Network Complexity of Foodwebs

Russell Standish
School of Mathematics and Statistics, University of New South Wales

Abstract

In previous work, I have developed an information theoretic complexity measure of networks. When applied to several real world food webs, there is a distinct difference in complexity between the real food web, and randomised control networks obtained by shuffling the network links. One hypothesis is that this complexity surplus represents information captured by the evolutionary process that generated the network.

In this paper, I test this idea by applying the same complexity measure to several well-known artificial life models that exhibit ecological networks: Tierra, EcoLab and Webworld. Contrary to what was found in real networks, the artificial life generated foodwebs had little information difference between itself and randomly shuffled versions.

Introduction

In Standish (2005), I developed a method for computing the information complexity of a network. In Standish (2010a), I refined and generalised the method to overcome a problem with higher complexity values of empty and full networks relative to partially filled networks of the same degree, as well as taking account of link weights. Coupled with some new algorithms for computing automorphism group size, this network complexity measure is practical for networks of several thousand nodes.

In Standish (2010a), I studied several published datasets of natural networks, including a number of foodwebs available from the Pajek website, and the neural network of *C. elegans* (see Table 1). In most cases, these networks exhibited significantly heightened complexity values compared with those of control networks obtained by shuffling the links in a random fashion. This leads to the hypothesis that evolutionary processes tend to produce networks with a *complexity surplus* (Δ) compared with random assembly processes.

In this work, I apply the same methods to networks created by artificial life evolutionary systems, in particular the interaction network of Tierra (Ray, 1991) and the foodwebs of EcoLab (Standish, 1994) and Webworld (Caldarelli et al., 1998).

Complexity as Information

The notion of using information content as a complexity measure is fairly simple. In most cases, there is an obvious *prefix-free* representation language within which descriptions of the objects of interest can be encoded. There is also a classifier of descriptions that can determine if two descriptions correspond to the same object. This classifier is commonly called the *observer*, denoted $O(x)$.

To compute the complexity of some object x, count the number of equivalent descriptions $\omega(\ell, x)$ of length ℓ that map to the object x under the agreed classifier. Then the complexity of x is given in the limit as $\ell \to \infty$:

$$\mathcal{C}(x) = \lim_{\ell\to\infty} \ell \log N - \log \omega(\ell, x) \tag{1}$$

where N is the size of the alphabet used for the representation language.

Because the representation language is prefix-free, every description y in that language has a unique prefix of length $s(y)$. The classifier does not care what symbols appear after this unique prefix. Hence $\omega(\ell, O(y)) \geq N^{\ell - s(y)}$. As ℓ increases, ω must increase as fast, if not faster than N^ℓ, and do so monotonically. Therefore $C(O(y))$ decreases monotonically with ℓ, but is bounded below by 0. So equation (1) converges.

To use this formalism with networks, we need to fix two things: how to decide when two networks are identical, and a prefix-free representation language, which will be used to count the representations of a given network. In this context, ignoring any link weights, two networks are considered identical if the nodes of one can be placed over the nodes of the second one, such that the links correspond exactly. They are topologically identical. We ignore any labels on the nodes or links.

Network bitstring representation

To represent the network as a bitstring, we need to store the node count (n) and link count (l), as well as representation of the adjacency matrix. The initial part of the string has $w = \lceil \log_2 n \rceil$ '1' bits, followed by a single '0' stop bit. Following that are w bits representing the value of n in binary.

