Towards an Empirical Measure of Evolvability

Joseph Reisinger
joeraii@cs.utexas.edu

Kenneth O. Stanley
kstanley@cs.utexas.edu

Risto Miikkulainen
risto@cs.utexas.edu

Dept. of Computer Sciences
University of Texas at Austin
1 University Station, C0500
Austin TX, 78712-0233

ABSTRACT

Genetic representations that do not employ a one-to-one mapping of genotype to phenotype are known as indirect encodings, and can be much more efficient than direct encodings for complex problems. Increasing a representation's capacity to facilitate effective search, i.e. its *evolvability*, has long been a goal of Evolutionary Computation. However, currently no benchmarks exist to measure evolvability. One reason is that it is difficult to decouple a representation's capacity to evolve under any fitness function, i.e. the latent evolvability, and its *performance* on a specific benchmark. Towards this goal, a method is proposed in this paper that measures the representation's ability to extract invariant properties from a changing fitness function. The test is applied to three distinct representations and it is able to distinguish all three. Ultimately, this test can serve as the foundation for performing controlled experiments determining what factors contribute to evolvability.

Categories and Subject Descriptors

G.1.6 [**Optimization**]: Global Optimization; I.2.8 [**Artificial Intelligence**]: Problem Solving, Control Methods, and Search

General Terms

experimentation, measurement, performance

Keywords

genetic algorithms, evolvability, representations, indirect encodings, genetic regulatory networks, development

1. INTRODUCTION

Evolvability is the genome's ability to *adaptively* organize how mutations affect the phenotype, through a many-to-one genotype-phenotype mapping [6, 17, 21]. This kind of adaptation arises through the interactions between the genetic operators and the genetic representation (i.e. the genotype-phenotypic map). Although it is generally believed that improving the evolvability of artificial evolutionary systems would benefit evolutionary computation (EC), to date no standardized test exists to measure it.

Two primary reasons motivate an empirical measure of evolvability. First, such a test can determine whether or not representations exist that are evolvable across many fitness functions. Second, such a test would make it easier to study what factors contribute to evolvability, which in turn will allow EC researchers to methodically construct representations more amenable to artificial evolution. Ultimately this may lead to a more fundamental understanding of why indirect encodings seem to be more evolvable in general.

Determining exactly what factors affect evolvability is important because artificial developmental systems that are based on biology (e.g. models with multiple cells, genetic regulatory networks, etc.) are computationally expensive, and it is therefore important to find the most tractable representation with high evolvability. Such models may provide valuable insight into more complex systems [6]. This paper takes a first step in this direction, focusing on relatively simple genotype-phenotype mappings.

Unfortunately, directly measuring evolvability is difficult for several reasons:

1. Evolvability in natural evolution is the capacity for an adaptive response to a *dynamic* environment (fitness function)[8], however in artificial evolution, the fitness function is in many cases static, so there is little selection pressure for "evolvability" in the biological sense.

2. It is possible to overlook the effects of convergence in a trained population. A population with training on a certain domain may have acquired some evolvability with respect to that domain, but may be more converged than a randomly initialized population, and thus may perform less efficiently. This is also influenced by the location of the population in the genetic space, with respect to the target function.

3. Finally, it is easy to confuse search performance and evolvability. For example, a particular encoding may solve a simple problem more efficiently, but not exhibit any traits associated with evolvability. Moreover, the traits that facilitate effective long-term evolution may not directly benefit evolution in the short-term.

This paper describes a principled method to address these difficulties and establish a direct test for a representation's contribution to evolvability (disregarding genetic operators as a source of evolvability). The evolvability of a particular representation acquired during the course of evolution can

GECCO'05, June 25–29, 2005, Washington, DC, USA.

be measured as the degree to which the representation enables evolution to identify and exploit underlying factors inherent in the fitness function, for example fundamental symmetries or modular structures. This measure can be used to augment more theoretical analysis, for example through analyzing representations that induce neutral networks [4].

This paper is organized into four main sections: a description of natural and artificial evolvability, the evolvability measure, the representations tested, and the results.

2. EVOLVABILITY

Evolvability has been defined several ways, e.g. the "ability to respond to a selective challenge" [8], the "ability of random variations to sometimes produce improvement" [21], etc. Evolvability can be intuitively understood as adaptive organization of the genotype-phenotype mapping such that the search operators produce more favorable phenotypes (i.e. choose a mapping with local neighborhoods around good solutions that are amenable to search). In this sense, evolvability is an aspect of the representation (genotype to phenotype mapping) and the search algorithm employed. Although the phrase "evolvable representation" is commonly used in EC literature, representations generally only have a *capacity* to become evolvable: evolvability emerges over the course of evolution with a specific fitness function, and is defined within the terms of that function. Therefore, in this paper the term *latent evolvability* will be used to describe the representation's underlying capacity for becoming evolvable, and *acquired evolvability* will be used to refer to its evolvability learned in response to a particular fitness function. The test defined in this paper will measure acquired evolvability, and use that as a proxy for latent evolvability.

