# Wains: A Pattern-Seeking Artificial Life Species

**Abstract** We describe the initial phase of a research project to develop an artificial life framework designed to extract knowledge from large data sets with minimal preparation or ramp-up time. In this phase, we evolved an artificial life population with a new brain architecture. The agents have sufficient intelligence to discover patterns in data and to make survival decisions based on those patterns. The species uses diploid reproduction, Hebbian learning, and Kohonen self-organizing maps, in combination with novel techniques such as using pattern-rich data as the environment and framing the data analysis as a survival problem for artificial life. The first generation of agents mastered the pattern discovery task well enough to thrive. Evolution further adapted the agents to their environment by making them a little more pessimistic, and also by making their brains more efficient.

#### Amy de Buitléir\*,\*\*

Athlone Institute of Technology

#### Michael Russell<sup>†</sup>

Athlone Institute of Technology

#### Mark Daly<sup>†</sup>

Athlone Institute of Technology

#### Keywords

Artificial life, agent, diploid, wain, data mining

A version of this paper with color figures is available online at http://dx.doi.org/10.1162/ artl\_a\_00074. Subscription required.

#### I Introduction

*Data mining*, also known as *knowledge discovery from data* (KDD), is the process of discovering interesting and useful patterns in large databases [15, p. 33; 30, p. 7]. It is distinct from performing a search or query on a database; in data mining we are looking for information when the relevant factors and relationships are unknown. The goal can be descriptive (e.g., to model and understand the data) or predictive (using some of the variables to predict other variables), or both [10, p. 5].

There is a growing need for data mining to solve business problems in many sectors. Consider the telecommunications industry, for example: As telecom networks become more automated and capable of self-organization and configuration, they produce ever increasing amounts of data. It is necessary to identify patterns that could foreshadow future problems, to diagnose possible conflicts between configuration changes made by automated agents, and to recommend (or automatically implement) solutions. More traditionally, data mining can be used to identify fraud, to identify network faults, and in customer profiling for more effective marketing [29].

Artificial life has been used for data mining. Techniques modeled on insect behavior are common because "*at some level of description* it is possible to explain complex collective behavior by assuming that insects are relatively simple interacting entities" [3]. In particle swarm optimization (PSO), each candidate solution to a problem is called a *particle*. A particle moves in the solution space according to simple mathematical rules, influenced by its local best known position and the positions of its neighbors [1, Section 1.4]. Ant colony optimization (ACO) is modeled on the behavior of ants finding a

<sup>\*</sup> Contact author.

<sup>\*\*</sup> Software Research Institute, Athlone Institute of Technology, Athlone, Ireland. E-mail: amy@nualeargais.ie

<sup>†</sup> School of Engineering, Athlone Institute of Technology, Athlone, Ireland. E-mail: mrussell@ait.ie (M.R.); mdaly@ait.ie (M.D.)

trail between their colony and a source of food; it has been used primarily for supervised classification [1, Section 1.5; 7; 22]. Techniques modeled on the sorting behavior of ants have been used for data clustering [22]. Another technique, which is not (necessarily) insect-based, is the prey model: When a predator encounters prey, it must decide whether to attack or continue searching for better or easier prey. This technique has been used to reduce the dimensionality of clustered data [9]. Autonomous agents that do not use swarm intelligence, or that are not directly modeled on biological life forms, have also been used for data mining; Ultsch's DataBots are an example of the latter [28].

The use of autonomous software agents (artificial life) in data mining seems likely to increase. Cao argues persuasively that there is a synergy between these two streams of research. They face mutual challenges such as distributed, parallel, and adaptive learning. Both require applications that can understand and represent the interactions between components in their domain [5, Chapter 1].

Many artificial life techniques used for data mining require the user to evolve a life form that can (1) survive and (2) perform the required task. This evolution is time-consuming, and is not guaranteed to be successful. It might be easier and faster to choose from an array of pre-evolved species of agents and adapt them to analyzing a particular data set. To give a real-world parallel, suppose we need an organism that can herd sheep. Evolving such an organism from a one-celled life form would require too many generations to be practical. Even starting from a sophisticated animal such as a rabbit would take too long. And though we know it *is* possible to evolve a suitable organism (sheepdogs do exist), there is no guarantee that any particular approach will be successful. It is much easier to take an existing animal, such as the dog, which has the behaviors we need, and train it to do the required task.

Consider the skills that make a sheepdog so useful on the farm, such as heading (circling the sheep), using the eye (out-staring a sheep), shedding (separating one sheep from the flock), leading, and catching an animal with an open mouth (without biting). These behaviors did not evolve to make the dog useful on the farm, but to make it a successful predator [6]. Assuming the dog has intelligence and a suitable temperament, it can be trained, and these behaviors can be adapted to our needs.

Inspired by this analogy, we consider the characteristics that would allow an artificial life species to adapt readily to data mining tasks. We begin by defining the *tasks* we would like them to perform:

- Discover interesting and previously unknown patterns (to assist with cluster analysis).
- Discover unusual combinations (anomaly detection).
- Find correlations (association-rule mining).
- · Classify new data.
- Make predictions based on incomplete data or about future events (predictive analytics).

Next, we identify the skills required to perform those tasks. This list summarizes the *requirements* for our artificial life species:

- The ability to survive long enough to produce viable offspring.
- The ability to discover previously unknown patterns in data.
- The ability to classify new data based on known patterns.
- The ability to make decisions based on the patterns recognized, and take appropriate action for survival.
- The ability to adapt to changes in the patternicity of the data.
- The ability to learn during its lifetime.

Recall that the skills that make a sheepdog useful were developed for its own survival, not for herding sheep. If we interpret the list above as a set of *survival* skills, it suggests that we need a species of agents that live in, and subsist on, data. In a sense, any artificial life population can be

said to live in a universe of data; the agents are computer programs or subroutines, and all programs do is manipulate data. However, these requirements call for a fundamentally different type of interaction between the agents and the data they analyze. If data analysis is a survival problem, the environment cannot be a simulation. The environment must be real—real data. There should be no layer of abstraction between the agents and their environment.

If a species of agents with these skills already existed, how could we train them to analyze a particular data set? One possibility is to introduce the data into the environment as a new food source, and gradually reduce the availability of the old food source, or data. The agents should quickly learn to take advantage of the new food source. We can then analyze the behavior of the agents, and peer into their brains, to see the patterns they have discovered. This is one of the reasons we identified the ability to adapt to changes in patternicity as a required skill. Of course, this skill is also useful if the data to be analyzed changes patternicity over time.

We can summarize the list of required skills by saying that we want not just artificial life, but artificial life endowed with artificial intelligence (AI). Unfortunately, we are not aware of any existing artificial species that has all of the skills listed above. Two systems that do combine artificial life and AI were developed by Larry Yaeger and Steve Grand. Yaeger developed PolyWorld [13, 31–34], a cross-platform (Linux, Mac OS X) program that uses artificial life with aspects of artificial intelligence. Grand developed Creatures [11, 12], one of the first commercial games to combine artificial life with machine learning. However, neither of these projects was designed with data mining in mind; as a result, the agents do not have the level of intelligence that we require.