Dataset	nodes	links	$\mathcal{C}$	$e^{\langle \ln \mathcal{C}_{\mathrm{ER}} \rangle}$	$\Delta = \mathcal{C} - e^{\langle \ln \mathcal{C}_{\mathrm{ER}} \rangle}$	$\frac{\lvert \ln \mathcal{C} - \langle \ln \mathcal{C}_{\mathrm{ER}} \rangle \rvert}{\sigma_{\mathrm{ER}}}$
celegansneural	297	2345	442.7	251.6	191.1	29
celegansmetabolic	453	4050	25421.8	25387.2	34.6	∞
lesmis	77	508	199.7	114.2	85.4	24
adjnoun	112	850	3891	3890	0.98	∞
yeast	2112	4406	33500.6	30218.2	3282.4	113.0
baydry	128	2138	126.6	54.2	72.3	22
baywet	128	2107	128.3	51.0	77.3	20
cypdry	71	641	85.7	44.1	41.5	13
cypwet	71	632	87.4	42.3	45.0	14
gramdry	69	911	47.4	31.6	15.8	10
gramwet	69	912	54.5	32.7	21.8	12
Chesapeake	39	177	66.8	45.7	21.1	10.4
ChesLower	37	178	82.1	62.5	19.6	10.6
ChesMiddle	37	208	65.2	48.0	17.3	9.3
ChesUpper	37	215	81.8	60.7	21.1	10.2
CrystalC	24	126	31.1	24.2	6.9	6.4
CrystalD	24	100	31.3	24.2	7.0	6.2
Everglades	69	912	54.5	32.7	21.8	11.8
Florida	128	2107	128.4	51.0	77.3	20.1
Maspalomas	24	83	70.3	61.7	8.6	5.3
Michigan	39	219	47.6	33.7	14.0	9.5
Mondego	46	393	45.2	32.2	13.0	10.0
Narragan	35	219	58.2	39.6	18.6	11.0
Rhode	19	54	36.3	30.3	6.0	5.3
StMarks	54	354	110.8	73.6	37.2	16.0

Table 1: Complexity values of several freely available network datasets, as reported in Standish (2010a). For each network, the number of nodes and links are given, along with the computed complexity $\mathcal{C}$. In the fourth column, the original network is shuffled 1000 times, and the logarithm of the complexity is averaged ($\langle \ln \mathcal{C}_{\mathrm{ER}} \rangle$). The fifth column gives the difference between these two values, which represents the information content of the specific arrangement of links. The final column gives a measure of the significance of this difference in terms of the number of standard deviations ("sigmas") of the distribution of shuffled networks. In two examples, the distribution of shuffled networks had zero standard deviation, so ∞ appears in this column.

Knowing the value of n, the number of bits needed to represent l is $\lceil \log_2 L \rceil$, where $L = (n(n-1)/2)$ so l is stored in a field of that width.

For the final part of the string, the linkfield, we can represent the adjacency matrix such that a '1' bit in position $i(n-1)+j$-th represents a link from node i to j if $j < i$ or from i to $j+1$ if $j > i$, where nodes are numbered $0 \ldots n-1$, $i < n$ and $j < n-1$. However, this representation is not efficient — given l, there must be exactly l '1' bits in the linkfield, ie it is one of the permutations of l '1' bits and $L-l$ '0' bits. We can enumerate the $\begin{pmatrix} L \\ l \end{pmatrix}$ permutations, and choose the rank of our linkfield in the enumeration as the encoding of the linkfield. This is known as rank encoding (Myrvold and Ruskey, 2001). One of the effects of choosing this encoding is that both an empty and a full network have just one possible linkfield, so will have a rank encoding of 0, representable in 0 bits, as we already know whether a network is empty or full from the values of n and l. Hence, the full and empty networks are the simplest networks for given n and l.

Weighted links

Whilst the information contained in link weights might be significant in some circumstances (for instance the weights of a neural network can only be varied in a limited range without changing the overall qualitative behaviour of the network), of particular theoretical interest is to consider the weights as continuous parameters connecting one network structure with another. For instance if a network X has the same network structure as A, with b links of weight 1 with a network structure B and the remaining $a-b$ links of weight w, then we would like the network complexity of X to vary smoothly between that of A and B as w varies from 1 to 0. Görnerup and Crutchfield (2008) introduced a similar measure.

The most obvious way of defining this continuous complexity measure is to start with normalised weights $\sum_i w_i = 1$. Then arrange the links in weight order, and compute the complexity of networks with just those links of weights less than w. The final complexity value of a network $X = N \times L$, where N is the set of nodes, and L the set of links with associated weights $w_i, \exists i \in L$, is obtained by integrating:

$$\mathcal{C}(X = N \times L) = \int_0^1 \mathcal{C}(N \times \{i \in L : w_i < w\}) dw \quad (2)$$

Obviously, since the integrand is a stepped function, this is computed in practice by a sum of complexities of partial networks.

