Genetic Game of Life

John S McCaskill and Norman H Packard

European Centre for Living Technology

Abstract

Families of 2D cellular automata (CA) with various symmetries, starting with the semi-totalistic rules that include Conway’s Game of Life (GoL), are extended to allow evolution by genetic information associated with individual live cells, that specify local CA rule variants. A newly born live cell’s genome is copied (potentially with mutation) from one of the live neighbour cells (there are three in the GoL) and is destroyed when the cell dies. Just as biological evolution constantly discovers innovations in the space of chemical and physical functionalities that it controls, we explore how the addition of genetic information to the game of life can display the same type of discovery process, where the genetically controlled innovations are now spatially inhomogeneous local modifications to the GoL rules. The complex CA dynamics of the GoL becomes the default “chemistry and physics”, and local genetic modifications that persist are the innovations discovered by evolution.

We investigated evolution for four successively more differentiated symmetry cases in the nearest neighbour rules: semi-totalistic, corner-edge totalistic, 8-rotation symmetric, and physical 2D symmetric (4-rotations and 4-reflections). We discovered a family of deterministic rules, avoiding stochastic choices of ancestor for genetic inheritance, which yield deterministic models (like the GoL) in the absence of mutation. Systematic genetic variations near to the game of life rule are investigated and found to produce signs of computational complexity with an abundance of glider and glider gun structures.

We examined the impact on evolution for both neutral selection and genetically determined fecundity and mortality, including population interaction effects. The dynamics are analysed for novelty by normalized activity statistics on both genes and connected component patterns. The genetic evolution is analysed by fast ongoing genealogy construction and population weighted activity statistics. The spatial structure is captured using hash encoded quadtrees, relying on bit-parallel 64-bit integer algorithms, of the connected components, which are also mapped through time. This allows the tracking of moving objects such as gliders which change shape as they move. Activity statistics allow novelty in spatial patterns to be analysed. A novel genetic tracking of the dynamical history of live genes allows an efficient recognition of periodic dynamical structures such as gliders which transport information.

# Introduction

Interest in cellular automata (CA) as models of emergent complexity began with von Neumann’s famous 29-state cellular automaton capable of universal construction [von Neumann Ref 1] and became widespread with Conway’s discovery of the simple two-dimensional (2D) Game of Life CA (GoL) [Conway Ref 2] and Wolfram’s analysis of computational complexity classes in 1D CAs [Wolfram Ref 3], which was extended to 2D CAs [Packard Ref 4]. Von Neumann’s and Conway’s life-like CAs are defined with strong relaxation to the ground state in order to facilitate computation by rational design, and in fact universal computation has been proven by construction in both cases [*Authors?* Ref 5,6]. Both however involve fundamentally unprotected computations, in which perturbations, in the form of even the simplest travelling patterns, will almost certainly destroy not only the computed result but the carefully crafted computing architecture as well. Because of this, and despite the widespread continuing interest in novel computational structures in the GoL and related CAs, there is a major jump to evolving systems in which computation needs to survive robustly in the presence of potential interactions with many competitors. This paper is concerned with bridging this gap and we stay deliberately as close as possible to the deterministic CAs that underlie computation in the GoL by retaining completely deterministic computation apart from random and rare mutational changes, while supporting locally determined genetically encoded rule changes that enable evolution.

As analysed by Eppstein [Eppstein Ref 7], the family of semi-totalistic 2D binary CAs (with the next state of a cell depending only on the state and the number of its live neighbours and not on their detailed configuration) can perhaps better be characterized by the …

[this paragraph should discuss the background of computational complexity classes as it applies to 2D CAs. It should include reference to e-machines and Crutchfield [Crutchfield Ref 8].

[Another paragraph is required to explain the relationship with spatial games (Nowak) and with CAs used to describe evolution in reaction diffusion systems [Boerlijst and Hogeweg Ref 9] and in cellular and amphiphile models (Ising and Potts models [Ikegami and Hogeweg Ref 10, McCaskill, Packard et. al. Ref 11]. In this paragraph reference the other GPU paper on GoL plus evolution and contrast with adding additional states/colors to GoL (e.g. immigration game).

The GoL [Conway Ref 2, *Modern Author* Ref 12] is a deterministic dynamical system that takes two dimensional spatial patterns of binary states (‘live’ or ‘dead/empty’) to new patterns as time progresses discretely, through the action of a local rule; Each site’s state at time *t+1* is dependent on its state and the states of its nearest neighbours at time *t*. The game of life is a semi-totalistic cellular automaton (CA) rule, as defined above, and can be denoted by the code B3/S23 in standard notation [[Eppstein](https://www.ics.uci.edu/~eppstein/ca/wolfram.html) Ref 7, + Golly Ref 13]. A *live* cell *survives* (a state “1” at a site at time *t* persists to time *t+1*) if there are either 2 or 3 neighbours *alive* in the 8 cells surrounding the site on a square lattice (otherwise *dying*, *i.e.* changing to state “0” at time *t+1*), and a *dead* or *empty* “0” state cell undergoes birth (transitions to “1” at time *t*+1) only if there are exactly 3 neighbours *alive* at time *t*.

