Just Use Distributions: Eliminating The Noise of Sampling-Based Fitness Estimation

Robert McKay [1,1], zhenyue Qin [2], and Tom Gedeon [2]

[1]Australian National University
[2]Affiliation not available

November 8, 2023

Posted on 5 Oct 2020 — CC-BY-NC-SA 4 — https://doi.org/10.36227/techrxiv.13042289.v2 — e-Prints posted on TechRxiv are preliminary reports that are not peer reviewed. They should not b...

Abstract

This paper uses distributional analysis to disentangle the effects of fitness landscape and stochasticity in Wagner's Gene Regulatory Network Based evolutionary scenario. Further advantages of the analysis include the ability to accurately identify global optima and potential extensions to approximate the distributional fitness more accurately than stochastic sampling at lower computational cost.

This is an inital draft, and will be updated as further content is written.

Just Use Distributions: Eliminating The Noise of Sampling-Based Fitness Estimation

Zhenyue Qin, Tom Gedeon, R.I. (Bob) McKay

***Abstract*—Discrete gene regulatory networks (GRNs) play a vital role in the study of robustness and modularity. A common method of evaluating the robustness of GRNs is to measure their ability to regulate a set of perturbed gene activation patterns back to their unperturbed forms. Usually, perturbations are obtained by collecting random samples produced by a predefined distribution of gene activation patterns. This sampling method introduces stochasticity, in turn inducing dynamicity. This dynamicity is imposed on top of an already complex fitness landscape. So where sampling is used, it is important to understand which effects arise from the structure of the fitness landscape, and which arise from the dynamicity imposed on it. Stochasticity of the fitness function also causes difficulties in reproducibility, analyticity, and other post-experimental analyses.**

In this paper, we develop a deterministic distributional fitness evaluation by considering the complete distribution of gene activity patterns, so as to avoid stochasticity in fitness assessment. This fitness evaluation facilitates repeatability. Its determinism permits us to determine theoretical bounds on the fitness, and thus to determine whether the algorithm has reached a global optimum. We also unveil some properties of desirable GRNs that lead them to be robust and modular. We conclude by discussing a number of obscure phenomena remaining to be understood in the future.

I. Introduction

II. Related Work and Motivation

In this section, we present a brief overview of the emergence of modularity, of GRNs and of Espinosa-Soto and Wagner's GRN Model [1]. Then, we introduce the motivation of this paper.

Judging by citations, of explanations for the emergence of modularity, the parsimony-based model of modularity emergence of Clune et al. [2] has recently been dominant. There is little doubt of its importance for evolving modular engineering systems, but its relevance to biology is moot. Parsimony pressures have long been studied in genetic programming [3], yet are notoriously difficult to tune – too strong a pressure (relative to the primary objective) and one is left with tiny but highly unfit solutions; too weak, and complexity runs riot. In biology, as Clune et al. argue, modularity is ubiquitous. Yet there is no obvious mechanism to tune the many different parsimony pressures required to explain its widespread emergence. Clune et a. [2] avoid this through use of the highly engineered NSGA-II multi-objective evolutionary algorithm [4], which is perfectly fine for engineering modularity in artificial systems, but questionable as a model of evolution based on natural selection(NSGA-II uses population-wide computations that would require a 'hidden hand' in biological systems). Thus at least in understanding biology, it is worth re-examining alternative explanations.

All authors are with the Research School of Computer Science, Australian National University, Canberra, ACT 2600, Australia (e-mails: zhenyue.qin@anu.edu.au, tom@cs.anu.edu.au, robert.mckay@anu.edu.au).

Espinosa-Soto and Wagner [1] traced the emergence of modularity to specialisation. Their work is based on Wagner's GRN model, which has witnessed wide application in a variety of computational biological studies [5]–[7]. It is important to acknowledge that their system also included a parsimony mechanism, but in the mutation operator, not as a second objective. This parsimony mechanism is not sufficient on its own to generate modularity, specialisation is required; equally important, the parsimony of the mutation operator requires little tuning, a wide range of values suffice.

A. Gene Regulatory Network (GRN)

A GRN is a collection of molecular regulators that coordinate interactions between genes (including both the protein-coding DNA sequences and regulatory non-coding DNA sequences), RNAs and proteins [8]. GRNs are central to the operation of all known forms of cellular life (eukaryota, bacteria and archaea [9]) and viruses [10]. The network structure of GRNs demonstrates a high level of modularity, considered to be a key contributor to robustness [11].

B. Espinsa-Soto and Wagner's GRN Model

The model of [1] abstracts cellular homeostasis, in which a cell can recover from small perturbations to a target state and recover that state. It consists of two components, one or more targets, and the GRN itself. A target is a set of gene activation patterns, represented by a vector of N binary values, with $+1$ and -1 respectively representing activity and inactivity. Figure 1 depicts two target activation patterns consisting of 10 genes that happen to have a modular structure of two modules of five genes each; the activation in the first module of the patterns is identical (shared), but opposite in the second (divergent).

The gene state is regulated by a GRN g, which controls the activation pathway of the organism. For a pattern of N genes, it is abstracted as a ternary N^2 transition matrix $g = [g_{ji}]$ with entries over $\{-1, 0, 1\}$, representing repression, independence or activation of gene i by gene j. A gene activity pattern regulated by this network is a Boolean row vector $s = [s^0, ..., s^{N-1}]$. The state transition is modelled by:

$$A(g, s) = \sigma[g.s] \tag{1}$$

where $\sigma(x) = 1$ if $x > 0$, $\sigma(x) = -1$ otherwise (applied elementwise).

