Efficient Hypomixability Elimination in Recombinative Evolutionary Systems

Keki M. Burjorjee
kburjorjee@pandora.com

Abstract

We submit that adaptation in recombinative evolutionary systems is powered by an implicit form of computation called Hypomixability Elimination. We describe hypomixability elimination, and provide evidence that it can be performed efficiently by recombinative evolutionary systems. Specifically we show that hypomixability elimination in a simple genetic algorithm can be used to obtain optimal bounds on the time and queries required to solve a subclass ($k = 7, \eta = 1/5$) of a familiar computational learning problem: PAC-learning parities with noisy membership queries; where $k$ is the number of relevant attributes and $\eta$ is the oracle’s noise rate. We show that a simple genetic algorithm that treats the noisy membership query oracle as a fitness function can be rigged to PAC-learn the relevant variables in $O(\log(n/\delta))$ queries and $O(n \log(n/\delta))$ time, where $n$ is the total number of attributes and $\delta$ is the probability of error. To the best of our knowledge, this is the first time optimally efficient non-trivial computation has been shown to occur in an evolutionary algorithm.

The optimality result and indeed the implicit implementation of hypomixability elimination by a simple genetic algorithm depends crucially on recombination. This dependence yields a fresh, unified explanation for sex, adaptation, speciation, and the emergence of modularity in recombinative evolutionary systems. Compared to other explanations, Hypomixability Theory is exceedingly parsimonious. For example, it does not assume deleterious mutation, a changing fitness landscape, or the existence of building blocks.

1 Introduction

In recent years, theoretical computer scientists have become increasingly perturbed by the problem posed by evolution. The subject of consternation is a system thought to have computed the encoding of every biological form to have lived, that, as luck would have it, represents information the way a Turing Machine might—digitally; in strings drawn from a quaternary alphabet—and manipulates information in ways that are well understood (e.g. meiosis) or amenable to abstraction (e.g. natural selection). Contemplating what this computational system has achieved given the resources at its disposal leaves one awestruck. Yet, theoretical computer science, for all its success in other areas, has not identified anything particularly efficient or arresting about evolution. We have on our hands

---


1 For the purposes of this paper, an efficient algorithm is one whose upper bounds (on, say, time and queries) match or approach the best known lower bounds for the problem at hand. For example, comparison sorting $n$ numbers in $O(n \log n)$ time is clearly efficient. An algorithm that solves the problem in $O(n \log^2 n)$ time is relatively efficient, whereas one that does it in $O(n^2)$ time is considered inefficient. Given this “definition”, the line between efficient and inefficient computation is of course fuzzy. For our purposes, however, fuzzy lines of this kind are more revealing than the objective one that, regardless of the problem at hand, deems an algorithm efficient if and only if its complexity is polynomially bounded.

2 The Computational Theories of Evolution Workshop held at the Simons Institute, Berkeley, CA from March 17 – March 21, 2014 brought together researchers from multiple disciplines to discuss the issue. Presentations available at http://simons.berkeley.edu/workshops/abstracts/326
what might be called a computational origins problem—while our physical origins have been worked out, our computational origins remain a mystery. Referring to this problem, Valiant speculates that future generations will wonder why it was not regarded with a greater sense of urgency [32].

If the computational origins problem is cause for reflection within Theoretical Computer Science, it is doubly so for Evolutionary Computation theorists. The promise of Evolutionary Computation was twofold: 1) That the computational efficiencies at play in natural evolution would be identified and 2) That these efficiencies would be harnessed, via biomimicry, and used to efficiently procure solutions to human problems. One might expect these outcomes to be realized in order or simultaneously. In a curious twist, however, the field seems to be making good on the second part of its promise, but has not delivered on the first. This twist makes evolutionary algorithms interesting from a theoretical standpoint. Find something efficient about one of the more bioplausible ones, and a piece of the computational origins puzzle may fall into place.

A crucial piece of the puzzle, and the subject of an ongoing debate amongst evolutionary biologists, is the part played by sex (i.e. mixing). Sex seems downright contradictory given the prevalence of widespread epistasis in biological genomes. If interaction between genes (epistasis) is the norm, not the exception [30], what sense does it make to break up groups of alleles that collectively confer an advantage [2]? Compared to previous answers, the one we provide—Hypomixability Elimination—is very parsimonious, relying only on the following two assumptions:

**Hypomixability**: There exists a small collection of loci, not necessarily adjacent to each other, whose alleles in the population of parents confer a disadvantage on average when mixed.

**Generative Elimination**: The adaptive elimination of hypomixability in some collection of loci with respect to the parent population engenders hypomixability elsewhere.

The second assumption is necessary to ensure that the advantage conferred by sex is ongoing, i.e. is not exhausted by the one-time elimination of hypomixability. We emphasize that our theory does not rely on any of the following:

1. The existence of so called building blocks—single genes or tightly linked sets of genes that confer an advantage regardless of the context in which they occur [14, 11, 33].
2. A changing fitness landscape, whether due to parasitism or changes in the external environment [13, 20].
3. Deleterious mutation [26, 19].

The following section introduces the idea of hypomixability elimination. In subsequent sections we demonstrate that the implicit implementation of hypomixability elimination by sexual evolution can be computationally efficient by analyzing a simple genetic algorithm with uniform crossover and deriving optimal bounds on the time and queries required to PAC-learn a subclass ($k = 7, \eta = 1/5$) of a non-trivial computational problem: learning parities with noisy membership queries [31, 10], where $k$ is the number of relevant attributes, and $\eta$ is the probability that the membership query (MQ) oracle misclassifies a query (i.e. returns a 1 instead of a 0, or vice versa). To the best of our knowledge, this is the first time efficient, not to mention optimally efficient, non-trivial computation is shown to occur in a evolutionary algorithm. The result hinges on a straightforward symmetry argument and an empirically reached conclusion with a $p$-value less than $10^{-900}$.

2 Hypomixability Theory

We begin with a primer on schemata and schema partitions$^3$ [11, 24]. For any positive integer $k$, let $[k]$ denote the set $\{1, \ldots, k\}$. For any positive integer $n$, a schema (plural, schemata) of the set

$^3$We rely only on the mathematical tools, not the conclusions, of Schema Theory [14].
Figure 1: A tabular depiction of the spartition $[\{1, 2, 4\}]_5$ of order three. The table headings give the templates of the schemata comprising the partition. The elements of each schema in the partition appear in the column below its template.

<table>
<thead>
<tr>
<th>00•0*</th>
<th>00•1*</th>
<th>01•0*</th>
<th>01•1*</th>
<th>10•0*</th>
<th>10•1*</th>
<th>11•0*</th>
<th>11•1*</th>
</tr>
</thead>
<tbody>
<tr>
<td>00000</td>
<td>00010</td>
<td>01000</td>
<td>01010</td>
<td>01000</td>
<td>10010</td>
<td>11000</td>
<td>11010</td>
</tr>
<tr>
<td>00001</td>
<td>00011</td>
<td>01001</td>
<td>01011</td>
<td>01001</td>
<td>10011</td>
<td>11001</td>
<td>11011</td>
</tr>
<tr>
<td>00100</td>
<td>00110</td>
<td>01100</td>
<td>01110</td>
<td>01100</td>
<td>10110</td>
<td>11100</td>
<td>11110</td>
</tr>
<tr>
<td>00101</td>
<td>00111</td>
<td>01111</td>
<td>01101</td>
<td>01101</td>
<td>11011</td>
<td>11110</td>
<td>11111</td>
</tr>
</tbody>
</table>

of genotypes $\{0,1\}^n$ is a subset of $\{0,1\}^n$ that is traditionally represented by a schema template. A schema template is a string in $\{0,1,\ast\}^n$. As will become clear from the formal definition below, the symbol $\ast$ stands for ‘wildcard’ at the positions at which it occurs. Given a schema template $X \in \{0,1,\ast\}^n$, let $[X]$ denote the schema represented by $X$. Then,

$$ [X] = \{ x \in \{0,1\}^n \mid \forall i \in [n], X_i \neq x_i \Rightarrow X_i = \ast \} $$

Let $I \subseteq \{1, \ldots, n\}$ be some set of integers, and let $\langle I \rangle_n$ denote the set of strings $\{X \in \{0,1,\ast\}^n \mid X_i \neq \ast \iff i \in I\}$. Then $I$ represents a partition of $\{0,1\}^n$ into a set of $2^{|I|}$ schemata, denoted $[I]_n$, whose templates are given by $\langle I \rangle_n$. The elements of $I$ are said to be the defining loci of the schema partition. We shorten the mouthful schema partition to spartition.

