Neutrality, Robustness, and Evolvability in Genetic Programming

Ting Hu and Wolfgang Banzhaf

Abstract Redundant mapping from genotype to phenotype is common in evolutionary algorithms, especially in genetic programming (GP). Such a redundancy leads to neutrality, a situation where mutations to a genotype may not alter its phenotypic outcome. The effects of neutrality can be better understood by quantitatively analyzing its two observed properties, i.e., robustness and evolvability. In this chapter, we summarize our previous work on this topic in examining a compact Linear GP algorithm. Due to the choice of this particular system we can characterize its entire genotype, phenotype, and fitness networks, and quantitatively measure robustness and evolvability at the genotypic, phenotypic, and fitness levels. We then investigate the relationship between robustness and evolvability at those different levels. Technically, we use an ensemble of random walkers and hill climbers to study how robustness and evolvability are related to the structure of genotypic, phenotypic, and fitness networks and influence the evolutionary search process.

1 Introduction

In evolutionary algorithms in general, but especially in genetic programming (GP), a redundant genotype-to-phenotype mapping is common where multiple unique genotypes map to the same phenotype [1, 12, 16, 23, 25]. A related notion of *neutrality* has been put forward to describe the mutational connectivity amongst those geno-

Ting Hu
Department of Computer Science, Memorial University, St. John's, NL A1B 3X5, Canada
e-mail: ting.hu@mun.ca

Wolfgang Banzhaf
Department of Computer Science, Memorial University, St. John's, NL A1B 3X5, Canada
BEACON Center of the Study of Evolution in Action, Michigan State University, East Lansing, MI 48824, USA
e-mail: banzhaf@mun.ca

types mapped to the same phenotype [2]. Specifically, if a single-point mutation changes one genotype to another without altering the phenotypic outcome, the mutation is called neutral. Redundancy and neutrality are different but closely related concepts. Redundancy is needed for, but does not guarantee, neutrality. It is possible that although a phenotype can be represented by multiple genotypes, these genotypes cannot traversed from one to the other through single-point mutations. In such a case, any mutations applied to those genotypes will change their phenotype.

Neutrality appears as an embedded property of many evolutionary algorithms and its influence on evolution has seen many debates in the field of evolutionary computation. On the one hand, neutrality may seem to hamper the evolutionary search since neutral mutations are not phenotypically effective [7, 27]. On the other hand, neutrality is considered beneficial for the search by providing a buffer against deleterious mutations [32] and, more importantly, by offering mutational potential through expanding neutral genotypic regions which are not subject to selection pressure [14, 29]. These two aspects relate neutrality to two notions that have drawn much attention in studies on both computational and natural evolution, namely to robustness and evolvability.

Robustness describes the resilience of an evolutionary system in the face of constant genetic and environmental perturbations, while *evolvability* captures the ability for generating novel and adaptive phenotypes. These two properties may seem contradictory at first glance, but are commonly observed coexisting in living organisms and are both results of neutrality.

The interplay between robustness and evolvability has been a focus of research in evolutionary biology. Both theoretical [20, 30] and empirical studies [10, 19, 21] have been put forth to elucidate the relationship between them. Using RNA molecules, some argued that neutral mutational connections constrain evolution since evolution yields phenotypes which are genotypically abundant even when they are not the most fit [8]. While others argued that robustness could facilitate evolvability and long-term innovation could only emerge in the presence of the mutational robustness [6, 9].

The relationship between robustness and evolvability is system-dependent, and it is crucially influenced by the distribution of genotypic redundancy and the mutational interconnections among phenotypes [17, 25]. Robustness promotes evolvability only if genotypic redundancy leads to more connections to different phenotypes.

A quantitative understanding of the relationship between robustness and evolvability can help resolve conflicting reports and clarify outstanding research questions. *Genotype networks*, a.k.a., neutral networks, provide a general framework for quantitatively characterizing robustness and evolvability, and have found applications in a wide array of systems [6, 11, 22, 24, 26].