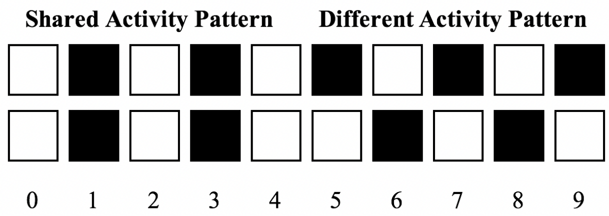


Fig. 1: A target consisting of two gene activation patterns, where white and black squares represent active (+1) and inactive (−1) genes

The model focuses on the evolution of the $N \times N$ GRN matrix, generally by an evolutionary algorithm. This can lead to a terminological confusion. In the modeled biology, there are N genes in the activation pattern; but considered at the evolutionary algorithm level, the evolving chromosome consists of N^2 genes. Where we need to distinguish these, we refer to the former as a "pattern gene", and the latter as a "network gene" or "network node". Figure 2 presents a flow chart of the fitness evaluation in the model.

In the rest of this paper, we will concentrate on the work of [1] with the target of figure 1, where an evolutionary system evolved first for 500 generations with a target consisting solely of the first activation pattern, and then for the remainder of the time (1500 further generations) with both activation patterns as target.

C. Anomalous Behaviours of the GRN Model and Our Motivation

Our initial work on this GRN model under typical genetic algorithm settings revealed a number of anomalies [12]. In summary, despite relatively fit, modular GRNs emerging in simulated evolutions, they could often be readily improved in both vitness and modularity by manually removing all inter-module connections, as in Figure 3. Yet evolutionary search does not find these improvements, despite mutation biases appearing to favour finding them. Figure 4 reveals that this is not due to discontinuous gradients: starting with the most robust/fittest GRN from the final generation of a typical run and removing non-modular edges one-by-one reveals a path of steadily improving fitness to a fully modular GRN. This phenomenon occurs even in runs incorporating elitism.

III. Distributional Fitness Evaluation

We hypothesised that the anomalies discussed in Section II-C might arise from the stochasticity of the sampling process of [1]: that in a population converged close to a local optimum, using order-based (tournament) selection, small stochastic variations in fitness might make it difficult to follow weak gradients. To evaluate this hypothesis, we need to separate the effects of the underlying fitness landscape from the effects of stochastic sampling. Fortunately, this is not hard to do, both in principle, and in this case, in practice.

In common with other GRN robustness models [13]–[15], Espinosa-Sot & Wagner[1] sample perturbations stochastically,

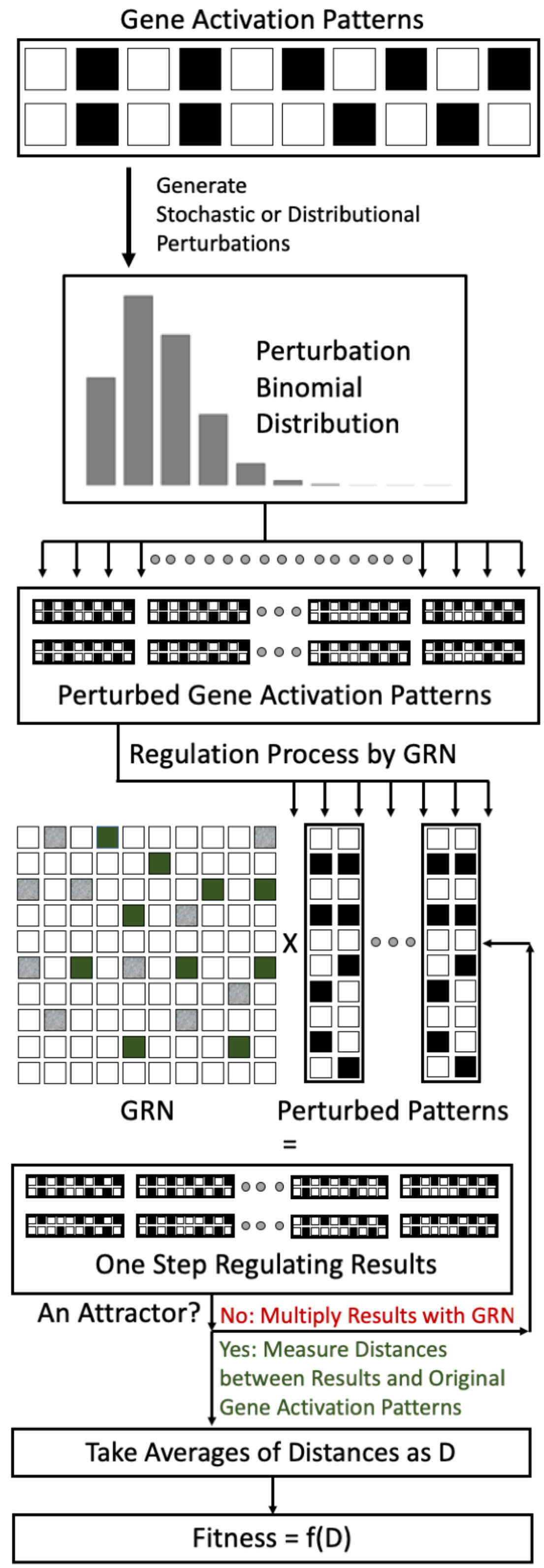
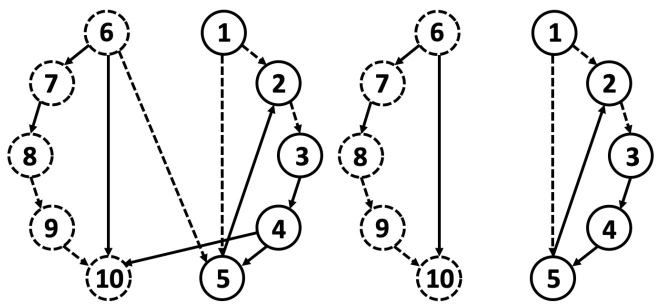
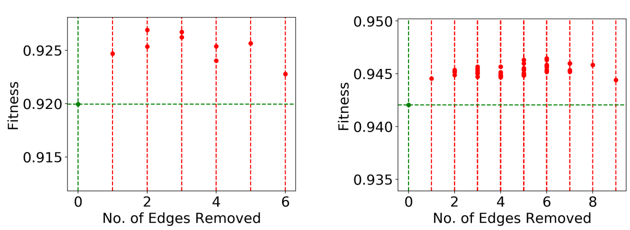