Counting the representations

In principle, one could compute the complexity of a network by enumerating all bitstrings for a given n and l, and counting the number of bitstrings that represent the target network. However, this algorithm is highly combinatoric, and only really feasible for small networks. However, the number of representations can also be computed by dividing the total number of possible renumberings of the nodes ($N!$) by the size of the automorphism group, for which several practical algorithms exist (McKay, 1981; Standish, 2010b; Darga et al., 2008). Even though each of these algorithms is NP-complete, in practice they tend to perform quite well for networks up to several thousands of nodes. Where each algorithm performs poorly, one of the other algorithms performs well, so a hybrid algorithm that runs each algorithm in parallel, and returning the result of the first algorithm to complete, performs extremely well.

ALife models

Tierra

Tierra (Ray, 1991) is a well known artificial life system in which self reproducing computer programs written in an assembly-like language are allowed to evolve. The programs, or *digital organisms* can interact with each via template matching operations, modelled loosely on the way proteins interact in real biological systems. A number of distinct strategies evolve, including parasitism, where organisms make use of another organism's code and hyperparasitism where an organism sets traps for parasites in order to steal their CPU resources. At any point in time in a Tierra run, there is an interaction network between the species present, which is the closest thing in the Tierra world to a foodweb.

Tierra is an aging platform, with the last release (v6.02) having been released more than six years ago. For this work, I used an even older release (5.0), for which I have had some experience in working with. Tierra was originally written in C for an environment where ints were 16 bits and long ints 32 bits. This posed a problem for using it on the current generation of 64 bit computers, where the word sizes are doubled. Some effort was needed to get the code 64 bit clean. Secondly a means of extracting the interaction network was needed. Whilst Tierra provided the concept of "watch bits", which recorded whether a digital organism had accessed another's genome or vice versa, it did not record which other genome was accessed. So I modified the template matching code to log the pair of genome labels that performed the template match to a file.

Having a record of interactions by genotype label, it is necessary to map the genotype to phenotype. In Tierra, the phenotype is the behaviour of the digital organism, and can be judged by running the organisms pairwise in a tournament, to see what effect each has on the other. The precise details for how this can be done is described in Standish (2003).

Having a record of interactions between phenotypes, and discarding self-self interactions, there are a number of ways of turning that record into a foodweb. The simplest way,

which I adopted, was sum the interactions between each pair of phenotypes over a sliding window of 100 million executed instructions, and doing this every 20 million executed instructions. This lead to time series of around 2000 foodwebs for each Tierra run.

In Tierra, parsimony pressure is controlled by the parameter SlicePow. CPU time is allocated proportional to genome size raised to SlicePow. If SlicePow is close to 0, then there is great evolutionary pressure for the organisms to get as small as possible to increase their replication rate. When it is one, this pressure is eliminated. In Standish (2004b), I found that a SlicePow of around 0.95 was optimal. If it were much higher, the organisms grow so large and so rapidly that they eventually occupy more than 50% of the soup. At which point they kill the soup at their next Mal (memory allocation) operation. In this work, I altered the implementation of Mal to fail if the request was more than than the soup size divided by minimum population save threshold (usually around 10). Organisms any larger than this will never appear in the Genebanker (Tierra's equivalent of the fossil record), as their population can never exceed the save threshold. This modification allows SlicePow = 1 runs to run for an extensive period of time without the soup dying.

EcoLab

EcoLab was introduced by the author as a simple model of an evolving ecosystem (Standish, 1994). The ecological dynamics is described by an n-dimensional generalised Lotka-Volterra equation:

$$\dot{n_i} = r_i n_i + \sum_j \beta_{ij} n_i n_j, \tag{3}$$

where n_i is the population density of species i, r_i its growth rate and β_{ij} the interaction matrix. Extinction is handled via a novel stochastic truncation algorithm, rather than the more usual threshold method. Speciation occurs by randomly mutating th ecological parameters (r_i and β_{ij}) of the parents, subject to the constraint that the system remain bounded (Standish, 2000).

