

Visualizing Evolutionary Dynamics of Self-Replicators Using Graph-based Genealogy

Chris Salzberg¹, Antony Antony¹, and Hiroki Sayama²

¹ Section Computational Science, Universiteit van Amsterdam, The Netherlands

² Dept. of Human Communication, University of Electro-Communications, Japan
{chris,antony}@phenome.org sayama@hc.uec.ac.jp

Abstract. We present a general method for evaluating and visualizing evolutionary dynamics of self-replicators using a graph-based representation for genealogy. Through a transformation from the space of species and mutations to the space of nodes and links, evolutionary dynamics are understood as a flow in graph space. Mapping functions are introduced to translate graph nodes to points in an n -dimensional visualization space for interpretation and analysis. Using this scheme, we evaluate the effect of a dynamic environment on a population of self-reproducing loops. Resulting images visually reveal the critical role played by genealogical graph space partitioning in the evolutionary process.

1 Introduction

Research on artificial self-replication has resulted in a variety of systems exhibiting complex evolutionary dynamics[11]. Of crucial interest in the analysis of these systems are the localized events (interaction, mutation) that collectively decide the path of global trends in evolution. Few attempts have been made to visualize the topology of this transition-space in a general way. For example, the method proposed by Bedau and Brown[2] circumvents this problem and purports to characterize the evolutionary activity of genotypes through their relative concentration in a population. While useful as a global indicator, this approach overlooks the very fluctuations which enable evolution to occur: these are the genealogical *links* relating distinct species through their ancestry. Without these links, a gap remains between the global dynamics we observe and the localized interactions which trigger their emergence.

In this paper we attempt to bridge this gap. We do so by introducing a method to transform the space of self-replicator species and their mutations to an abstract graph space where nodes and links represent species and mutations, respectively. Within this new space, temporal evolution of populations and their genealogical connectivity is conceptualized as a *flow*, with individual replicator species classified according to their reproductive capability on a scale between *source* and *sink*. We then show an example of graph-based genealogy visualization applied to a simple self-replicating cellular automaton, the “Evoloop”[9], to demonstrate the effectiveness of our method in capturing the critical role played by genealogical graph space partitioning in the evolutionary process.

2 Graph-based Genealogy Analysis

We derive a general, graph-based picture of evolution from a set of basic definitions. In what follows we limit ourselves within asexually reproducing systems where reproduction occurs via binary fission.

2.1 Definitions

Our method assumes an arbitrary system of self-replicators evolving over time in a spatial domain \mathbf{C} which we call *configuration space*. This may be a cellular automata space[5, 9], the memory and CPU of a computer[7, 13], or any other well-defined domain. The intrinsic structure of individual self-replicators in this domain is described uniquely by a sequence of digits from an arbitrary alphabet which we call an *identifier*. Identifiers may be gene sequence, program code, or any other well-defined modular form that specifies the replicator’s evolutionary identity. We call the space of these identifiers Θ .

To each replicator in configuration space we associate a position $r \in \mathbf{C}$ and identifier $\theta \in \Theta$; the pair (r, θ) uniquely describes an *instance*. We group replicators with the same identifier together as a *species*. To keep track of different species we assign an arbitrary but unique index k to each, which is used for this purpose only. We define $\theta(k)$ as a mapping from species index k to its actual identifier.

The creation of a new replicator is defined as a *birth* and is described by the *parent* species θ_p , *child* species θ_c , time of birth t_b , and location of birth r_b . A birth for which $\theta_p \neq \theta_c$ indicates mutation has occurred. Borrowing the idea by Bedau and Brown[2], we summarize the time evolution of birth events over the domain \mathbf{C} by the birth “trigger” function δ_b :

$$\delta_b(\theta_p, \theta_c, t_b, r_b) = \begin{cases} 1 & \text{if a new replicator of } \theta_c \text{ is born from a parent} \\ & \text{of } \theta_p \text{ at time } t_b \text{ at position } r_b, \\ 0 & \text{otherwise.} \end{cases} \quad (1)$$

In what follows, we use the function δ_b as our fundamental quantity. A more complete description would also include an analogous death trigger function δ_d identifying the event of replicator death; for simplicity we omit this function here. Note that most existing analysis of artificial models implicitly assume tree-based genealogy [7, 9, 13]; equation (1) makes no such assumption. We do however assume that we may identify parent and child in a unique and unambiguous manner, and that births may be precisely tracked in both time and space. While practically impossible in real biological systems, such complete data collection can be done for artificial evolutionary models.