The order of some spartition $[I]_n$ is simply cardinality of $I$. It is easily seen that spartitions of lower order are coarser than spartitions of higher order; more specifically, that a spartition of order $k$ is comprised of $2^k$ schemata.

**Example 1.** The index set $\{1, 2, 4\}$ induces an order three spartition of $\{0,1\}^5$ into eight schemata as shown in Figure 1.

For any distribution $D$ over $\{0,1\}^n$ and any spartition $[I]_n$, the projection of $D$ onto $\langle I \rangle_n$, denoted $\Xi_I[D]$, is a distribution over $\langle I \rangle_n$ defined as follows: For any $X \in \langle I \rangle_n$,

$$ \Xi_I[D](X) = \sum_{x \in [X]} D(x) $$

For any product set $A = A_1 \times \ldots \times A_n$ and any distribution $D$ over $A$, let $D_i$ denote the $i^{th}$ univariate marginal distribution (over $A_i$) of $D$, and let $\Pi[D]$ denote the product of marginals distribution of $D$. That is, for all $x \in A$,

$$ \Pi[D](x) = \prod_{i \in [n]} D_i(x_i) $$

The following lemma is trivially obtained:

**Lemma 2.** Let $A = A_1 \times \ldots \times A_n$ be a product set, and let $D$ be some distribution over $A$. Then for any $x \in A$,

$$ D(x) > 0 \Rightarrow \Pi[D](x) > 0 $$

For any distribution $D$ over some product set we say that $D$ is in equilibrium if $\Pi[D] = D$. If $D$ is not in equilibrium, it is said to be in disequilibrium. For any boolean expression $b$, let $[b]$ denote 1 if $b$ is true and 0 otherwise. Unless otherwise specified, the range of a fitness function is $\mathbb{R}_{\geq 0}$. 

3
Another difference is that we focus on mixability with respect to parent distributions, not general populations.

particular

so on. By defining mixability with respect to a particular set of loci defined by Livnat et al. have no equivalent in our formulation. These measures are defined, not with respect to some marginal distribution over a set of loci entails non-unit mixability of the loci.

\[
\text{Definition 3 (Partial Override Distribution). Let } \mathcal{D} \text{ be a distribution over } \{0,1\}^n, \text{ let } \mathcal{I} \subseteq [n] \text{ be some index set, let } \mathcal{I}' = [n] \setminus \mathcal{I}, \text{ and let } \mathcal{D}' \text{ be some distribution over } (\mathcal{I})_n. \text{ The partial override distribution } \Lambda[\mathcal{I}, \mathcal{D}; \mathcal{D}'] \text{ is a distribution over } \{0,1\}^n \text{ defined as follows: for any } x \in \{0,1\}^n,
\]

\[
\Lambda[\mathcal{I}, \mathcal{D}; \mathcal{D}'](x) = \left( \sum_{X \in \langle \mathcal{I} \rangle_n} [x \in X] \mathcal{D}(X) \right) \left( \sum_{X \in \langle \mathcal{I}' \rangle_n} [x \in X] \cdot \Xi_{\mathcal{I}'}[\mathcal{D}](X) \right)
\]

2.1 Mixability

We assume that mixing occurs via uniform crossover (i.e. free recombination), in other words, that loci are unlinked.

\[
\text{Definition 4 (Mixed fitness of a schema). Let } \mathcal{D} \text{ and } \phi \text{ be some distribution and fitness function over } \{0,1\}^n \text{ respectively. For any } \mathcal{I} \subseteq [n], \text{ let } X \in (\mathcal{I})_n \text{ be some schema template, and let } X \in X_n \text{ be a random variable defined as follows: for any } i \in [n], \text{ if } i \in \mathcal{I}, \text{ then } X_i = X_i', \text{ otherwise } X_i \sim \mathcal{D}_i. \text{ Then the mixed fitness of } X \text{ with respect to } \mathcal{D} \text{ and } \phi, \text{ denoted } \Phi(X; \mathcal{D}, \phi), \text{ is defined as follows:}
\]

\[
\Phi(X; \mathcal{D}, \phi) = E[\phi(X)]
\]

Observe that unless } \mathcal{I} = [n], \text{ knowledge of the full joint distribution } \mathcal{D} \text{ is not required to calculate } \Phi(X; \mathcal{D}, \phi). \text{ Knowing } \mathcal{D}_i \text{ for all } i \in [n] \setminus \mathcal{I} \text{ suffices. This observation yields the following lemma.}

\[
\text{Lemma 5. For any fitness function } \phi \text{ and distribution } \mathcal{D} \text{ over } \{0,1\}^n, \text{ let } \mathcal{I} \subseteq [n]. \text{ Then for any } X \in (\mathcal{I})_n,
\]

\[
\Phi(X; \mathcal{D}, \phi) = \Phi(X; \Pi[\mathcal{D}], \phi)
\]

The next definition introduces the notion of mixability. Briefly, the mixability of some set of loci } \mathcal{I} \text{ gives the change in expected fitness when mixing occurs amongst the loci in } \mathcal{I}. \text{ This quantity is calculated with respect to full mixing of the loci not in } \mathcal{I}.

\[
\text{Definition 6 (Mixability). For any fitness function } \phi \text{ and distribution } \mathcal{D} \text{ over } \{0,1\}^n, \text{ and any } \mathcal{I} \subseteq [n], \text{ let } X \in (\mathcal{I})_n \text{ be a random variable drawn from the distribution } \Xi_{\mathcal{I}}[\mathcal{D}]. \text{ The mixability of } \mathcal{I} \text{ with respect to } \mathcal{D} \text{ and } \phi, \text{ denoted } M(\mathcal{I}; \mathcal{D}, \phi), \text{ is defined as follows:}
\]

\[
M(\mathcal{I}; \mathcal{D}, \phi) = \frac{\Phi(\propto; \mathcal{D}, \phi)}{E[\Phi(X; \mathcal{D}, \phi)]}
\]

Mixability less than, greater than, and equal to 1 is called hypomixability, hypermixability, and unit mixability respectively.\(^4\)

\[
\text{Definition 7 (Mixed Fitness Variance). Let } \phi \text{ be a fitness function and } \mathcal{D} \text{ a distribution over } \{0,1\}^n. \text{ For any } \mathcal{I} \subseteq [n], \text{ let } X \in (\mathcal{I})_n \text{ be a random variable drawn from the distribution } (\Pi \circ \Xi_{\mathcal{I}})[\mathcal{D}]. \text{ The mixed fitness variance of } \mathcal{I} \text{ with respect to } \mathcal{D} \text{ and } \phi, \text{ denoted } V(\mathcal{I}; \mathcal{D}, \phi), \text{ is defined as follows:}
\]

\[
V(\mathcal{I}; \mathcal{D}, \phi) = \text{Var}[\Phi(X; \mathcal{D}, \phi)]
\]

Part 1 of the following lemma shows that there is a straightforward relationship between the mixed fitness variance and mixability of a set of loci. Part 2 shows that disequilibrium in the multivariate marginal distribution over a set of loci entails non-unit mixability of the loci.

\(^4\)Comparing the definitions above to the ones provided by Livnat et al. [21], who also use the term mixability, one can spot similarities and differences. Beginning with the similar definitions, } \Phi(\propto; \mathcal{D}) \text{ as defined here is similar (if not identical) to the mixability measure } M_1 \text{ defined by Livnat et al. [21]. On the other hand, the mixability measures } M_2, \ldots, M_n \text{ defined by Livnat et al. have no equivalent in our formulation. These measures are defined, not with respect to some particular set of loci } \mathcal{I} \text{ as in our formulation, but with respect to all pairs of loci, all triples, all sets of four loci, and so on. By defining mixability with respect to a particular set of loci, our formulation gains crucial expressive power. Another difference is that we focus on mixability with respect to parent distributions, not general populations.
Lemma 8. Let $\phi : \{0,1\}^n \to \mathbb{R}_0^+$ be some fitness function, let $\mathcal{D}$ be some distribution over $\{0,1\}^n$, let $\mathcal{I} \subseteq [n]$ be some index set and let $\mathcal{D} = \Xi_{\mathcal{I}}[\mathcal{D}]$. Then $\mathcal{I}$ has unit mixability with respect to $\mathcal{D}$ if the mixed fitness variance of $[\mathcal{I}]_n$ is zero or the loci in $\mathcal{I}$ are in equilibrium. In other words,

1. $V(\mathcal{I}; \mathcal{D}, \phi) = 0 \Rightarrow M(\mathcal{I}; \mathcal{D}, \phi) = 1$
2. $\Pi[\mathcal{D}] = \mathcal{D} \Rightarrow M(\mathcal{I}; \mathcal{D}, \phi) = 1$

Proof. For a proof of part 1, observe that $V(\mathcal{I}; \mathcal{D}, \phi) = 0$ entails that there exists $c \in \mathbb{R}_0^+$ such that

$$\forall \mathcal{X} \in \langle \mathcal{I} \rangle_n, (\Pi \circ \Xi_{\mathcal{I}})[\mathcal{D}](\mathcal{X}) > 0 \Rightarrow \Phi(\mathcal{X}; \mathcal{D}, \phi) = c$$

(1)

In other words, there exists a $c \in \mathbb{R}_0^+$ such that $\Phi(\mathcal{X}; \mathcal{D}, \phi) = c$ for every schema $\mathcal{X} \in \langle \mathcal{I} \rangle_n$ that has non-zero probability mass under $\Pi \circ \Xi_{\mathcal{I}}$. This gives us that $\Phi(\mathcal{X}; \mathcal{D}, \phi) = c$.

