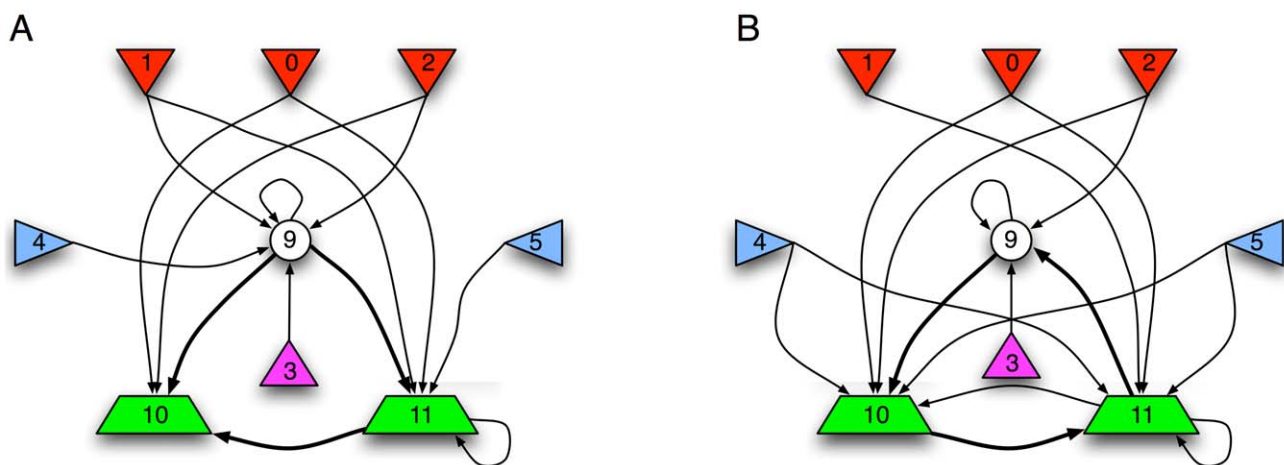
Figure 6. Two evolved HMG networks. The shapes represent the 9 Markov variables (bits) at time $t = 49,000$ that are active in the network (bits 6, 7, and 8 are connected to the network, but are not functional at generation 49,000 and not rendered here). The central feed-forward circuit for navigation is rendered in bold arrows. Color codes and numbering as in Fig. 2. **A**: The network evolved in our focus experiment that achieved 88% of possible fitness. **B**: Another network that evolved in an independent run, and that implements a variant of the hierarchical temporary memory algorithm that creates an expectation of future sensory signals. In contrast to the controller that evolved in panel (A), this one uses a feed-back strategy between memory and motors. This controller achieves 74% of maximal fitness within a random maze environment.
doi:10.1371/journal.pcbi.1002236.g006

(b9 = 1) or right (b9 = 0). Once the animat is moving along a wall, bit 9 is set to 1 and the animat continues moving in the same direction keeping in mind the value of the left motor (b11). In a sense, the motor bit b11 is also used as memory here, as indicated by the auto-feedback. If bit 5 indicates an obstacle to the right, bit 11 is set, which forces bit 10 off in turn. If bit 4 indicates an obstacle to the left on the other hand, bit 9 is set to 0 which causes bit 11 to turn off and bit 10 to turn on. Once the animat is in front of the next doorway, it moves forward through the door. Thus, we see that this animat effectively uses the integration of different streams of information (door sensor, retina, lateral sensors, and current state of motion) to compute behavior that is appropriate in the given environment most of the time. Reaching 88% of "maximal" fitness is fairly remarkable, as a hand-written optimal controller reaches only 93% of maximal fitness (data not shown) because we force our controllers to be minimally stochastic.

In another run that achieved a fitness of 74%, a related but fundamentally different algorithm evolved to achieve almost the same functionality (wiring depicted in Fig. 6B). The central part of this algorithm, which is a version of the "hierarchical temporal memory algorithm" [59], is implemented by a feedback loop between the motors 10 and 11 and internal bit 9 (bold arrows in Fig. 6B), as opposed to the feed-forward loop seen in Fig. 6A. Because the animat can read from its motor bits, it can keep track of how it is currently moving, and make decisions based on this state as well as the state of the internal variable bit 9. Temporal memory is achieved by creating a basic expectation (bit 9 set to one) of encountering a door beacon that will be pointing it to the left (bit 3 = 0). If instead it encounters a door pointing to the right (bit 3 = 1), it changes that expectation (bit 9 = 0) and maintains it in memory until it moves in the correct (right) direction. Once this happens, the expectation is changed back to anticipating a beacon pointing it to the left, but the animat does not immediately react to this expectation because bit 9 is ignored as long as the animat moves to the right.