2.2 Genealogy on a Graph

The basis for our genealogy analysis is a transformation from configuration space \mathbf{C} to a directed graph \mathbf{G} which we call a “genealogy graph”. Although not

strictly necessary, we assume time and space to be discrete hereafter. Definitions describing this graph cover a time window from t_i to t_f , which is written as $\mathbf{T} = (t_i, t_f)$. We assign a node in the graph for each species, and associate it with a unique index k and an initial population $P(k, t_i)$. Directed edges in this graph represent ancestral links created with the birth of replicators: following detection and identification, a parent node k is assigned to the newborn node l . The pair of species $(\theta(k), \theta(l))$ are henceforth distinguished as parent and child relative to the directed link between them.

Note that nodes in \mathbf{G} represent *groups*, not *instances*, of self-replicators. Here we aim at systems of what Szathmary calls “limited heredity replicators” [12], for which the number of possible different types is about equal to or smaller than their population so that the same type may be realized many times during evolutionary exploration processes. Nodes may thus have multiple incoming links corresponding to mutations undergone by instances of several distinct species. For different systems where each birth always produces a novel, almost unique type, each node has one (and only one) incoming link so that conventional tree-based genealogy becomes more relevant.

To track the evolution of graph-based genealogy, we introduce a traversal frequency function $F(k, l, \mathbf{T})$ describing the number of link traversals (births) from node k (parent) to node l (child) in the interval \mathbf{T} , derived from δ_b as:

$$F(k, l, \mathbf{T}) = \sum_{t'=t_i}^{t_f} \sum_{r \in \mathbf{C}} \delta_b(\theta(k), \theta(l), t', r). \quad (2)$$

From the traversal frequency, we derive three important quantities: the number of incoming link traversals $I(k, \mathbf{T})$, outgoing link traversals $O(k, \mathbf{T})$ and buckle (self-link) traversals $B(k, \mathbf{T})$ in the interval \mathbf{T} :

$$I(k, \mathbf{T}) = \sum_{l \neq k} F(l, k, \mathbf{T}), \quad (3)$$

$$O(k, \mathbf{T}) = \sum_{l \neq k} F(k, l, \mathbf{T}), \quad (4)$$

$$B(k, \mathbf{T}) = F(k, k, \mathbf{T}). \quad (5)$$

In transforming from δ_b to $F(k, l, \mathbf{T})$, information about spatial distribution and genetic details of replicators have been lost. However, population dynamics and genealogy for our system — the quantities of interest for our analysis — are well-described. Moreover, the emphasis in this formalism is on the dynamics of evolutionary *connectivity* rather than on global cumulative trends. The advantage to such an approach is that we retain statistical properties of individual species within the context of their ancestral links; these links play a critical role in the emergent phenomena which we observe.

2.3 Evolution as Flow

The framework defined above inspires a change in the way we understand the dynamics of evolution. The alternative we propose here is the idea of *evolution*

as *flow*. In this picture, an evolutionary system is composed of a subset of nodes in genealogical graph space, each of which represents a distinct type of replicator to which a population and collection of active incoming and outgoing links are attributed. Self-replication and variation correspond to the traversals of self-links and outgoing links in graph space, respectively, which induce a collective motion of the global population.

To reflect the above ideas in terms of a *flow*, we derive a quantity which we call the *production*:

$$\text{Prod}(k, \mathbf{T}) = \text{O}(k, \mathbf{T}) + \text{B}(k, \mathbf{T}) - \text{I}(k, \mathbf{T}) \quad (6)$$

According to this definition, replicator species which frequently construct other species as well as replicators of their own species will have a high production; those which are frequently constructed by other species but fail to self-replicate will have negative production. Note that the production is not the same as the fitness of a species; in addition to self-reproduction (B), it also considers the existence of outgoing and incoming links (O and I) and thus emphasizes genealogical connectivity. Borrowing terminology from physics and network theory, we associate a node with highly positive production to an evolutionary *source*, whereas a node with highly negative production we call an evolutionary *sink*. Ranking nodes in this way quantifies the role species play in the evolutionary process. The balance in equation (6) is hence between the capability of a species to *produce* versus the tendency to *be produced*.

To visualize this evolutionary flow in a genealogy graph, we introduce a vector mapping function $\mathbf{M} : \Theta \mapsto \mathbf{R}^n$ from the space of species identifiers to an n -dimensional possibility space. This vector mapping function consists of n real-valued functions $\{M_j : \Theta \mapsto \mathbf{R}; \quad j = 1, \dots, n\}$, which, as a whole, map a species $\theta(k)$ to a point \mathbf{x}^k in an n -dimensional space. The point $\mathbf{x}^k = (x_1^k, \dots, x_n^k)$ represents the co-ordinates of the k th species in this new visualization space.