In genotype networks (Figure 1), vertices represent unique genotypes and mutational connections are represented as edges between pairs of genotypes. A genotype network is comprised of all genotypes that encode the same phenotype. Mutations within a genotype network are neutral by definition. Multiple genotype networks representing different phenotypes can also be connected through non-neutral single-point mutations. Genotype networks quantitatively characterize the distribution of

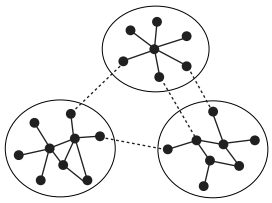

Fig. 1 Schematic diagram of genotype networks. Each vertex represents a genotype and all genotypes encoding the same phenotype define one genotype network. An edge links two vertices if the two genotypes can be transferred from one to another through a single point mutation. Single point mutations can also connect genotypes from different phenotypes, shown in dashed lines.

genotypic redundancy among phenotypes, i.e., over-represented phenotypes have larger genotype networks and under-represented phenotypes have smaller networks. Genotype networks also capture the mutational potential among different phenotypes using different edges representing non-neutral mutations between genotypes that belong to two phenotypes.

Phenotype and fitness networks can be constructed in a similar way by representing phenotypes (or fitness values) as vertices and their mutational connections as edges. By building networks at these different levels, we are enabled to take a close look at the relationship of robustness and evolvability at the genotypic, phenotypic, and fitness levels.

Most existing studies of neutrality in evolutionary algorithms look at the effect of neutrality on the evolutionary search indirectly, i.e., they ask whether neutrality by a redundant representation improves or hampers the search ability of an evolutionary algorithm. Very little has been done to quantitatively measure robustness and evolvability directly and to study their relationship and influence on evolution dynamics.

In this chapter, we discuss the use of genotype networks to quantitatively analyze robustness and evolvability in a Linear Genetic Programming system. Linear GP has a compact representation and here is intentionally amenable to an exhaustive enumeration of all possible genotypes and phenotypes. We characterize its genotype, phenotype, and fitness networks and their properties, and examine the diffusion and dynamics of an evolutionary population on those networks. We report on a quantitative examination of neutrality and elucidate the relationship of robustness and evolvability in GP. We hope that our analysis can find application in other GP instances and in other evolutionary algorithms, that it provides a better understanding of evolutionary mechanisms, and that it will eventually inspire new and more sophisticated evolutionary algorithms [3, 15].

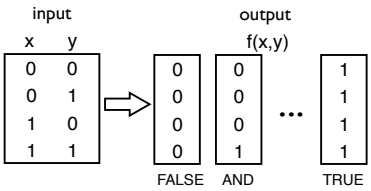

Fig. 2 The phenotype of a LGP program is defined as the Boolean relationship it encodes, represented as the four-digit output of the ordered two-variable inputs.

2 Linear GP Algorithm

2.1 Representation

Linear Genetic Programming is a branch of Genetic Programming (GP) where the chromosomal representation is a set of instructions that are executed sequentially [5]. Although LGP follows a linear instructional structure, it is very powerful and capable of modeling complex nonlinear relationships among multiple attributes. LGP has gained increasing popularity due to its fast speed of program execution and individual evaluation [4, 13, 18, 28].

Here, we consider a two-input one-output Boolean function (Boolean circuit) modeling problem. Each LGP instruction is comprised of one return value, two operands, and one Boolean operator producing the return value from the operands. Registers $\mathtt{R}_1$ and $\mathtt{R}_2$ store the two Boolean input values. Register $\mathtt{R}_0$ takes a default initial Boolean value and its finciteal value after the execution of all instructions is returned as the LGP program's output. To enhance the computational capacity of LGP programs, we add an extra calculation register $\mathtt{R}_3$. Calculation registers $\mathtt{R}_0$ and $\mathtt{R}_3$ can serve as both return or operands, whereas input registers $\mathtt{R}_1$ and $\mathtt{R}_2$ are read-only and can only serve as operands with their input content being protected from overwriting. The Boolean operator in each LGP instruction is chosen from a pre-defined operator set $opr = \{\mathtt{AND}, \mathtt{OR}, \mathtt{NAND}, \mathtt{NOR}\}$. An example Boolean LGP program with a length $L = 4$ can be given as:

$$\begin{aligned}
\mathtt{R}_3 &= \mathtt{R}_2 \ \ \mathtt{AND} \ \ \mathtt{R}_3 \\
\mathtt{R}_0 &= \mathtt{R}_1 \ \ \mathtt{OR} \ \ \mathtt{R}_2 \\
\mathtt{R}_3 &= \mathtt{R}_2 \ \mathtt{NAND} \ \mathtt{R}_1 \\
\mathtt{R}_0 &= \mathtt{R}_3 \ \mathtt{NAND} \ \mathtt{R}_0
\end{aligned}$$