The interaction matrix is a candidate foodweb, but has too much information. Its offdiagonal terms may be negative as well as positive, whereas for the complexity definition (2), we need the link weights to be positive. There are a number of ways of resolving this issue, such as ignoring the sign of the off-diagonal term (ie taking its absolute value), and antisymmetrising the matrix by subtracting its transpose, then using the sign of the offdiagonal term to determine the link direction.

For the purposes of this study, I chose to subtract just the negative β_{ij} terms from itself and its transpose term β_{ji}. This effects a maximal encoding of the interaction matrix information in the network structure, with link direction and weight encoding the direction and size of resource flow. The effect is as follows:

- Both β_{ij} and β_{ji} are positive (the *mutualist* case). Neither offdiagonal term changes, and the two nodes have links pointing in both directions, with weights given by the two offdiagonal terms.

- Both β_{ij} and β_{ji} are negative (the *competitive* case). The terms are swapped, and the signs changed to be positive. Again the two nodes have links pointing in both directions, but the link direction reflects the direction of resource flow.

- Both β_{ij} and β_{ji} are of opposite sign (the *predator-prey* or *parasitic* case). Only a single link exists between species i and j, whose weight is the summed absolute values of the offdiagonal terms, and whose link direction reflects the direction of resource flow.

Webworld

Webworld is another evolving ecology model, similar in some respects to EcoLab, introduced by Caldarelli et al. (1998), with some modifications described in Drossel et al. (2001). It features more realistic ecological interactions than does EcoLab, in that it tracks biomass resources. It too has an interaction matrix called a *functional response* in that model that could serve as a foodweb, which is converted to a directed weighted graph in the same way as the EcoLab interaction matrix. I used the Webworld implementation distributed with the *EcoLab* simulation platform Standish (2004a).

Results

Methods and materials

Tierra was run on a 512KB soup, with SlicePow set to 1, until the soup died, typically after some 5×10^{10} instructions have executed. Some variant runs were performed with SlicePow=0.95, and with different random number generators, but no difference in the outcome was observed.

The source code of Tierra 5.0 was modified in a few places, as described in the Tierra section of this paper. The final source code is available as tierra.5.0.D7.tar.gz from the *EcoLab* website hosted on SourceForge (http://ecolab.sf.net).

The genebanker output was processed by the eco-tierra.3.D13 code, also available from the *EcoLab* website, to produce a list of phenotype equivalents for each genotype. A function for processing the interaction log file generated by Tierra and producing a timeseries of foodweb graphs was added to Eco-tierra. The script for running this postprocessing step is process_ecollog.tcl.

The EcoLab model was adapted to convert the interaction matrix into a foodweb and log the foodweb to disk every 1000 time steps for later processing. The Webworld model