Fig. 2: Flow Chart of the Fitness Evaluation in [1]'s GRN model. Dark untextured squares represent activation, grey texture repression and white lack of influence.



(a) Before Removal (b) After Removal

Fig. 3: Illustration of deterministically removing all the inter-module connections of a GRN. Solid and dashed circles represent different modules. Solid and dashed arrows stand for activation and repression respectively.



(a) With Elitism (b) Without Elitism

Fig. 4: Fitness increases continuously while removing inter-module connections from a fit GRN. Green dots represent the initial fitnesses, and red dots show the resulting fitnesses from inter-module connection removal.

then study the recovery of the original pattern. They use a binomial model: 500 perturbations of the locations in the pattern are identically and independently sampled, with a probability of being perturbed of $p = 0.15$, the recovery by each GRN genrates a reward based on the level of recovery, then the reward is averaged over the sampled perturbations. Thus we can compute the expected fitness of a GRN by tracing its behaviour over all 1024 possible perturbations, and weighting appropriately. This produces a deterministic fitness metric, and at a computational cost $1024/500$ (i.e. roughly double) that of [1]. We call this method distributional fitness evaluation. The underlying idea is extensible beryond discrete GRNs to a wide range of computational studies of discrete netwroks, including other genetic networks [16], Boolean neural networks [17], and Hopfield networks [18].

A. *Definition and Advantages of Distributional Fitness Evaluation*

Partially following the ideas of [1], we can extend the (one step) action $A(g, s)$ of an $N \times N$ GRN g on an activation state s of length N of equation 1 to its recursive application as

$$A^0(g, s) = s$$
$$A^{t+1}(g, s) = A(g, A^t(g, s))$$

We define an elementary perturbation e of length N as a vector of $\{-1, 1\}$, so that a perturbation of a target state s (also of length N) in the sense of [1] is the pairwise product $e \odot s$. Following Boolean usage, the weight $w(e)$ of an elementary perturbation is the number of -1 values.

[1] follows the regulatory process for $t_0 = 20$ steps:

$$G(g, e \odot s) = \begin{cases} s & \text{if } A^t(g, e \odot s) = s \text{ for } t < t_0 \\ A^{t_0}(g, e \odot s) & \text{otherwise} \end{cases} \tag{2}$$

and uses two auxiliary functions to weight contributions:

$$f(g) = 1 - e^{(-3 \cdot g)}$$
$$\gamma(x) = (1 - x)^5$$

Putting it together, we evaluate the effectiveness of GRN g in recovering state s as

$$F(g, s) = f\left(\sum_{n=0}^{N} p_n \cdot \frac{1}{|E_n|} \sum_{e \in E_n} \gamma\Big(H\big(G(g, e \odot s), s\big)\Big)\right) \tag{3}$$

where E_n is the set of elementary perturbations of length N and weight n, p_n is the probability $p_n \sim B(N, p)$ from the binomial distribution, and H is the Hamming distance.

Compared with the random sampling of perturbations of [1], distributional fitness evaluation offers the following advantages:

1) Determinism: evaluating the fitness of a GRN multiple times will give the same fitness each time, while preserving essentially the same fitness landscape as sampling. This allows us to disentangle the effects of the fitness landscape itself, and that of noise.
2) Global Optimum Analysis: we can determine bounds on the fitness, and test whether those bounds are achieved; with a stochastically evaluated fitness, it is infeasible to determine whether further improvement is possible.
3) Speed Optimisation: Due to the determinism of fitness evaluation, cacheing of previously computed fitness may reduce wasted computation.

B. *Upper Bounds on Distributional Fitness*

An obvious upper bound for the fitness of a GRN in the scenario of [1] is for the GRN to return all perturbations of a target to the corresponding target. When there is only a single target, this is attainable (as we shall see), and for the 10×10 GRNs that are the focus of this study, is readily evolved. However we follow [1] in using a two-stage evolution, in the second stage of which there are two targets. In this case, this upper bound is not attainable. It is easy to see why. Consider the first target of Figure 1. If it is perturbed by an elementary perturbation of weight 0 (i.e. it is unperturbed), then we would expect a GRN to readily recover it (it isn't required to do anything). But consider the second target perturbed by an elementary perturbation whose first five locations are 1, with the last five being -1. The resulting perturbed target is identical to the previous one. Hence the GRN must map it to the same end result: which is a Hamming distance of 5 from its target state. In fact, for every perturbation of target 1, there is a perturbation of target 2 that gives the same starting state for the GRN. At most one of them can be returned to its target by the GRN. Which choice is best?