2.2 Genotype, phenotype, and fitness

In our LGP system, the *genotype* is an unique LGP program. To enable exhaustive enumeration of the entire genotype space, we set a fixed length of $L = 4$ for all LGP programs. The total number of possible instructions is $2 \times 4 \times 4 \times 4 = 2^7$, and thus, the total number of possible four-instruction programs, i.e. genotypes, is $(2^7)^4 = 2^{28} > 268$ million. We see that even for a small problem instance and a short fixed program length the genotype space can be quite large.

We define the two-input, one-output Boolean function $f : \mathbf{B}^2 \to \mathbf{B}$, where $\mathbf{B} = \{\texttt{TRUE}, \texttt{FALSE}\}$ represented by a LGP program as its *phenotype*. The total number of possible phenotypes is thus $2^{2^2} = 16$. A phenotype can be represented by a set of outputs observed across each of the 2^2 possible combinations of Boolean inputs (Figure 2). Compared to the large genotype space, the phenotype space is very small. This suggests a high redundancy in the mapping from genotype to phenotype, i.e., a large number of different genotypes should map to the same phenotype (approximately 16.7 million genotypes per phenotype, on average).

Based on a predefined phenotypic target, fitness can be assigned to each of the 16 phenotypes. We define the *fitness* of a phenotype to be the Hamming distance between its four-digit binary vector and that of the target. While this is technically the error between the two functions, we use the term fitness for this quantity, despite it being minimized. There are five possible fitness values, for example if the target phenotype is `TRUE` (i.e., $\langle 1111 \rangle$) the fitness of phenotype `FALSE` (i.e., $\langle 0000 \rangle$) is 4. The phenotype `x OR y` (i.e., $\langle 0111 \rangle$) has an improved fitness of 1. The mapping from phenotype to fitness is redundant again, i.e., from 16 phenotypes to five fitness values, but depends on which phenotype is set as the target. Redundancy between phenotype and fitness is less strong (approximately 3.2 phenotypes per fitness value, on average).

A single-point mutation to a genotype changes any one of the four elements of an instruction and replaces it with a randomly chosen possible allele. Single-point mutations that do not alter the phenotypic outcome are called neutral mutations. Mutations that lead to a change of phenotype are called non-neutral mutations.

3 Genotype, phenotype, and fitness networks

Our Boolean LGP system now has 16 genotype networks, each corresponding to a particular phenotype. The distribution of genotypic redundancy is highly uneven, with the largest genotype network `FALSE` having more than 60 million genotypes ($> 23\%$ of the entire genotype space) and the smallest genotype networks `x == y` and `x XOR y` having less than 25 thousand genotypes ($\ll 1\%$ of the entire genotype space). The distribution of the sizes of genotype networks is shown in Figure 3. Note that phenotype `FALSE` has more genotypes than `TRUE`, simply because all registers are initialized as `FALSE` before any computation. Programs whose execu-

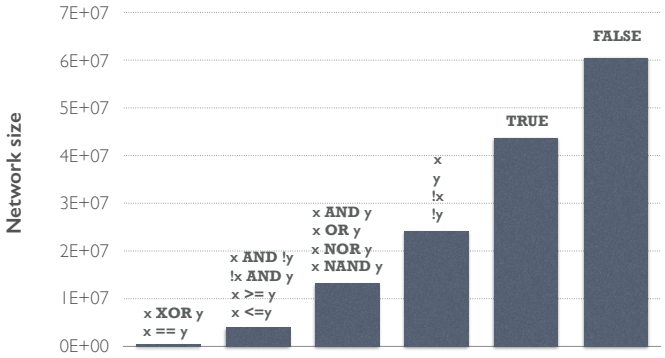

Fig. 3 Distribution of the size of genotype networks. Due to the symmetry of Boolean relationships, multiple phenotypes can have the same number of underlying genotypes. The size of genotype networks ranges from 25 thousand to 60 million.