The application of Lemma 2 to expression (1) yields

$$\forall \mathcal{X} \in \langle \mathcal{I} \rangle_n, \Xi_{\mathcal{I}}[\mathcal{D}](\mathcal{X}) > 0 \Rightarrow \Phi(\mathcal{X}; \mathcal{D}, \phi) = c$$

which gives us that for any random variable $\mathcal{X}$ drawn from the distribution $\Xi_{\mathcal{I}}[\mathcal{D}]$, $E[\Phi(\mathcal{X}; \mathcal{D}, \phi)] = c$.

For a proof of part 2, observe that

$$\Phi(\mathcal{X}; \mathcal{D}, \phi) = \sum_{\mathcal{X} \in \langle \mathcal{I} \rangle_n} \Phi(\mathcal{X}; \mathcal{D}, \phi). (\Pi \circ \Xi_{\mathcal{I}})[\mathcal{D}](\mathcal{X})$$

The premise of part 2 entails that $(\Pi \circ \Xi_{\mathcal{I}})[\mathcal{D}] = \Xi_{\mathcal{I}}[\mathcal{D}]$, which gives us that $\Phi(\mathcal{X}; \mathcal{D}, \phi) = E[\Phi(\mathcal{X}; \mathcal{D}, \phi)]$, where $\mathcal{X}$ is a random variable drawn from the distribution $\Xi_{\mathcal{I}}[\mathcal{D}]$. □

In evolutionary systems, it is parents, not individuals in the the general population, that get mixed. Thus, when we speak of the mixability of some set of loci in some generation it is always with respect to the parent distribution. To be clear, the parent distribution in some generation is simply the distribution from which parents are sampled:

Example 9. For any fitness function $\phi$ and distribution $\mathcal{D}$ over $\{0,1\}^n$ such that $\mathcal{D}$ represents a general population with at least one individual with positive fitness. If selection is fitness proportional, then the parent distribution of $\mathcal{D}$, denoted $S_\phi[\mathcal{D}]$, is as defined follows: For any $x \in \{0,1\}^n$,

$$S_\phi[\mathcal{D}](x) = \frac{\phi(x) \mathcal{D}(x)}{\sum_{y \in \{0,1\}^n} \phi(y) \mathcal{D}(y)}$$

As the following example shows, mixability with respect to the general population can be very different from mixability with respect to the parent population.

Example 10. Let $\mathcal{I} = \{1,2\}$ and let $\phi : \{0,1\}^3 \to \{0,1\}$ be a fitness function that returns the exclusive-or (XOR) of the first two of its inputs. Let $\mathcal{D}$ be the uniform distribution over $\{0,1\}^3$. $\mathcal{D}$ is in equilibrium; so by part 2 of Lemma 8, $M(\mathcal{I}; \mathcal{D}, \phi) = 1$. On the other hand, $\Phi(\mathcal{X}; S_\phi[\mathcal{D}], \phi) = 1/2$ and $E[\Phi(\mathcal{X}; S_\phi[\mathcal{D}], \phi)] = 1$, so $M(\mathcal{I}; S_\phi[\mathcal{D}], \phi) = 1/2$.

Non-unit mixability in parent populations is of particular interest. Briefly, non-unit mixability in some set of loci $\mathcal{I}$ entails that there exists an “adjustment” to the marginal distributions over the loci in $\mathcal{I}$ of the parent population that would eliminate the non-unit mixability $\mathcal{I}$ while raising the expected fitness of the offspring population. A formal statement and proof of the above is as follows:

Theorem 11 (Non-unit mixability entails opportunity for adaptation). For any distribution $\mathcal{D}$ and fitness function $\phi$ over $\{0,1\}^n$, and any $\mathcal{I} \subseteq [n]$, if $M(\mathcal{I}; \mathcal{D}, \phi) \neq 1$, then there exists a distribution $\mathcal{D}^*$ over $\{0,1\}^n$ such that $\mathcal{D}^*$ is in equilibrium, $M(\mathcal{I}; \Lambda[\mathcal{I}, \mathcal{D}^*; \mathcal{D}], \phi) = 1$, and for any independent random variables $X \sim \Pi[\mathcal{D}]$, and $Y \sim (\Pi \circ \Lambda)[\mathcal{I}, \mathcal{D}^*; \mathcal{D}]$, we have that $E[\phi(Y)] > E[\phi(X)]$.
Proof. The proof follows straightforwardly from Lemma 8. Part 1 of this lemma entails that
\[ M(I; D, \phi) \neq 1 \Rightarrow V(I; D, \phi) \neq 0 \]
This result and the premise of non-unit mixability entails that there exists \( X^{max}, X^{min} \in \langle I \rangle_n \) such that
\[
(\Pi \circ \Xi_I)[D](X^{min}) > 0 \tag{2}
\]
\[
\Phi(X^{max}; D, \phi) > \Phi(X^{min}; D, \phi) \tag{3}
\]
\[
\forall X \in \langle I \rangle_n, \Phi(X^{max}; D, \phi) \geq \Phi(X; D, \phi) \tag{4}
\]
Equations (2-4) in turn entail that
\[
\Phi(X^{max}; D, \phi) > \sum_{X \in \langle I \rangle_n} \Phi(X; D, \phi) \cdot (\Pi \circ \Xi_I)[D](X) \tag{5}
\]
Note that the right hand side of equation (5) is \( E[\phi(X)] \). For any \( X \in \langle I \rangle_n \), let \( D^* \) be defined as follows:
\[
D^*(X) = [X = X^{max}]
\]
As all the probability mass of \( D^* \) is concentrated on just one schema template in \( \langle I \rangle_n \), \( D^* \) is in equilibrium, which gives us that \((\Pi \circ \Lambda)[I, D^*; D] = \Lambda[I, D^*; \Pi[D]]\), which in turn entails that the left hand side of (5) is \( E[\phi(Y)] \). That \( M(I; \Lambda[I, D^*; D], \phi) = 1 \) follows from the Conclusion 2 of Lemma 8 and the fact that \( D^* \) is in equilibrium. \( \square \)

Theorem 11 shows that if the marginal distributions of loci in \( I \) is all one has control over it is always possible to eliminate non-unit mixability amongst the loci in \( I \) in a way that raises the expected fitness of the offspring population. There is, however, a sense in which theorem 11 is not “tight”. Let \( H \) denote the entropy operator, and note that \( H((\Pi \circ \Lambda)[I, D^*; D]) = H(D^*) + \sum_{i \in I} H(D_i) \). Choosing \( D^* \) such that the entropy \( H(D^*) = 0 \), as we did in the proof of Theorem 11, minimizes \( H((\Pi \circ \Lambda)[I, D^*; D]) \). Is it possible choose \( D^* \) with less of a toll on the entropy? The answer depends on \( \phi \). For example, if \( \phi \) is the boolean function XOR over only the loci in \( I \) (an XOR \(|I|\)-junta), the answer seems to be no. If \( \phi \) is the boolean function OR over only the loci in \( I \) (an OR \(|I|\)-junta), the answer is yes. One could, for example, pick a distribution \( D \) over \( \langle I \rangle_n \) such the \( D_i(x) = [x = 1] \) for some \( i \in I \), and \( D_j \) is uniformly distributed for all \( j \in I \), such that \( i \neq j \).

2.2 Sampling Mixability

We turn now to the practical feasibility of identifying a set of loci with non-unit mixability. Considering the size of typical search spaces, the precise mixability of a partition \([I]_n\) with respect to a parent distribution \( D \) and fitness function \( \phi \) is, of course, unknowable. However, given some number of parents sampled from the parent distribution, the mixability can be estimated. For any \( p \in [0, 1] \), let \( B(p) \) denote the Bernoulli distribution with parameter \( p \). A simple estimator for the mixability of a set of loci is given below.