In this first phase of the research project, we attempted to create a life form satisfying the requirements listed above. The structure of this article is as follows: Section 2 describes the environment. Section 3 describes how the agents sense their environment, make decisions, and learn from their experiences. It also describes how they eat and lose energy, mate, rear children, and play. Section 4 describes how the agents are constructed from their genes. Section 5 describes the experimental procedure. Section 6 analyzes the results obtained. Section 7 summarizes our findings and describes some future avenues for research based on this project.

## 2 The Wain Ecosystem

For this project, we created a computational ecosystem<sup>1</sup> with two types of objects:

- Wains,<sup>2</sup> an artificial agent.
- Numerals, a potential food source or toy, which consists of a 28 × 28 grayscale image of a handwritten numeral. Each numeral has different characteristics and uses. The numerals are chosen at random from the MNIST database of handwritten digits [20, 19].

A program counter is used to schedule events; the units of time are called *clock ticks*.

#### 3 The Wain

We wanted wains to become "smarter," both during an individual's lifetime and over generations. To support this goal, the environment had to be challenging. Wains have a choice of responses to each situation and learn through trial and error. The young begin to build a mental map of their environment while still under parental care and protection. We attempted to model processes known to work in nature wherever it seemed practical to do so; this philosophy guided many of our decisions, such as the use of Darwinian evolution rather than Lamarckian evolution.

I The code is available from the authors, under an open-source license.

<sup>2</sup> Wain (rhymes with "rain," or alternatively, with "mean") is a word for "child," commonly used in Donegal and Northern Ireland.

This section describes the brain structure, appearance, and metabolism of wains. It also describes how they interact with their environment: how they make decisions and learn from their mistakes, and how they eat, mate, rear children, and play.

# 3.1 Brain Structure

The brain structure for wains was designed in advance; evolution was allowed to fine-tune the parameters and discover the factors that should influence a wain's decisions. The brain consists of two parts, a *classifier* and a *decider*, as illustrated in Figure 1. Each external input to the wain's senses is a  $28 \times 28$  array containing a grayscale image of either a handwritten numeral or a fellow wain. The classifier receives this input as a one-dimensional vector of length 784, and categorizes the input vector as belonging to one of the patterns that the wain knows. The decider chooses the course of action based on the pattern ID, along with the wain's internal status. These components are described below in more detail.

We allowed evolution to control multiple aspects of the brains of wains in the hope that improvements to the brain would occur over generations. The brain receives two types of inputs: internal and external. The number of external inputs is specified by the *extenception capacity gene*. When deciding how to react to its environment, a wain must also take into account its own status. This internal data includes factors such as the wain's current hunger, passion, and boredom levels. The number of internal inputs is specified by the *interception capacity gene*. (A complete list of genes is provided in Table 1 in Section 5.)

# 3.2 Learning Patterns

A wain may receive tens of thousands of unique input vectors during its lifetime; it must be able to discover and recognize patterns in order to develop general guidelines for making decisions. This is the job of the classifier, which is a modified Kohonen self-organising map (SOM) [17, 18]. Kohonen developed the SOM as a computational method for analyzing high-dimensional data.

A SOM maps the input vectors onto a regular grid (usually two-dimensional) in which each node has a weight vector representing a model of the input data. (The components in a SOM are usually referred to as nodes rather than neurons.) Kohonen's technique ensures that any topological relationships within the input data are also represented in the grid. The training of the grid is *unsupervised*; no target pattern is required.

The algorithm for implementing a SOM is straightforward. For each input vector (input pattern) presented, the following steps are performed:

1. A *winning node* is selected. The node chosen is the one that is most similar (in some sense) to the input vector. Often Euclidean distance is used as a measure of similarity.



Figure I. A schematic diagram of a wain brain.



Figure 2. Samples of different styles of the numeral 2, from the MNIST database: (a) with loop, (b) plain.

- 2. The weight vector of the winning node is adjusted to make it slightly more similar to the input vector.
- 3. The weight vectors of all nodes within a given radius of the winning node are also adjusted to make them slightly more similar to the input vector, by an amount that is smaller the further the node is from the winning node.

Step 3 ensures that as more and more inputs are received, nodes that are physically close respond to similar patterns in the input data; as a result, the map preserves the topology of the input data. However, the goal is for wains to discover and remember patterns that are useful to their survival, not to decide whether a handwritten 2 is more similar to a 9 or a 3. Since topology preservation is not required for this project, the modified SOM used in the wain classifier omits step 3; this allows faster processing. The other modification made to the SOM was to periodically erase the least useful node so that the classifier can learn a new pattern.

Although the classifier is working with images of handwritten numerals and other wains, we do not require that objects be classified into exactly 11 patterns (10 for the digits from 0 to 9 plus one for wains); the goal is not to implement a numeral recognition system. The wain genome allows for extra patterns so that wains can learn common variants of the objects they encounter. For example, a wain might identify two separate patterns for the numeral 2; one written with a loop (see Figure 2a), and one without (see Figure 2b). The number of patterns that a wain can recognize is specified by the *pattern capacity gene*.

Evolution is free to create wains that can remember as many patterns as desired. However, as will be discussed in Section 3.6, the energy cost of living is partly based on capacity to remember patterns; this was done to prevent the evolution of excessively large, inefficient brains. Each wain encounters numerals in a different order, so one wain might identify handwritten 3s as pattern #7, while another wain identifies them as pattern #2.

When a wain is born, it has no experience of the world, and therefore it must be able to learn patterns very quickly. The patterns that a wain recognizes in its environment are fluid categories, which shift or become broader or narrower according to the inputs received during its lifetime. The rate at which the SOM modifies each pattern node is specified by the *pattern learning rate gene*.

Later in life, the wain has a workable set of patterns that have helped it to survive. It must continue to learn, but it may not be practical to make dramatic changes to its view of the world based on one experience. For example, if a wain eats a 3, believing it to be edible, but experiences a reduction in energy, it could be that the environment has changed so that 3s are now poisonous—but it is more likely that the 3 was misidentified. If the wain decides that 3s are now poisonous based on a few bad experiences, it may be neglecting a valuable energy resource. For this reason, the pattern learning rate was designed to decay over time. This rate of decay is specified by the *pattern learning rate decay gene*.

The number of patterns that a wain can remember is fixed at conception. If a pattern turns out not to be useful, it may be better to forget it and start fresh. The number of matches for

each pattern is tracked. At intervals specified by the *Edelman cycle gene*,<sup>3</sup> the least useful pattern (the one that has the fewest matches) is discarded, and the node is randomized so that it can learn a new pattern. This creates competition between patterns, so that only useful patterns persist. The number of matches for all patterns is then cleared, so that the new pattern has a fair chance to compete for survival.