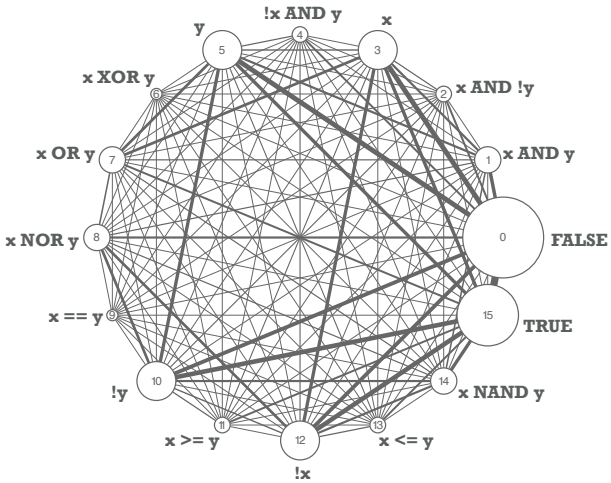

Fig. 4 The phenotype network. Vertices represent phenotypes and edges link two phenotypes if there exist at least one pair of genotypes mapped to the two phenotypes that can be transferred from one to the other through one single-point mutation. Vertex size is proportional to the total number of genotypes mapped to a corresponding phenotype. Edge width is proportional to the total number of one point mutations that change genotypes of one phenotype to another.

tion does not change the content of the output register R_0 will output `FALSE`. The heterogeneous distribution of genotype networks suggests that some phenotypes are over-represented and some are under-represented. Random sampling and initializing genotypes likely will generate over-represented phenotypes. If a phenotypic target is under-represented, the search task will be relatively more difficult.

A phenotype network can be further constructed by representing a genotype network, i.e., a phenotype, as a vertex, and connecting two phenotypes using an edge if there exist at least one pair of genotypes of those two phenotypes that can be

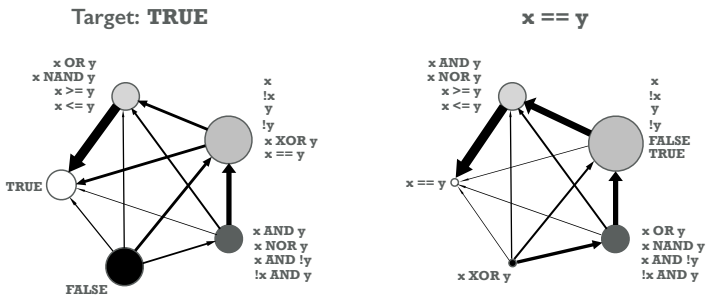

Fig. 5 The fitness networks with target phenotype `TRUE` (left) and `x == y` (right). Each vertex is a fitness value with white representing the best fitness zero and black representing the worst fitness four. The vertex size is again proportional to the total number of underlying genotypes. When selection prevents worsening fitness, point mutations may become irreversible, and the mutational transitions among fitness values, represented by edges, is now directed. Edge width is proportional to the total number of point mutations changing genotypes from one fitness value to another.

transferred from one to the other through a single-point mutation. Figure 4 shows the phenotype network in our setting. Phenotypes as vertices are numbered using the decimal values corresponding to their binary strings, labeled with their represented Boolean relationships. The phenotype network of our LGP system here is a complete graph, meaning that every vertex is connected directly to any other vertex. However, the connections are also highly heterogeneous, reflected by the varying width of edges. This suggests that a phenotype has varying mutational potentials to access other phenotypes. For instance, random mutations to genotypes of phenotype `!y` more likely lead to phenotypes `y`, `FALSE`, and `TRUE`, and less likely to phenotypes `x OR y` and `x` .