Conway’s Game of Life has become a canonical example of a complex system, with simple local rules that produce complex dynamics. It has a rich phenomenology of dynamics from special initial conditions, documented in massive catalogue projects [Ref 14,15] and other articles [Ref 16,17]. Specially engineered initial states can have extremely long transients, occupying large regions of space, and indeed the Gol has been shown to support universal computation [Ref 18,19].

Notwithstanding these properties, the GoL has not been a good model for studying the *emergence of complexity*, for two main reasons. The first is that starting from random initial state patterns on a finite compact domain, it is well known that the GoL almost always settles down to a combination of isolated static and simply periodic structures which are individually of limited spatial extent. In fact, this behaviour is so robust that as in a sandpile, Bak has shown that random isolated birth events cause the relaxed state to self-organize to a critical state where there is a powerlaw of cascade magnitudes to [Bak Ref 20] . The second is that the complexity generated by the GoL is not robust, in the sense that perturbations destroy functionally complex structures. Even when a complex dynamical structure happens to be produced by a random initial condition, it is typically destroyed by glider that perturbs it.

In biology, genetics is coupled to real-world physics and chemistry, enabling evolution to produce a complex biosphere. In the present work, we use the GoL to provide an interesting model of a rich but very simple “physics” or “chemistry”. We then enhance the GoL to include genetic information, with the aim of understanding how complexity may emerge from this simple version of evolution.

The coupling of GoL to genetic information has already been attempted in various ways [Refs], but a systematic investigation of near GoL evolving dynamics is still outstanding. We add genetics to the GoL by associating a genome with all live cells. Genetic inheritance is ensured by a newly born live cell’s genome being copied (potentially with mutation and recombination) from one or more of the live neighbour cells (there are three in the GoL) and being deleted when the cell dies. In this article we focus on the simplest case of mutation and asexual reproduction without recombination.

The genome of a live cell can encode local departures from the GoL rule, making the system spatially inhomogeneous cellular automaton. For the specification of a genetic GoL system, we must:

* Specify how departures from the GoL are determined by genes
* Specify how genes are propagated from one time-step to the next

Starting from random initial state patterns on a finite compact domain, it is well known that the GoL almost always settles down to a combination of isolated static and simply periodic structures or gliders which are individually of limited spatial extent [Ref 5,6]. Although specially engineered initial states can have extremely long transients, occupying large regions of space, and indeed the Gol has been shown to support universal computation [Ref 7,8], the absence of complex interconnected pattern persistence starting from random initial conditions means that it is not a good candidate for the emergence of complexity.

However, because of its rich dynamics from special initial conditions, documented in massive catalogue projects [Ref 9,10] and other articles [Ref 11,12], it would appear to provide an interesting model of a rich but very simple “physics” or “chemistry” that may be coupled to biological evolution through genetic information. The coupling of GoL to genetic information has already been attempted in various ways [Ref 13-15], but a systematic investigation of near GoL evolving dynamics is still outstanding. Here, we enhance the deterministic GoL dynamics to create an evolutionary system, by associating a genome with all live cells and deriving deterministic inheritance rules that may be genetically neutral or sequence dependent. Genetic inheritance is ensured by a newly born live cell’s genome being copied (potentially with mutation and recombination) from one or more of the live neighbour cells (there are three in the GoL) and being deleted when the cell dies. In this article we focus on the simplest case of mutation and asexual reproduction without recombination.

[We can examine genetics associated with the unmodified cellular automaton dynamics.]

This investigation is motivated by a fundamentally interest in the interaction between computational complexity and evolution. Whereas the GoL’s absence of

# Genetic Game of Life Models

The genome of a live-cell contains inherited information, copied from its ancestors, which may be used to track the flow of information in the GoL and the wider family of CA models investigated here. The genome of a live cell also may encode local departures from the GoL rule, making the system a spatially inhomogeneous cellular automaton. The local state of a cell is described completely by the presence or absence of a genome (the live/empty (1/0) state in the GoL) and its 64-bit sequence. The restriction to finite length 64-bit sequences is not fundamental but enables fast computation using machine integers on intel processors.

As in the GoL, we restrict attention in this paper to a local cellular automaton on a 2D square lattice, with the Moore 8-cell nearest neighbourhood, so that the discrete dynamics are completely defined locally by specifying the next state *ci,j*(*t+1*) of a cell at lattice site (*i,j*) at time *t+1*, in dependence on the previous state of the cell *ci,j*(*t*) and of its 8 nearest neighbours. The GoL rule is only a semi-totalistic rule, because its next state depends on the central state and not only on the sum *s9* of 9 neighbours (including the central state), the next state being 1 for *s9* == 3, *ci,j*(*t*) for *s9*==4, and 0 otherwise. For the rest of this paper, we used the preferred notation with the 8-neighbor sum *s=s8* and note that, for the semi-totalistic rules like GoL, the next state is a function of (*c,s*) where *c*=*ci,j*(*t*). As in biology, we do indeed want to retain the distinct dependence on the central state *c*, and therefore do not consider more symmetric fully totalistic rules in this paper, but we do consider a hierarchy of decreasingly symmetric rules, starting from the family of semi-totalistic rules.