**Example 12 (Simple Sampling Mixability).** Let \( X^1, \ldots, X^r \in \{0, 1\}^n \) be the set of parents in some generation \( t \). For any \( I \in [n] \), let \( Y^1, \ldots, Y^r \in \{0, 1\}^n \) and \( Z^1, \ldots, Z^r \in \{0, 1\}^n \) be conditionally independent random variables defined as follows: for all \( i \in [n] \), and all \( j \in [r] \), \( Z^i_j = X^i_j \) if \( i \in I \), otherwise \( Z^i_j \sim B(\frac{1}{r} \sum_{k \in [r]} X^k_j) \), and for all \( i \in [n] \) and all \( j \in [r] \), \( Y^i_j \sim B(\frac{1}{r} \sum_{k \in [r]} Y^k_j) \), then the canonical sampling mixability of \( I \) in generation \( t \) with respect to some fitness function \( \phi \), denoted \( \bar{M}_s(I) \), is an estimator defined as follows:
\[
\bar{M}_s(I) = \frac{\hat{a}}{b(I)}
\]
where \( \hat{\alpha} = \sum_{j \in [r]} \phi(Y^j) \) is an estimator for the numerator in Definition 6, and \( \hat{b}(I) = \sum_{j \in [r]} \phi(Z^j) \) is an estimator for the denominator.

### 2.2.1 Spartial Coarseness and Statistical Significance

Like all estimators, the simple sampling mixability estimator \( \hat{M}_n(I) \) is just a random variable, and, as such, has an expected value, \( E[\hat{M}_n(I)] \), and a variance \( \text{Var}[\hat{M}_n(I)] \). Let us informally examine how \( \text{Var}[\hat{M}_n(I)] \) changes as we remove elements from \( I \), i.e. as we coarsen the spartition \( [\mathcal{I}]_n \). As elements are removed, the variance of \( \hat{\alpha} \) stays the same, whereas it is reasonable to conjecture that the variance of \( \hat{b}(I) \) decreases because one is putting “more wood behind fewer arrows”. Thus, it is reasonable to conjecture that for any \( I' \) such that \( I' \subset I \) we have that \( \text{Var}[\hat{M}_s(I')] \leq \text{Var}[\hat{M}_s(I)] \).

Indeed, it is reasonable to expect any “well behaved” estimator of mixability to have this property. Coarsening schema partitions, in other words, seems to shorten confidence intervals around mixability estimates.

### 2.3 The Unit Mixability Principle

The **Principle of Unit Mixability** states that under free recombination (i.e. uniform crossover), evolution efficiently and adaptively eliminates statistically significant non-unit mixability in any small set of loci \( I \) with respect to a parent population. Non-unit mixability is eliminated by sending the multivariate marginal distribution over \( I \) of future parent populations to equilibrium.

We submit that big, positive deviations from unit mixability with respect to parent distributions are rare in the real world. One needs to begin with a rather pathological distribution \( D \) to obtain a parent distribution \( (S_\delta[D] \) in the event that selection is fitness proportional) such that there exists some set of loci with sizable hypomixability. Thus, for all practical purposes, the Principle of Unit Mixability predicts the efficient, adaptive, equilibrium-seeking elimination of statistically significant hypomixability if it arises in some coarse spartition with respect to some parent distribution.

### 2.4 Parallelism in Evolutionary Systems

Observe that a naive (i.e. non-evolutionary) enforcement of the unit mixability principle by some algorithm that scans for coarse schema partitions with non-unit mixability scales poorly with \( n \). There are \( \left( \begin{array}{c} n \\ k \\ \end{array} \right) \) spartitions of order \( k \). When \( n = 10^6 \), say, the number of spartitions of order \( k = 1 \ldots 7 \) are on the order of \( 10^6, 10^{14}, 10^{17}, 10^{22}, 10^{27}, 10^{33}, \) and \( 10^{38} \) respectively. Is evolution capable of performing hypomixability elimination as described above within timeframes that are not equally astronomical?

Partial support for an affirmative answer appears in a previous paper [8], where it is shown that for any spartition \( [\mathcal{I}]_n \), infinite population evolution over \( \{0,1\}^n \) using fitness proportional selection, homologous recombination, and standard bit mutation implicitly induces infinite population evolution over the set of schemata in \( [\mathcal{I}]_n \).

More formally, let the operators \( S \) and \( \Xi \) be as defined in the previous section. For any transmission function \( T \) and distribution \( D \) over \( \{0,1\}^n \), let \( \mathcal{V}_T[D] \) be a distribution over \( \{0,1\}^n \) defined as follows:

For any \( z \in \{0,1\} \),

\[
\mathcal{V}_T[D](z) = \sum_{x \in \{0,1\}^n} \sum_{y \in \{0,1\}^n} T(z \mid x,y) D(x) D(y)
\]

Then, for any spartition \( [\mathcal{I}]_n \), distribution \( P \) over \( \{0,1\}^n \), fitness function \( \phi : \{0,1\}^n \to \mathbb{R}^+ \), and transmission function \( T \) over \( \{0,1\}^n \) that models homologous recombination followed by canonical bit mutation, there exist probability distributions \( Q \) and \( R \) over \( \{0,1\}^n \), probability distributions \( \mathcal{P}', \mathcal{Q}', \mathcal{R}' \) over \( \{\mathcal{I}\}_n \), fitness function \( \phi' : \mathcal{I}_n \to \mathbb{R}^+ \), and transmission function \( T' \) over \( \mathcal{I}_n \) such
that the following diagram commutes5:

\[
\begin{array}{ccc}
P \xrightarrow{S_{\phi}} Q \xrightarrow{V_T} R \\
\varepsilon_{T} ~ & | & ~ \varepsilon_{T} \\
P' \xrightarrow{S_{\phi'}} Q' \xrightarrow{V_{T'}} R'
\end{array}
\]

The proof of the above is constructive. That is, \(\phi'\) and \(T'\) are precisely specified. Crucially, there is no restriction on the index set \(I\). In other words, the above holds true simultaneously for all spartitions of \(\{0,1\}^n\).

2.5 A Need for Science, Practiced With Rigor

Formal proof for the occurrence of efficient hypomixability elimination beyond that referenced in the previous section—specifically formal proof pertaining to evolutionary algorithms with finite populations—is difficult to provide, not least because the analysis of evolutionary algorithms with finite populations is notoriously unwieldy. We have argued previously that resorting to the scientific method [29] is a necessary and appropriate response to this hurdle [6].

2.5.1 Parsimony and Unification

For a science to be viable, it must be parsimonious (i.e. based on weak assumptions). Hypomixability theory assumes the existence of one or more coarse schema partitions that are statistically significantly hypomixable with respect to the parent population. It assumes, further, that the adaptive elimination of such hypomixability engenders statistically significant hypomixability in other coarse schema partitions (the need for this assumption is explained in Section 8). These assumptions are exceedingly weak, especially when compared to the strong assumptions that have previously been made in Evolutionary Computation and Evolutionary Biology [5, Chapter 2]. Curiously, hypomixability theory does more with less. It is not necessary, for example, to have a theory of sex [20] that is separate from a theory of adaptation, which in turn is separate from a theory of speciation, each theory with its attendant (possibly conflicting) assumptions. All three phenomena as well as the emergence of modularity [21, 23] can be explained by one, unified, parsimonious theory.

2.5.2 Testable Predictions

A hallmark of rigorous science is the ongoing making and testing of predictions. Predictions found to be true lend credence to the hypotheses that entail them. The more unexpected a prediction, the greater the credence owed the hypothesis if the prediction is borne out [29, 28].

The work that follows validates a prediction entailed by hypomixability theory. As we explain in the next section, hypomixability theory predicts that a genetic algorithm with uniform crossover (UGA) can be used to construct an algorithm that efficiently solves the problem of learning parities with a noisy MQ oracle for small but non-trivial values of \(k\), the number of relevant attributes, and \(\eta \in (0,1/2)\), the probability that the oracle makes a classification error (returns a 1 instead of a 0, or vice versa) on a given query. Such a result is unexpected because of the absence of modules/building blocks under a uniform distribution over the search space. To the best of our knowledge, the only theory in a position to predict a result similar to the one we obtain is the theory of sex proposed by Livnat et al. [21]. We have more to say about this theory in Section 9.