Let us now look at our information-theoretic constructions as a function of evolutionary time. The quantity Φ_{MC} is expensive to calculate so they and other measures were calculated along the LOD of each population every 500 generations up to generation 50,000. Each genome was evaluated by testing the controller it spawns for 1,000 world-time steps in 10 control mazes (each tested 10 times, see Methods) in order to even out chance achievements (animates can achieve high fitness by chance due to the stochastic nature of their controllers). We show the evolution of three

information integration and three information processing measures over time (for the same run whose fitness evolution is depicted in Fig. 5) in Fig. 7. As fitness increases, all measures we plot here increase at first, but quickly become stagnant when fitness flattens out (see Fig. 5). Important changes are apparent in all measures when the capacity to use the door beacon for navigation emerges around generation 15,000. To see differences in the measure's abilities to predict fitness, we need to analyze how well these complexity proxies correlate with fitness across our set of 64 runs.

Statistics

In order to test whether fitness correlates with a complexity proxy, we calculate the (nonparametric) Spearman rank correlation coefficient of the "final" fitness (the fitness of the genome at generation 49,000 on the LOD, see Methods) with the value of that variable measured at generation 49,000. We chose generation 49,000 as final time because the organism from this generation is guaranteed to represent the common line of descent of the 300 individuals in any particular run (see Methods). While we have correlation data of fitness with each variable along the LOD every 500 generations for each run, these points are not independent, and therefore cannot be used in order to assess the statistical significance of the correlation. The correlation of final fitness (across 64 independent samples) with each of the different information-theoretical candidates for functional complexity is shown in Fig. 8. Note that the highest control fitness achieved across the 64 runs is $f_{ctrl} = 88.27 \pm 0.78$, or almost 90% of perfect performance (see Methods for our definition of fitness). That data point (for the run shown in Fig. 5, giving rise to the controller depicted in Fig. 6A) is indicated in red in Fig. 8. The run that evolved the controller shown in Fig. 6B is colored green in Fig. 8.

For all measures, we observe positive and highly significant correlations with fitness (Fig. 8 and Table 1). The best correlation is achieved for the integrated information measure Φ_{MC} ($R = 0.937$), followed by the information integration across the atomic partition Φ_{atom} (Spearman's $R = 0.784$), while the correlation with I_{pred} is weaker ($R = 0.63$). Likewise, I_{total}, which does not attempt to separate out the integration of different streams of information correlates only weakly with fitness ($R = 0.335$). The atomic processed information SI_{atom} [Eq. (12), $R = 0.553$] and the integration $\mathcal{I}$ both contribute to the strong correlation of Φ_{atom} with fitness, as Φ_{atom} is a sum of $\mathcal{I}$ and SI_{atom} as per Eq. (14). The integration measures Φ_{atom}, Φ_{MC}, and $\mathcal{I}$ also correlate well with
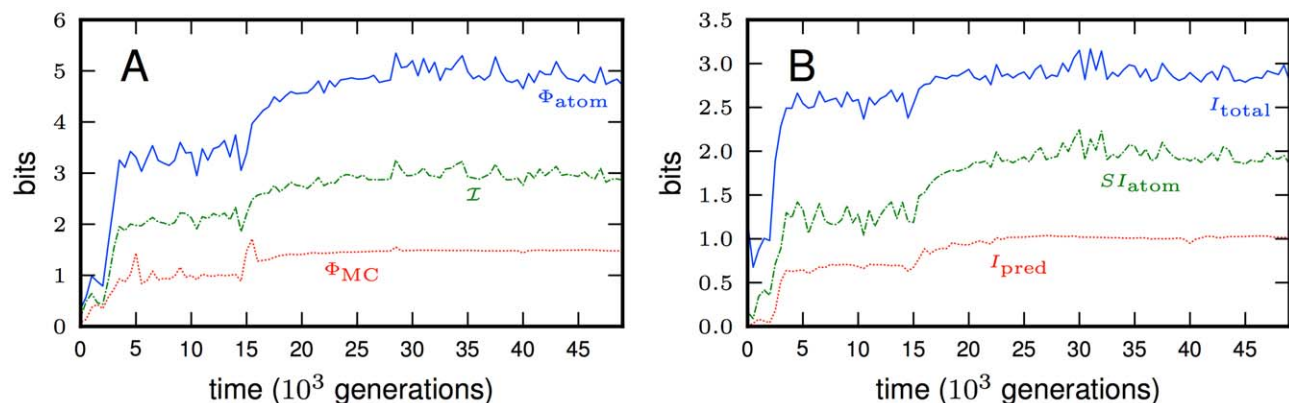
Figure 7. Information-based measures of complexity. A: Three Φ related measures of information integration for genomes on the LOD of the same run as shown in Fig. 5. Blue line: Φ_{atom} defined in (15), green: $\mathcal{I}$ defined in (13) and red: Φ_{MC}. B: Three information processing measures for the same experiment as (A): Blue: total information I_{total} (11), green: atomic information SI_{atom} (12), and red: predictive information I_{pred} (1).
doi:10.1371/journal.pcbi.1002236.g007
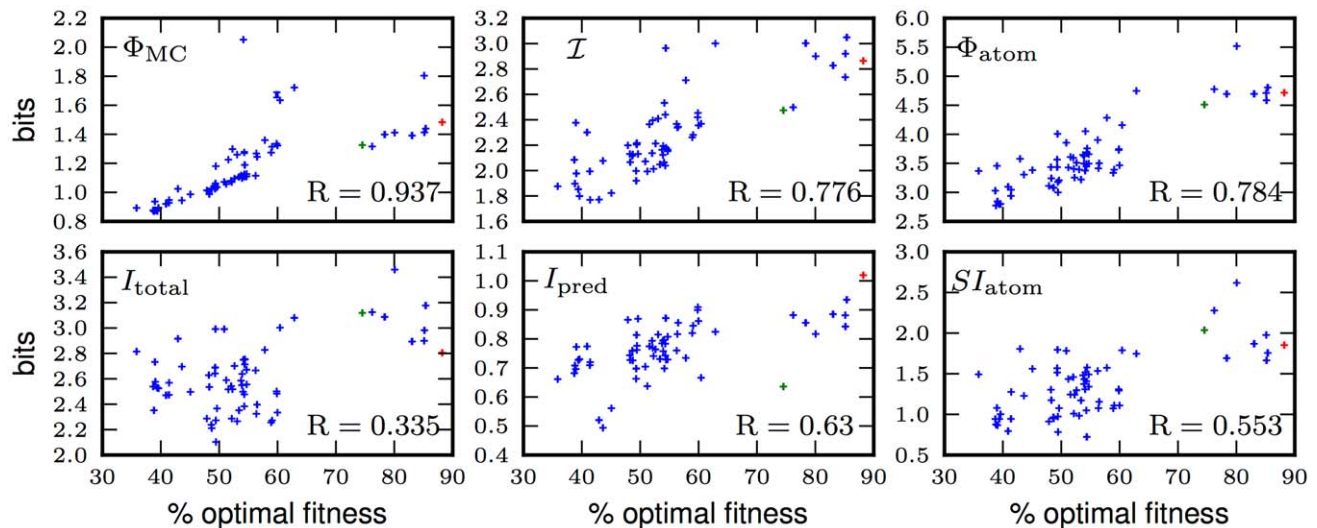
Figure 8. Correlation of information-based measures of complexity with fitness. Φ_{MC}, $\mathcal{I}$, Φ_{atom}, I_{total}, I_{pred}, and SI_{atom} plotted against f_{ctrl} (as a percentage of optimal fitness) using the final fitness (generation 49,000) on the LOD trajectory for 64 independent runs. R indicates Spearman's rank correlation coefficient.
doi:10.1371/journal.pcbi.1002236.g008