For the specification of the CA dynamics of the Genetic GoL system, we must:

1. Specify how live/empty next states are determined by the configuration of live neighbours and their genomes (departure from GoL rules are possible), and
2. Specify how the information in the genomes attached to live states are propagated when the next state is live.

In the following subsections, we consider these aspects in successive extensions involving symmetry breakings on the undistinguished configurations in the GoL.

Since we wish to commence this study near to the GoL rules, which involve only the sum of live neighbours s = 2,3, it is illuminating to study this case first for fixed rule departures without genetic determination. As we shall see, some of the interesting evolutionary phenomena revealed by the Genetic Game of Life are already captured by this simplest case. Naively, one would expect 2x2 = 4 distinguished states (*c,s*) = (0 or 1, 2 or 3) that can possibly lead to live states and hence 24=16 different genetic extensions. We perform a detailed survey of the additional choices available for coupling genetics with the dynamics in this first case and implement these in a unified computer program to explore the model properties.

Although a natural first object of study for genetically dependent rules is the traditional extended GoL symmetry class of 218 semi-totalistic rules distinguishing 18 states (2 central states times the sum *s* of live neighbours ranging from 0 to 8), the sparsity of such rules with properties near to the GoL led us to also consider the broader families of rules with lower symmetries, distinguishing up to 64 local neighbourhood states (the maximum for which a 64-bit genome can encode the rule). The full set of 512=28+1 distinguished local states (9 cells), giving rise to 2512 possible rules, is too large to explore initially, especially with our restricted length genomes, and is physically less appealing because it does not take spatial symmetry into account. In between the 18 and 512 state extremes, we identify and implement three intermediate symmetries (cases 3-5 below). This leads to the following six cases, of which we implement and study the first five in this paper, not considering the last case (6) without spatial symmetry:

1. Semi-totalistic rules with s=2,3 (gene dynamics with fixed homogeneous rules) (4)
2. Semi-totalistic rules s=1-8 with LUTs for birth and survival determined by genes (16)
3. Quarter-totalistic rules s=1-8, se=0-4 with distinct corner and edge counts (46)
4. Eight-rotation symmetric genetically encoded rules for s=2-6 (64)
5. Spatial 2D symmetric genetically encoded rules (4-rot’n,4-refl’n) for s=0-4 (64)
6. Fully distinguished nearest neighbours without symmetries (512).

The numbers in parentheses are the number of distinguished local configurations that can possibly lead to a live cell (“1”) in the specified family of local CA rules. These configurations divide into the two equal-sized subsets, birth and survival: The birth rules for central current state “0” and the survival rules for current state “1” define the exceptions to the default rule which is next state “0”.

Rather than always to allow all possible rules in a given symmetry model, it is also of interest to consider restricted models in which a global constraint is introduced so that only a subset of the possible local states can be specified for active rules. In this work, we consider a description of such constraints limited to a semi-totalistic framework of permissions. For example, a birth-survival mask would have 1-bits signifying permission for a subset of positions corresponding to particular *s* values for either survival or birth. While the genes may contain other entries, the model would not allow these genes to enable birth or survival for local state configurations that correspond to *s* values for which there is a 0-bit in the birth-survival mask. For example, with the 16-bit birth-survival mask 0x0406, in the semi-totalistic case 2, only at most GoL rules are allowed, and with the mask 0x0606 the genetically encoded extension of the s=2,3 case 1 model family is specified.

How such LUTs are encoded in the genome must be decided by the model. In this work, we consider two types of encodings:

1. a direct position-dependent encoding assigning specific genome bits (or possibly contiguous sets of bits for redundant encodings) at certain positions to specific LUT entries
2. a modular variable-length position-independent encoding in which the genome encodes (at any block-aligned position) the local states which result in a next state of “1”.

In this paper, we restrict our attention to genomes of maximum length 64, and often use the term genome to refer to this full sequence, reserving the term gene to refer to a specifier of part of the rule-table.

For example, for the semi-totalistic case (2), if we exclude the special cases of spontaneous birth (B0 *i.e.* birth for s=0) and lone survival (S0) then there are 8+8=16 distinguished states that may be independently part of an active next-state ruleset. In this case, genes may specify any look up table (LUT) depending only on the central state *c* and neighbour sum *s* via two separate subsets of s-1 values for survival (1→1) and birth (0→1). We may also employ multiple bits (nc) to encode each LUT entry for an active rule, for example with only one of the possible nc gene patterns being active, then genomes of length 16 nc are required. In this paper we only consider this option for the semi-totalistic case, where there is sufficient length in the genome to allow the range of values nc=1,2, or 4. The alternative modular encoding that we employ is in this case to encode the values of *s* (requiring 3 bits) for which the next state is live, as well as the central state *c* (for survival or birth) i.e. 4 bits in total per entry. The standard game of life would require at least 3\*4 = 12 bits to be specified, 0xb23 in hexadecimal notation, so that 64-bit genomes such as 0xaaaaaaaa22221111 or any other combination of only the three digits 1,2,a would encode the GoL. Note that, since the existence of a 4-bit code in the genome for a particular survival-birth and s-1 value suffices to turn that LUT rule on, additional copies are redundant, allowing for mutational error resistance, especially when the birth-survival masks makes a significant number of entries ineffective. The sequence 0x0000000000000a21, encodes the GoL local rule if the birth-survival mask is 0 for s=1.