# 3.3 Making Decisions

The core of the decider is a basic Hebbian<sup>4</sup> neural network. Each row in the weight matrix represents an action (eat, mate, play, ignore), and each column represents one of the pattern IDs produced by the classifier. The values in the weight matrix can be negative (for undesirable actions), zero (for neutral actions), or positive (for desirable actions). The input vector consists of the following information, which is called the *context*:

- The wain's current hunger (b), boredom (b), and passion (p) levels, each of which is a number between 0 and 1. (The wain's energy level, e, is related to the hunger level by the equation e = 1 b.)
- A list of values, where each element is zero except the element whose index corresponds to the pattern ID; this element has the value 1.

The values of the four output neurons represent the perceived desirability of each of the potential actions (eat, mate, play, ignore), and are used to calculate a proportional vote for each action, as follows:

$$votes = \begin{cases} 100 + 1000 \times output, & output \ge 0\\ 1, & output < 0 \end{cases}$$
(1)

The final decision is made by weighted random selection, using the votes as weights. This ensures that desirable actions are chosen far more often than actions that are perceived to be undesirable, but also that the wains will occasionally take risks. An action that had a bad outcome under one set of circumstances may have a good outcome in a different situation.

There is one case where the brain makes a decision without consulting the decider. If the wain's energy is below 0.1, the brain will choose to eat any object that the wain encounters. We implemented this to reduce the possibility that a young adult would starve before it learned to eat. However, this feature acted in less than 0.08% of all encounters, so it could be disabled in the future.

# 3.4 Learning to Make Better Decisions

If a wain chooses an action other than "ignore," it will know afterward if that was a good decision or a bad decision, and should adjust its future behavior accordingly. To support this, the wain's happiness is measured as a function of its energy, passion, and boredom:

$$happiness = e \times 7 + (1 - p) \times 3 + (1 - b)$$
(2)

The parameters 7 and 3 were chosen after some experimentation, but they clearly reflect the order of priorities: eating, mating, then playing.

The wain's happiness is calculated before a decision is made, and again after the action is taken. If the happiness has increased, the decider's neural network is trained with the context vector, using a positive learning rate. Otherwise, a negative learning rate is used for training. However, a wain

<sup>3</sup> Named for Gerald Edelman, who put forth the theory of *neural Darwinism*, which proposes that a type of natural selection occurs within the brain, forming new connections between neurons, and pruning connections that are found to be undesirable.

<sup>4</sup> A type of unsupervised learning where, if the firing of one cell regularly contributes to the firing of another, the strength of this connection tends to increase over time [16].

should not permanently alter its behavior based on one experience; it may have misidentified the pattern, or the result might be different under other conditions. The rate at which the decider reinforces decisions with positive outcomes is specified by the *positive learning rate gene*. The rate at which it reinforces decisions with negative outcomes is specified by the *negative learning rate gene*.

Wains need to be able to respond to changes in their environment. To allow this, the strength of each association (positive or negative) between a pattern and a possible course of action is reduced slightly at each time step. As a result, the associations will decay over time unless reinforced. The rate of decay is specified by the *decider aging rate gene*.

## 3.5 Appearance

Wains have a genetically controlled appearance; this provides a mechanism for wains to judge how closely related they are to potential mating partners, and allows for the possibility of kin selection. The appearance of a wain is a  $28 \times 28$  grayscale image; the value of each pixel is specified by an *appearance gene*. The appearance of the wains in the starter population is shown in Figure 3a; this shape was designed to be easy for the wains to distinguish from numerals. Wains learn to distinguish between objects that are suitable for mating with (i.e., others of their species), and objects that are not. If the wain is currently raising a child, a small temporary modification is made to the wain's image, as shown in Figure 3b. This feature was introduced to allow wains to identify mates that are likely to be receptive.

Over time, mutation and recombination cause the appearance of some wains to differ from that of the initial population. Thus, the appearance of wains provides a rough guide of the genetic variation in the population. If eventually they were to diverge into separate species (i.e., subpopulations whose genetic differences are such that offspring from cross-mating would not be viable), wains would be able to distinguish between their own kind and the other in the same way they distinguish between numerals.

## 3.6 Eating and Metabolism

Wains have an energy level, *e*, between 0 and 1. Some numerals are edible; eating them provides energy to a wain. Others are mildly poisonous; eating them decreases a wain's energy. (Eating poisonous food will only be fatal if it reduces the wain's energy to zero or below.) If a wain does not eat a numeral, it will not receive the energy gain (or loss). Since a numeral can be both a food item and a toy as described in Section 3.9, the optimal choice of action will depend on both how hungry the wain is and how bored it is. The assignment of numerals as edible or poisonous is part of the software configuration, and can be adjusted during a trial.



Figure 3. Appearance of wains in the initial population: (a) default, (b) while rearing a child. Note the additional white pixels in the upper left of (b).

Once a wain's energy reaches 1, it is full; continuing to eat is not beneficial, but it is not harmful. Everything a wain does, even just being alive, has an associated energy cost. At regular intervals, wains lose some energy through a *metabolism tax*, denoted  $e_{\text{metabolism}}$  and given by

$$e_{\text{metabolism}} = e_{\text{iq}}(n_{\text{ex}} + 10 \, n_{\text{pat}} + n_{\text{int}}) \tag{3}$$

where

emetabolism is the metabolism tax,

- $e_{iq}$  is a multiplier relating the brain complexity to its metabolic costs,
- $n_{\rm ex}$  is the number of external inputs to the wain's brain,
- $n_{\text{pat}}$  is the maximum number of patterns the wain's brain can differentiate,
- $n_{\rm int}$  is the number of internal inputs to the wain's brain.

This metabolism tax is determined by the complexity and processing requirements of the brain; this prevents wains from evolving excessively large, inefficient brains. If a wain's energy reaches 0, it dies and is removed from the population. In this implementation, we chose not to let dead wains be a food source for other wains, because we wanted to encourage numeral pattern recognition instead of scavenging. (In the first few generations, we anticipated that there would be a high death toll, and we were concerned that wains might ignore the numerals in favor of an easier source of food.)

#### 3.7 Mating and Reproduction

In biology, sexual reproduction may be adaptive. For example, it may support repair of chromosome damage [2, 8, 24], removal of deleterious mutations from the gene pool [25], or protection against parasites (the "Red Queen hypothesis") [14, 26].<sup>5</sup> It may be also beneficial for artificial life. Calabretta [4] found that "diploid genotypes create more variability in fitness in the population than haploid genotypes" and recommend diploid reproduction when good results for both average and peak fitness are desired. Smith and Goldberg [27] also found an increase in diversity for diploid populations, adding that "diploidy embodies a form of temporal memory that is distributed across the population."