To answer this question, consider the binomial probabilities of elementary perturbations by weight: to two decimal places: $< 0.20, 0.35, 0.28, 0.13, 0.04, 0.01, 0.00, 0.00, 0.00, 0.00, 0.00 >$. So perturbations of weights 0, 1 and 2 carry the most influence in equation 3, and for higher weights the influence decreases monotonically. Now for two elementary perturbations to conflict in this way, they must be identical in the first five places (since the two targets are identical there), and inverse on the last five (since the two targets are inverse there). So the choice is easy: the elementary perturbation with the least weight (therefore 0, 1 or 2) in the second half should be mapped back to the corresponding target. In this case, the Hamming distance between the regulated and original patterns is 0, and the result of function γ in Formula 3 is 1. Conversely, for second-half weights of 3, 4 or 5, the GRN will regulate the pattern to the opposite form. In such cases, the Hamming distance is 5 and γ returns 0.03125.

In summary, specialising equation 3 to the targets s_1 and s_2 of Figure 1 gives

$$\begin{aligned} &F(g) = \\ &f\Bigg(\sum_{n=0}^{10} B(n; 10, 0.15)\cdot \\ &\frac{1}{\binom{10}{n}} \sum_{e \in E_n : w_{6,10}(e) < 3} \gamma\Bigg(H\big(G(g, e \odot s_1), s_1\big) \\ &+ H\big(G(g, e \odot s_2), s_2\big)\Bigg)\Bigg) \end{aligned} \tag{4}$$

where $w_{6,10}(e)$ denotes the weight of (number of -1's in) the second half of e.

Substituting values 1 and 0.03125 into Equation 4 and rearranging gives

$$\begin{aligned} &F(g) = f\Bigg(\sum_{n=0}^{10} B(n; 10, 0.15)\cdot \\ &\binom{10}{n} \cdot \Bigg(\sum_{e : w_{6,10}(e) < 3} \cdot 1 + \sum_{e : w_{6,10}(e) > 2} \cdot 0.03125\Bigg)\Bigg) \end{aligned} \tag{5}$$

Table I summarises how many elementary perturbations of each total weight have second half weights < 3 (resp. > 4). Substituting its values into Equation 5 gives a fitness bound to four decimal places of 0.9462.

C. Comparing Runs

The treatments examined in this paper use a mixture of stochastic and distributional evaluation. How can we fairly compare them? In the long run, if the same individual is repeatedly re-evaluated using stochastic evaluation, the mean fitness must converge to the distributional fitness, in any particular case the stochastically evaluated fitness may be above or below the distributional fitness of the same individual. This would just induce noise in any comparisons (itself undesirable). However there is a further complication. We typically wish to compare the end-of-run best fitness achieved, which introduces a bias: even if the same individual is produced as that best individual: since it was chosen as the best individual in the generation, in a stochastic evaluation run its fitness is more likely to have been stochastically evaluated in the upper part of the fitness distribution. To elminate this problem, for all comparisons, in all tables, and in all figures, except where explicitly mentioned, we always present the distributionally-evaluated fitness for an individual, even if it is evolutionarily evaluated using stochastic evaluation. This has the effect that for stochastic evaluation runs, figures showing the course of evoluation are not actually showing the fitness that was used in the evolution. The effect on a specific run can potentially be substantial. However our results are typically averaged over 100 runs for a treatment. In this case, the stochastic variations should largely cancel out. .

TABLE I: Numbers of Unrecoverable Elementarty Perturbations by Weight.

Weight	No. of Perturbations	Unrecoverable
0	$\binom{10}{0}$	0
1	$\binom{10}{1}$	0
2	$\binom{10}{2}$	0
3	$\binom{10}{3}$	$\binom{5}{3}$
4	$\binom{10}{4}$	$\binom{5}{3}\cdot\binom{5}{1}+\binom{5}{4}$
5	$\binom{10}{5}$	$\binom{5}{3}\cdot\binom{5}{2}+\binom{5}{4}\cdot\binom{5}{1}+\binom{5}{5}$
6	$\binom{10}{6}$	$\binom{5}{3}\cdot\binom{5}{3}+\binom{5}{4}\cdot\binom{5}{2}+\binom{5}{5}\cdot\binom{5}{1}$
7	$\binom{10}{7}$	$\binom{5}{3}\cdot\binom{5}{4}+\binom{5}{4}\cdot\binom{5}{3}+\binom{5}{5}\cdot\binom{5}{2}$
8	$\binom{10}{8}$	$\binom{10}{8}$
9	$\binom{10}{9}$	$\binom{10}{9}$
10	$\binom{10}{10}$	$\binom{10}{10}$

D. Fitness Structure

IV. Distributional GRN Model

In this section, we present D-GRN, a distributional framework for Wagner's GRN model under distributional fitness evaluation, and briefly delineate the mutation and recombination operators.

A. Algorithmic Framework of D-GRN

Algorithm 1 presents a high-level structure for a typical haploid genetic algorithm; however the mutation and recombination operators follow those of [1].