In this work, we find that the semi-totalistic case is too coarse an encoding of CA rules, to allow significant genetic evolution of complex structures beyond the classic game of life. We explore both the fixed length and variable length encodings for each of the cases 2-5 above, with details as specified in Appendix 1 We investigate the more differentiated rules and find that they produce a range of interesting dynamics. When coupled to a genetic population, these differentiated rules rapidly evolve to proliferate unless very strongly constrained (by the birth-survival mask introduced above).

The key focus in this paper is on sequence- and possibly population-dependent selection mechanisms that attribute an increasing cost to more prolific (less GoL-like) and more specific rule specification, so that survival mandates complex dynamics close to the game of life. We compare these with rule-independent selection mechanisms that interact with the GoL rules only in so far as birth or survival

## Semi-totalistic rules involving only s=2,3

### Live state determination

Since the GoL starting from random compact patterns of live states almost certainly relaxes to a set of unconnected simple patterns or periodic structures, with new live states only being produced in a small number of contexts, it is not as it stands a good substrate for evolution.

In contrast, the conventional approach generalizing the GoL, requiring that the birth and survival rules form a single interval of neighbourhood sum values, with lower and upper limits in the sum variable *s,* and restricting the possible rule-tables to a family with members specified by four integers SlSuBlBu (lower and upper values of the neighbourhood sum for survival and birth, respectively) i.e. to 362=1296 possible rules. Most of these rule-tables either lead to strong proliferation of live states or their extinction, and in order to allow genetic encoding to deliver novel dynamics of interest it turns out to be important to further dissect the rule-tables in the vicinity of the GoL rule 2333.

Genetic modifications that reduce the number of configurations resulting in live states will further restrict the potential for ongoing evolution, which requires more active rules (more neighbourhood state configurations leading to live cells) than the GoL. The most parsimonious first choice is, as for the GoL, to continue to consider only rules with next live states for *s =* 2 or 3 live nearest neighbours, *i.e.* distinguishing 4 neighbourhood states as candidates for a live next central state *c:* (*c*,*s*){(0,2),(0,3),(1,2),(1,3)}*.* Transitions to live central cell states from the first two states are referred to as birth, and those involving the last two as survival, independently of the new value of *s*. Since there are 4 starting states and 2 predicted outcomes (live or not) for the next central cell state, there are 24 such rules, corresponding to the subsets of the starting states that give rise to a live next state. The GoL rule has survival for s=2,3 and birth for s=3, corresponding to the subset *RGoL*= {(0,3),(1,2),(1,3)}. Since we need more active rules, it is logical to begin with letting the genes control the missing birth rule (0,2)⇒1 for *s*=2. If the proliferation induced by this extra birth process should prove too strong, then one could counter this by removing one or more of the elements of *RGoL*. This compensation could either be fixed or dependent on the genes. Since there are a number of intermediate and hybrid cases, we summarize the various options that we have investigated in Table 1.

In order to distinguish the genetic dependency from uniform changes in the rules, we split the survival and birth processes into two optionally executed stages, the first depending on the selective genetics (Sg and Bg) and the second genetically independent (*i.e.* enforced, Sf and Bf), as shown in Table 1. The first 8 binary options in Table 1 give rise to 144 different cases. For birth, all of the 24=16 cases are different, in contrast with the case in survival, where there are only 3x3=9 cases. Even if birth is enforced, the genomes of the live neighbours may still have a vital impact on the future dynamics by determining which of them becomes the ancestor of the newly born genome. In addition to the distinction of birth and survival depending on the state of the central cell, there is another possibility opened up by the genetics which is not distinguished in the binary GoL: Instead of simply remaining alive, the genome of the next state may be overwritten by one of the neighbouring genomes according to a birth process. We label this binary option O2/3 as it may be allowed independently for s=2,3. This corresponds to the well-studied Moran model of population genetics [Ref needed? If so, Crow]. There are thus 576 different genetic extension models, even before one considers details of the genetic dependency.