The general concept of evolvability applies to both natural evolution and in artificial evolution; however latent evolvability is not important in nature: sudden shifts in the environment factors occur, but the genome is not restarted with a random configuration with no acquired evolvability. Also in nature, evolvability can be selected for directly, since more evolvable representations are better able to survive in the long-term (the environment is not static); genomes that are able to react to changing environments quickly (i.e. have high acquired evolvability) are more likely to survive. However, in artificial evolution, less selection pressure exists to generate "evolvable" phenotypes because a single "perfect" phenotype usually exists: outside of coevolution or incremental evolution [7], the fitness function is generally not adaptive. If the fitness function is static, there is little need for evolvability, since any solution with high fitness, even one with low evolvability, is likely to survive. Thus if a representation will be ultimately used in a domain with a static fitness function, it is more important to increase latent evolvability, since no evolvability will be acquired during evolution. Determining how to measure latent evolvability, and what representations exhibit a high latent evolvability will be a theme for future work.

From studies of biological organisms it is clear that many factors affect evolvability, including:

- *modularity*: the organization of genotypic and phenotypic features into reusable components [11, 13].
- *duplication and divergence*: structure A is duplicated yielding structure B, then B is free to change function without resulting in a loss of A [13].
- *canalization*: organizing the effects of mutations such that some features become more resistant to change [19, 20].
- *developmental constraints*: constraining phenotypic variation against the appearance of less fit individuals [12].
- *hidden genetic variation*: the developmental system builds up phenotype-neutral mutations that can be expressed *en masse* with a certain selection pressure [9].

Whether or not these factors are products of selection for evolvability or necessary preconditions for it is not yet well understood. However, even simple developmental encodings such as GRNs and RNA folding can exhibit these features (see [2] for generic canalization in GRNs, and [6] for a general overview in the RNA folding case). Thus the contribution to evolvability of these factors can be measured without modeling a full developmental system.

3. MEASURING EVOLVABILITY

One way to measure evolvability in artificial evolution is to compare the performance of several representations on a standard benchmark test. However, such a test would confuse the effects of the representations' latent evolvability and their performance on the given task. Such a test is fine comparing the effectiveness as an optimization algorithm, but does not give an accurate measure of the relative evolvability of each representation.

Instead, acquired evolvability can be empirically measured as the representation's ability to retain and generalize information learned about a changing domain. Specifically, by varying the amount of information provided during a training period (i.e. training on an adaptive fitness function) and measuring the resulting *change* in efficiency during a test phase, evolvability can be determined. If the representation exhibits a large difference in efficiency between cases where little information is provided, and cases where information about the underlying problem is plentiful, then the representation can be said to have acquired evolvability for the domain. Conversely, no change would indicate that the representation does not make use of the information provided.

This test separates evolvability from benchmark performance and furthermore from the effects of genetic variance (by comparing only trained populations). Separating performance and genetic variance is necessary because direct encodings can perform well on some tasks, even on re-evolution, although they exhibit no evolvability. In other words, this separation allows systematic study of what is good for evolution in the long-term (evolvability), instead of the short-term (finding the optimal solution).

Since acquired evolvability is being measured in a manner uncoupled from the performance on a particular benchmark, it is important to ask what it means to have high evolvability, but to perform relatively poorly at a certain task? Certainly if the representation were poor for many tasks, it might not be particularly useful in general. However, representations with high evolvability should perform better, on average, on problems with adaptive fitness functions, such as those found in coevolution or incremental evolution, since adaptability is more directly rewarded. Ultimately, the goal of this line of work is to develop an encoding that has both high evolvability and is efficient in solving many tasks, but since performance is the best understood of the two goals,

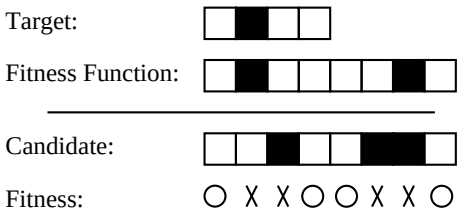

Figure 1: Calculating specific symmetry. The fitness function is composed of the target function concatenated with its mirror image (in order to ensure bilateral symmetry). The candidate solution is rewarded for characters matching the fitness function, but only when the character that is bilaterally-symmetric also matches.

this paper emphasizes the effort to raise evolvability, as opposed to pure benchmark performance.

The domain used to demonstrate the test procedure is binary string symmetry. A distinction must be made between learning *general symmetry* of any target string that is symmetric about the midpoint, and *specific symmetry* where one half of the target string is defined explicitly, and this half is mirrored in order to complete the full target string. This distinction is important because the degree to which the representation learns general symmetry from instances of specific symmetry is taken as the measure for acquired evolvability. Fitness is calculated as the number of characters in the phenotype that match both the target function and the character that is bilaterally-symmetric (figure 1). Phenotypic characters that only match the target function do not count towards fitness, so that only matching symmetries are rewarded.

This domain was chosen for two reasons: first, the fitness calculation is computationally simple to perform, and second, there is a clear underlying structure to the problem that does not change, even if the target is changed (i.e. target-independent bilateral symmetry). Thus, if the representation is capable, there is an opportunity for exploiting this invariance (general bilateral symmetry) in order to become more evolvable, and furthermore, no other opportunities exist to do so.