In true sexual reproduction, however, only half of the population can bear children. In an effort to gain the potential benefits of sexual reproduction while avoiding this limitation, wains are diploid and reproduce sexually (i.e., each parent contributes a set of chromosomes), but they have only one sex. Wains have a passion level, p, between 0 and 1. When a wain encounters another wain, if it chooses to mate, it loses a small amount of energy for the time investment of *flirting*. If both partners choose to mate, and the wain that is randomly selected to be the carer is not currently rearing a child, the passion level of both wains is set to zero and a child is produced.

The wain genome consists of a sequence of building instructions encoded as a series of bytes. A single instruction begins with a byte indicating the type of instruction, followed by one or two bytes of data. (The available set of instructions is discussed in Section 4.) Each wain has two complete sequences of instructions, either one of which would be sufficient to create a wain. By loose analogy with biology, an instruction is called a *gene*, and the various settings for a particular instruction are called *alleles*. When two wains mate, they each donate one string of genetic material to the offspring. Again by analogy with biology, we call this single string a *gamete*. The first step in producing the gamete is to make copies of the two strings of genes from the parent, and to perform one or more of the operations listed below, in decreasing order of probability:

• Crossover: Breaking the strings at corresponding locations, and swapping the tails.

<sup>5</sup> Meirmans and Strand provide a useful summary [23].

- *Cutting and splicing*: Breaking the sequences at non-corresponding locations and swapping the tails, thereby ending up with two sequences of different length.
- Mutation: Randomly altering a bit in one of the sequences.

In biology, crossover is the most common of these operations, and it normally occurs only at gene boundaries. However, both crossover and cutting and splicing can occasionally work within a gene, resulting in duplications, deletions, and other defects [21, pp. 142–144]. Such events are rare, but can be "significant over the course of evolution" [21, p. 141]; for this reason we chose to allow them to occur in wain genetics. After the two sequences have been created, one of them is randomly selected as the gamete, and the other is discarded. The offspring receives one gamete from each parent, and thus ends up with two sequences of instructions. Thus, the offspring contains a mixture of genetic information from both parents.

Since the two sequences of instructions in a wain's genome are generally not identical, before constructing the offspring the sequences are merged into a single sequence of instructions, which we call the *blueprint*. As in biology, when the genetic instructions at corresponding locations differ, one instruction may take precedence over the other (i.e., have dominance), or the result may be a blending of the two instructions (i.e., there is incomplete dominance). These dominance relationships are applied to each corresponding pair of genes to compile the blueprint, which is then used to create the offspring. This will be discussed in more detail in Section 4.1.

# 3.8 Child Rearing

When two adults mate, each donates a fraction of its current energy to the resulting child. In addition, the carer donates a fraction of all the food it eats to the child until the child is mature. In both cases, the fraction is specified by the *devotion gene*.

After a child is born, it remains with the carer until it is mature. The age of maturity is specified by the *maturation time gene*. During this time, the child builds a set of patterns based on its experiences, as described in Section 3.2. However, it does not make decisions, nor make any adjustments to its decision matrix (as described in Section 3.4), until it is mature.

# 3.9 Play

Wains have a boredom level, *b*, between 0 and 1. A wain's boredom level is increased slightly, by a userconfigurable amount, at the same time the metabolism tax is applied. Some numerals are fun; playing with them reduces the wain's boredom. Others are boring; they either increase the wain's boredom or have no effect. Wains are configured to find each other slightly boring so that mating will usually be a more attractive option than playing. Ignoring a numeral or another wain has no effect on boredom. The assignment of numerals as fun or boring is part of the software configuration, and can be adjusted during a trial.

Once a wain's boredom reaches 0, continuing to play is not beneficial, but it is not harmful. A bored wain will not experience any ill effects; the option to play with objects was merely introduced to give the wains a richer life, which could eventually drive evolution to produce better brains.<sup>6</sup>

# 4 Wain Genetics

Since the focus of this project was the brain, we designed wains with a very simple body and metabolism; most of the genetic traits relate to the brain. The wain genome consists of instructions encoded as a series of bytes. Any byte that cannot be interpreted as part of one of the other gene sequences will be treated as a no-op instruction, which has no effect. This ensures that all gene sequences are valid and can be used to construct a wain.

<sup>6</sup> We have not determined if boredom and play are truly beneficial; this is proposed for future work (see Section 7.1).

# 4.1 Genetic Dominance

As discussed in Section 3.7, dominance relationships must be defined to handle the situation when the alleles at corresponding locations in the two gene sequences differ. Non-homologous gene combinations result in a no-op gene in the blueprint, which has no effect. For most homologous gene combinations, a type of genetic blending was implemented by taking the average of the two values. For the exteroception capacity gene, interoception capacity gene, and pattern capacity gene, the minimum of the two values was used in the hope that it would result in smaller, more efficient brains. We wanted to ensure that the wains would demonstrate the ability to learn and recognize patterns within the time frame allotted for this phase of the research project; the minimum of the two values for the maturation time gene was used to encourage younger mating and faster evolution.

# 5 Experimental Setup

# 5.1 Generating a Starter Population

For each trial, a small number (from 200 to 1000) of gene sequences was generated using the order and parameters specified in Table 1. (The same parameters were used for all trials.) Each gene sequence was duplicated, and a wain was constructed from the resulting pair of (identical) sequences. These wains formed the starter population for a trial run. The range for the maturation time gene was chosen based on the experience in early trials. The ranges for the pattern capacity gene, pattern learning rate gene,

Instruction	Parameter		
Devotion gene	A random integer between 0 and 255.		
Maturation time gene	A random integer between 100 and 500.		
Exteroception capacity gene	784, allowing input of 28 $\times$ 28 grayscale images.		
Interoception capacity gene	3 (for energy, passion, boredom).		
Pattern capacity gene	A random integer between 11 and 25.		
Pattern learning rate gene	A random integer between 90 and 110, which encodes for a learning rate between 0.90 and 1.10.		
Pattern learning rate decay gene	A random integer between 230 and 255, which encodes for a decay rate between 0.90 and 1.00.		
Edelman cycle gene	A random integer between 50 and 100,000. This range was chosen arbitrarily.		
Positive learning rate gene	A random integer between 10 and 30, which encodes a learning rate between 1 and 3.		
Negative learning rate gene	A random integer between 10 and 30, which encodes a learning rate between 1 and 3.		
Decider aging rate gene	A random integer between 230 and 255, which encodes an association aging rate between 0.90 and 1.00.		
Appearance gene	A sequence of 784 genes, encoding the image shown in Figure 3a.		

#### Table I. Gene sequence for the starter population.

Table 2. Setup for final wain trial.

ltem	Value	
Initial population	100	
Minimum population	50	
Metabolic cycle	1000 ticks	
e <sub>iq</sub>	0.000033	
Pcrossover	0.1	
Pcut-and-splice	0.01	
Pmutation	0.001	

and pattern learning rate decay gene were chosen after experimenting with a standalone implementation of the modified SOM. The ranges for the positive learning rate gene, negative learning rate gene, and decider aging rate gene were chosen after experimenting with a standalone Hebbian neural net.