In the interests of further limiting and analysing the extent of rule departures from the GoL, we also record here for completeness two further binary options Nr and Ns, which enforce GoL rules if respectively the previous transition rule was a non GoL rule or the current state was last produced by a non GoL transition. In near GoL simulations colouring cells by departures from the GoL rules in these two ways allows an assessment of both the potential and effective impact of the modified rules on the dynamics.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **transition** | **nr** | **c** | **s** | **S/B g/f** |
| selective genetic birth for 3 live neighbours | 0 | 0 | 3 | Bg3 |
| selective genetic birth for 2 live neighbours | 1 | 0 | 2 | Bg2 |
| enforce birth for 3 live neighbours | 2 | 0 | 3 | Bf3 |
| enforce birth for 2 live neighbours | 3 | 0 | 2 | Bf2 |
| selective survival for 3 live neighbours | 4 | 1 | 3 | Sg3 |
| selective survival for 2 live neighbours | 5 | 1 | 2 | Sg2 |
| enforce survival for 3 live neighbours | 6 | 1 | 3 | Sf3 |
| enforce survival for 2 live neighbours | 7 | 1 | 2 | Sf2 |
| birth overwrite for 3 live neighbours | 8 | 1 | 3 | O3 |
| birth overwrite for 2 live neighbours | 9 | 1 | 2 | O2 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| enforce GoL rule if last rule was non GoL rule | 10 | 0/1 | 2/3 | Nr |
| enforce GoL rule if last state change was performed by non GoL rule | 11 | 0/1 | 2/3 | Ns |

Table 1 Options for the control of **genelife** restricted to 2 or 3 live neighbours. The first column of the table records the transition processes extending the GoL rules. Of the 4096 options opened up by this table, only 9/16 i.e. 2304 of them are distinct because only 3/4 of the selective/enforced survival options are distinct. The remaining columns consist of an index number nr, the central cell state c to which the transition applies, the sum s of live neighbours, the transition notation where S and B stand for survival and birth and the subscripts g and f for genetic and enforced. All options except for nr 4,5 have been realized in the genelife software.

### Genetically selective birth/survival and choice of ancestor

We shall return to the genetic extension of the unmodified GoL shortly, but it is convenient to first consider the simpler process of birth involving only two individual genes. Since the simplest hyperactive modification to the GoL involves the additional B2 birth process, we begin the structural analysis of the genetic extension with this simplest case of two live neighbour birth. Even in this simple case there are still several decisions to be made associated with the choice of ancestor, even after the choices in Table 1 have been made.

Fundamentally, models of selection distinguish two modes of selection:

1. Selection based on comparing fitness as a property of a single individual, independent of the presence of other individuals in the neighbourhood. This has the property of well-ordering all the genetic sequences (by fitness), with transitivity in comparisons ensured: *i.e.* A>B and B>C implies A>C.
2. Selection based on a contest or tournament between individuals in which fitness depends on the other individual involved, there is no transitive ordering of genomes, and in population terms the fitness of a genome is population density dependent. Perhaps the best-known example of strongly not well-ordered fitness is in the scissors-paper-stone game A >B>C>A.

We shall consider examples of both these modes in this work.

Secondly, selection models distinguish neutral selection, in which genetic differences do not influence the survival or birth outcome, and non-neutral selection. We shall address both neutral and non-neutral cases, since even the locally neutral evolution cases can generate interesting structure in the tournament selection case. An important simple special case of tournament selection is selection which only distinguishes whether two genomes are the same or different and makes a neutral choice of which will be the ancestor for offspring between them if this criterion is fulfilled:

1. Birth only if two genomes are the same (neutral selection in that case)
2. Birth only if two genomes are different (neutral selection in that case)

#### Deterministic resolution of neutral selection

In general, if two genomes are different and a neutral outcome is sought, then some other mechanism must be invoked to choose an ancestor for the newly born genome. The conventional population genetics approach of choosing one of them randomly adds a major source of stochasticity to the otherwise deterministic GoL. It turns out there are a number of possible alternatives:

1. Random choice of live neighbours for birth
2. Distinguish live neighbours for birth by their position in the configuration
3. Examine the neighbourhoods of live neighbours to distinguish them

Both (ii) and (iii) suffer from potential ambiguity if the live neighbours remain identical under the distinction. We obviously would wish to preserve a certain degree of spatial symmetry in both the alternatives (ii) and (iii). In the appendix A, we catalogue and illustrate the different configurations of live neighbours for the non-trivial cases of s = 2,3,4. The cases s=0,1 are very simple by comparison, and the cases 5,6,7,8 can be obtained simply by exchanging zeros and ones in the figures. For the GoL B3 rule, we note that there is a very simple generic principle for choosing a single ancestor among the three live neighbours positionally, and one that does not break any of the spatial symmetries considered: choose the one at the “most different” position. This most different position is indicated in green in the figure in the appendix. Generally, it turns out that two of the three positions are related to each other by more symmetries than the different one. Now this is very good news, because it means that a deterministic inheritance scheme for neutral selection based on spatial position can be achieved without breaking spatial symmetries. Because approach (iii) is incomplete for the many cases when the live neighbours themselves have equal numbers of live neighbours, and because a realization of (ii) that works for B3 has been found, we do not pursue (iii) further in this paper.