By varying how quickly the specific instances of bilateral symmetry change (the *target drift rate*), training sessions can be constructed that provide variable amounts of information about general symmetry. For example, a representation trained on only a single instance of bilateral symmetry has no pressure to exploit general symmetry as a survival strategy. Therefore the representation might not exhibit acquired evolvability, even though it has the capacity for it. Conversely, if the target function is changed slowly, then there *is* a pressure for such learning. Since the optimal level of change for a given representation is not known ahead of time, several trials are run with the target drift rate varied between 0.0 to 0.5 (0-50% change per generation), i.e. ranging from static evaluation to evaluation on random noise. The idea that an adaptive target function may facilitate evolvability is due to Lipson et. al. [11].

After the initial training period, the target function is completely randomized (maintaining bilateral symmetry), and the *efficiency* is measured by running for 100 generations and then averaging the best fitness of each trial at every epoch. This procedure ensures that the result is based

- For each value of the target drift rate:
  - Evolve for a fixed number of generations (150), randomly changing the target after each generation.
  - Reset to a new target function.
  - Evolve without changing the target function for a fixed number of generations (100).
  - Calculate the *efficiency* for this trial as the average of the best fitness individual over the last 100 generations.
- Calculate the acquired evolvability as the variance of efficiency over all drift rates.

Figure 2: Summary of the method employed to measure acquired evolvability.

not only on the quality of the final solution reached, but also on how long it took the population to produce that solution. The number of training generations is set at 150, during which time the target function is allowed to change. In this paper, target drift rates ranging from 0 to 0.5 are tested in a total of nine trials: 0.0, 0.01, 0.05, 0.1, 0.15, 0.2, 0.3, 0.4, and 0.5, in order to cover a full range of high- and low-information cases. See figure 2 for a summary.

Finally, acquired evolvability can be identified qualitatively from the shape of the average efficiency vs. target drift rate curve. The more pronounced the "bell-curve" shape, the higher the evolvability, since there is a significant difference between the trials where information about general symmetry is provided (i.e. 1% - 10% drift), and trials where little or no information is provided (i.e. 0% drift, and 50% drift). Conversely, a flat line represents no evolvability, since both high-information and low-information cases provide the same result. The "tails" of the bell-curve are fixed at 0% (no information due to an unchanging fitness function) and 50% (no information due to the fitness function changing too quickly). However, the location of the "peak" of the bell-curve depends on the representation, although in all three representations tested in this paper, the peak fell between 1% and 10% target drift. That the representation should be able to extract the most information from this case can be understood intuitively: the target is changing slowly enough for the population to adapt when it drifts, but fast enough to keep the population from converging on a specific target, providing information about general symmetry.

Based on this qualitative analysis, acquired evolvability can be quantified as the variance of efficiency over all trials,

$$E = \sum_t (X_t - \mu)^2 / N(t),$$

where X_t is the efficiency of trial t, μ is the average over all the trials, and $N(t)$ is the number of trials. High variance corresponds to a steeper bell-curve and higher evolvability. Conversely, low variance corresponds to a flat-line, or no difference in performance between low-information and high-information cases, thus little evolvability. By using this metric, representations can be assigned a numerical score for acquired evolvability and thus ranked empirically.

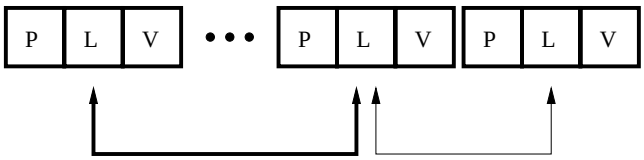

Figure 3: The modified direct genetic representation. Each gene consists of a phenotypic trait (P), and a linkage (L) and variance (V) parameter. In this example the first two traits are linked very strongly and the second two traits are linked weakly. The addition of the linkage and variance parameters allows the genotype to encode information about general symmetry.

4. CANDIDATE ENCODINGS

Three encodings are tested using this method (for parameters see appendix):

1. A direct-encoding, where the genotype and phenotype are the same, represented by a binary string.
2. A modified direct encoding, containing secondary parameters affecting how mutations should occur for each character, and at what rate.
3. An encoding based on genetic regulatory networks, where the phenotype is read as the network state after a set number of iterations.

These three encodings were chosen to cover as broad a range as possible, and in particular the modified direct encoding is the simplest possible representation able to learn the underlying dynamics of the problem. It also serves as a basis for controlled testing of features found in natural developmental systems (e.g. developmental variation and modularity). By distinguishing these encodings, the proposed test is shown to produce useful insights for future research.

4.1 Direct

The direct encoding tested in this paper consists of a binary string representing both the genotype and phenotype. Standard point mutation (bit flip) and one-point crossover are employed, and each phenotypic character has the same chance of being mutated. This encoding serves as a baseline to calibrate the test for evolvability. Any evolvability it exhibits must be strictly due to genetic variance inherent in the population, since the only other modifiable parameters are the phenotypic characters undergoing evolution. In other words, since the genotype is not extensible, there is no space to store information learned about the fitness function in general; only specific instances can be stored.