Note that we have put no requirements on the nature of the functions M_j , hence they can be many-to-one and thus the point \mathbf{x}^k is not necessarily unique to species $\theta(k)$. The optimal choice of functions will map highly connected nodes — groups of species related by strong ancestral links — close to each other in the visualization space. The transformation from graph links to lines connecting the points \mathbf{x}^k will then be more easily understood when viewed in \mathbf{R}^n . Note that we have chosen this graph-space mapping for its simplicity and generality; many others exist and could also be used.

3 Example: Evoloop with Dynamic Environment

In this section we apply the proposed method to a simple self-replicating cellular automaton, the “Evoloop” [9]. Definitions discussed in Section 2.1 are linked to model-specific events governing self-replication in the Evoloop CA. From these definitions we construct a mapping from CA space to a 2D visualization space in the manner outlined in Section 2.3. The introduction of a dynamic environment [1] is used to demonstrate the insight gained by this technique.

3.1 Defining Trigger Functions

As our chosen model, the Evoloop satisfies the three conditions necessary for evolution to occur: replication, variation (mutation) and differential fitness (competition)[3]. This model is also simple and scalable, taking the form of 9-state cellular automata with a von Neumann neighbourhood. Structurally, the Evoloop is divisible into two basic components: an inner and outer sheath of square or rectangular shape and a gene sequence of moving signal states. Coordination of the duplication process is controlled via the sequence of genes within the external sheath; mutations occur through extrinsic interaction, leading to a change in the gene sequence of offspring loops. This results in a uniquely emergent process of evolution, one which — due to their robustness and high replication rate — generally favours smaller-sized loops[9].

Our analysis begins with a model-specific definition for *birth*. We use for this purpose the umbilical cord dissolver, a state which appears upon closure of the arm and then retracts towards the parent loop. The local configuration highlighted in the second frame of Fig. 1 signals that such an event has occurred. At birth, loops are assigned a *genotype* $g \in \Gamma_g$ corresponding to the configuration of genes in their gene sequence and a *phenotype* $p \in \Gamma_p$ describing their size (length and width). According to earlier definitions, the space of identifiers Θ is the space of all possible combinations of genotype and phenotype, hence $\Theta = \Gamma_g \times \Gamma_p$ (with a constraint applied so that phenotype must be large enough to contain accompanying genotype). Each $\theta \in \Theta$ contains the necessary information to reconstruct a loop in the exact configuration as it was when it was born. Given the time t_b of the middle frame of Fig. 1 and the location r_b of the umbilical cord dissolver, the birth trigger function $\delta_b(\theta_p, \theta_c, t_b, r_b)$ now has a precise meaning for the Evoloop model. With δ_b defined, graph-based parameters derived in Section 2.2 are now described.

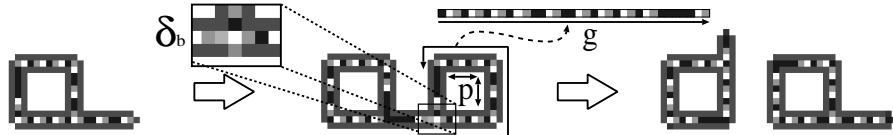


Fig. 1. Self-replication of the Evoloop. g is the genotype, p the phenotype (length and width), and δ_b (middle frame) the configuration which triggers a birth event.

3.2 Mapping to a 2D Space

The fact that $\Theta = \Gamma_g \times \Gamma_p$ presents a convenient separation for constructing mapping functions onto a 2D visualization space. We associate x and y axes with genotype g_k and phenotype p_k for identifier $\theta(k)$, respectively, so that

$$\mathbf{x}^k = \mathbf{M}(\theta(k)) = \begin{pmatrix} M_x(\theta(k)) \\ M_y(\theta(k)) \end{pmatrix} = \begin{pmatrix} M_x(g_k) \\ M_y(p_k) \end{pmatrix}. \quad (7)$$

For the phenotype-map, we apply a simple transformation $M_y(p = l \times w) = \sqrt{lw}$. Details of the genotype-map are omitted here as they involve model-specific parameters and weighting factors. We instead point to an important feature: that this function is heavily size-dependent yet also highly affected by changes in genotypical configuration. Points in 2D space hence line up near the diagonal, with mutation-induced changes in gene positioning leading to spatial separation in the x - y plane between distinct species. For more details we refer to [8].