each other (data not shown). The difference in the correlation coefficients for Φ_{MC} and I_{pred} is highly significant ($p = 0$ in a Fisher r-to-z transformation test).

A clear separation between runs that achieved high ($> 70\%$ of maximal fitness) and low fitness ($< \approx 60\%$) is apparent in Fig. 8, indicating the difference between controllers that can or cannot access the information in the door beacon, which in turn requires the evolution of at least a single bit of memory. However, while it is not possible to achieve fitness in excess of 70% without using the information from the door beacon, one run utilized this information without exceeding 60% fitness, as determined via a knock-out analysis of the Markov variables.

Discussion

We have characterized several different information-theoretic measures in terms of their ability to reflect the complexity of information processing and integration in discrete dynamical systems. In order to discuss non-trivial examples of networks that are functional, we evolved computational networks that control an animat's behavior in a maze, and tested whether an increase in appropriate behavior is correlated with the putative proxies for complexity. The "brains" we evolved capture the essence of what it means to be successful in the maze environment: they can navigate arbitrary mazes of the type they are confronted with, and perform equally well with random versions of mazes that they

never encountered during their evolution. In particular, they integrate the sensory information from several sources appropriately, and when they evolve memory they are able to implement it in a variety of ways, including a variant of the hierarchical temporal memory paradigm [59]. We find that a standard measure that has been used to characterize complex robot behavior in the past [36], the predictive information I_{pred}, usually correlates well with fitness but sometimes fails to do so. We found examples where the failure to be predictive of fitness is associated with the evolution of memory (for example, the run indicated in green in Fig. 8), but also examples where this is not the case (e.g., the run that achieved the highest fitness, shown in red in Fig. 8). We hypothesize that when memory emerges, the integration of information from memory with the other signal streams is best reflected by measures of information integration such as Φ_{atom}, $\mathcal{I}$, and Φ_{MC}. Indeed, it is possible to show under fairly general assumptions that measures like I_{pred} can maximize fitness under the condition that no other information is used by an agent (such as acquired or inherited information [27], see also Text S1). Thus, while we expect that I_{pred} performs worse and worse as a predictor of complex function as more and more memory is utilized for navigation, in some cases I_{pred} turns out to perform very well counter to this expectation. It is currently unclear what is at the origin of this difference in predictive performance of I_{pred}.