---

5This diagram was mistakenly called a coarse-graining by Burjorjee and Pollack [8]. The mistake was corrected in a later paper [4], where the idea of coarse-graining was explained in detail. As \(\phi'\) in the commutative diagram is not invariant to \(T\), no coarse-graining of evolutionary dynamics was previously shown. What was actually shown is arguably more interesting—implicit parallel computation.
3 PAC-Learning Parities with Noisy Membership Queries

The problem of learning parities is a refinement of the learning juntas problem [25], so we approach the former problem by way of the latter. For any boolean function \( f \) over \( n \) variables, a variable \( i \) is said to be relevant if there exist binary strings \( x, y \in \{0, 1\}^n \) that differ only in their \( i \)th coordinate such that \( f(x) \neq f(y) \). Variables that are not relevant are said to be irrelevant. For any non-negative integer \( n \) and any integer \( k \leq n \), a \( k \)-junta is a function \( f : \{0, 1\}^n \to \{0, 1\} \) such that for any integer \( j \leq k \) only \( j \) of the \( n \) inputs to \( f \) are relevant. These \( j \) relevant variables are said to be the juntas of \( f \). The function \( f \) is completely specified by its juntas (characterizable by the set \( J \subseteq [n] \) of junta indices) and by a hidden boolean function \( h \) over the \( j \) juntas. The output of \( f \) is just the output of \( h \) on the values of the \( j \) relevant variables (the values of the irrelevant variables are ignored). The problem of identifying the relevant variables of any \( k \)-junta \( f \) and the truth table of its hidden boolean function is called the learning \( k \)-juntas problem.

The learning \( k \)-parities problem is a refinement of the learning \( k \)-juntas problem where it is additionally specified that \( j = k \), and the hidden function \( h \) is the parity (i.e. xor) function over \( k \) inputs. In this case, the function \( f \) is completely specified by its juntas.

An algorithm \( A \) is said to solve the learning \( k \)-parities problem if for any \( k \)-parity function \( f : \{0, 1\}^n \to \{0, 1\} \) whose juntas are given by the set \( J \subseteq [n] \) and any \( \delta \in (0, 1/2) \), \( A(n, k, \delta) \) outputs a set \( S \subseteq [n] \) such that \( \Pr(S \neq J) \leq \delta \).

A noisy MQ oracle \( \phi \) behaves as follows. For any string \( x \in \{0, 1\}^n \), \( \phi \) returns \( \neg f(x) \) with probability \( \eta \in (0, 1/2) \) and \( f(x) \) with probability \( 1 - \eta \). The parameter \( \eta \) is called the noise rate. Bounds derived for the time and query complexity of \( A^\phi \) with respect to \( n \) and \( \delta \) speak to the efficiency with which \( A^\phi \) solves the problem. The algorithmic learning described here is Probably Approximately Correct learning [18] with the inaccuracy tolerance \( \epsilon \) set to zero: Probably Correct (PC) learning, if you will.

3.1 The Noise Model

Blum and Langley [3] give a simple binary search based method that learns a single relevant variable of any \( k \)-junta in \( O(n \log n) \) time and \( O(\log n) \) queries with noise free membership queries. As explained in Section 3.1 of a paper by Mossel and O’Donnell [25], once a single relevant variable is found, the method can be recursively repeated at most \( k2^k \) times to find the remaining relevant variables. In an MQ setting, the introduction of noise does not complicate the situation much if the corruption of a given answer is independent of the corruption of previous answers. In this case, noise can be eliminated to an arbitrary level simply by sampling the MQ oracle \( p(\frac{1}{1-2\eta}) \) times, where \( p(\cdot) \) is some polynomial, and taking the majority value.

An appropriate departure from independent noise is the random persistent classification noise model due to Goldman, Kearns, and Schapire [12] wherein on any query \( x \), the oracle operates as follows: If the oracle has been queried on \( x \) before, it returns its previous answer to \( x \); otherwise, it returns the correct answer with probability \( 1 - \eta \). Stating matters colorfully, the noise “freezes” on genesis. We refer to it, therefore, as freezing noise. An oracle with freezing noise appears deterministic from the outside, so clearly the corruption caused by this kind of noise cannot be undone by querying the oracle multiple times and taking the majority value.

While the freezing noise model is appropriate for a membership query oracle, this form of noise tends to make analysis difficult. Fortunately, if it is extremely unlikely that an algorithm \( A \) will query the oracle twice or more with the same value, then \( A \) can be treated as if it is making calls to an MQ oracle with random independent classification noise [10]. The analysis in the following sections takes advantage of this loophole. As \( n \) gets large it becomes extremely unlikely that the membership query oracle will be queried twice or more on the same input. Therefore, the analysis treats the noise as if it were independent.
3.2 Epistasis, NK Landscapes, k-juntas, Freezing Noise, and Parsimony

Typical run-time analyses of evolutionary algorithms assume a separable fitness function (one-max, weighted linear pseudo-boolean functions, the sphere function, etc.). The assumption of separability (i.e., non-epistasis) is a strong one, and as such, limits the applicability of the results of such analyses. In contrast, k-juntas with freezing noise come with much weaker assumptions, we claim. So the results obtained are more broadly applicable.

Our claim proceeds from the observation that rampant unstructured epistasis is well modeled by freezing noise. For a case in point, observe that a 0-junta (a constant function) with freezing noise can be modeled by an NK landscape \([17, 1]\) with \(K = N - 1\). For \(k > 0\), k-juntas with freezing noise cannot be modeled within Kauffman’s NK framework, but can be modeled perfectly well within Altenberg’s Generalized NK Maps framework [1].

A generalized NK map \(F\) is the mean of \(m\) fitness components \(F_1, \ldots, F_m\) defined as follows:

\[
F(x) = \frac{1}{m} \sum_{i=1}^{m} F_i(x_{j_1(i)}, x_{j_2(i)} \ldots, x_{j_{p_i}})
\]

where \(p_i\) gives the number of input variables of \(F_i\) and \(\{j_1(i), \ldots, j_{p_i}\} \subset [N]\) specifies the location of these variables in \(x\).

Situating k-juntas with freezing noise within Altenberg’s framework allows one to appreciate the nature of the departure from rampant unstructured epistasis when \(k > 0\).

The observation that highly epistatic fitness functions with a small amount of structure are more probable than separable or near-separable (i.e., low-epistasis) fitness functions completes our argument for the claim. The former class of fitness functions are more parsimonious i.e., less presumptive, than the latter.

3.3 Information Theoretic Lower Bound

For any positive integer \(k\), a simple information theoretic argument shows that it is not possible to PAC-learn the relevant variables of k-juntas in less than \(O(n \log n)\) time or less than \(O(\log n)\) queries; not possible, in other words, to PAC-learn the relevant variables in \(o(n \log n)\) time or \(o(\log n)\) queries. The argument relies on Shannon’s source coding theorem, part of which states that if \(N\) i.i.d. random variables each with entropy \(H(X)\) are transmitted in less than \(NH(X)\) bits, it is virtually certain that information will be lost [22].

Let us consider the minimum time and queries required to learn just one relevant variable. Observe that the oracle can transmit at most one bit per query and that the time required by \(\mathcal{A}\) to generate each query is \(\Omega(n)\). Finally recall that the entropy of a random variable \(X\) that can take an arbitrary value in \([n]\) is \(\Omega(\log n)\). Thus, by Shannon’s source coding theorem, the transmission of the index of a single relevant variable with an arbitrarily small possibility of error takes \(\Omega(\log n)\) queries and \(\Omega(n \log n)\) time.

3.4 An Initial Experiment

We ran the SGA in Algorithm 1 with uniform crossover (\(asex = false\)), population size \(m = 1500\), bitstring length \(n = 1000\), and per bit mutation rate \(p_{mut} = 0.004\) for \(\tau = 200\) generations using \(\phi_f\) as the fitness function where \(f\) is a 7-parity function over \(\{0, 1\}^{1000}\) with juntas in \(\mathcal{I} = \{45, 224, 295, 385, 696, 799, 838\}\), and \(\phi\) is an MQ oracle with noise rate \(\eta = 1/5\). The panel at the top of Figure 2 shows the frequency of the bit 1 at each locus in the initial generation and every fortieth subsequent generation. The chart at the bottom shows the simple sample mixability (see Example 12) of the juntas with respect to the parent population in each generation. As the panel on top shows, by generation 200 all the relevant loci and none of the irrelevant loci have gone to fixation. Indeed, one can simply “read off” the juntas of \(f\) by returning the index of loci in generation 200.
Algorithm 1: Pseudocode for a simple genetic algorithm with uniform crossover. The population is stored in an \( m \) by \( n \) array of bits, with each row representing a single genotype. \textsc{shuffle}(\cdot) randomly shuffles the contents of an array in-place, \textsc{rand}() returns a number drawn uniformly at random from the interval \([0,1]\), \textsc{ones}(a,b) returns an \( a \) by \( b \) array of ones, and \textsc{rand}(a,b) < c resolves to an \( a \) by \( b \) array of bits each of which is 1 with probability \( c \).