B. Recombination

In [12], we introduced and experimentally investigated diagonal recombination. Given two parental GRNs $A_1[1, \ldots, 10]$ and $A_2[1, \ldots, 10]$, diagonal recombination proceeds by first sampling a pivot point i from $\{1, \ldots, 10\}$, then preserving the two sub-matrices $A_1[1 \ldots i-1, 1 \ldots i-1]$ and

Algorithm 1: Algorithm Framework of D-GRN

Input: algorithm hyperparameters
population size Π of size
mutation rate M of rate (capital μ)
crossover rate X of rate (capital χ)
generations Γ of number
per-location perturbation rate E of rate (capital ϵ)
asymmetric stepsize Z of size (capital ζ)
evolutionary tournament size T of size (capital τ)
Output: robust networks
// Initialisation
crossover size $N_\chi = X.\Pi/2$;
generate Π feasible networks randomly;
save generated individuals in the population P;
get distributional fitness of each individual $I_i \in P$;
// Loop until terminating condition
for $i = 1$ *to* Γ **do**
    Reset $P' = \Phi$;
    // Crossover
    for $j = 1$ *to* N_χ **do**
        for $l = 1$ *to 2* **do**
            for $k = 1$ *to* T **do**
                uniformly sample individual $I_{j,l,k} \sim P$;
            select $I_{j,l} = \text{argmax}_{k \in \{1,\ldots,\text{T}\}} F(I_{j,l,k})$;
        generate $I_{j,3}$ and $I_{j,4}$ from $I_{j,1}$ and $I_{j,2}$ by diagonal recombination;
        $P' = P' \bigcup \{I_{j,3}, I_{j,4}\}$;
    // Copying
    for $j = 2N_\chi + 1$ *to* Π **do**
        uniformly sample individual $I_j \sim P$;
        $P' = P' \bigcup \{I_j\}$;
    // Mutation
    foreach *individual* I_i *in* P' **do**
        uniformly sample real $r \sim [0, 1)$;
        if $r < M$ **then**
            biasedly mutate I_i;
    // Updating
    get distributional fitness of each individual $I_i \in P'$;
    update $P = P'$;

$A_1[i \ldots 10, i \ldots 10]$, while exchanging the remainder of corresponding locations between A_1 and A_2. This is illustrated in figure 5. We note that if i is sampled as 1, the corresponding recombination is a null operation.

C. Mutation

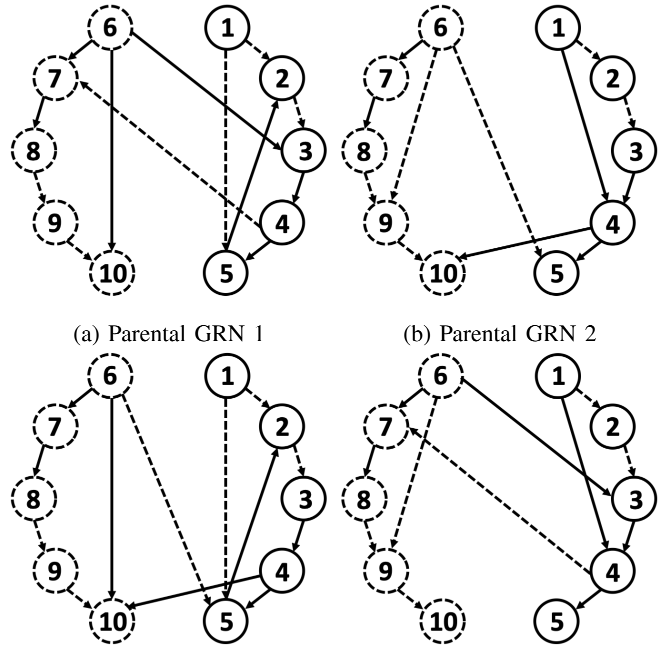
The mutation operator of [1] biases the edge density to a specific, relatively low, level. A node in the network has a probability μ to mutate every generation; if it mutates, it can either lose or gain an interaction. In matrix terms, the probability for each row to have a changed value is μ (corresponding in the case of size 10 targets with $\mu = 0.2$ to a per-individual mutation rate, i.e. sampling a nonzero value from B(10,0.2), of approximately 0.9). The probability for a node to lose an interaction (a nonzero value to change to zero) is defined as

$$p(u) = \frac{4r_u}{4r_u + N - r_u} \tag{6}$$

where N is the number of genes in a gene activation pattern of a target, and r_u is the number of regulators of gene u [1], i.e. nonzero values in column u. Conversely, the probability for a gene u to gain an interaction (i.e. for a nonzero value in row u to become nonzero) is $1 - p(u)$. The neutral point of this bias can be computed as:

$$p(u) = 1 - p(u) \Rightarrow r_u = \frac{N}{5}$$

The bias acts to maintain the sparsity of the network at around this value, which research in computational biology suggests is essential to induce modularity [19].



(a) Parental GRN 1 (b) Parental GRN 2
(c) Offspring GRN 1 (d) Offspring GRN 2

Fig. 5: Illustration of the diagonal recombination, with the pivot index as 5. Solid and dashed circles represent different modules. Solid and dashed arrows stand for activation and repression respectively.

V. Experimental Setup

All simulation code was implemented in Java 1.8.0 and Python 2.7.10. All programs are publicly available at (to come). We use the Mann-Whitney significance test in all comparisons.

A. Parameter Tables and Explanations

The specific targets T used in the experiments appear in Table II. Evolutionary and other simulation parameters for the main body of experiments are specified in Table III and explained in Table IV. A few variations on these will be specified in the relevant context.

TABLE II: Gene Activity Patterns (Target)

Target Pattern	Introduction Stage
+1 -1 +1 -1 +1 -1 +1 -1 +1 -1	0
+1 -1 +1 -1 +1 +1 -1 +1 -1 +1	500

TABLE III: Parameters of the Evolutionary Simulations

Pattern Size 10	GRN Size 100	Initial Density 0.2
# Perturbations 2^N or 500	Perturbation Rate 0.15	Population Size 100
Mutation Rate 0.2	Activation Rate 0.5	Crossover Rate 0.2
Crossover Type Diagonal	Selection (Size) Tournament (3)	Reproduction Rate 0
Max. Generation 2000	Trials per Treatment 100	Significance Test Mann Whitney