At the start of early wain trials, food provided abundant energy. The logs were monitored to verify that the wains were learning to distinguish between the different kinds of objects in their environment, and make appropriate decisions. A little at a time, the amount of energy provided by food was reduced, to attempt to drive the population to make even smarter decisions. After each adjustment, the population size was monitored to ensure that the population was still stable.

## 6 Results and Interpretation

This section presents the results obtained from the final trial with wains. The setup for this trial is shown in Tables 2 and 3. The values chosen for  $p_{crossover}$ ,  $p_{cut-and-splice}$ , and  $p_{mutation}$  are not biologically realistic, but were chosen so that evolution would happen quickly enough to be observed during the time allotted for this phase of the research. The energy values were based on results achieved in earlier trials. We anticipated that the numerals 3 and 8 would be easily confused. By making one of them edible and one poisonous, we ensured that misidentification had a cost; in this way we hoped to drive the wains to develop good identification skills. We also chose some numerals to be both edible and fun, in the hope that the wains would develop more complex decision rules that take their current energy into account.

## 6.1 Population Stability

As shown in Figure 4, the population was self-sustaining. After the starter population was created, no wains were added except through birth. At clock ticks 744138, 752352, 791676, and 806168, the poisonous numerals (7, 8, and 9) were made successively more poisonous, to see how the wains would cope. (Table 3 shows the values for these numerals.) As shown in Figure 5, the first three changes seemed to have no effect, but after the last change, a dip was observed in the population size, followed by a recovery. The poison levels remained at the new (higher) settings, but the wains had adapted. Section 6.5 will provide some insight into how they adapted. No changes were made to the boredom relief offered by any numeral during this trial.

## 6.2 Eating Patterns

Figure 6 shows how the first generation of wains successfully learned to distinguish between edible and inedible foods. The time period shown is from the first generation, so this graph reflects learning

Table 3. Nume	eral characte	ristics for	final	wain	trial.

		Boredom				
Object	Energy	relief	Comment			
0	1.0	0	Edible, boring			
I	0.8	-0.I	Edible, fun			
2	0.6	-0.2	Edible, fun			
3	0.3	-0.3	Edible, fun			
4	0.2	-0.6	Edible, fun			
5	0.1	-0.8	Edible, fun			
6	0	-1.0	Edible, fun			
7	-0.05 (time 0 to 593023) -0.06 (time 593024 to 744137) -0.08 (time 744138 to 752351) -0.09 (time 752352 to 791675) -0.1 (time 791676 to end)	-0.2	Poisonous, fun			
8	-0.08 (time 0 to 593023) -0.09 (time 593024 to 744137) -0.11 (time 744138 to 752351) -0.12 (time 752352 to 791675) -0.15 (time 791676 to 806167) -0.2 (time 806168 to end)	0	Poisonous, boring			
9	-0.11 (time 0 to 593023) -0.12 (time 593024 to 744137) -0.13 (time 744138 to 752351) -0.14 (time 752352 to 791675) -0.2 (time 791676 to 806167) -0.3 (time 806168 to end)	0	Poisonous, boring			
Wain	-0.05	0.1	Poisonous, boring			

during a single lifetime.<sup>7</sup> During this period, each wain encountered 52 examples of each handwritten numeral, on average. Some of the wains in the initial population died quickly. If these wains were less adept than average at distinguishing numerals, after their death the accuracy of the population as a whole would rise, even if the remaining individuals made no further improvement. To allow for this effect, data excluding the wains that died during the period shown is also plotted; these are the lines labeled "filtered."

Figure 7 shows the eating pattern over a longer period of time. Only wains aged between  $11 \times 10^5$  and  $12 \times 10^5$  clock ticks are included. The age range was restricted so that the data would not be

<sup>7</sup> The average life span of wains in the first generation was  $3.87 \times 10^5$  clock ticks. However, wains do not die of old age; they can live indefinitely if they can avoid starvation. Twenty-four of the first-generation wains lived for more than  $32 \times 10^5$  clock ticks.



Figure 4. Wain population growth. The time span shown includes 12 generations of wains. The gray vertical band indicates a series of adjustments to the toxicity of poisonous numerals.

affected by changing demographics in the population (e.g., a sudden influx of young wains reaching maturity at the same time). The time span shown includes 12 generations of wains. As can be seen from the graph, overall, the wains tend to eat edible numerals and avoid the poisonous ones. In particular, they avoid trying to eat each other, as shown in Figure 8. If a wain does try to consume another wain, the first wain loses some energy; this mimics the waste of energy that would occur in the biological world.

Eating a poisonous numeral can be classified as a mistake on the part of a wain, but *not* eating an edible numeral is not necessarily a mistake. If a wain is not hungry, eating will not make it happier, so the numeral may have more value as a toy (by reducing the wain's boredom level, increasing its



Figure 5. Population changes in response to a harsher environment. The four vertical lines indicate adjustments to the toxicity of poisonous numerals.



Figure 6. First-generation wain eating patterns. Graph shows the fraction of encounters where the wains decided to eat (or try to eat) the object.

happiness). Or, if the numeral is boring (i.e., playing with it will not reduce boredom), then ignoring it is just as sensible as eating it.

However, wains do make mistakes, as can be seen from the number of poisonous numerals eaten. The situations in which a wain would eat a poisonous numeral are listed below:

- 1. Inexperience: The wain has not yet learned which numerals are poisonous.
- 2. Misidentification: One of the wain's patterns matches two or more numerals, at least one of which is edible.
- 3. Taking risks: As discussed in Section 3.3, the element of randomness ensures that the wains will occasionally choose a decision that is unlikely to have a good outcome.
- 4. Starvation: As discussed in Section 3.3, if the wain's energy is below 0.1, the brain will choose to eat any object that the wain encounters.

Figure 8 shows that misidentification plays a significant role. Only wains aged between  $11 \times 10^5$  and  $12 \times 10^5$  clock ticks are included, so the wains have roughly the same amount of life experience. By the time a wain reaches  $11 \times 10^5$ , it has encountered thousands of numerals,<sup>8</sup> so inexperience is not a factor. As shown in this graph, handwritten 8s, which are poisonous, are eaten far more often than other poisonous numerals. One reason for this may be confusion between 8s and 3s, which are edible. Similarly, handwritten 4s, which are edible, are eaten far less often than other edible numerals. Confusion between 4s and 9s, which are poisonous, may account for this.