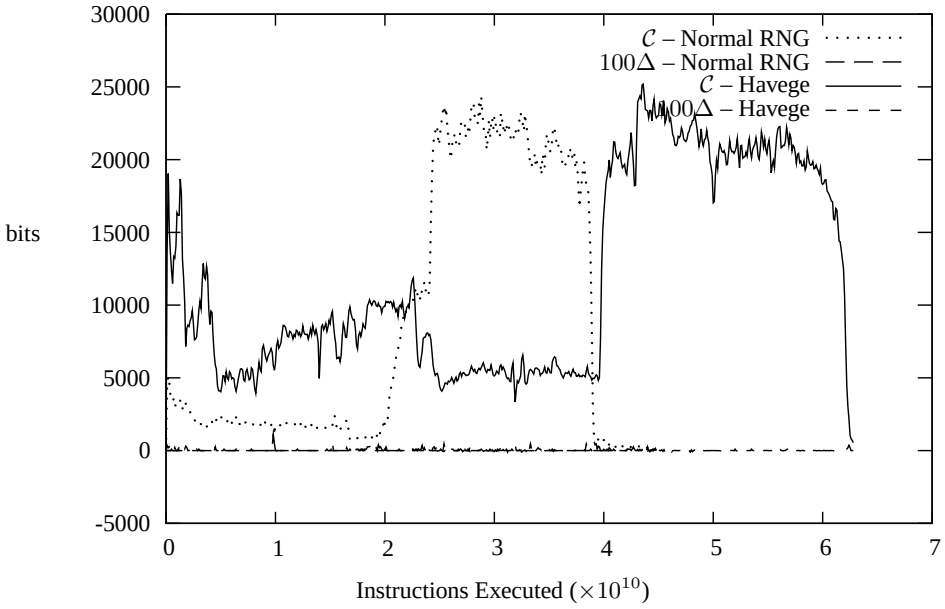

Figure 1: Complexity of the Tierran interaction network for SlicePow=0.95, and Δ, exaggerated by a factor of 100. Two different random number generators were used, Havege and the normal linear congruential generator supplied with Tierra.

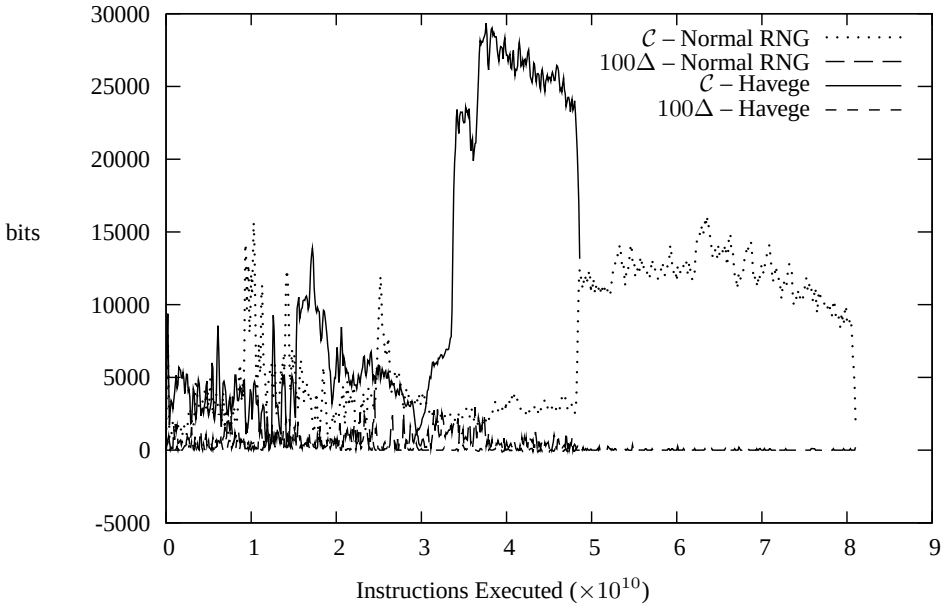

Figure 2: Complexity of the Tierran interaction network for SlicePow=1, and Δ, exaggerated by a factor of 100. Two different random number generators were used, Havege and the normal linear congruential generator supplied with Tierra.