B. Population Initialisation

A population in our simulated evolution consists of 100 individuals, each being a GRN as defined in Eq 1. During initialisation, each individual in the population will randomly generate 20 edges with arbitrary directions for its GRN. The choice of number 20 comes from Eq. 6 that biases towards sparse networks as indicated in Subsection IV-C, Specifically, Eq. 6 shows each gene has 2 regulators and there are 10 genes, therefore the total edge number will be biased towards 20 in a GRN. Nonetheless, since the edge number lies in the range of 0 to 100 (the edges are directional), the probability mass function of edge numbers within a GRN, although it will have a peak at $p(e = 20)$, where p stands for probability and e represents edge number, the function curve will skew to the right. Formally,

$$\sum_{e=0}^{20} p(e) < \sum_{e=21}^{100} p(e)$$

TABLE IV: Explanations of simulation parameters

Target Pattern	pattern to be perturbed then recovered
Introduction Stage	generations where target is introduced
Pattern Size	N, number of locations in an activation pattern
GRN Size	$N \times N$, the size of each GRN (evolved genotype)
Initial Density	initial density of edges in the GRN
# Perturbations	number of perturbations of each target[a]
Perturbation Rate	expected proportion of corrupted genes
Population Size	the number of individuals in the population
Mutation Rate	probability GRN node gains/loses an interaction[b]
Activation Rate	proportion of new interactions that are activations[c]
Crossover Rate	proportion of individuals that are crossed over[d]
Crossover Type	the tupe of crossover (recombination) used
Selection (size)	the type of selection and size when relevant
Reproduction Rate	proportion of old generation randomly copied on
Max. generation	the generation when the simulation will terminate
Trials per Treatment	number of trials for comparing treatments
Significance Test	statistical test used in comparing treatments

a For distributional evaluation, this will be 2^N.
b For compatibility with the terminology of [1]. In EC terms: the per-gene mutation rate in the evolving GRN is $1/N$ of this the per-individual mutation rate is N times this
c Gained interactions are either activations or repressions.
d Some crossovers may be ineffective, see subsection IV-B.

As a result, the expected number of edges within a GRN may not be exactly 20, despite the bias. We conducted experiments in which we did not apply selection pressure during the simulated evolution; the average GRN edge number converged to approximately 22.

C. Selection Scheme: Tournament vs Proportional Selections

In evolutionary computation, there are two prevalent selection schemes: tournament and proportional. Although [1] uses proportional, we use tournament in this paper because it gives similar results to proportional selection, but is better suited to incremental evaluation, a strategy we plan to use in extending this work to larger targets.

D. Modularity Metric

We adopt the normalised Q scoring system to quantify modularity in a GRN, ultimately based on the definition proposed by Newman [20]. Briefly, this quantity is defined as the difference between, on the one hand, the ratio of the number of edges in the network connecting nodes within modules to the total number of edges, and on the other, the expected value for the same quantity for a randomly connected network with the same edge density [21]. Formally, Q is calculated as

$$Q = \sum_{i=1}^{K} [\frac{l_i}{L} - (\frac{d_i}{2L})^2] \tag{7}$$

where i represents one of the K potential modules within a network, L is the total number of connections in a network, l_i stands for the number of interactions in the module i, and d_i is the sum of degrees of all the nodes in module i [1]. The value Q will lie in the range of $[-0.5, 0.5]$ for our module structure.

Unfortunately the expected value of Q varies according to the density of edges. In order to eliminate the effects of variations in total edge numbers within a GRN, and provide fair comparisons, we normalise Q as in Equation 8, following the spirit of [21].

$$Q_n = \frac{Q - Q_{ran}}{Q_{max} - Q_{ran}} \tag{8}$$

where Q is the modularity Q value obtained from Equation 7 for the specific network $\{V, E\}$, Q_{ran} is the average Q value of 10,000 random networks with the same number of vertices V and edges E as the network $\{V, E\}$, and Q_{max} stands for the maximum Q value in these 10,000 random networks. This normalised Q_n shows us how modular our network is by comparing it to sampled highly modular and random networks with the same attributes [1].

VI. Experimental Results and Analysis: Stochastic vs Deterministic Evaluation

In this set of experiments, we compare the behaviour of the algorithm using stochastic evaluation with that using distributional evaluation. From the perspective of the particular domain, this helps to understand to what extent aspects of the behaviour (for example, emergence or non-emergence of modularity) are a consequence of the fitness landscape of the problem, and to what extent they derive from the noise effects of random sampling imposed upon that landscape. From an evolutionary biology perspective, they allow us to compare the behaviour of small population and perturbation sample sizes (computationally tractable but generally biologically implausible) against the smoothing effect of effectively infinite perturbation samples. Because a key effect of increasing either population size or perturbation sample size is to smooth behaviour, this can help to gain some insight into what we might expect from more biologically realistic (but computationally infeasible) population sizes. Finally, from a methodological perspective, this section illustrates what is feasible if the distribution underlying the noise is small enough to be directly computable (in this case, the GRN target is sufficiently short). Without further theoretical advances, distributional evaluation would not be computationally tractable if the target, and thus the distribution, was much larger.

We conducted 100 independent evolutionary simulations for distributional and for stochastic fitness evaluation. We collected the fittest GRNs in each generation and evaluated their fitnesses and modularities. We remind readers that these results are reported using the distributional fitness, even for runs using stochastic evaluation. An important consequence is that we know, even for stochastic runs, whether we have actually found a true optimum.