4.2 Modified Direct

The modified direct encoding is similar to the direct encoding, with the exception that each phenotypic trait also has a *linkage* and *developmental variance* parameter (figure 3). The same evolutionary operators are used.

The linkage parameter is a value from 0 to 1 which describes how single mutations affect multiple traits. If two traits have very similar linkage parameters, then they will most likely be mutated together (to the same value). The function used to determine this probability is

$$E(\ell_1, \ell_2) = e^{-T*(\ell_1-\ell_2)^2}, \quad (1)$$

where T is the *tolerance* parameter controlling what linkage distances are effective, and ℓ_1 and ℓ_2 are the two linkage parameters being compared. This function is a Gaussian distribution with a maximum of 1.0 occurring when $\ell_1 = \ell_2$, i.e. when the two linkage parameters are equal.

The linkage parameter allows the genetic representation to store knowledge it has gained of the fitness function in a way that does not necessarily directly affect fitness. This type of parameter is known as a *neutral trait* [16], and can be considered a form of *strategy parameter*, commonly found in Evolutionary Strategies (ES) literature[15]. This neutrality is non-trivial, however, since although the linkage parameters only affect how future mutations are structured, the effects of a linkage mutation will manifest over time as a fitness change. In the case of learning specific symmetry, an *optimal* linkage pattern exists linking each phenotypic trait on the string to the corresponding trait on the second half that is bilaterally symmetry (i.e. general symmetry).

The developmental variance parameter describes the trait's propensity to be mis-expressed during fitness evaluation. Each phenotypic trait has a probability of being mutated (flipped) during development with a probability equal to the variance parameter. Combining developmental variance with multiple evaluations of a single genotype yields an average of the local search space around the solution, biased by the linkage parameters. That is, if the linkage is bilaterally symmetric, then only bilaterally symmetric variants will be included in the average. Tying linkage directly to fitness is important for rewarding solutions that have learned the "correct" parameter linkage for the target problem.

Two versions of the modified direct encoding were tested, one with developmental variance taking place as described above (the standard implementation), and a second with the effects of developmental variance randomized. Thus the effects both parameters have on evolvability can be measured independently; linkage by comparing the second version to direct, and variance by comparing both modified encodings.

4.3 Genetic Regulatory Network

The third encoding tested in this paper is based on a simple model of the processes inherent to genetic regulatory networks (GRNs), particularly those found in eukaryotes. GRN encodings are often used in EC research because they are considered to have high evolvability, which makes them a particularly interesting encoding to test.

4.3.1 GRN Encodings

Recent studies in EC highlight the use of GRN encodings, for example, the work of Eggenberger [5] and Bongard [3], as well as in theoretical biology, for example in the work of Hogeweg [10]. These methods are all *multicellular*, where the GRN state evolution over time describes cell division and differentiation. Since complex morphologies consisting of many cells can be computationally intensive to model, the implementation tested in this paper is limited to a single GRN, which corresponds to single-celled eukaryotes, such as yeast.

Despite the relative simplicity of single GRN models, they have several features which make them important for studying evolvability. First, GRNs are capable of organizing as *scale-free* networks, i.e. a mix between random and hierarchical organizations [1]. Scale-free networks exhibit many properties that are important for evolvability, such as modu-

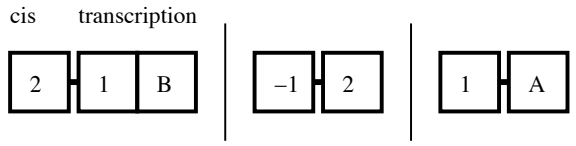

Figure 4: The GRN genetic representation. Each gene has a cis region and a transcription region, as well as an innovation number (not depicted). The GRN specified by this genome is shown in figure 5.

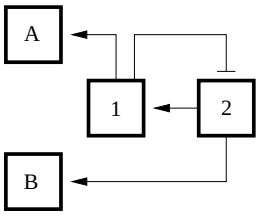

Figure 5: The resulting network from the genome shown in figure 4. Arrows represent promotion, and the flat connection between product 1 and 2 is inhibitory. A and B are environmental tfs, i.e. they determine the state of the phenotype but do not perform any regulatory function like 1 and 2. GRN representations are able to represent complex networks of interactions compactly.

larity and robustness against failure due to single mutations. Second, GRNs inherently support the "duplication and divergence" scheme whereby a gene is first duplicated, and the copy then begins to develop some new function, and slowly diverges from the original [22]. This phenomenon allows new function to develop while still preserving the old regulatory structures. Duplication and divergence is implicated as a source of modularity, which is necessary for evolvability [14]. Finally, complex network models in general have been shown to exhibit canalization [2].