For visualization of species according to eq. (7) we use the following scheme:

- Evolutionary *sources* ($\text{Prod}(k, \mathbf{T}) > 0$) are mapped to circles.
- Evolutionary *sinks* ($\text{Prod}(k, \mathbf{T}) < 0$) are mapped to triangles.
- Sizes of sources and sinks are scaled relative to the magnitude $|\text{Prod}(k, \mathbf{T})|$.
- Links between species are represented by lines whose thickness is determined by the cumulative transversal frequency over the time window \mathbf{T} . Arrow heads are omitted here to make the plots concise. In the case of bidirectional links, thickness of the higher-frequency direction is used.

3.3 Results

Figure 2 demonstrates results on an 800×800 grid, beginning with a single loop of species 9f4945/8×8 (genotype/phenotype) using the compressed hexadecimal notation presented in [1]. Data is binned over periods of $\mathbf{T} = 10\text{K}$ iterations. The observed evolution towards small-sized species is expected, however Fig. 2 also reveals the paths through which smaller species achieve domination. In particular, the second frame shows that the majority of well-traversed paths (drawn in black) lead to species of a smaller phenotype. The many kinks in this path indicate the existence of species which are neither source nor sink. These “transient” species [1] replicate a different species than their own, playing the role of intermediary between stable species in the process of evolution; this can be understood as the active mutation discussed by Ikegami[4]. Due to their low numbers, many traditional methods of population analysis would miss these species, yet they play a crucial role in shaping the evolution of populations.

For comparison, we contrast the above result with a different case using the dynamic environment made of “persistent dissolver”, another kind of dissolving

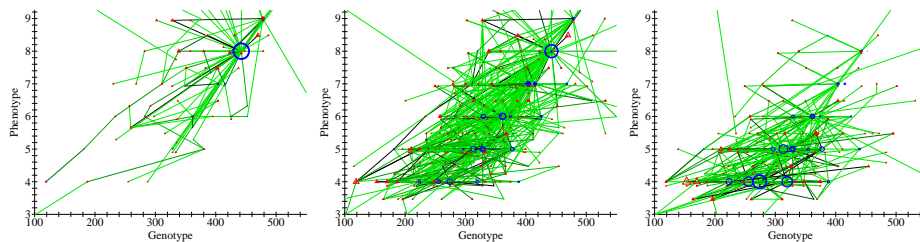


Fig. 2. Evolution in genealogy graph space. Each image covers 10K iterations. Sequence runs from $\mathbf{T} = (0\text{K}, 10\text{K})$ to $\mathbf{T} = (20\text{K}, 30\text{K})$. Visualization scheme described in text.

state that lasts for a substantially longer timescale, introduced in [1]. The effects of this new environment have been found to encourage diversity leading to speciation, punctuated equilibrium, and evolutionary “bottlenecking”. Fig. 3 depicts evolution of the same species, now coupled with the dynamic environment. Loop species are observed to collectively explore a broader portion of genealogical graph-space, roaming to both smaller and larger sizes. We notice for instance that the original size-8 loop evolves into a larger size-9 loop, then to another, different size-8 loop (identified as k , l and m in Fig. 3). A strong direct link and many lower frequency links form the path between these species. Hence lateral motion in graph-space (evolution to other species of the same size) has been achieved via loops of other sizes. The fact that this graph-space exploration occurs can be understood in terms of the partitioning of graph-space. A comparison between Fig. 2 and 3 reveals that a number of high-frequency (black) links leading to smaller loops have been cut due to local extinction caused by the dynamic environment. The system has thus explored a much more diverse subset of graph-space. While previously observed[1] via direct data analysis, the visualization of Fig. 3 offers a more complete representation of this behaviour.