Input: \( m \): population size
Input: \( n \): length of bitstrings
Input: \( \tau \): number of generations
Input: \( p_{\text{mut}} \): per bit mutation probability
Input: \( \text{asex} \): (flag) perform asexual evolution

1. \( \text{pop} \leftarrow \text{rand}(m,n) < 0.5 \)
2. for \( t \leftarrow 1 \) to \( \tau \) do
3.   \( \text{fitnessVals} \leftarrow \text{evaluate-fitness}(\text{pop}) \)
4.   \( \text{totalFitness} \leftarrow 0 \)
5.   for \( i \leftarrow 1 \) to \( m \) do
6.     \( \text{totalFitness} \leftarrow \text{totalFitness} + \text{fitnessVals}[i] \)
7.   end
8.   \( \text{cumFitnessVals}[1] \leftarrow \text{fitnessVals}[1] \)
9.   for \( i \leftarrow 2 \) to \( m \) do
10.  \( \text{cumFitnessVals}[i] \leftarrow \text{cumFitnessVals}[i-1] + \text{fitnessVals}[i] \)
11. end
12. for \( i \leftarrow 1 \) to \( 2m \) do
13.   \( k \leftarrow \text{rand}() \ast \text{totalFitness} \)
14.   \( \text{ctr} \leftarrow 1 \)
15.   while \( k > \text{cumFitnessVals}[:\text{ctr}] \) do
16.     \( \text{ctr} \leftarrow \text{ctr} + 1 \)
17.   end
18.   \( \text{parentIndices}[i] \leftarrow \text{ctr} \)
19. end
20. \( \text{SHUFFLE} (\text{parentIndices}) \)
21. if \( \text{asex} \) then
22.   \( \text{crossOverMasks} \leftarrow \text{ones}(m,n) \)
23. else
24.   \( \text{crossOverMasks} \leftarrow \text{rand}(m,n) < 0.5 \)
25. end
26. for \( i \leftarrow 1 \) to \( m \) do
27.   for \( j \leftarrow 1 \) to \( n \) do
28.     if \( \text{crossMasks}[i,j] = 1 \) then
29.       \( \text{newPop}[i,j] \leftarrow \text{pop}[\text{parentIndices}[i],j] \)
30.     else
31.       \( \text{newPop}[i,j] \leftarrow \text{pop}[\text{parentIndices}[i+m],j] \)
32.     end
33. end
34. end
35. \( \text{mutationMasks} \leftarrow \text{rand}(m,n) < p_{\text{mut}} \)
36. for \( i \leftarrow 1 \) to \( m \) do
37.   for \( j \leftarrow 1 \) to \( n \) do
38.     \( \text{newPop}[i,j] \leftarrow \text{xor}(\text{newPop}[i,j], \text{mutMasks}[i,j]) \)
39. end
40. end
41. \( \text{pop} \leftarrow \text{newPop} \)
42. end
43. end
Figure 2: *Top Panel:* Frequencies of the bit 1 at each locus in generations 0, 40, 80, 120, 160, and 200 when Algorithm 1 with uniform crossover (asex = false), $m = 1500$, $n = 1000$, $p_{\text{mut}} = 0.004$ was run with fitness function $\phi_f$, where $f$ is a 7-parity function over \{0,1\}, $f$ with juntas \{45, 224, 295, 385, 696, 799, 838\}, and $\phi$ is a membership query oracle with noise rate $\eta = 1/5$. The black dots give the location of the juntas of $f$. The horizontal dotted lines mark the frequencies 0.05 and 0.95. *Bottom Figure:* Simple sample mixability (see Example 12) of the juntas with respect to the parent population in each generation.
where the frequency of the bit 1 is less than 0.05 or greater than 0.95. Note that an odd number of loci in $I$ are fixed at 1. This is because the parity function chosen rewards odd parity.

Our goal, going forward, is to obtain bounds on the time and queries required to recover the juntas of any 7-parity function with probability of error less than $\delta$ for arbitrarily large $n$ and arbitrarily small $\delta$.

4 Main Result and Approach

We present an evolutionary computation based algorithm that probably correctly learns $k$-parities in $O(\log(n/\delta))$ queries and $O(n\log(n/\delta))$ time given access to a MQ oracle with a noise rate of $1/5$. Our argument is comprised of two parts. In the first, we define a form of learning for the learning parities problem called Piecewise Probably Correct (PPC) learning and show that an algorithm that PPC-learns $k$-parities in $O(n)$ time and $O(1)$ queries can be used in the construction of an algorithm that PC-learns $k$-parities in $O(n\log(n/\delta))$ time and $O(\log(n/\delta))$ queries. In the second part we rely on a symmetry argument and a hypothesis testing based rejection of two null hypotheses at the $10^{-900}$ level of significance to conclude that for $\eta = 1/5$, a UGA can PPC learn 7-parities in $O(n)$ time and $O(1)$ queries.

5 PPC to PC learning

An algorithm $A$ with access to some oracle $\phi$ is said to piecewise probably correctly (PPC) learn $k$-parities if for any $k$-parity $f$, whose juntas are given by $J \subseteq [n]$, $A(n,k)$ outputs a set $S$ such that for any $x \in [n]$, $\Pr(\neg(x \in S \iff x \in J)) \leq 1/2$. That is, the probability that $A$ misclassifies $x$ is less than or equal to 1/2.

Given that it takes $O(n)$ time to formulate a query, PPC-Learning in $O(n)$ time and $O(1)$ queries is clearly optimal. The following theorem shows a close relationship between PPC-learning $k$-parities and PC-learning $k$-parities

**Theorem 13.** If $k$-parities is PPC-learnable in $O(n)$ time and $O(1)$ queries, then for any $\delta \in (0, 1/2)$, $k$-parities is PC learnable in $O(n\log(n/\delta))$ time and $O(\log(n/\delta))$ queries, where $\delta$ is the probability of error.

Given the information theoretic lower bound on learning relevant variables of a parity function with respect to $n$, for any positive integer $k$, Theorem 13 states that the PPC-learnability of $k$-parities in $O(n)$ time and $O(1)$ queries entails that $k$-parities are PC-learnable with optimal efficiency with respect to $n$. The proof of the theorem relies on the two well-known bounds. The first is the additive upper Chernoff bound [18]:

**Theorem 14 (Additive Upper Chernoff Bound).** Let $X_1, \ldots, X_r$ be $r$ independent bernoulli random variables, let $\hat{\mu} = \frac{1}{r}(X_1 + \ldots + X_r)$ be an estimator for the mean of these variables, and let $\mu = E[\hat{\mu}]$ be the expected mean. Then, for any $\epsilon > 0$, the following inequality holds:

$$\Pr(\hat{\mu} > \mu + \epsilon) \leq e^{-2r\epsilon^2}$$

The second is the union bound [18], which is as follows:

**Theorem 15 (Union Bound).** For any probability space, and any two events $A$ and $B$ over that space,

$$\Pr[A \cup B] \leq \Pr[A] + \Pr[B]$$

Crucially, the events $A$ and $B$ need not be independent.
Proof of Theorem 13. Let $\mathcal{A}$ be an algorithm that PPC-learns $k$-parities in $O(n)$ time and $O(1)$ queries with a per attribute error probability $\delta' < 1/2$. For any $k$-parity function $f$ over $n$ variables whose juntas are given by $J$, let $S_1,\ldots,S_r$ be sets output by $\mathcal{A}$ on $r$ independent runs, and let $S$ be a set defined as follows

$$x \in S \iff \frac{1}{r} \sum_{i=1}^{r} [x \in S_i]\{1\} > \frac{1}{2}$$

That is $x \in S$ iff $x$ appears in more than half the sets $S_1,\ldots,S_r$. We claim that $\Pr(S \neq J) \leq \delta$ if

$$r > \frac{2}{(1-2\delta')^2} \log \left( \frac{n}{\delta} \right)$$

Considering that it takes $O(nr)$ time to compute $S$ given $S_1,\ldots,S_r$, Theorem 13 follows straightforwardly from the claim. For a proof of the claim observe that for each $x \in [n]$ we have exactly two cases: (i) $x \in J$ and (ii) $x \notin J$.