Introducing fitness further groups genotypes to build a higher-level fitness network. Since the fitness function is defined as the Hamming distance between the target phenotype and the reference phenotype, the assignment of fitness values and thus the structure of the fitness network depend on the setting of the phenotypic target. Figure 5 shows two fitness networks using different target phenotypes. When selection is present and rejects mutations that worsen the fitness, the fitness network becomes directed, where single-point mutations are only accepted if fitness is improved or remains the same. Again, we observe heterogeneous mutational potential for transitioning from one fitness level to another.

4 Quantitative analysis of robustness and evolvability

Using the framework of genotype and phenotype networks, robustness and evolvability can be quantitatively analyzed. In the context of RNA genotypes and their secondary structure phenotypes, it has been argued that the paradoxical tension of mutational robustness and evolvability can be solved by distinguishing robustness

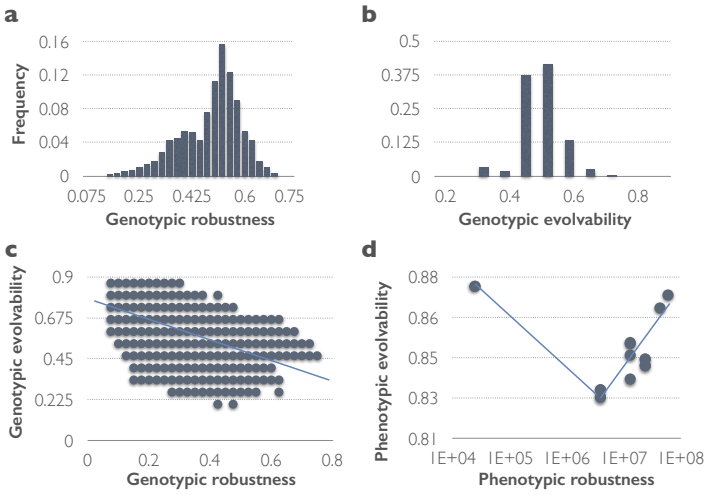

Fig. 6 Robustness and evolvability of genotypes and phenotypes. A typical and representative phenotype `x>=y` is chosen to show the genotypic properties in its genotype network. Subfigures a) and b) show the distributions of genotypic robustness and evolvability, and c) shows the scattered plot and correlation of genotypic robustness and evolvability. The correlation of phenotypic robustness and evolvability is shown in d). The fitted lines provide a visual guidance of correlations.

and evolvability at genotypic and phenotypic levels [31]. The relationship of robustness and evolvability can be different at those two levels. We discuss the quantitative analysis of robustness and evolvability of a genotype and a phenotype in the following subsections.

4.1 Genotypic robustness and evolvability

The robustness of a genotype can be measured as the fraction of its neutral neighbors among all neighbors [31]. This definition follows the intuition that if a random single-point mutation to a genotype likely leads to a different genotype but retains the same phenotype, this genotype can be regarded as robust.

The measure of the evolvability of a genotype, on the other hand, should reflect the innovation ability of a genotype. It is defined as the fraction of the number of phenotypes that are accessible through non-neutral single-point mutations to a genotype to the number of all phenotypes [31].

We now look at the distribution of genotypic robustness and evolvability within a genotype network. Note that all phenotypes have very similar behavior, so we only show results of one typical and representative phenotype `x >= y`. If the genotype network of `x >= y` is visualized with more robust genotypes located more towards the center, the bi-modal distribution of genotypic robustness (Figure 6a) suggests a dense core and a thick periphery of the network. The genotypic evolvability (Fig-

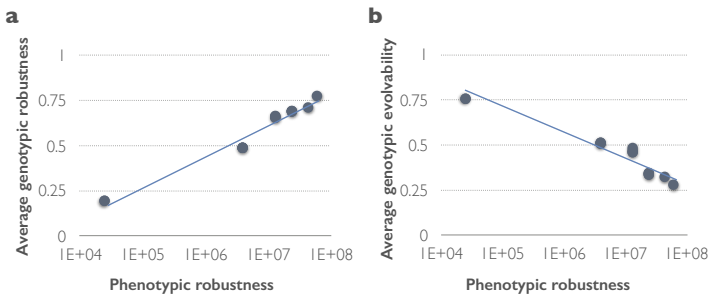

Fig. 7 The correlation of phenotypic robustness and the average genotypic (a) robustness and (b) evolvability. The fitted lines provide a visual guidance of correlations.