4.3.2 Implementation

In this paper, a GRN consists of a concentration of *transcription factors* (tfs) in some arbitrary substrate, and a set of fuzzy if-then rules for reading the current state and altering the production of one or more tfs. These rules are encoded directly into the genome itself. A single gene consists of a cis-regulatory (cis) region paired with a list of transcription factors (figure 4). The cis region contains the precondition for producing the proteins specified in the tf list. It is composed of promoter sites, which facilitate protein transcription, and inhibitor binding sites, which inhibit production. Proteins in the environment bind to these sites and affect the production of the tfs specified in the tf list. The transcription factors produced can either be *environmental* or *regulatory*. Environmental tfs correspond to the "outputs" of the GRN (and thus have a set number, one per phenotypic character). At each time-step during the simulation their concentrations are read and stored as entries in the output vector. The regulatory tfs bind back to the cis region to determine the current state of the network. New regulatory tfs can arise through mutation, however the number of environmental tfs is fixed.

A key feature of biological GRNs is the degree to which promoter binding is "fuzzy." That is, an entire family of similar proteins are capable of binding to a specific promoter site, allowing complex GRNs to form with a compact specification. This fuzziness is captured in the artificial GRN encoding as *binding efficacy*, which depends on how compatible the two proteins are.

- Set all tf concentrations to a random value between 0 and 1, initially.
- Set all production rates to 0.0, initially.
- For a fixed number of steps:
  - For each gene, calculate the production rate based on the concentrations of the proteins binding at each promoter and inhibitor site (eqn. 2).
  - For each tf in each gene, increase or decrease that tf's concentration based on the production rate (eqn. 3). Clamp the concentration $[0, 1]$.
  - Decay the tf concentrations by a constant factor.
- Record the absolute concentrations of the environmental tfs.

Figure 6: GRN activation method.

Initially the environment contains a set concentration of tfs (random). When a tf in the environment matches (within some tolerance level based on the binding efficacy) either a promoter or an inhibitor region, it binds to that region and affects the production rate of the transcription factors encoded in that gene (figure 5). The production rate is calculated in two steps. First, the promoter efficacy of a single gene is calculated by

$$P(G) = \sum_{c \in C} \sum_{v \in V} \text{sign}(c) * \text{concentration}(v) * E(c, v), \quad (2)$$

where C is the set of all binding sites in the cis region of gene G, V is the set of all regulatory tfs in the environment, and E is the binding efficacy, given in equation 1. Then the tfs specified by that gene are added to the environment, with concentrations given by

$$\Delta\text{concentration}(v) = P(G)/\text{length}(C). \quad (3)$$

The newly produced proteins are added to the existing environmental concentrations. This cycle continues for a predefined number of iterations, at the end of which the environmental tfs are arranged in order and the phenotype is read as the concentrations thresholded $\{0, 1\}$ (figure 6).

The GRN encoding uses the NeuroEvolution of Augmenting Topologies (NEAT) method for performing crossover on variable length genomes [18]. Initially each genome contains 40 genes (one for each environmental tf), but new genes can arise through mutation. Each gene consists of a cis region, transcription region, and a unique *innovation number*. Innovation numbers record when new *regulatory structure*, i.e. a new character in a cis or regulatory region, or a new gene, is introduced to a genome. During crossover, the innovation numbers are lined up so that similar coding regions can be crossed over correctly, regardless of the amount of genotypic complexification (i.e. artificial synapsis is performed).

4.3.3 Variants

In addition to the standard implementation, three variants of the GRN encoding are compared:

1. A variant with the tolerance parameter set to one half of the baseline value. The tolerance parameter controls how similar a tf must be to a promoter/inhibitor site in order to bind to it. A decrease in tolerance means more proteins are compatible with more sites, and thus the overall network connectivity increases. If the tolerance value is set too low the network will become noisy and unable to perform meaningful computation. On the other hand if the tolerance value is set too high, then few interactions will take place.

2. A variant with initially two regulators in each cis region instead of one. Increasing the size of the cis region has a similar effect to decreasing the tolerance (increase connectivity), however since the cis region is under genetic control, the connectivity can be fine tuned. The main cost is a two-fold increase in the initial genome size, which may make search inefficient.

3. A variant with non-random initial tf state. The baseline GRN has a randomized initial tf state, in order to encourage the GRN function to become robust (i.e. applicable to many environments). The non-random version does not select for this robustness, and therefore may exploit the initial tf state to encode its phenotype. Testing this variant will help determine the extent to which reliance on epigenetic factors (e.g. initial tf state) may affect evolvability.

Comparing the evolvability of the four GRN variants will help determine what factors affect the GRN's ability to extract and exploit information about the target problem.

5. RESULTS

Each encoding along with several parameter variants were tested with target drift rate values of 0.0, 0.01, 0.05, 0.1, 0.15, 0.2, 0.3, 0.4, and 0.5. Though the optimal drift rate may differ from encoding to encoding, this range should be sufficient to capture the full range of learning response. For each target drift rate setting, the encoding was tested 40 times, to insure a useful average. An overview of the acquired evolvability scores is given in figure 7. All tests for statistical significance were done using the standard Analysis of Variance (ANOVA) method, which gives the probability that any pairing of two means is statistically significant.

The direct encoding *performs* better during the test phase than either of the other two representations (an average efficiency of about 91%; figures 8, 9, and 11), but exhibits little efficiency variance, less than 0.00001697 (figure 7). The means contributing to this variance are not statistically significant ($P = 0.59$), thus indicating that there is no acquired evolvability. Since there is no genetic capacity for the direct representation to record information about the domain, any level of variance must be due to sampling bias or the population's genetic variance. In general, representations that do not show statistically significant differences in efficiency have acquired no evolvability during the test, and thus may have little or no latent evolvability.