Case i) $[x \in J]$: Let $\hat{\mu}_x$ be a random variable defined as follows:

$$\hat{\mu}_x = \frac{1}{r} \sum_{i=1}^{r} [x \notin S_i]\{1\}$$

Theorem 14 entails that for any $\epsilon > 0,$

$$\Pr(\hat{\mu}_x > E[\hat{\mu}_x] + \epsilon) \leq e^{-2r\epsilon}$$

Note that $E[\hat{\mu}_x] = \delta'$. So by the premise of Theorem 13, $E[\hat{\mu}_x] < 1/2$. Setting $\epsilon = 1/2 - \delta'$ in the expression above yields

$$\Pr\left(\hat{\mu}_x > \frac{1}{2}\right) \leq e^{-\frac{1}{2}r(1-2\delta')^2}$$

Thus, $\Pr(x \notin S) \leq e^{-\frac{1}{2}r(1-2\delta')^2}$.

Case ii) $[x \notin J]$: An argument similar to the one above with $\hat{\mu}_x$ defined as follows yields $\Pr(x \in S) \leq e^{-\frac{1}{2}r(1-2\delta')^2}$.

$$\hat{\mu}_x = \frac{1}{r} \sum_{i=1}^{r} [x \in S_i]\{1\}$$

By combining the two cases we get that for all $x \in [n]$, $\Pr(\neg(x \in S \iff x \in J)) \leq e^{-\frac{1}{2}r(1-2\delta')^2}$. The application of the union bound yields

$$\Pr(\neg(1 \in S \iff 1 \in J) \lor \ldots \lor \neg(n \in S \iff n \in J)) \leq ne^{-\frac{1}{2}r(1-2\delta')^2}$$

In other words, $\Pr(S \neq J) \leq ne^{-\frac{1}{2}r(1-2\delta')^2}$. Finally, setting $ne^{-\frac{1}{2}r(1-2\delta')^2} < \delta$ and taking logarithms yields the claim.
6 Symmetry Analysis

For any positive integer $m$, let $D_m$ denote the set $\{0, \frac{1}{m}, \frac{2}{m}, \ldots, \frac{m-1}{m}, 1\}$. Let $G$ be a UGA with a population of size $m$ and binary genotypes of length $n$. A hypothetical population is shown in Figure 3. The 1-frequency of some locus $i \in [n]$ at some time step $t$ is a value in $D_m$ that gives the frequency of the bit 1 at locus $i$ at time step $t$ (in other words the number of ones in the population of $G$ at locus $i$ in generation $t$ divided by $m$, the size of the population).

Let $f$ be a $k$-junta over $\{0, 1\}^n$ whose juntas are given by $J$ and let $h$ be the hidden function of $f$ such that $h$ is symmetric, i.e. for any permutation $\pi : [n] \rightarrow [n]$ and any element $x \in \{0, 1\}^n$, $h(x_1, \ldots, x_k) = h(x_{\pi(1)}, \ldots, x_{\pi(k)})$. Consider a noisy MQ oracle $\phi_f$ that internally uses $f$. Let $G$ be a UGA that uses $\phi_f$ as a fitness function, and let $1^t_i$ be a random variable that gives the 1-frequency of $G$ at time step $t$, then for any time step $t$, any loci $i, j \in J$ and any loci $i', j' \in [n] \setminus J$, an appreciation of algorithmic symmetry (the absence of the positional bias in uniform crossover and the fact that $h$ is symmetric) yields the following conclusions:

**Conclusion 16.** $\forall x \in D_m$, $\Pr(1^t_i = x) = \Pr(1^t_j = x)$

**Conclusion 17.** $\forall x \in D_m$, $\Pr(1^t_i = x) = \Pr(1^t_{i'} = x)$

Which is to say that for all $i, j \in J$, $1^t_i$ and $1^t_j$ are drawn from the same distribution, which we denote $p_t$, and for all $i', j' \in [n] \setminus J$, $1^t_i$ and $1^t_{j'}$ are drawn from the same distribution, which we denote $q_t$. (It is not to say that $1^t_i$ and $1^t_j$ are independent, or that $1^t_i$ and $1^t_{j'}$ are independent.) Appreciating that the location of the juntas of $f$ is immaterial to the 1-frequency dynamics of the relevant and irrelevant loci yields the following conclusion:

**Conclusion 18.** For all $t$, $p_t$ and $q_t$ are invariant to $J$ provided that $|J|$ remains constant

Finally, if it is known that that the per bit probability of mutation is not dependent on the length of the genotypes, then appreciating that the non-relevant loci are just “along for the ride” and can be spliced out without affecting the 1-frequency dynamics at other loci give us the following conclusion:
Conclusion 19. For all $t$, $p_t$ and $q_t$ are invariant to $n$.

6.1 Note on our Use of Symmetry

This section is a lightly modified version of Section 3 in an earlier paper [6]. We include it here because our case for the use of symmetry arguments remains the same.

A simple genetic algorithm with a finite but non-unitary population of size $m$ (the kind of GA used in this paper) can be modeled by a Markov Chain over a state space consisting of all possible populations of size $m$ [27]. Such models tend to be unwieldy [15] and difficult to analyze for all but the most trivial fitness functions. Fortunately, it is possible to avoid modeling and analysis of this kind, and still obtain precise results for non-trivial fitness functions by exploiting some simple symmetries introduced through the use of uniform crossover and length independent mutation.

A homologous crossover operation between two genotypes of length $n$ can be modeled by a vector of $n$ random binary variables $\langle X_1, \ldots, X_n \rangle$ representing a crossover mask. Likewise, a mutation operation can be modeled by a vector of $n$ random binary variables $\langle Y_1, \ldots, Y_n \rangle$ representing a mutation mask. Only in the case of uniform crossover are the random variables $X_1, \ldots, X_n$ independent and identically distributed. This absence of positional bias [9] in uniform crossover constitutes a symmetry. Essentially, permuting the bits of all genotypes using some permutation $\pi$ before crossover, and permuting the bits back using $\pi^{-1}$ after crossover has no effect on the overall dynamics of a UGA. If, in addition, the random variables $Y_1, \ldots, Y_n$ that model the mutation operator are identically distributed (which is typical), conditionally independent given the per bit mutation rate, and independent of the value of $n$, then in the event that the values of genotypes at some locus $i$ are immaterial to the fitness evaluation, the locus $i$ can be “spliced out” without affecting allele dynamics at other loci. In other words, the dynamics of the UGA can be exactly coarse-grained [4].

These conclusions flow readily from an appreciation of the symmetries induced by uniform crossover and length independent mutation. While the use of symmetry arguments is uncommon in Theoretical Computer Science, symmetry arguments form a crucial part of the foundations of Physics and Chemistry. Indeed, according to the theoretical physicist E. T. Jaynes “almost the only known exact results in atomic and nuclear structure are those which we can deduce by symmetry arguments, using the methods of group theory” [16, p331-332]. Note that the conclusions above hold true regardless of the selection scheme (fitness proportionate, tournament, truncation, etc), and any fitness scaling that may occur (sigma scaling, linear scaling etc). “The great power of symmetry arguments lies just in the fact that they are not deterred by any amount of complication in the details”, writes Jaynes [16, p331]. An appeal to symmetry, in other words, allows one to cut through complications that might hobble attempts to reason within a formal axiomatic system. Of course, symmetry arguments are not without peril. However, when used sparingly and only in circumstances where the symmetries are readily apparent, they can yield significant insight at low cost.

7 Statistical Hypothesis Testing

For any positive integer $n$, let $f$ be a 7-parity function over $\{0,1\}^n$, and let $\phi_f$ be a noisy MQ oracle such that for any $x \in \{0,1\}^n$ $\Pr(\phi(x) = \neg f(x)) = 1/5$ and $\Pr(\phi(x) = f(x)) = 4/5$. Let $G(n)$ be the simple genetic algorithm given in Algorithm 1 with genotypes of length $n$, population size $m=1500$, uniform recombination (asex = false), and per bit mutation probability $p_{mut} = 0.004$. Let $A^\phi(n)$ be an algorithm that runs $G(n)$ for 800 generations using $\phi_f$ as the fitness function and returns a set $S \subseteq [n]$ such that $i \in S$ if and only if the 1-frequency of locus $i$ at generation 800 exceeds $1/2$.

Claim 20. $A^{\phi_1}$ PPC-solves the learning 7-parities problem in $O(n)$ time and $O(1)$ queries.

Argument. Let $D'_m$ be the set

$$\{ x \in D_m \mid 0.05 < x < 0.95 \}$$
Figure 4: The 1-frequency dynamics over 3000 runs of the first (left figure) and last (right figure) loci of Algorithm 1 with $m = 1500$, $n = 8$, $\tau = 800$, $p_{\text{mut}} = 0.004$, and $\text{asex} = \text{false}$ using the membership query oracle $\phi_f^*$, described in the text, as a fitness function. The dashed lines mark the 1-frequencies 0.05 and 0.95.