That I_{pred} ultimately has to fail as a predictor of fitness when memory is used can be seen in the limiting case of navigating entirely by memory. In that case, any correlation between sensory inputs and motor actions would be purely accidental, in particular if the sensory data that ultimately predict appropriate motor actions are not in the immediate past. However, a non-Markovian version of I_{pred} that takes sensorial data from more distant time steps into account could conceivably perform well even in this case.

On the other hand, measures of information integration could still be elevated even when navigating by memory, as the motor units are driven by streams of information emanating from within, rather than without. However, as sensorial information is not integrated, measures of information integration should be lower when navigating entirely by memory as opposed to navigating via

Table 1. Spearman's rank correlation coefficients (R) and significance (p-value) between different candidate measures of functional complexity with "final fitness", using the values achieved at generation 49K of the LOD (an approxiSmation of the most recent common ancestor) for 64 independent runs.

	Φ_{MC}	Φ_{atom}	$\mathcal{I}$	SI_{atom}	I_{total}	I_{pred}
R	0.937	0.784	0.776	0.553	0.335	0.63
p	4.1×10^{-30}	1.8×10^{-14}	4.8×10^{-14}	2.1×10^{-6}	6.8×10^{-3}	2.4×10^{-8}

doi:10.1371/journal.pcbi.1002236.t001

sensors complemented by memory. At the same time, a brain that dreams rather than acts has vanishing predictive information (as the sensor inputs as well as the motor units have vanishing entropy). Yet, integrated information could still be high, and thus reflect complex information processing in the brain even in the absence of behavior. In this respect, measures of integrated information are a good candidate for a quantitative measure of consciousness, as advocated earlier [39,40,44,50]. We note, however, that evolving functional networks with high Φ is not easy. For our 12-bit controllers, $\Phi_{MC} < 2$ bits almost always, and the main complex is significantly smaller than the network size, usually only comprising sensor and motor variables, and occasionally the memory bit when it is used.

Clearly, how useful Φ is as a measure of brain complexity let alone consciousness rests on testing it on more complex networks that enable complex behavior in simulated environments that are both deep and broad. Evolving networks that rely heavily on memory, and that have the capacity to observe *their own state* [64] and integrate that information with the sensorial stream, would be particularly useful in this respect. Ultimately, we expect that measures of information integration can then turn into predictors of fitness or function rather than the other way around. Indeed, the functional complexity of biological organisms (measured in terms of fitness) can only be estimated in the rarest of cases when we have a full understanding of what makes an organism successful in its particular niche.

In future work, we hope to evolve animats in more complex environments that require more broad and versatile use of memory, to thoroughly test the hypothesis that information integration measures outperform pure processing measures such as predictive information in complex tasks. Furthermore, we plan to test whether animats evolve *information matching* [65], that is, whether the integrated informational structure generated by an adapted complex fits, or matches, the informational structure of its environment. As it is possible to determine in detail how information about the world is represented within the Markov brains of these animats, the evolution of such creatures should demonstrate that evolution can move beyond representation-free AI [66] towards autonomous intelligence.

Methods

Agent embodiment

Of the six sensors shown in Fig. 2, three are obstacle detection bits (binary sensors that indicate that an obstacle is in front of it (bits 0–2 encode front, left-front, and right-front, respectively), as well as a "door beacon" (bit 3) that indicates whether the *next* opening will be to the right (bit 3 = '1') or else in front or left (bit 3 = '0') of the opening that the animat is currently passing through. This bit can be used to navigate more successfully in the maze, by keeping this bit in memory and integrating this information with the other sensors. The next opening-direction information is not available after the animat passes through the previous opening. Because the animat cannot turn, it is important to detect whether an animat has hit a lateral wall. Detectors 4 and 5 each return '1' if there is a wall to the left or right respectively.

For example, in Fig. 4, the opening-direction bit (bit 3 = '1') in the door just after the starting location indicates that the subsequent opening is to the right. After reaching this door and stepping through it, the sensor bit is set to bit 3 = '0' indicating that the next opening is to the left or in front (in this case, in front). Therefore, efficiently navigating the maze requires memorizing this bit when the animat is in an opening and acting on that information until the animat can see the next opening. Two output

bits (motors) control each animat's movement: the animat moves right if only bit 10 is on, left if only bit 11 is on, and forward if both are on. The animat has four internal bits (circles 6–9 in Fig. 2) that it can use for information memorization and integration.

Hidden Markov gates

The table depicted within each HMG in Fig. 3 represents the gate's function in terms of a stochastic finite state machine. The binary state of each HMG's inputs corresponds to a row in its probability table. These probabilities are encoded within the genes that specify the network, as described in Text S2. To determine how those probabilities generate an output from an input, first a random number between 0 and the sum of the elements of that row is generated. Comparing this random number to the cumulative sum of the numbers in that row selects an element in the row whose column index corresponds to the binary state of that gate's outputs. The OR operator is used to combine the outputs from multiple gates which output to the same bit.