Two versions of the modified direct encoding were tested: the standard implementation (modified), and a variant with random developmental variance (modified-random), as discussed in discussed in Section 4.2. The variant exhibits no evolvability (variance is not statistically significant, $P = 0.56$), similar to the direct encoding (figure 7). However,

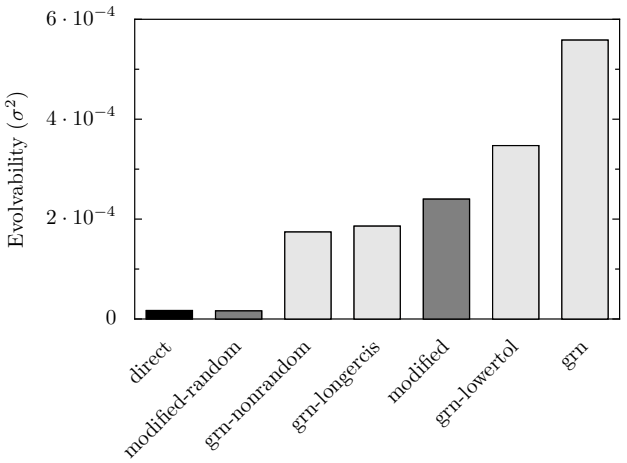

Figure 7: Overall comparison of all encodings and parameter variants by acquired evolvability. The light-gray colored bars represent GRN-based variants, the dark-gray bars represent modified direct variants, and the black bar is the baseline direct-encoding. "grn" refers to the baseline GRN implementation, "grn-lowertol" to a GRN variant with the tolerance parameter halved, "grn-longercis" to a GRN variant with the initial cis region length doubled, "grn-nonrandom" to a variant with all initial tf levels set to 1.0. "Modified" refers to the modified direct baseline, "modified-random" to the modified direct implementation with random developmental variance, and "direct" to the direct encoding. The GRN encoding is the most evolvable, and the evolvability of the modified encoding depends heavily on the form of the developmental variance.

the baseline implementation exhibited more than 14 times the evolvability of the direct encoding (variance is statistically significant, $P < 0.0001$), though performing on average more than 10% worse (figure 8). This result indicates that developmental variance plays a large role in determining evolvability, and in particular that this variance must be meaningful with respect to the target fitness function, otherwise there is no evolvability increase.

Four parameter variants of the GRN encoding were tested: the standard implementation (grn), from section 4.3, an implementation with the tolerance parameter set to one-half of the baseline value (lowertol), an implementation with initially two promoters in each cis region, instead of one (longercis), and an implementation with non-random initial tf state and all tfs initialized to 1.0 (nonrandom). Evolvability is summarized in figure 7 and efficiency vs. target drift in figure 9. With an acquired evolvability of 0.000558 (statistically significant, $P < 0.0001$), the baseline GRN encoding is almost 33 times more evolvable than the direct encoding. The lower tolerance parameter variant performs slightly worse than the baseline GRN, and the other two variants (longer cis, and nonrandom initialization) both exhibit less evolvability than the modified direct encoding. The latter two variants exhibit efficiency variance that is not statistically significant (and therefore no acquired evolvability), though only by a small margin ($P = 0.52$). The absolute performance of the GRN encoding is similar on average to the modified direct encoding, on average 5-10% worse than the direct encoding.

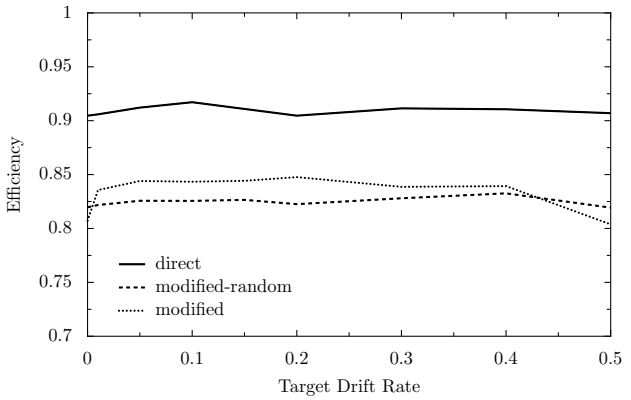

Figure 8: Comparison of two modified direct variants for efficiency. Refer to figure 7 for variant descriptions. The direct encoding and modified-random show little variance, while the modified direct baseline encoding has distinct dips towards the low-information target drift settings (0.0 and 0.5). This is consistent with the "bell-curve" shape.