Note that the hidden function of $f$ is invariant to a reordering of its inputs and the per bit probability of mutation in $G$ is constant with respect to $n$. Thus, Conclusions 16, 17, 18, and 19 hold. Consider the following two null hypotheses:

$$H_0^p : \sum_{x \in D'_{1500}} p_{800}(x) \geq \frac{1}{2}$$

$$H_0^q : \sum_{x \not\in D'_{1500}} q_{800}(x) \geq \frac{1}{2}$$

We seek to reject $H_0^p \lor H_0^q$. Assume $H_0^p$ is true, then for any independent random variables $X_1, \ldots, X_{3000}$ drawn from the distribution $p_{800}$, and any $i \in [3000]$,

$$\Pr(X_i \in D'_{1500}) \geq 1/2$$

which entails that

$$\Pr(X_i \not\in D'_{1500}) < 1/2$$

The independence of the random variables entails that

$$\Pr(X_1 \not\in D'_{1500} \land \ldots \land X_{3000} \not\in D'_{1500}) < \left(\frac{1}{2}\right)^{3000}$$

Let $f^*$ be the 7-parity function over $\{0, 1\}^8$ whose juntas are given by the set $\{1, \ldots, 7\}$. Figures 4a and 4b show the 1-frequency of the first and last loci, respectively, of $G(8)$ given the fitness function $\phi_f^*$ in 3000 independent runs, each 800 generations long. Thus, the chance that the 1-frequency of

---

6To rerun the experiment and examine the results, visit https://github.com/burjorjee/evolve-parities/tree/foga-2015 and follow instructions.
the first locus of $G(8)$ is in $D_{1500} \setminus D'_{1500}$ in generation 800 of all 3000 runs, as seen in Figure 4a, is less than $(1/2)^{3000}$. As $(1/2)^{3000} < 10^{-903}$, we can reject hypothesis $H^p_0$ at the $10^{-903}$ level of significance.

Likewise, if $H^q_0$ is true, then for any independent random variables $X_1, \ldots, X_{3000}$ drawn from the distribution $q_{800}$, and any $i \in \{3000\}$,

$$\Pr(X_i \notin D'_{1500}) \geq 1/2$$

which entails that

$$\Pr(X_1 \in D'_{1500} \land \ldots \land X_{3000} \in D'_{1500}) < \left(\frac{1}{2}\right)^{3000}$$

Thus, the chance that the 1-frequency of the last locus of $G(8)$ could be in $D'_{1500}$ in generation 800 of all 3000 runs, as seen in Figure 5b, is less than $(1/2)^{3000}$. We thus reject hypothesis $H^q_0$ at the $10^{-903}$ level of significance.

Each $p$-value is less than a Bonferroni adjusted critical value of $10^{-900}/2$, so we reject the global null hypotheses $H^p_0 \lor H^q_0$ at the $10^{-900}$ level of significance. We are left with the following conclusions:

$$\sum_{x \in D'_{1500}} p_{800}(x) < \frac{1}{2}$$

$$\sum_{x \notin D'_{1500}} q_{800}(x) < \frac{1}{2}$$

The observation that running $G(n)$ for 800 generations takes $O(n)$ time and $O(1)$ queries completes the argument.

### 7.1 Other values of $k$ and $\eta$?

The PC-learning result obtained above pertains only to $k = 7$ and $\eta = 1/5$. We expect that the proof technique used here can be used to derive identical bounds with respect to $n$ and $\delta$ for other values of $k$ and $\eta$ as long as these values remain small. This conjecture can only be verified on a case by case basis with the proof technique provided. The symmetry argument used in the proof precludes the derivation of bounds with respect to $k$ and $\eta$ as is typically done in the computational learning literature. Our goal, however, is not to derive such bounds for all $k$ and $\eta$, but to verify a prediction entailed by hypomixability theory, and in doing so to give the first proof of efficient computational learning in an evolutionary algorithm on a non-trivial learning problem. That the computational learning is optimally efficient is a welcome finding.

### 7.2 Asexual Evolution Does Not Solve Parities

As asexual evolution does not mix genotypes, it cannot be expected to capitalize on hypomixability, and hence cannot be expected to solve $k$-parities for any value of $k$. Nevertheless, for the sake of completeness, and to gain insight into the behavior of asexual evolution on $\phi_f$, where $f$ is some 7-parity function, we repeated the experiment described earlier in this section with recombination turned off. Note that the symmetry analytic conclusions 16, 17, 18, and 19 of Section 6 continue to hold. Figure 7.1 shows the 1-frequencies of the first and last loci of $f$ of Algorithm 1 over 75 runs using $\phi_f$ as a fitness function as before and $m = 1500$ and $n = 8$ as before. The values of $\tau$ and $asex$ were changed to 10000 and true respectively. In other words, we greatly reduced the number of runs, greatly increased the length of each run, and disabled recombination. As the figure shows, the first locus did not go to fixation even once during the 75 runs, despite an increase in run length from 800 to 10000 generations.
8 Adaptation and Speciation

A one-time elimination of hypomixability, even if optimally efficient, cannot explain the sustained adaptation that is observed in natural and artificial evolutionary systems. Fortunately, this problem is easily fixed by making a weak assumption. We assume that hypomixability elimination is generative. That is, we assume that the elimination of statistically significant hypomixability in some small collection of loci engenders statistically significant hypomixability in one or more small collections of loci elsewhere. With this assumption, adaptive hypomixability elimination can go from being a one-time occurrence to an ongoing phenomenon.

The heuristic that emerges (Generative Hypomixability Elimination) is non-local because it does not make use of neighborhood information. It is noise-tolerant because it is sensitive only to the average fitness values of coarse schemata, and it is general-purpose because it relies on very weak assumptions about the distribution of fitness over a search space. Proof of concept can be found in previous works [5, 6]. In these publications hypomixability elimination was hypothesized to be entropy minimizing (hence the name Generative Fixation [5]). The current theory that hypomixability elimination is entropy maximizing subject to the unit-mixability constraint corrects this error.

The unmistakable seed of a theory of speciation lies in the observation that there are 64 different ways to adaptively eliminate hypomixability in the 7-parity problem, no two of which are sexually compatible. While a panmictic population always pursues just one way to eliminate hypomixability in some set of loci, island/spatial models may be able to pursue two or more (sexually incompatible) ways in parallel. Each way is a branch and each branch may sprout its own branches.

9 The Emergence of Modularity

It has been noted that modularity in sexual evolution can emerge through changes in allele frequencies under sex and selection [21, 23]. Such emergence can be readily observed in the results shown in the top panel of Figure 2. Consider locus 45, the first of the seven juntas. In generation 0, the alleles 0

---

Footnote:

7For small values of k, an OR k-junta is an example of a fitness function where the evolutionary elimination of statistically significant hypomixability is not accompanied by the fixation of any juntas.
and 1 at locus 45 have identical selection coefficients because the expected parity of the six remaining juntas in a genotype picked at random from the population is approximately $1/2$. By generation 160, the expected parity of these six juntas in a genotype picked at random from the population is close to 1. So allele $0$ at locus 45 has a higher selection coefficient than allele $1$.

The emergence of modularity during sexual evolution is offered up by Livnat et al. [21] as a reason to go along with hypotheses that make an \textit{a priori} assumption of modularity. They remark, “It is not necessary to ask, then, how mixability [i.e. emergence of modularity in their context] supports the Fisher/Muller hypothesis, the deterministic mutation hypothesis, or other hypotheses that rely on separate effects.” According to Livnat et al., the support rendered to such theories by the emergence of modularity is obvious. In any event, such theories are not put on trial.

We are of a different mind. Based on the computational complexity results presented earlier in this paper, we submit that the emergence of modularity (an outcome of hypomixability elimination) is not a sideshow, but sexual evolution’s \textit{main} act; theories of evolution based on \textit{a priori} assumptions of modularity, in effect, swoop in to explain what happens \textit{after} the most difficult part of the show is over. What remains to be done at that point is the computational equivalent of \textit{mopping up}. Necessary work, to be sure, but certainly not as computationally remarkable or difficult to explain.

10 Conclusion

The computational origins problem was introduced and for the first time, optimally efficient non-trivial computation was demonstrated in an evolutionary system. This demonstration serves as validation of a prediction entailed by hypomixability theory—a parsimonious theory that provides a unified explanation for sex, adaptation, speciation, and the emergence of modularity in evolutionary systems.

Acknowledgements

Many thanks to Lev Reyzin for answering numerous questions about Computational Learning Theory.

References