Genetic encoding of network structure

Networks are encoded within circular genomes that are given by a sequence of unsigned characters [0,255]. Each gene encodes a single HMG and its connection to other gates via the Markov variables, as well as the state-transition probabilities that define the gate. Details about the interpretation of the genome and its translation into a network are given in Text S2. Each HMG can have at most 4 inputs, and at most 3 outputs. If more than one HMG writes into a single Markov variable, these outputs are combined via an OR operation but we allow at most 3 write-attempts into a single Markov variable. If in the sequential interpretation of the genome an HMG requests to write to a variable that already has three connections, that HMG's connection will instead be routed to the nearest available variable. The same restrictions exist if an HMG tries to read from a variable that already has 3 read connections.

Fitness calculation

The animat's fitness in a maze m is determined by:

$$g(m) = \sum_{t=0}^{T} \left(\frac{D - d_t}{D} + L_t \right) \quad (16)$$

where T is the number of time steps ($T = 300$), d_t is the shortest path distance to the last doorway in the maze from the animat's position at time t, D is the maximum shortest path from all locations in the maze to the last doorway, and L_t is the number of times the animat has passed the last doorway. The maze environment is periodic so that if the animat goes past the end of the maze, the environment is the same as the beginning of the maze.

The stochastic nature of the controller implies that the $g(m)$ measured in one run through a single maze m may not be a reliable estimate of the genome's fitness because chance decisions could lead to either too high or too low fitness. We therefore define the animat's selection fitness by:

$$f(m) = \left(\prod_{i=1}^{10} \frac{g^{(i)}(m)}{g_{\text{opt}}(m)} \right)^{\frac{1}{10}} \quad (17)$$

where $g^{(i)}(m)$ is the ith stochastic realization of the animat's fitness in maze m, and $g_{\text{opt}}(m)$ is the maximum fitness attainable in that maze. The geometric mean of 10 evaluations helps to ensure the

reproducibility of the animat's fitness, and make it a better predictor of the long-term success of the lineage it represents.

In order to ensure that the genomes have evolved the ability to navigate through *general* mazes of this type (rather than adapting to a single particular maze), a set of 10 control mazes are used to calculate the *control fitness*:

$$f_{\text{ctrl}} = \frac{1}{100} \sum_{m=1}^{10} \sum_{i=1}^{10} \frac{g^{(i)}(m)}{g_{\text{opt}}(m)} \qquad (18)$$

The control fitness uses the arithmetic rather than the geometric mean in order to better track performance. The geometric mean in Eq. (17) allows for the elimination of controllers that ever completely fail at a single instance (as fitness is then multiplied by zero). The arithmetic mean in Eq. (18) is a better numerical indicator for the power of the strategy, as a single failure does not result in a vanishing control fitness.

Evolution and Genetic Algorithm

64 populations of 300 individuals were evolved for 50,000 generations. For the purpose of selection, a single maze was randomly generated for each run, given a set of boundary conditions. Every 100 generations a new maze was generated for each run so that the animats would not adapt to a specific instance of the problem. To implement selection, the top three individuals from each generation (the elite) were copied into the next generation without mutation, unless their fitness was determined to be zero after re-testing. The remaining places in the population were filled by roulette-wheel selection with mutations [67]. This implies that the number of offspring that any parent places into the next generation is proportional to the relative fitness advantage (or disadvantage) it holds with respect to the average population fitness. However, no individual could place more than 10 offspring into the next generation. The genomes (described in Text S2 and Fig. S1) were changed via a variety of processes from generation to generation. Single loci were copied with a probability of $\mu_{\text{sitecopy}} = 0.025$, deleted with probability $\mu_{\text{sitedel}} = 0.05$, a random value inserted after a loci with probability $\mu_{\text{siteins}} = 0.025$, replaced with a uniformly drawn random number $\in [0,255]$ with probability of $\mu_{\text{siteuniform}} = 0.05$, or increased/decreased by a random number $\in [-10,10]$ (restricted to the range $[0,255]$ if necessary) with probability of $\mu_{\text{site up/down}} = 0.05$. Whole genes where duplicated with $\mu_{\text{Gdup}} = 0.005$, deleted with $\mu_{\text{Gdel}} = 0.01$, and a random gene inserted with $\mu_{\text{Gins}} = 0.005$.

Finally, all mutation rates were normalized such that the whole genome mutation rate is equal to one change per genome per generation on average. This has the consequence that the "expressed" genome fraction (fraction with functioning start codon giving rise to HMGs connected to the main network) decreases with evolutionary time. Around 50,000 generations, the amount of expressed genes is of the order of 15% of the total genome size (on average about 200 of 3,000 loci are expressed in an evolved genotype).