Figure 9 shows the correlations between poisonous and non-poisonous numerals, using the same data as in Figure 8. Recall from Figure 8 that the number of attempts to eat other wains is consistently low. As will be shown, wains have a very clear pattern for members of their species, so any attempts to eat other wains are likely to be due to risk-taking rather than misidentification. The highest correlation involving an adult wain is with the numeral 3. This value, 0.636, is a suitable cutoff; any correlations above this line suggest misidentification. The highest two correlations are between 4s and 9s and between 8s and 3s, providing further support that the wains may be confusing these numerals. The

<sup>8</sup> At each clock tick, a numeral is offered to one of the wains in the population. As a rough estimate, we can assume that there are 250 wains in the population, so by the time a wain reaches  $11 \times 10^5$ , it will have been offered approximately 4,400 numerals.



Figure 7. Wain eating patterns. Graph shows the fraction of encounters where the wains decided to eat (or try to eat) the object.

only other correlation above the threshold is between 4s and 7s; confusion between these numerals is also plausible, as will be shown.

There is no direct way to find out which numeral a wain thinks a particular image is. The question is meaningless, because the wain has no concept of numerals; all it knows are patterns. If a wain chooses to eat an image containing a handwritten numeral, it is because that image is a reasonable match for one of the wain's patterns, and in the wain's experience, images that match that pattern are usually edible.



Figure 8. Detailed wain eating patterns. Graph shows the fraction of encounters where the wains decided to eat (or try to eat) a numeral, or a wain. Thick lines indicate edible objects; thin lines indicate poisonous objects.



Figure 9. Correlations between poisonous and non-poisonous objects. Correlations above the horizontal line indicate pairs of objects that the wains are likely to confuse.

However, there is an indirect way to gain some insight into the mistakes that wains make. Figure 10 shows the SOM of a young adult wain. Each node in the SOM has a vector of 784 weights; these have been converted into a  $28 \times 28$  matrix of grayscale values, allowing us to see the patterns that this wain has identified. The wains do not actually receive sensory inputs in two dimensions; the appearance of objects in their environment is presented to the SOM as a one-dimensional vector, so they are unaware that pixel 1 and pixel 29 are adjacent (in the same column in subsequent rows). The figure shows that some of the patterns identified by this wain are ambiguous, and that confusion of 3s with 8s, and 4s with 9s or even 7s, is plausible. This is confirmed by Figure 11, which shows the actual matches made by this wain.



Figure 10. A typical SOM. This the SOM of a young adult wain. Node (j) is a clear match for  $\times$ , the appearance of a wain. Most of the numerals are well defined. As is typical, there are two nodes that might match a 2: node (k) would match a 2 with a loop, while node (b) would match a plain 2 (or a heavily slanted 1). There is no clear representation of the numeral 4; nodes (h) and (n) are probably the best match for a 4, but (h) is a better match for a 9, and (n) is a better match for a 7. Node (d) might match a 5 or a 6.



Figure 11. Associating numerals with patterns. This histogram shows the patterns to which various numerals have been matched. This data is for the wain whose SOM is shown in Figure 10.

#### 6.3 Mating Patterns

As can be seen in Figure 12, wains quickly learned to prefer other wains, rather than numerals, as potential mating partners. Only wains aged between  $11 \times 10^5$  and  $12 \times 10^5$  clock ticks are included in the graph. So far, no cause for the temporary increase in flirtations with objects at clock tick  $21 \times 10^5$  has been determined.



Figure 12. Wain flirting patterns. Graph shows the fraction of encounters where the wain decided to mate (or try to mate) with the object.

Recall from Section 3.7 that when a wain encounters another wain, if it chooses to mate, it loses a small amount of energy for the time investment of flirting. If a flirtation is unsuccessful, the unlucky suitor ends up with less energy and no reduction in passion, and therefore has a lower happiness level. It appears that wains have learned to flirt when reproduction is likely, as demonstrated by the fact that they generally flirt in less than one-third of encounters, yet the population is thriving. There are three situations in which a wain's flirtation would be unsuccessful:

- 1. The other wain has chosen not to mate.
- 2. The other wain has been selected to be the carer, and it is currently rearing a child.
- 3. This wain has been selected to be the carer, and it is currently rearing a child.

A wain currently has no way to know another wain's passion level; if it did, it could learn to anticipate the other wain's decision, which would help it avoid situation 1. A wain's appearance indicates if it is raising a child. Wains do see each other's appearance, so they could use that information to avoid situation 2. In order to do that, they would need to have two separate SOM patterns, one for a wain with a child, and one without. However, in the >500 brain scans performed to date on wains, the only instances where there are multiple  $\times$ -like patterns in the SOM are those that would differentiate between mutant wains and normal wains, as will be discussed in Section 6.6.

A wain does not have a sensory input indicating if it is currently raising a child; however, it does know its current passion level. Since the passion level is reset to zero after mating, it could be used as a very rough measure of the likelihood that this wain is currently raising a child. It is a rough measure because a low passion level only indicates that the wain has mated recently; it does not indicate whether the wain was the carer or the non-carer in the last mating. Flirting only when its passion level is high would help a wain avoid situation 3.

But is this strategy being used? Figure 13 shows the values for one element of the decision weight matrix: the element that relates the wain's passion level to the likelihood that the wain will flirt, given an opportunity. As explained in Section 3.3, the weight matrix is multiplied with the vector of sensory inputs to create a weighted list of possible actions. Negative weights are possible; if the mating weight is negative, the wain may still flirt, but only rarely.

Over half of wains flirt only rarely and without regard for their passion levels; they have a negative weight at this location in the weight matrix. Somewhat less than half of the population do seem to follow the strategy outlined above; that is, they are unlikely to flirt unless their passion levels are high. There



Figure 13. Wain mating weights. Graph shows the values for one element of the decision weight matrix: the element that correlates the wain's passion level with the likelihood that it will flirt, given an opportunity. The statistics are taken from the wains that were alive at time 3308000.



Figure 14. Wain play patterns. Thick lines indicate fun objects (those that reduce boredom); thin lines indicate boring objects.

are a few wains in the population that have a very high (>4) weight, which means that even a low passion level makes them very likely to flirt. They have been given the nickname "Don Wains."<sup>9</sup>

## 6.4 Play Patterns

A wain never suffers any ill consequences from playing with a numeral, and suffers only a minor increase in boredom for playing with another wain. At best, the object will lower its boredom levels. At worst, the wain loses a potentially better opportunity, such as eating the object (if it is edible) or mating with it (if it is another wain). Figure 14 shows the play patterns for wains. Only wains aged between  $11 \times 10^5$  and  $12 \times 10^5$  clock ticks are included in the graph. Note that the rates for the numerals in this graph are roughly in the inverse order of Figure 8, which suggests that the wains prefer to play with the numerals that they are most reluctant to eat.