A. Effects of Stochastic Evaluation on Evolutionary Efficiency

TABLE V: Mean Best Fitnesses and Modularity Q Values of Fittest GRNs (over 100 runs) from the Final Generation, for Distributional vs Stochastic Fitness Evaluation

	Distributional		Stochastic		p-value
	Mean	SD	Mean	SD	
Fitness	0.9395	0.0183	0.9370	0.0202	1.04×10^{-12}
Distributional Equivalent			0.9256		
Modularity	0.9785	0.2840	0.9513	0.3011	0.5016

1) Results: Table V shows the outcomes for fitness and modularity in the final generation, while figure 6 shows the evolution of fitness and modularity over the generations.

Distributional fitness evaluation leads to a tiny, but statistically highly significant, increase in final-generation fitness when compared with stochastic evaluation. Thus, at least with these evolutionary settings, the dynamicity imposed by stochastic evaluation has a deleterious effect. The fitness effect is mirrored by a small, but non-significant, improvement in modularity. Distributional evaluation achieves a (distributional) fitness only 0.0067 below the theoretical bound of 0.9462, while distributional fitness evaluation is 0.0206 below. Thus while the fitness difference is small, the difference from the attainable optimum is a factor of three larger. For completeness, in the table we also show the fitness that the stochastic best individuals recorded using stochastic evaluation (in fact the difference is small). Figure 6 shows that fitness and modularity evolve very similarly for the two evaluation methods up to the change of target. Subsequently, the fitness graphs remain similar (although stochastic fitness is always slightly below distributional), but the modularity from stochastic evaluation increases more slowly after the target change than the distributional. It is tempting to speculate that the latter might result from reduced search eagerness due to the stochasticity of the fitness function, but if that is the reason, it is surprising that the effect on fitness is not more pronounced (in fact, it seems imperceptible).

B. Deterministic Fitness Evaluation with Complete Sampling Can Help Better Analyse GRN Edge Functions

VII. CHARACTERISTICS OF THE GRN MODEL

VIII. DISCUSSION

IX. CONCLUSION

It is essential to identify characteristics of nature-inspired algorithms that lead to solutions of high modularity. Our expectation that such features exist relies heavily on the ubiquity of modular structures in biological evolution. In this paper, we studied one one well-accepted scenario that leads to the emergence of modularity in Wagner's GRN model, namely gene specialization. We showed that this problem can be better simulated using complete sampling of all perturbations of gene activation patterns instead of stochastical sampling. This enables the quantification of fitness upper bounds, and permits us to study under what conditions evolution attains the global optimum. Future studies will explore the functions evolved by network edges, particularly those connectiong between modules. The aim is to remove most inter-module edges spontaneously by evolutionary algorithms, to automatically obtain highly modular solutions.

REFERENCES

[1] C. Espinosa-Soto and A. Wagner, "Specialization can drive the evolution of modularity," *PLoS computational biology*, vol. 6, no. 3, p. e1000719, 2010.

[2] J. Clune, J.-B. Mouret, and H. Lipson, "The evolutionary origins of modularity," *Proc. R. Soc. B*, vol. 280, no. 1755, p. 20122863, 2013.

[3] B.-T. Zhang and H. Mühlenbein, "Balancing accuracy and parsimony in genetic programming," *Evolutionary Computation*, vol. 3, no. 1, pp. 17–38, 1995. [Online]. Available: https://doi.org/10.1162/evco.1995.3.1.17

[4] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: Nsga-ii," *IEEE Transactions on Evolutionary Computation*, vol. 6, no. 2, pp. 182–197, 2002.

[5] A. Bergman and M. L. Siegal, "Evolutionary capacitance as a general feature of complex gene networks," *Nature*, vol. 424, no. 6948, p. 549, 2003.

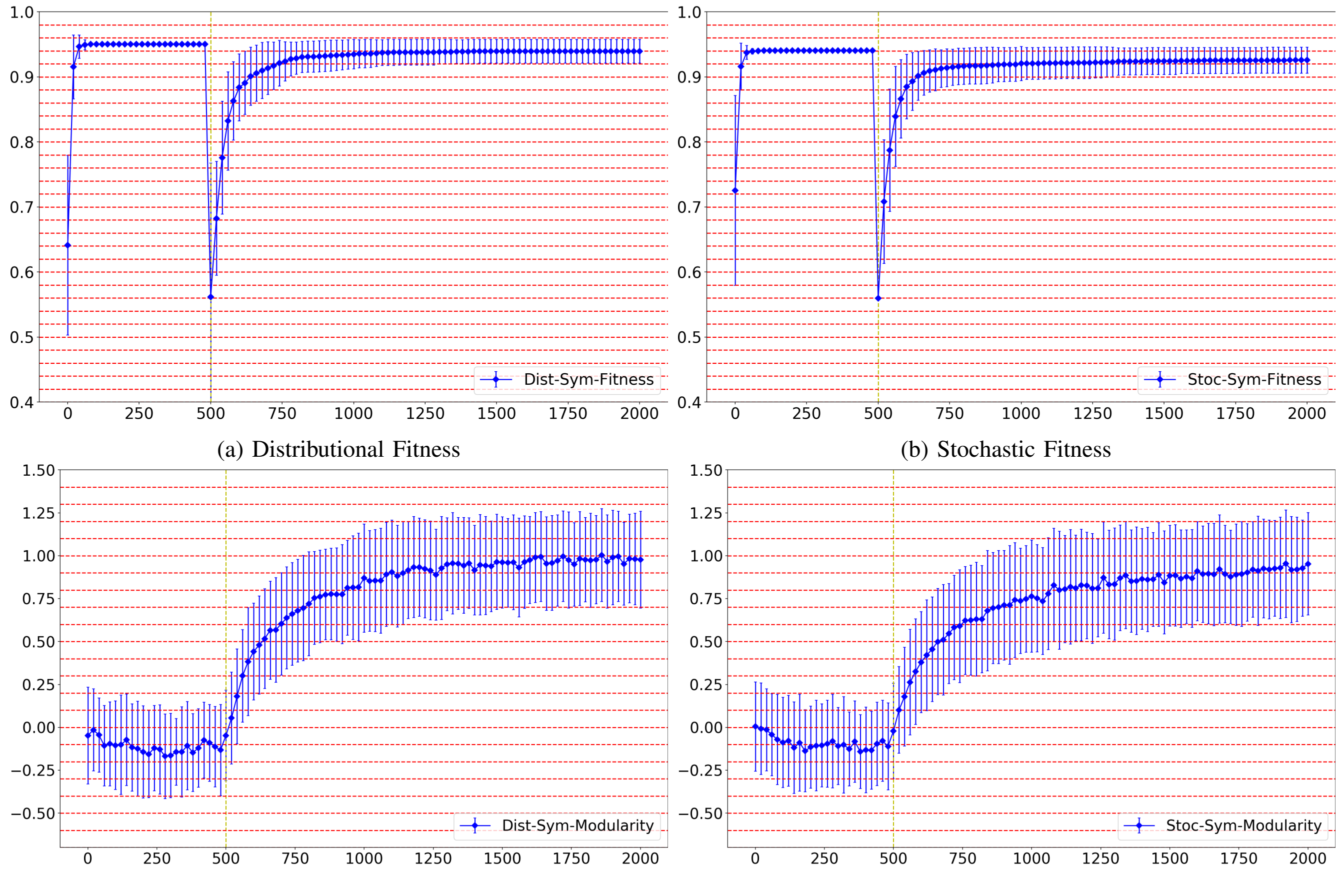
[6] R. B. Azevedo, R. Lohaus, S. Srinivasan, K. K. Dang, and C. L. Burch, "Sexual reproduction selects for robustness and negative epistasis in artificial gene networks," *Nature*, vol. 440, no. 7080, p. 87, 2006.