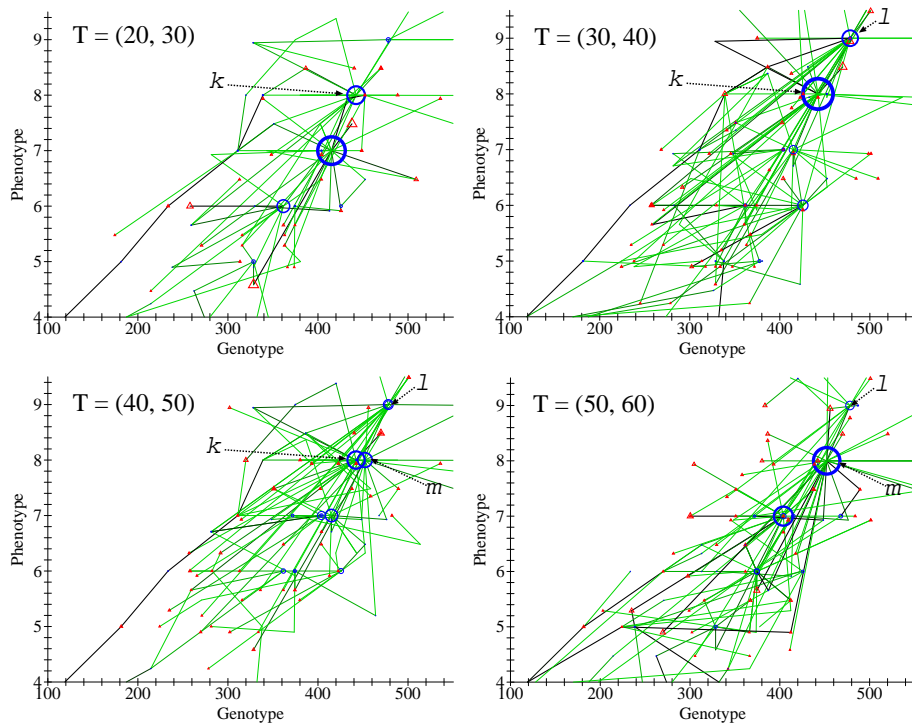


Fig. 3. Evolution in genealogy graph space with persistent dissolver. Each image covers 10K iterations. Sequence reads row-wise left to right, from $\mathbf{T} = (20\text{K}, 30\text{K})$ to $\mathbf{T} = (50\text{K}, 60\text{K})$. Species $k = 9\text{f}4945/8 \times 8$, $l = 9\text{f}a4a29/9 \times 9$, $m = 13\text{e}9251/8 \times 8$.

4 Conclusion

Our approach in this paper was to emphasize the importance of genealogical connectivity in visualizing the evolutionary dynamics of self-replicators. Results obtained with the Evoloop demonstrated the potential for genealogical complexity in even a simple CA model. Applied to this model, the proposed method was shown to highlight critical evolutionary paths in genealogical graph space.

Related work has been performed by a number of groups. Schuster and Fontana [10] focus on adaptive trajectories of RNA sequences through RNA “shape”-space using the concept of a “relay series”. A key distinction is that the relay series is *target-oriented*; gene sequences which do not form part of the relay series are not visualized. This is not true of the method presented here. Lenski et al. [6] explore the evolutionary origin of complex features. Genealogical analysis in this work is also exact; there are no “missing links” in evolutionary paths. Yet the focus in [6] is on functional genomics for the Avida model rather than evolutionary dynamics for a general system; here the latter target is emphasized.

Current work focuses on applying the visualization presented here to a new class of Evoloop species with multiple graph-space attractors. For more information on this project, see: <http://artis.phenome.org>.

References

1. A. Antony, C. Salzberg, and H. Sayama. A closer look at the evolutionary dynamics of self-reproducing cellular automata. Accepted pending revisions.
2. M. Bedau and C. Brown. Visualizing evolutionary activity of genotypes. *Artificial Life*, 5:17–35, 1999.
3. D. Dennett. *Encyclopedia of Evolution*, pages E83–E92. Oxford University Press, New York, 2002. ed. Pagel, M.
4. T. Ikegami. Evolvability of machines and tapes. *Artificial Life and Robotics*, 3:242–245, 1999.
5. C. Langton. Self-reproduction in cellular automata. *Physica D*, 10:135–144, 1984.
6. R. Lenski, C. Ofria, R. Pennock, and C. Adami. The evolutionary origin of complex features. *Nature*, 423:139–144, 2003.
7. T. Ray. An approach to the synthesis of life. In *Artificial Life II*, volume XI of *SFI Studies on the Sciences of Complexity*, pages 371–408. Addison-Wesley Publishing Company, Redwood City, California, 1991.
8. C. Salzberg, A. Antony, and H. Sayama. in preparation.
9. H. Sayama. A new structurally dissolvable self-reproducing loop evolving in a simple cellular automata space. *Artificial Life*, 5:343–365, 1999.
10. P. Schuster and W. Fontana. Chance and necessity in evolution: lessons from RNA. *Physica D*, 133:427–452, 1999.
11. M. Sipper. Fifty years of research on self-replication: An overview. *Artificial Life*, 4:237–257, 1998.
12. E. Szathmary. A classification of replicators and lambda-calculus models of biological organization. In *Proceedings: Biological Sciences*, volume 260, pages 279–286, 1995.
13. G. Yedid and G. Bell. Macroevolution simulated with autonomously replicating computer programs. *Nature*, 420:810–812, 2002.