A somewhat weaker, but still valid procedure that generalizes (ii) to other numbers of live neighbours (e.g. 2,4, etc) is to recognize that the choice of the most different position for B3 can be broken down into three steps: a) find a canonical representation of the pattern of live neighbours which represents all symmetric versions of the pattern (under one of the chosen symmetries above) b) specify the absolute position of the chosen position in this canonical representation c) transform this position relative to the canonical representation back to the “orientation” of the particular starting configuration. It turns out that since the canonical rotation is mapped symmetrically to each possible instance that even making a simple choice such as the first position in the canonical representation gives rise to a positional inheritance rule with symmetry preserving properties. However non-trivial genetic dynamics such as genetic rotors for GoL oscillators or still lifes are possible.

We illustrate this principle first for the case of 8-rotation symmetry, which turns out to play a pivotal role in the analysis, and then extend it to other (more physical) symmetries. The distinguished configurations for s=2,3,4 are shown in the left column of figures A1-A3 for the 4,7,10 canonical rotations. These are simply and efficiently defined as the 8-rotation of the 8-bit binary pattern of live neighbours that has the smallest numerical value. All the different configurations for s=0-8 live neighbours given by the binomial coefficients 8Cs (1,8,28,56,70,56,28,8,1) reduce to (1,1,4,7,10,7,4,1,1) configurations distinguishable up to 8-rotation symmetry. Note that these numbers only differ from 8Cs/8 for s=0,2,4,6,8 and because of the 6 ambiguous canonical bit patterns 00000000, 00010001, 00110011 and 01010101, 01110111, and 11111111 (*i.e.* patterns that can be rotated into themselves with less than 8 single steps). For these patterns only, an alternative rule must be found to choose the ancestor if we allow B0,B2,B4,B6 or B8 rule extensions. In our implementation for these special cases, we coded the following 8 disambiguation options which are mostly deterministic but include one spatially and one genetically random option:

1. random choice: this involves a departure from determinism for these cases only
2. ignore problem and choose selected bit of canonical configuration: accepting minimal asymmetry induced by these comparatively infrequent (for s=1-7) cases.
3. disallow birth: effectively modifies the rules and is like excluding these rules from the table
4. choose lesser in value of genes if different (otherwise it makes no difference) i.e. revert to non-neutral genetic model in these (rare) cases only
5. similar to 4, choose gene with least number of ones and if same, then lesser in value
6. choose a recombinant AND of all genes involved in this case
7. choose a default ancestor such as the gene coding for the Game of Life in these cases only
8. generate a random gene to give birth to for these ambiguous instances

The choice 6 is potentially minimally disruptive, effectively just reducing the rate of departure from GoL dynamics, and is better in most circumstances than option 5 that is also symmetric but creates a non-trivial correlation between dynamics and genetic change. However, depending on the investigation, each of the techniques has its strengths and weaknesses.

#### Models of selective difference

## Genetically encoded semi-totalistic birth and survival rules for s=1-8

### Live state determination

### Genetically selective birth/survival and choice of ancestor

## Semi-totalistic rules s=1-8 with LUTs for birth and survival determined by genes (16)

For the semi-totalistic case, if we exclude the special cases of spontaneous birth (B0 *i.e.* birth for s=0) and lone survival (S0) then there are 8+8=16 distinguished states that may be independently part of an active next state ruleset. Thus, there are 216 CA rulesets, and these may be encoded by a binary genome of length 16 with one bit per LUT entry. In this paper, we restrict our attention to genomes of maximum length 64, and often use the term genome to refer to this full sequence, reserving the term gene to refer to a specifier of part of the rule-table. We may also employ multiple bits (ncoding) to encode each LUT entry for an active rule, for example with only one of the possible ncoding gene patterns being active, then genomes of length 16x ncoding are required. A modular variable length encoding could be to encode the values of s (requiring 3 bits) as well as the central state (for survival or birth) for which the next state is live i.e. 4 bits in total per entry. The standard game of life would require 3\*4 = 12 bits to be specified. Longer genomes could contain the same entry repeatedly allowing for mutational error resistance.

Discuss omission of s=0 and consequences (cf Eppstein many cases with S0 allowed).

## Quarter-totalistic rules s=1-8, se=0-4 with distinct corner and edge counts (46)

Bring in the excel file counting the weighted distances, showing that each s,se configuration is different, therefore it warrants consideration in the totalistic family of rules.

## Eight-rotation symmetric genetically encoded rules for s=2-6 (64 configs)

Justify and explain canonical rotation construction.

## Spatial 2D symmetric genetically encoded rules (4-rot’n,4-refl’n) for s=0-4 (64)

## Fully distinguished 512 configurations

Postponed for future work, requires longer genomes for investigation.

# Genome propagation

## Movement vs copying

Whereas the indistinguishability of “1” states in binary CAs means that it is not possible to distinguish movement from death and rebirth, in this paper with genetic information attached to the live states, it is possible in principle. Is there a meaningful assignment of a subset of GoL birth or survival transitions to movement? It would make a difference if mutation were deemed not to occur for transitions involving movement. Also, it might be appropriate to make the choice of an ancestor sensitive to the interpretation of movement *vs* birth: e.g. to minimize the number of births needed to maintain the dynamics. For example, an isolated rod of three live states is a GoL oscillator between vertical and horizontal configurations. In the deterministic most different ancestor canonical assignment of ancestors from three live neighbours, the central gene is copied to two new sites so that (without mutation) the rod becomes genetically homogeneous in one step. This process is clearly a copy mechanism. On the other hand, in the 0-bit canonical assignment of ancestors, the two peripheral genes circulate anti-clockwise and this is more naturally understood as a process of motion and as such should be carried out without mutation.