Knock-out analysis

In order to determine the importance and role of individual variables in the brain's operation, we perform "knock-outs" on the variables to test their effect on the Markov animat's performance. Four types of per-bit knockouts were used: replace the value that the variable takes on by 'always read 0', 'always read 1', 'always write 0', and 'always write 1'. Some brains use variables with fixed values on purpose, in order to select certain rows from the probability tables with certainty. Such variables can be detected when only one of the two read-knockouts (read-zero or read-one) reduce the fitness of the controller. Variables that actually store and/or process information

will lead to reduced fitness by both knockouts. Motor variables that are not read from are unaffected by the read knockouts but are affected by the write knockouts. Similarly, write-knockouts from sensor variables do not affect fitness, while read-knockouts do.

To determine the function of individual HMGs, first each HMG was deleted from the controller to see if it changed the fitness. This identified unique important HMGs, but sometimes the results were masked by redundant HMGs. Then, each entry in the probability table for each HMG was "knocked out" by replacing the corresponding allele by zero or 255 [see Eq. (1) of Text S2 for the effect of this replacement]. This data combined with the input distribution for each HMG was used to determine the role of any particular HMG in the brain, and how it worked together with the other HMGs to control the animat.

Line of descent

For each run the line of descent (LOD) was obtained [68] by tracing back the fittest organism in the population backwards towards the randomly constructed ancestral sequence used to begin each experiment (encoding on average 12 HMGs, see Text S2). Seen from the point of view of the ancestral sequence, each following generation creates a branching tree with some lines eventually becoming extinct and other branches surviving. Because we simulate an asexual population in a single niche, only a single line of descent can ultimately remain because of competitive exclusion between members of the same species [69]. This line can be identified from following the lineage back from *any* of the 300 organism present in the final generation (generation 50,000) back to the origins (300 individual lines of descent). Going back ten generations, say, to 49,990, there will be fewer lines because some lines coalesced going backwards (branched going forward). The further backwards one moves on this "tree of descent", the more lines coalesce until the last common ancestor (LCA) of the entire population that was alive at generation 50,000 has been reached. Because of the single-niche environment, the 300 lines coalesce very quickly, and are virtually guaranteed to have coalesced to a single line when going back to generation 49,000, which is the "final" generation we study in our simulations, and defines the "final fitness". The organism at generation 49,000 of the LOD is not guaranteed to be the LCA, but it is guaranteed to be the ancestor of all organisms present in the final generation. Thus, the LOD records the evolutionary history of the experiment mutation by mutation, and allows us to reconstruct the evolutionary path that led to the adapted type. Fitness as well as complexity measures were calculated for organisms on the LOD every 500 generations.

Supporting Information

Figure S1 Genetic encoding of animat controllers. **A**: In this example, two HMGs encoded by two genes can read from and write to several of the 12 Markov variables, indexed 0–11. The top row shows the Markov variables at time t that the HMGs can read from while the row below shows how the HMGs write into those variables to update their state at $t+1$. **B**: The genome is a circular sequence of loci that carry unsigned integers *allele* $\in [0,255]$ and encode the input output structure of each HMG as well as the connectivity between them and the state transition tables that determine each HMG's function. Colors denote different functional sections of the gene. **C**: Causal influence of the Markov variables induced by the two HMGs. Presence of an arrow between variables a and b implies that a may change the state of b in a single time step. Absence of an arrow implies that the variables cannot influence each other within a single time step.
(PDF)

Text S1 Relationship between different forms of I_{pred}.
(PDF)

Text S2 Genetic encoding of network structure and function.
(PDF)

Text S3 Relationship between *EI* and *SI*.
(PDF)

Video S1 This movie shows the trajectory of an evolved animat traveling through the maze after 2,000 generations of evolution in the top panel, and the inner workings of its Markov network brain in the lower panel. At this point in evolutionary history, the animat has learned to move forward whenever it stands in front of an opening, but otherwise performs a random walk. The fitness at this time point is $15.8 \pm 0.6\%$ of maximal. The animat in the maze is depicted with a triangle, and the trail it leaves reflects the activation pattern of its four internal nodes and its motor outputs, as described in Fig. 4. The brain state (lower panel) shows all HMGs (U0–U10) and the probabilities in the state-transition tables as percentages (colored in shades of gray). Input bits (labeled iB) and output bits (labeled oB) are green if true and blue if false. The red element in each table indicates the element of the table selected at that time step based on the values of the input bits and the probabilities in that row. In other words, a table element turning red indicates which state of the HMG was selected as a function of the input. This is akin to a pattern of neuronal firings as a function of the inputs.
(MP4)

Video S2 The trajectory and brain states of an evolved animat at generation 14,000. At this point, the animat has acquired the capacity to maintain a direction of travel and move opposite to the direction indicated by the lateral contact sensor. Its movement with respect to the door opening is still random. The fitness at this time point is $47.9 \pm 0.7\%$ of maximal.
(MP4)

Video S3 The trajectory and brain states of an evolved animat at generation 49,000. By this time, the animat has evolved the capacity to use the information provided by the door beacon by storing it in bit 9, and move purposefully in the indicated direction after emerging from the previous door. Because of its high fitness, the animat traverses the maze five times, but does not always take the same trajectory every time, illustrating the stochasticity of its decisions. The fitness at this time point is $88.2 \pm 0.7\%$ of maximal.
(MP4)

Acknowledgments

Author Contributions

Conceived and designed the experiments: JAE AH CK GT CA. Performed the experiments: JAE AH. Analyzed the data: JAE AH. Contributed reagents/materials/analysis tools: NC. Wrote the paper: JAE CK GT CA.