# 6.5 Wain Evolution

## 6.5.1 Learning Rate Genes

Recall from Section 3.4 that the *positive learning rate gene* controls the rate at which the decider reinforces decisions with positive outcomes, while the *negative learning rate gene* controls the rate at which it reinforces decisions with negative outcomes. Figure 15 shows how the values of these two genes changed over time. A slow but steady reduction in the value of the positive learning rate gene can be seen. The effect of this would be to make wains less likely to assume an action is wise because it had a good outcome on one or two occasions; instead, they wait for more evidence.

Another change that can be seen in Figure 15 may explain how wains adapted to the increases in toxicity of the poisonous numerals, described in Section 6.1. Immediately after this change was made, the value of the negative learning rate gene increased significantly, and eventually leveled off. The effect of this would be to make the wains more likely to avoid actions that had bad outcomes. Taking the changes to both of these genes into account, it seems that the wains have evolved a slightly more pessimistic view of their environment than was present in the initial population.

<sup>9</sup> A nod to the legendary lover, Don Juan.



Figure 15. Evolution of decider component in the brains of wains. The gray vertical band indicates a series of adjustments to the toxicity of poisonous numerals. The time span shown includes 12 generations of wains.

#### 6.5.2 Pattern Capacity Gene

As discussed in Section 3.6, the metabolism tax paid by wains is partly dependent on the pattern capacity of the SOM, which is determined by the pattern capacity gene. If a wain can reduce the number of patterns that it learns, without sacrificing its ability to identify enough edible food to survive, then it has an evolutionary advantage. As shown in Figure 16, wains have one fewer patterns now than the initial population did. The population continues to thrive, however, as evidenced by Figure 4.



Figure 16. Evolution of pattern capacity in the brains of wains. The solid line shows the pattern capacity for wains as a function of time. The downward-sloping dashed line is a linear regression for the first  $7 \times 10^5$  ticks; it shows a reduction in the number of patterns stored, making the SOM more efficient. The more horizontal dashed line is a linear regression for the rest of the time period; it suggests that the reduction will continue, but more slowly. Dotted lines are displayed one standard deviation above and below each linear regression. The time span shown includes 12 generations of wains.



Figure 17. The solid line shows the Edelman cycle for wains as a function of time. The graph shows a trend toward lengthening the Edelman cycle. The dashed line is the linear regression; the two dotted lines are one standard deviation above and below. The graph suggests a slight trend toward lengthening the Edelman cycle. The time span shown includes 12 generations of wains.

## 6.5.3 Edelman Cycle Gene

Recall from Section 3.2 that at intervals specified by the Edelman cycle gene, the least useful pattern (the one that has the fewest matches) is discarded, and the node is randomized so that it can learn a new pattern. In the initial population, the cycle was set to a random integer between 50 and 100,000.



Figure 18. Wain appearance mutations: (a) normal; (b) normal rearing a child; (c) double  $\times$  mutation; (d) short southeast mutation; (e) reflected south mutation; (f) Picasso mutation.



Figure 19. Inheritance of a mutation. The numbers are the wain IDs.

Figure 17 shows how the cycle has evolved over time. The cycle increased, then leveled off, but appears to be increasing again.

## 6.6 Mutations

Given the high mutation rate used in this trial ( $p_{mutation} = 0.001$ ), it was not surprising that mutations appeared in the second generation. One area where it is particularly easy to observe the effect of mutations is in the appearance of the wains. Figure 18 shows some representative examples. Figure 19 shows the family history of one particular mutation. The *missing-south* mutation caused the sequence of appearance genes to be truncated, so that the lower half of the  $\times$  is missing. If the offspring inherit appearance genes for the lower half from only one parent, those genes will be expressed. Therefore, the missing-south mutation is a recessive trait, although it involves a sequence of genes rather than a single gene.

Once mutants appeared, it was to be expected that wains would develop ways to recognize them. Figure 20 shows a SOM with two separate patterns for wains: one for normal wains, and one for mutants. As this wain is itself a mutant, it now has a way to identify wains that are likely to be related to it. In time, this could lead to kin selection.

# 6.7 Risk-Taking and Evolution

After reviewing the results, it seems that wains too often take actions that are predicted to have bad outcomes. This happens because every action receives a minimum weighting of one, which was implemented so that the wains would occasionally take risks. An action that had a bad outcome



Figure 20. A mutant-detecting SOM. This wain's SOM shows two separate patterns for wains. This wain is itself a mutant of the reflected-south variety.

under one set of circumstances may have a good outcome in a different situation. One potential area for improving the decider would be to make the amount of risk-taking genetic; this should evolve wains with a better balance between taking risks and playing it safe.

# 7 Conclusions and Future Directions

The wains discovered patterns in the data (as discussed in Sections 6.2, 6.3, 6.4), and they thrived (Section 6.1). Not only did individual wains learn to make better decisions during their lifetime (Section 6.2), but changes were made to the brain over several generations that improved the decision-making ability of the wains (Section 6.5.1). The design of the classifier component of the brain makes it easy to discover the patterns identified by the wains (Figures 10 and 11). When poisonous numerals became more toxic, wains adapted (Section 6.1), primarily by modifying their learning rates through evolution (Section 6.5.1). Evolution also made the brains of wains more efficient, by reducing the number of patterns that the SOM stored, without affecting the wain's ability to identify sufficient food to survive (Section 6.5.2). After only 12 generations, the wain population has exhibited complex, interesting behavior.

Some future directions for research are identified below.

# 7.1 Further Testing

The wains were not told what kinds of patterns to look for, which suggests that, given time to evolve, they would be able to find patterns in large, complex data sets, where the researcher may not know what patterns exist, or where to look for them. We plan to test wains with a realistic data mining problem, to discover if they are a useful tool for data mining, and to identify their strengths and weaknesses compared to other data mining tools. We also plan to determine if boredom and play are truly beneficial (i.e., if they make the wains better at exploring their environment).

## 7.2 Larger Brains

To have more general application, the wains will need larger brains. The exteroception capacity (length of the input vector) should be large enough to accommodate the maximum length of the records in the data to be analyzed.

## 7.3 Richer Interactions

Wains might behave more intelligently if they had a period between childhood and adulthood, an adolescence during which they observed the choices made by their parents and learned from the outcomes. In this way, by the time they were independent, they would be pre-equipped to make better choices. Similarly, it may be useful to allow wains to observe their peers, to learn from the choices they make and the results obtained. These forms of cultural transmission may allow the species to explore their environment more quickly, resulting in faster data analysis.

# 7.4 Richer Environment

While maintaining the philosophy that "the data is the environment," the ecosystem could be made more biologically realistic by allowing the wains to move freely within the environment. This would give wains a measure of control over which objects and other wains they interact with. The data could be used to add an element of geography; different parts of the environment would have a different mix of resources for the wains to exploit. This could support isolation of populations, and eventual speciation. Each species would perhaps have different strengths and weaknesses, which could be exploited for data mining.