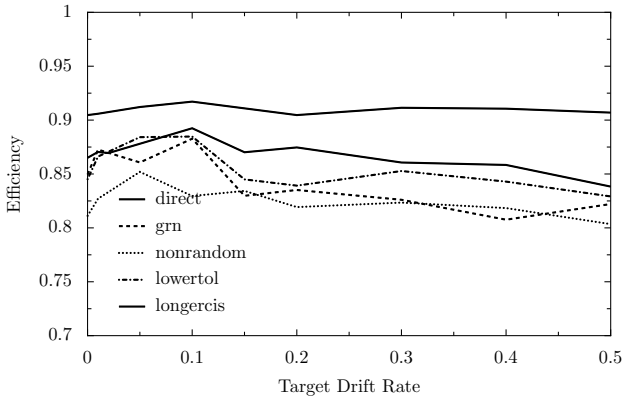

Figure 9: Comparison of the GRN variants for efficiency. Refer to figure 7 for variant descriptions. The direct encoding performs better on average at each target drift setting, but exhibits no variance in performance, unlike the GRN encodings.

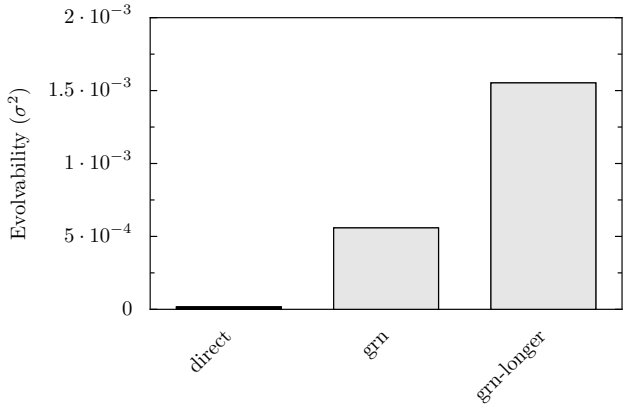

Figure 10: Comparison of the effects of training on the acquired evolvability of GRN representations. A training period length of 300 generations (grn-longer) leads to an effective tripling of evolvability.

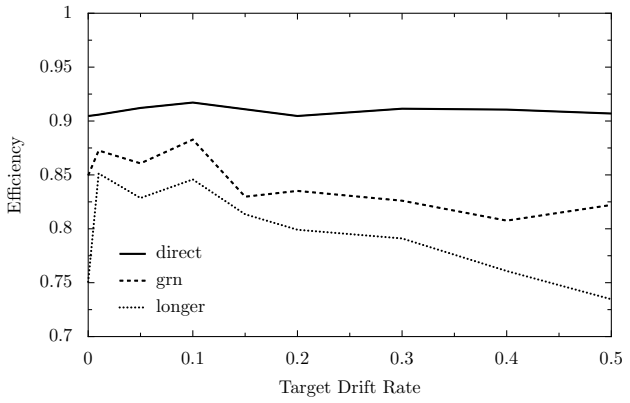

Figure 11: Comparison of the training period length GRN variants for efficiency. Refer to figure 7 for variant descriptions. The "grn-longer" plot refers to a GRN variant where the training period is 300 generations, and exhibits high efficiency variance.

Finally, the standard GRN implementation was run with a training period of 300 generations, instead of 150. This change resulted in nearly a 3-fold increase in the acquired evolvability (statistically significant, $P < 0.0001$), an effective 91-fold increase over the direct implementation (figure 10 for evolvability comparison, figure 11 for efficiencies). This result indicates that the length of training may play an important role in determining acquired evolvability.

6. DISCUSSION AND FUTURE WORK

Changing the fitness function over time (adaptive fitness function) plays an important role in evolution because a static fitness function does not provide any pressure to select for evolvability. Therefore, even if a representation is highly evolvable, there may be no pressure to become evolvable in many domains. This insight is important because it may help explain the tendency to see many "brittle" evolved solutions, and in general low evolvability in artificial evolution, even with developmental systems.

The results highlight several questions about evolvability:

- Can evolvability be used to counteract the effects of convergence? For example, although the GRN encoding has high acquired evolvability, when comparing a population that has been undergoing training for 150 generations to a randomly initialized population, the random population seems to always be more efficient at solving the problem. The most likely explanation is because the trained population has converged somewhat during evolution.

- How broadly can the measure proposed in this paper be applied to other tasks? The test for evolvability requires a fitness function that can be changed over time, in order to provide a variable amount of information about some invariant properties. What kinds of domains are amenable to such variation? One interesting direction would be to use coevolution as an adaptive benchmark, as it allows for theoretically unbounded increases in acquired evolvability.

- What factors influence acquired evolvability? For example, why does the version of the modified direct encoding with random developmental variance exhibit

no acquired evolvability? Both versions are the same genetically and thus have the same latent evolvability, however only the standard version acquires evolvability during training. The only difference is that "correct" developmental variance allows the linkage parameter to directly influence fitness.

Finally, there are three points that need to be addressed to improve the test procedure. First, work must be done to explore the relationship between acquired and latent evolvability. Is it sufficient just to repeat the test procedure on several domains, and average acquired evolvability? Second, a metric needs to be established to take into account the effect of training time on acquired evolvability. Does evolvability saturate as training time increases? Some encodings may become evolvable more slowly than others, but may have a greater capacity to do so. Each encoding should be tested not only on a range of target drift rate settings, but also on a range of training period lengths. Third, some context must be imposed on the relative result scale. What exactly does 33 times more evolvability mean, in practice? Is it significant, i.e., does it impact long-term evolution?