References

1. Badii R, Politi A (1997) Complexity: Hierarchical structures and scaling in physics, volume 6 of *Cambridge Nonlinear Science* Series. Cambridge, UK: Cambridge University Press.
2. Adami C (2002) What is complexity? Bioessays 24: 1085–1094.
3. Adami C (2004) Information theory in molecular biology. Phys Life Rev 1: 3–22.
4. Adami C (2009) Biological complexity and biochemical information. In: Meyers RA, ed. Encyclopedia of Complexity and Systems Science. Springer Verlag. pp 489–511.
5. Sporns O (2011) Networks of the Brain. Cambridge, MA: MIT Press.
6. Löfgren L (1977) Complexity of description of systems: A foundational study. Int J Gen Sys 3: 197–214.
7. Chaitin GJ (1979) Toward a mathematical definition of life. In: Levine RD, Tribus M, eds. The Maximum Entropy Formalism. MIT Press. pp 477–498.
8. Papentin F (1980) On order and complexity i: General considerations. J theor Biol 87: 421–456.
9. Papentin F (1982) On order and complexity ii: Application to chemical and biochemical structures. J theor Biol 95: 225–245.
10. Thomas RDK, Reif WE (1993) The skeleton space: A finite set of organic designs. Evolution 47: 341–360.
11. Thomas RD, Shearman RM, Stewart GW (2000) Evolutionary exploitation of design options by the first animals with hard skeletons. Science 288: 1239–1242.
12. Soloveichik D, Winfree E (2006) Complexity of self-assembled shapes. SIAM J Comput 36: 1544–1569.
13. Ahnert SE, Johnston IG, Fink TMA, Doye JPK, Louis AA (2010) Self-assembly, modularity, and physical complexity. Phys Rev E 82: 026117.
14. Kolmogorov A (1965) Three approaches to the quantitative definition of information. Probl Inf Trans 1: 4.
15. Lempel A, Ziv J (1976) On the complexity of finite sequences. IEEE Trans Inform Theory 22: 75–81.
16. Ebeling W, Jimenez-Montano M (1980) On grammars, complexity, and information measures of biological macromolecules. Math Biosci 52: 53–71.
17. Li M, Vitanyi P (1997) An introduction to Kolmogorov complexity and its applications. New York, NY: Springer Verlag.
18. Gell-Mann M, Lloyd S (1996) Information measures, effective complexity, and total information. Complexity 2: 44–52.
19. Adami C, Cerf NJ (2000) Physical complexity of symbolic sequences. Physica D 137: 62–69.
20. McShea DW (2000) Functional complexity in organisms: Parts as proxies. Biol Philos 15: 641–668.
21. Szostak JW (2003) Functional information: Molecular messages. Nature 423: 689.
22. Hazen RM, Griffin PL, Carothers JM, Szostak JW (2007) Functional information and the emergence of biocomplexity. Proc Natl Acad Sci U S A 104: 8574–8581.
23. Bennett C (1995) Universal computation and physical dynamics. Physica D 86: 268–273.
24. Lloyd S (2001) Measures of complexity: A nonexhaustive list. IEEE Control Syst Mag N Y 21: 7–8.
25. Taylor S, Tishby N, Bialek W (2007) Information and fitness. eprint arXiv.org:0712.4382.
26. Polani D (2009) Information: currency of life? HFSP J 3: 307–316.
27. Rivoire O, Leibler S (2011) The value of information for populations in varying environments. J Stat Phys 142: 1124–1166.
28. Bonchev DG (2009) Information theoretic complexity measures. In: Meyers RA, ed. Encyclopedia of Complexity and Systems Science Springer Verlag. pp 4820–4838.
29. Dehmer M, Mowshowitz A (2011) A history of graph entropy measures. Inf Sci 181: 57–78.
30. Koch C, Laurent G (1999) Complexity and the nervous system. Science 284: 96–98.
31. Haldane JBS (1932) The Causes of Evolution. London: Longmans.
32. Maynard Smith J (1969) Evolutionary Genetics. Oxford, UK: Oxford University Press.
33. Lenski R, Rose M, Simpson S, Tadler S (1991) Long-term experimental evolution in *Escherichia coli*. I. Adaptation and divergence during 2,000 generations. Am Nat 138: 1315–1341.
34. Donaldson-Matasci MC, Bergstrom CT, Lachmann M (2010) The fitness value of information. Oikos 119: 219–230.
35. Bialek W, Nemenman I, Tishby N (2001) Predictability, complexity, and learning. Neural Comput 13: 2409–2463.
36. Ay N, Bertschinger N, Der R, Guettler F, Olbrich E (2008) Predictive information and explorative behavior of autonomous robots. Eur Phys J B 63: 329–339.
37. Felleman DJ, Van Essen DC (1991) Distributed hierarchical processing in the primate cerebral cortex. Cereb Cortex 1: 1–47.
38. Hagmann P, Cammoun L, Gigandet X, Meuli R, Honey CJ, et al. (2008) Mapping the structural core of human cerebral cortex. PLoS Biol 6: e159.
39. Tononi G (2008) Consciousness as integrated information: a provisional manifesto. Biol Bull 215: 216–242.
40. Balduzzi D, Tononi G (2008) Integrated information in discrete dynamical systems: motivation and theoretical framework. PLoS Comput Biol 4: e1000091.
41. Slepian D, Wolf JK (1973) Noiseless coding of correlated information sources. IEEE Trans Inform Theory 19: 471–480.
42. Barrett AB, Seth AK (2011) Practical measures of integrated information for time-series data. PLoS Comput Biol 7: e1001052.
43. Rota GC (1964) The number of partitions of a set. Am Math Mon 7: 498–504.
44. Tononi G (2010) Information integration: its relevance to brain function and consciousness. Arch Ital Biol 148: 299–322.
45. Atick JJ (1992) Could information theory provide an ecological theory of sensory processing. Network 3: 213–251.