# Indirect local rule modifications

Regarding the transfer of genetic information, a deterministic selection mechanism is realized, so that mutation provides the only source of random variation in the dynamics. Four levels of perturbations on the B3/S23 standard GoL rules are investigated: (0) genetic selection on the GoL, with the genes not influencing the GoL rules (1) genetic selection on GoL-like rules, e.g. S2gb3gB2g3 ∈ {S(2g(b))(3g(b))/B(2g)3(g)}, with conditional rules depending on genes denoted by g and survival rules involving birth overwrites denoted by b (2) genetic modulations of the live neighbour counting process that allow different numbers of live neighbours a) for specific 2nd neighbour ring configurations b) using masks on 1st neighbours encoded by the existing live neighbours c) with the gene specifying the allowed numbers directly (3) limitations of the influence of arbitrary gene encoded rule departures in varying symmetries through the requirement that state changes induced by a non-GoL rule inhibit further rule departures in their neighbourhood until corrected by a regular GoL rule. Whereas many perturbations of the GoL rules either quickly die out or proliferate rapidly to fill space with reproducing structures, a family of interesting dynamical systems is found, and its tendency towards open-ended evolution analysed by means of activity statistics. Note, that while the standard B3/S23 game of life, starting from random patterns of 50% 1s and 0s, in most instances does not produce complex dynamics, the new family does.

Notation: Rule extensions  [Golly](https://en.wikipedia.org/wiki/Golly_(program)) open-source cellular automaton package

## Example: selection = 7

## Example: selection = 0

## Example: selection = 2

## Example: selection = 4

## Example: selection = 5

# Direct encoding of local CA rule by genes

## Example: selection = 8

## Example: selection = 9

# Discussion

# Extensions for further papers

## Incorporate specific search algorithms for gliders, spaceships and more

<https://www.ics.uci.edu/~eppstein/ca/search.html>

## Genetically controlled coupling to other GoL dynamics

### Example: selection = 24: 2 planes

### Example: selection = 16: 16 planes coupled pairwise controlled by genes

### Example: selection = 17: 16 planes coupled pairwise controlled by genes 2nd example

### Example: selection = 18: 16 planes coupled to subset of nearest planes controlled by genes

### Example: selection = 19: 16 planes coupled to subset of nearest planes controlled by genes 2nd example

## Genetically controlled 3D GoL

Carter Bay proposed investigated possible extension of the Game of Life to 3D, finding that amongst the possible semi-totalistic rules with 26 neighbours there were strong constraints for rules that exhibited the central properties of the game of life (with E=S and F=B in our notation):

*“Definition 1. A rule ElEuFlFu defines a "Game of Life" if and only if both of the following are true.*

*1. A glider must exist and must occur "naturally" if we apply ElEuFlFu repeatedly to primordial soup configurations.*

*2. All primordial soup configurations, when subjected to ElEuFlFu, must exhibit bounded growth.*

*(Here we define primordial soup as any finite mass of arbitrarily dense randomly dispersed living cells.)”*

In particular, 5≤Fl≤9 are hard constraints to ensure 1. and 2. and Carter focused on the range 4 to 7 as most relevant to GoL-likeness for both E and F. He found only two rules 4555 and 5766 to satisfy definition 1, and of these only the rule 5766 supported an extension of many 2D-GoL objects to 3D (by plane duplication). It seems that in 3D, as for 2D, the semi-totalistic rules provide very tight constraints without significant alternatives to the known GoL rules. Bay also discusses possible extensions to rules that distinguish the 26 neighbours into three classes: face-centered (6), edge-centered (12) and corner (8) sites. Given the preferred z-axis of our asymmetric 64xNxN space and the desire to relate 3D rules to 2D rules it makes sense to further distinguish the (xy) in-plane and out-of-plane sites yielding 5 classes with 4,2,4,8,8 members. The number of rules with the total sum in the range 4-7 is then …

### Example: selection = 16,20

# Appendix 1 Symmetry models and their properties

In this appendix we list the various symmetry models implemented in the *genelife* software and describe for each case the two encodings of genetic information for the local rule-tables of the cellular automaton: fixed-position and modular (variable position). In each case we show how the symmetry model can be encoded meaningfully with 64-bit genomes.