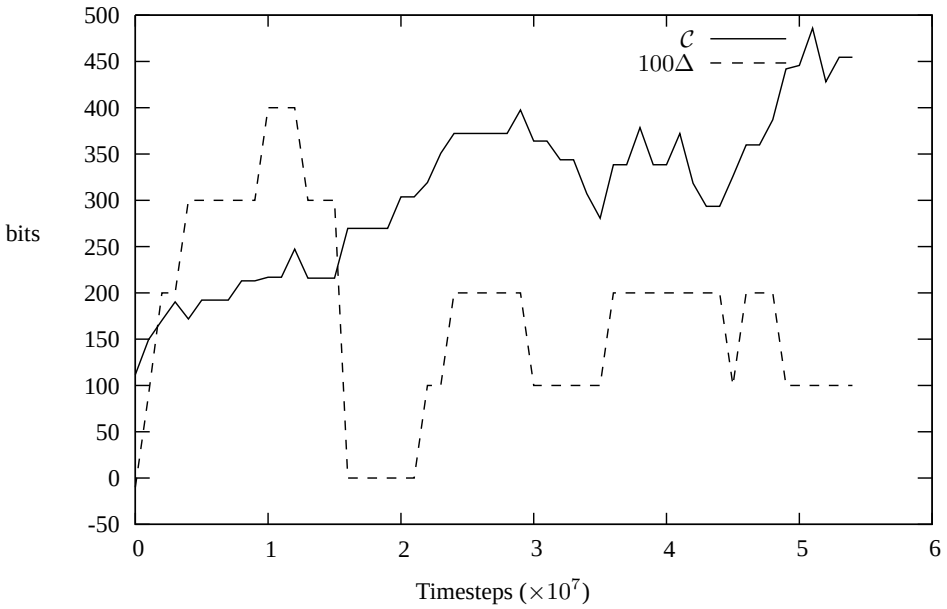

Figure 3: Complexity of EcoLab's foodweb, and Δ, exaggerated by a factor of 100, as described in the text.

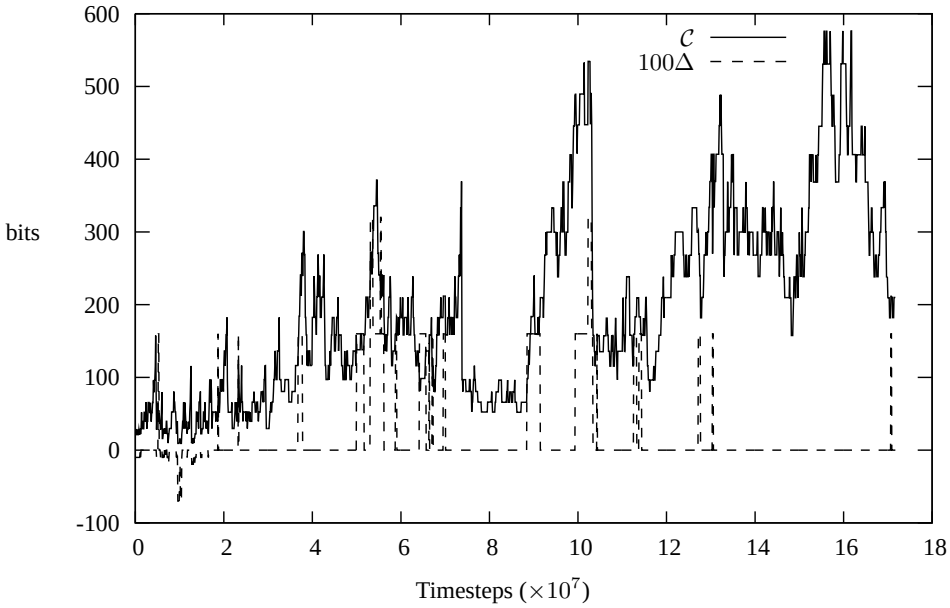

Figure 4: Complexity of Webworld's foodweb, and Δ, exaggerated by a factor of 100, as described in the text.

was adapted similarly. The model parameters were as documented in the included ecolab.tcl and webworld.tcl experiment files of the ecolab.4.D37 distribution, which is also available from the *EcoLab* website.

Finally, each foodweb, and 100 link-shuffled control versions were run through the network complexity algorithm (2). This is documented in the cmpERmodel.tcl script of ecolab.4.D37. The average and standard deviation of $\ln \mathcal{C}$ was calculated, rather than $\mathcal{C}$ directly, as the shuffled complexity values fitted a log-normal distribution better than a standard normal distribution. The difference between the measured complexity and $\exp\langle \ln \mathcal{C} \rangle$ (ie the geometric mean of the control network complexities) is what is reported as Δ in Figures 1–4.

Discussion

It can be seen from Figures 1–4, that none of the artificial life models studied generate substantially greater network complexities than do the control networks. By "substantially", I mean more than 10% of the total network complexity. The complexity difference that exists is nevertheless often statistically significant, albeit small (of the order of a few bits). By contrast, most of the 26 practical networks studied in Standish (2010a) exhibited substantially greater complexities than their controls, the exceptions being the David Copperfield adjective-noun adjacency dataset (0.98 bits), and the *C. elegans* metabolic network (which at 34.6 bits is about 0.1% of the total complexity).