## 7.5 More Realistic Ecology

Allowing wains and other artificial life species to coexist would make the ecology more complex and more biologically realistic, which could drive the wains to become more intelligent. Predator species and prey species might be introduced. Wains might be given a more complex biochemistry, requiring multiple types of nutrients available from diverse sources of food.

Some changes that might support the evolution of more intelligence include working with larger populations, imposing a maximum life span, opening up the genome to give evolution more control over wain design, and allowing the genetic code itself to evolve. Rates for mutation and crossover, which are currently fixed, could be allowed to evolve as well.

## Acknowledgments

This project was partly funded by Ericsson and the Software Research Institute (SRI). We would like to express our appreciation to Ericsson and the SRI staff.

#### References

- 1. Abraham, A., Grosan, C., & Ramos, V. (2006). Swarm intelligence in data mining. Berlin: Springer.
- Bernstein, H., Byerly, H. C., Hopf, F. A., & Michod, R. E. (1985). Genetic damage, mutation, and the evolution of sex. *Science*, 229(4719), 1277–1281.
- 3. Bonabeau, E., Dorigo, M., & Theraulaz, G. (1999). Swarm intelligence: From natural to artificial systems. Oxford, UK: Oxford University Press.
- Calabretta, R., Galbiati, R., Nolfi, S., & Parisi, D. (1996). Two is better than one: A diploid genotype for neural networks. *Neural Processing Letters*, 4(3), 149–255.
- 5. Cao, L. (2009). Data mining and multi-agent integration. Berlin: Springer-Verlag.
- 6. Dalton, C. (2012). Farm dogs. Available at http://www.teara.govt.nz/en/farm-dogs/3 (accessed March 2012).
- Dorigo, M., Maniezzo, V., & Colorni, A. (1991). Positive feedback as a search strategy (Technical Report No. 91-016). Milan: Politecnico di Milano.
- Gensler, H. L., & Bernstein, H. (1981). DNA damage as the primary cause of aging. The Quarterly Review of Biology, 56(3), 279-303.
- Giraldo, L. F., Lozano, F., & Quijano, N. (2011). Foraging theory for dimensionality reduction of clustered data. *Machine Learning*, 82(1), 71–90.
- 10. Gorunescu, F. (2011). Data mining concepts, models and techniques. Berlin: Springer.
- 11. Grand, S. (2001). Creation: Life and how to make it. London: Phoenix.
- Grand, S., & Cliff, D. (1997). Creatures: Entertainment software agents with artificial life. Autonomous Agents and Multi-Agent Systems, 1(1), 39-57.
- Griffith, V. (2007). YouTube–PolyWorld: Using evolution to design artificial intelligence. Available at http://www.youtube.com/watch?v=\_m97\_kL40x0 (accessed June 2012).
- 14. Hamilton, W. D., Axelrod, R., & Tanese, R. (1990). Sexual reproduction as an adaptation to resist parasites (a review). *Proceedings of the National Academy of Sciences of the U.S.A.*, 87, 3566–3573.
- 15. Han, J., & Kamber, M. (2011). Data mining: Concepts and techniques. Burlington, MA: Elsevier.
- 16. Hebb, D. O. (1949). The organization of behavior: A neuropsychological theory. New York: Wiley.
- Kohonen, T. (1982). Self-organized formation of topologically correct feature maps. *Biological Cybernetics*, 43(1), 59–69.
- 18. Kohonen, T. (2001). Self-organizing maps (3rd ed.). Berlin: Springer.
- 19. LeCun, Y., Bottou, L., Bengio, Y., & Haffner, P. (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86, 2278–2324.
- LeCun, Y., & Cortes, C. (2010). MNIST handwritten digit database. Available at http://yann.lecun.com/exdb/ mnist/ (accessed June 2012).

- 21. Lewin, B., Krebs, J., Goldstein, E., & Kilpatrick, S. (2009). Lewin's genes X. Sudbury, MA: Jones and Bartlett.
- Martens, D., Baesens, B., & Fawcett, T. (2011). Editorial survey: Swarm intelligence for data mining. Machine Learning, 82(1), 1–42.
- 23. Meirmans, S., & Strand, R. (2010). Why are there so many theories for sex, and what do we do with them? *Journal of Heredity*, 101(Suppl. 1), S3-S12.
- Michod, R. E., Bernstein, H., & Nedelcu, A. M. (2008). Adaptive value of sex in microbial pathogens. Infection, Genetics and Evolution, 8(3), 267–285.
- Muller, H. J. (1964). The relation of recombination to mutational advance. *Mutation Research/Fundamental* and Molecular Mechanisms of Mutagenesis, 1(1), 2–9.
- Neiman, M., & Koskella, B. (2009). Sex and the Red Queen. In I. Schön, K. Martens, & P. Dijk (Eds.), Lost sex (pp. 133–159). Berlin: Springer.
- Smith, R. E., & Goldberg, D. E. (1992). Diploidy and dominance in artificial genetic search. *Complex Systems*, 6(3), 251–285.
- Ultsch, A. (2004). Strategies for an artificial life system to cluster high dimensional data. In H. Schaub, F. Detje, & U. Brüggeman (Eds.), The Logic of Artificial Life: Abstracting and Synthesizing the Principles of Living Systems: Proceedings of the 6th German Workshop on Artificial Life, April 14–16, 2004, Bamberg, Germany. Berlin: AKA.
- Weiss, G. (2005). Data mining in telecommunications. In O. Z. Maimon & L. Rokach (Eds.), *Data mining and knowledge discovery handbook: A complete guide for practitioners and researchers* (pp. 1189–1201). New York: Springer Science & Business.
- 30. Witten, I. H., Frank, E., & Hall, M. A. (2011). Data mining: Practical machine learning tools and techniques. San Mateo, CA: Morgan Kaufmann.
- Yaeger, L. (1993). Computational genetics, physiology, metabolism, neural systems, learning, vision, and behavior or PolyWorld: Life in a new context. In C. G. Langton (Ed.), Artificial Life III, Proceedings of the Workshop on Artificial Life (pp. 263–298). Boulder, CO: Westview Press.
- 32. Yaeger, L., Griffith, V., & Sporns, O. (2008). Passive and driven trends in the evolution of complexity. In S. Bullock, J. Noble, R. Watson, & M. A. Bedau (Eds.), *Artificial Life XI: Proceedings of the Eleventh International Conference on the Simulation and Synthesis of Living Systems* (pp. 725–732). Cambridge, MA: MIT Press.
- 33. Yaeger, L. S. (2009). How evolution guides complexity. HFSP Journal, 3(5), 328.
- 34. Yaeger, L. S., & Sporns, O. (2008). Evolution of neural structure and complexity in a computational ecology. In L. Rocha, L. Yaeger, M. Bedau, D. Floreano, R. Goldstone, & A. Vespignani (Eds.), Artificial Life XI: Proceedings of the Eleventh International Conference on the Simulation and Synthesis of Living Systems (pp. 330–336). Cambridge, MA: MIT Press.