46. Nadal JP, Parga N (1994) Nonlinear neurons in the low noise limit: A factorial code maximizes information transfer. Network 5: 565–581.

47. Schneidman E, Still S, Berry II MJ, Bialek W (2003) Network information and connected correlations. Phys Rev Lett 91: 238701.

48. Barlow HB (1961) Possible principles underlying the transformation of sensory messages. In: Rosenblith WA, ed. Sensory Communication. Cambridge, MA: MIT Press. pp 217–234.

49. McGill WJ (1954) Multivariate information transmission. Psychometrika 19: 97–116.

50. Tononi G, Sporns O, Edelman G (1994) A measure for brain complexity: Relating functional segregation and integration in the nervous system. Proc Natl Acad Sci U S A 91: 5033–5037.

51. Lungarella M, Pegors T, Bullwinkle D, Sporns O (2005) Methods for quantifying the informational structure of sensory and motor data. Neuroinformatics 3: 243–262.

52. Lungarella M, Sporns O (2006) Mapping information ow in sensorimotor networks. PLoS Comput Biol 2: 1301–1312.

53. Ay N (2001) Information geometry on complexity and stochastic interaction. Preprint 95/2001, Max Planck Institut for Mathematics in the Sciences, Leipzig.

54. Ay N, Wennekers T (2003) Temporal infomax leads to almost deterministic dynamical systems. Neurocomputing 52-4: 461–466.

55. Ay N, Wennekers T (2003) Dynamical properties of strongly interacting markov chains. Neural Netw 16: 1483–1497.

56. Wilson SW (1991) The animat path to AI. In: Meyer JA, Wilson S, eds. From Animals to Animats. Cambridge, MA: MIT Press. pp 15–21.

57. Klyubin AS, Polani D, Nehaniv CL (2007) Representations of space and time in the maximization of information ow in the perception-action loop. Neural Comput 19: 2387–2432.

58. Koller D, Friedman N (2009) Probabilistic Graphical Models. Cambridge, MA: MIT Press.

59. Hawkins J, Blakeslee S (2004) On Intelligence. New York, NY: Henry Holt and Co.

60. George D, Hawkins J (2005) A hierarchical Bayesian model of invariant pattern recognition in the visual cortex. In: Prokhorov D, ed. Proceedings of the International Joint Conference on Neural Networks (IJCNN). IEEE, volume 3. pp 1812–1817.

61. George D, Hawkins J (2009) Towards a mathematical theory of cortical micro-circuits. PLoS Comput Biol 5: e1000532.

62. Riesenhuber M, Poggio T (1999) Hierarchical models of object recognition in cortex. Nature Neuroscience 2: 1019–1025.

63. Hintze A, Adami C (2008) Evolution of complex modular biological networks. PLoS Comput Biol 4: e23.

64. Adami C (2006) What do robots dream of? Science 314: 1093–4.

65. Tononi G, Sporns O, Edelman G (1996) A complexity measure for selective matching of signals by the brain. Proc Natl Acad Sci U S A 93: 3422–3427.

66. Brooks R (1991) Intelligence without representation. Artif Intell 47: 139–159.

67. Michalewicz Z (1999) Genetic Algorithms+Data Structures = Evolution Programs. New York: Springer Verlag.

68. Lenski RE, Ofria C, Pennock RT, Adami C (2003) The evolutionary origin of complex features. Nature 423: 139–144.

69. Hardin G (1960) The competitive exclusion principle. Science 131: 1292–1297.