The complete failure for several independent artificial evolutionary systems to be able to generate this complexity surplus weakens the case for the surplus as being due to operation of an evolutionary process. It is possible that this is another illustration of the difference between artificial evolutionary systems and natural evolutionary systems observed with Bedau-Packard statistics (Bedau et al., 1998). There is also the possibility that some systematic artifact skews the observational data towards more symmetric networks (which increases complexity values), however it seems implausible that networks collected by many different observers in many different fields should exhibit the same systematic error. More work needs to be done applying this complexity metric to both artificially evolved networks and observational data of naturally evolved networks to elucidate if this is artifact, or a real phenomenon.

Conclusion

In this work, I measured the network complexity of several artificially evolved foodwebs to see if I could reproduce the complexity surplus seen in empirical network data. In none of the artificial systems I studied was the complexity surplus substantial enough to be considered a real effect.

References

Bedau, M. A., Snyder, E., and Packard, N. H. (1998). A classification of long-term evolutionary dynamics. In Adami, C., Belew, R., Kitano, H., and Taylor, C., editors, *Artificial Life VI*, pages 228–237, Cambridge, Mass. MIT Press.

Caldarelli, G., Higgs, P. G., and McKane, A. J. (1998). Modelling coevolution in multispecies communities. *J. Theor, Biol.*, 193:345–358.

Darga, P. T., Sakallah, K. A., and Markov, I. L. (2008). Faster symmetry discovery using sparsity of symmetries. In *Proceedings of the 45st Design Automation Conference*, Anaheim, California.

Drossel, B., Higgs, P. G., and McKane, A. J. (2001). The influence of predator-prey population dynamics on the long-term evolution of food web structure. *J. Theor. Biol.*, 208:91–107.

Görnerup, O. and Crutchfield, J. P. (2008). Hierarchical self-organization in the finitary process soup. *Artificial Life*, 14:245–254.

McKay, B. D. (1981). Practical graph isomorphism. *Congressus Numerantium*, 30:45–87.

Myrvold, W. and Ruskey, F. (2001). Ranking and unranking permutations in linear time. *Information Processing Letters*, 79:281–284.

Ray, T. (1991). An approach to the synthesis of life. In Langton, C. G., Taylor, C., Farmer, J. D., and Rasmussen, S., editors, *Artificial Life II*, page 371. Addison-Wesley, Reading, Mass.

Standish, R. K. (1994). Population models with random embryologies as a paradigm for evolution. *Complexity International*, 2.

Standish, R. K. (2000). The role of innovation within economics. In Barnett, W., Chiarella, C., Keen, S., Marks, R., and Schnabl, H., editors, *Commerce, Complexity and Evolution*, volume 11 of *International Symposia in Economic Theory and Econometrics*, pages 61–79. Cambridge UP.

Standish, R. K. (2003). Open-ended artificial evolution. *International Journal of Computational Intelligence and Applications*, 3:167. arXiv:nlin.AO/0210027.

Standish, R. K. (2004a). Ecolab, Webworld and self-organisation. In Pollack et al., editors, *Artificial Life IX*, page 358, Cambridge, MA. MIT Press.

Standish, R. K. (2004b). The influence of parsimony and randomness on complexity growth in Tierra. In Bedau et al., editors, *ALife IX Workshop and Tutorial Proceedings*, pages 51–55. arXiv:nlin.AO/0604026.

Standish, R. K. (2005). Complexity of networks. In Abbass et al., editors, *Recent Advances in Artificial Life*, volume 3 of *Advances in Natural Computation*, pages 253–263, Singapore. World Scientific. arXiv:cs.IT/0508075.

Standish, R. K. (2010a). Complexity of networks (reprise). *Artificial Life*. submitted. arXiv: 0911.348.

Standish, R. K. (2010b). SuperNOVA: a novel algorithm for graph automorphism calculations. *Journal of Algorithms - Algorithms in Cognition, Informatics and Logic*. submitted, arXix: 0905.3927.