[7] E. Huerta-Sanchez and R. Durrett, "Wagner's canalization model," *Theoretical population biology*, vol. 71, no. 2, pp. 121–130, 2007.

[8] Y. Xiao, "A tutorial on analysis and simulation of boolean gene regulatory network models," *Current genomics*, vol. 10, no. 7, pp. 511–525, 2009.

[9] K. Nowick and L. Stubbs, "Lineage-specific transcription factors and the evolution of gene regulatory networks," *Briefings in Functional Genomics*, vol. 9, no. 1, pp. 65–78, 01 2010. [Online]. Available: https://doi.org/10.1093/bfgp/elp056

[10] A. Bensussen, C. Torres-Sosa, R. A. Gonzalez, and J. Díaz, "Dynamics of the gene regulatory network of hiv-1 and the role of viral non-coding rnas on latency reversion," *Frontiers in physiology*, vol. 9, pp. 1364–1364, Sep 2018, 30323768[pmid]. [Online]. Available: https://pubmed.ncbi.nlm.nih.gov/30323768



(a) Distributional Fitness
(b) Stochastic Fitness
(c) Distributional Modularity
(d) Stochastic Modularity

Fig. 6: Evolutionary Progress for each Generation, Means over 100 Trials. Left to Right: Distributional vs Stochastic Evaluation. Top: Fitness; Bottom: Modularity. Vertical bars represent one standard deviation. The vertical dashed line at generation 500 indicates addition of the second activation pattern.

[11] H. Kirsten and P. Hogeweg, "Evolution of networks for body plan patterning; interplay of modularity, robustness and evolvability," *PLoS computational biology*, vol. 7, no. 10, p. e1002208, 2011.

[12] Z. Qin, T. Gedeon, and R. B. McKay, "Anomalies in the behaviour of a modularity inducing problem domain," in *Artificial Life Conference Proceedings*. MIT Press, 2019, pp. 214–221.

[13] M. L. Siegal and A. Bergman, "Waddington's canalization revisited: developmental stability and evolution," *Proceedings of the National Academy of Sciences*, vol. 99, no. 16, pp. 10 528–10 532, 2002.

[14] J. Masel, "Genetic assimilation can occur in the absence of selection for the assimilating phenotype, suggesting a role for the canalization heuristic," *Journal of evolutionary biology*, vol. 17, no. 5, pp. 1106–1110, 2004.

[15] S. Ciliberti, O. C. Martin, and A. Wagner, "Robustness can evolve gradually in complex regulatory gene networks with varying topology," *PLoS computational biology*, vol. 3, no. 2, p. e15, 2007.

[16] S. Huang, "Gene expression profiling, genetic networks, and cellular states: an integrating concept for tumorigenesis and drug discovery," *Journal of molecular medicine*, vol. 77, no. 6, pp. 469–480, 1999.

[17] S. Bhide, N. John, and M. R. Kabuka, "A boolean neural network approach for the traveling salesman problem," *IEEE Transactions on Computers*, vol. 42, no. 10, pp. 1271–1278, 1993.

[18] K. Gurney, *An introduction to neural networks*. CRC press, 2014.

[19] R. D. Leclerc, "Survival of the sparsest: robust gene networks are parsimonious," *Molecular systems biology*, vol. 4, no. 1, p. 213, 2008.

[20] M. E. Newman and M. Girvan, "Finding and evaluating community structure in networks," *Physical review E*, vol. 69, no. 2, p. 026113, 2004.

[21] N. Kashtan and U. Alon, "Spontaneous evolution of modularity and network motifs," *Proceedings of the National Academy of Sciences*, vol. 102, no. 39, pp. 13 773–13 778, 2005.