1. **semi-totalistic case s = 2,3**. There are 4 possible configurations that can possibly result in a future live cell state, two for birth and two for survival, resulting in 16 possible rules. In this case, we extend these by considering detailed additional options for ancestry as in Table 1. We do not consider genetic encoding of rule tables in this case. The genetically encoded version of this case can be studied using the appropriate birth-survival mask under case 2.
2. **semi-totalistic case *s* = 1 to 8**. Distinguish configurations only by *s*. In addition to the 16-bit gene encoding (excluding *s*=0 entries) we also consider the variable gene length encoding involving entries of length 4-bits each. As discussed in the main text, there are at most 2\*8 LUT entries for active genes. The GoL is encoded as 0x…………0406 in the fixed length encoding and any combination containing each of the hexadecimal digits a,1,2 at least once in the modular encoding. Other digits do not affect the rule table if the corresponding digit is masked out in the birth-survival mask. Note that the variable genome length encoding allows the rule-tables of different genomes to be combined simply: any entry that is defined is valid irrespective of multiplicity and position.
3. **edge-corner quarter-totalistic case *s* = 1 to 7, *se* = 0 to *s***. Distinguish the four diagonal or corner sites (NW,NE,SE,SW) from the four edge-centred sites (N,E,S,W with total live count *se*) in an otherwise semi-totalistic rule-table, . The numbers of partitions of *s* into these two classes of sites for *s*=1…7 are 2,3,4,5,4,3,2 = 23 *i.e.* 2\*23=46 bits are required for a direct fixed gene length encoding. The variable length genome encoding as above would here require 7-bit entries (excluding as before the s=0 entries). Instead, we employ a condensed description in which a modular rule entry with eight bits encodes B/S, s and a mask for a subset of the first 4 of the at most 5 values of *se*. The configuration s=4, se=4 is encoded by the otherwise unused configuration 0x01 for survival and 0x81 for birth. In this way the GoL can be specified with just 3 module entries (*i.e.* 24 bits), and all possible rules with specific subsets of s active for s=1 to 7 can be specified in 56 bits.
4. **8-fold rotation symmetry case s = 2 to 6, *crot* = 0 to *crot*max(*s*).** Distinguish all configurat-ions that are not an 8-fold rotation of one another. This is both a simpler and somewhat smaller symmetry partition of the 8-neighbour configurations as *s* varies from 0 to 8 with 1,1,4,7,10,7,4,1,1 and a total of 36 distinguished configurations. Configurations can be quickly mapped to their symmetry class, by finding the rotation of 8 bits that yields the minimal numerical value (the latter identifies the different successive canonical rotations *crot* for each value of *s*). Restricting attention to the central *s*-range of 2-6 gives 32 distinguished configurations and 64 LUT entries for survival or birth. This fits neatly into a 64-bit integer genome. The variable length genome encoding would require 8-bit entries for individual rule cases, but as above this would require too many modules for rules like the GoL rule to be specified *i.e.* 8\*(4+2\*7)=144 bits. Instead, as in the previous case, we adopt an intermediate modular approach in which the possible up to 10 distinguished subcases of rules for each s are grouped as at most two groups of 5 bits, the latter being masked in each modular rule, and the rule specifies B/S, *s*, [*crot*/5][[1]](#footnote-2)◊, *crot* mod *5* mask with 1+3+1+5=10 bits in each module. In this way, only 6 modules are required to specify the GoL rule and this fits in a 64-bit genome.
5. **4-fold rotation and reflection symmetry in 2D for s = 0 to 4**. Distinguish all configurations that are not equivalent by 4-fold rotation and reflection symmetry in 2D. The numbers of these distinguished configurations for s = 0 to 8 are 1,2,6,10,13,10,6,2,1 *i.e.* in total 51. The minimum length encoding of a gene specifying this table with one bit per LUT entry would be 102. Instead we restrict attention in this work to the domain s = 0 to 4 which fits neatly into the 64 -bit genome with 2 by 32 bits required. Encoding with a variable length genome analogously to above would result in (1+3+4 = 8)-bit patterns needed to specify a LUT entry, and the standard GoL rule would require 8\*(6+2\*10) = 208 bits. Instead we make use of masks as above, dividing the at most 13 subcases of *crot* per *s* value into 3 masked sets of 6 configurations each requiring two bits to specify [*crot*/6]◊, 6 bits for a masked subset of *crot* mod 6 values, as well as the 1+3 bits for B/S and *s*. The GoL rule can thus be expressed modularly with 5 such 12-bit modules in 60 bits.
6. **Full asymmetry.** Distinguish arbitrary configurations ignoring symmetries: there are 256 binary neighbour patterns in the Moore neighbourhood, and hence 2512 possible rule-tables. We could encode these directly with a 512-bit genome or with a variable length genome by a sequence of 9-bit patterns specifying configurations (including the central site as a 9th bit) that give rise to live states. A maximum of 7 such rules would fit in a 64-bit genome and the standard GoL genome would have to be 9\*(C82+2\*C83)= 9\*(28+56)=756 bits in length. For completeness, we could also consider the continuous interval subset of rules defined by upper and lower bounds for the integer value of the 8-bit neighbor configuration for survival and birth, for which rule-tables can be specified by 2\*(8+8)=32 bits. We do not investigate this case 6 further in this work.

1. ◊ Here [x] represents the integer part of x. [↑](#footnote-ref-2)