7. CONCLUSION

Since the direct representation exhibits no significant variance in efficiency while the GRN and modified direct encodings do, the test proposed in this paper correctly contrasts relative acquired evolvability. Furthermore, this test elucidates many factors that influence evolvability, including the length of the training period, the genetic control of mutation effects (linkage parameter and GRN encoding), and the initial complexity of the representation (GRN initial cis region length). Only acquired evolvability in the general symmetry domain has been measured so far. In order to approximate a representation's latent evolvability, a broad range of test problems must be employed. In spite of this caveat, the acquired evolvability test provides insight into evolvability, and raises many interesting questions for future inquiry.

8. REFERENCES

[1] A.-L. Barabási and Z. N. Oltvai. Network biology: Understanding the cell's functional organization. *Nature Reviews Genetics*, 5:101–113, 2004.

[2] A. Bergman and M. L. Siegal. Evolutionary capacitance as a general feature of complex gene networks. *Nature*, 424:549–552, 2003.

[3] J. C. Bongard. Evolving modular genetic regulatory networks. In *Proc. of CEC 2002*, pages 1872–1877. IEEE Press, 2002.

[4] M. Ebner, M. Shackleton, and R. Shipman. How neutral networks influence evolvability. *Complexity*, 7(2):19–33, 2001.

[5] P. Eggenberger. Evolving morphologies of simulated 3d organisms based on differential gene expression. *Proc. of the 4th European Conf. on Artificial Life*, pages 205–213, 1997.

[6] W. Fontana. Modeling 'evo-devo' with rna. *BioEssays*, 24:1164–1177, 2002.

[7] F. Gomez and R. Miikkulainen. Incremental evolution of complex general behavior. *Adaptive Behavior*, 5:317–342, 1997.

[8] T. F. Hansen. Is modularity necessary for evolvability? remarks on the relationship between pleiotropy and evolvability. *BioSystems*, 69:83–94, 2002.

[9] J. Hermisson and G. Wagner. The population genetic theory of hidden variance and genetic robustness. *Genetics*, 168:2271–2284, 2004.

[10] P. Hogeweg. Evolving mechanisms of morphogenesis: On the interplay between differential adhesion and cell differentiation. *Jour. of Theo. Bio.*, 203:317–333, 2000.

[11] H. Lipson, J. Pollack, and N. Suh. On the origin of modular variation. *Evolution*, 56(8):1549–1556, 2002.

[12] J. Maynard-Smith, R. Burian, S. Kauffman, P. Alberch, J. Campbell, B. Goodwin, R. Lande, D. Raup, and L. Wolpert. Developmental constraints and evolution. *The Quarterly Review of Biology*, 60(3):265–287, 1985.

[13] R. A. Raff. *The Shape of Life: Genes, development, and the Evolution of Animal Form*. The University of Chicago Press, 1996.

[14] R. A. Raff and B. J. Sly. Modularity and dissociation in the evolution of gene expression territories in development. *Evolution and Development*, 2(2):102–113, 2000.

[15] I. Rechenberg. *Evolutionsstrategie '94*. Friedrich Frommann Verlag, 1994.

[16] J. Smith and T. Fogarty. Operator and parameter adaption in genetic algorithms. *Soft Computing*, 1:81–87, 1997.

[17] B. M. R. Stadler, P. F. Stadler, G. Wagner, and W. Fontana. The topology of the possible: Formal spaces underlying patterns of evolutionary change. *Journal of Theoretical Biology*, 213:241–274, 2001.

[18] K. O. Stanley and R. Miikkulainen. Evolving neural networks through augmenting topologies. *Evolutionary Computation*, 10(2):99–127, 2002.

[19] K. O. Stanley and R. Miikkulainen. A taxonomy for artificial embryogeny. *Artif. Life*, 9(2):93–130, 2003.

[20] C. H. Waddington. Canalization of development and the inheritance of acquired characters. *Nature*, 150:563, 1942.

[21] G. Wagner and L. Altenberg. Complex adaptations and the evolution of evolvability. *Evolution*, 50(3):967–976, 1996.

[22] C.-H. Yuh and E. H. Davidson. Modular cis-regulatory organization of endo16, a gut-specific gene of the sea urchin embryo. *Development*, 122:1069–1082, 1996.

APPENDIX

For all algorithms, population size is 50 and target string length is 40 bits. Selection is through a binary tournament. All parameters are encoded as doubles or unsigned integers, and crossover occurs only at gene boundaries.

Direct and modified direct have a 1% chance per gene per generation of being mutated. Modified direct genomes are evaluated 5 times with the fitness averaged. The linkage similarity tolerance is 150.

GRN fixed parameters: 10% weight mutation (Gaussian, variance = 1.0), 0.5% add new character, and 5% duplicate gene rate. The GRN model iterates for 10 steps (fixed), each tf concentration is decayed 10% per step, and there is no limit to the number of tfs. The tolerance parameter is evolvable, initially 1500 (750 for lowertol). The initial number of genes is 40 (each gene has a random promoter and a fixed environmental tf).