ure 6b) resembles a normal distribution, with the majority of genotypes being able to reach 50% of other phenotypes though single-point mutations. The genotypic evolvability and robustness are negatively correlated (Figure 6c). This negative correlation is weak ($R^2 = 0.015$) but highly significant ($p \ll 0.001$). This observation is in line with findings in RNA networks where at the genotypic level robustness and evolvability share an antagonistic relationship [31]. It is also intuitive that if random mutations to a genotype do not change its phenotype most of the time, this genotype may have less access to other different phenotypes.

4.2 Phenotypic robustness and evolvability

The robustness of a phenotype is defined as the size of its genotype network, i.e., the total number of unique genotypes that map to the phenotype. The more genotypes a phenotype has, the more robust it appears.

The definition of phenotypic evolvability has seen different proposals. It can be defined similarly to genotypic evolvability as the fraction of different phenotypes that can be reached via non-neutral single-point mutations from a given phenotype [31]. However, given the complete connectivity of our phenotype network (Figure 4), this phenotypic evolvability measure will assign the same value of 1 to all phenotypes.

Alternatively, the evolvability of a phenotype can be measured as the distribution of its mutational potential to other phenotypes [8]. Specifically, we use v_{ij} to denote the total number of non-neutral single-point mutations between phenotypes i and j. Letting

$$f_{ij} = \begin{cases} \frac{v_{ij}}{\sum_{k \neq i} v_{ik}}, & \text{if } i \neq j \\ 0, & \text{if } i = j \end{cases} \tag{1}$$

denote the fraction of non-neutral point mutations to genotypes of phenotype i that result in genotypes of phenotype j, we define the evolvability E of a phenotype i as

$$E_i = 1 - \sum_j f_{ij}^2. \tag{2}$$

A phenotype that has more equally distributed mutational potential to other phenotypes is regarded as more evolvable.

The correlation of phenotypic robustness and evolvability is shown in Figure 6d for all 16 phenotypes. Phenotypic robustness and evolvability are non-monotonically correlated, with median robust phenotypes having the lowest evolvability (`x AND !y` and `!x AND y`). Both the least (`x XOR y` and `x == y`) and most robust (`FALSE`) phenotypes are highly evolvable. Our results disagree with previous findings in evolutionary biology that either a monotonic positive [31] or negative [8] correlation is observed.

We also compare the measured properties across the genotypic and phenotypic levels. Figure 7 shows the average genotypic robustness and evolvability in relation to the phenotypic robustness. A strong and significant positive correlation ($R^2 = 0.98, p \ll 0.001$) is observed between the average genotypic robustness and the robustness of the corresponding phenotype (Figure 7a). Meanwhile, average genotypic evolvability is negatively correlated ($R^2 = 0.95, p \ll 0.001$) with phenotypic robustness (Figure 7b). This suggests that more robust phenotypes are comprised of more robust and less evolvable genotypes.

Note that at the level of fitness, robustness and evolvability can be defined similarly to the definition for phenotypes. However, fitness evolvability and robustness are no longer correlated (data not shown, $R^2 = 1.8 \times 10^{-4}, p = 0.91$).

5 Random walkers and hill climbers

We use an ensemble of random walkers and hill climbers to investigate how the structures of genotype, phenotype, and fitness networks influence evolutionary search. We test if the quantitative measures of robustness and evolvability provide insights into predicting the search dynamics. We perform two sets of simulations. In the first set, a genotype is allowed to randomly explore the genotypic and phenotypic spaces, i.e., as a random walker. In the second set of experiments, a specific target phenotype is chosen and a fitness value is thus assigned to each genotype. Hill climbers are only allowed to move from genotypes via non-deleterious single-point mutations.

5.1 Random walks through genotype networks

First we investigate how individual random walkers explore the genotypic space. We consider a representative phenotype `x >= y` and confine the random walking within its genotype network. By doing so, we enforce the selection pressure

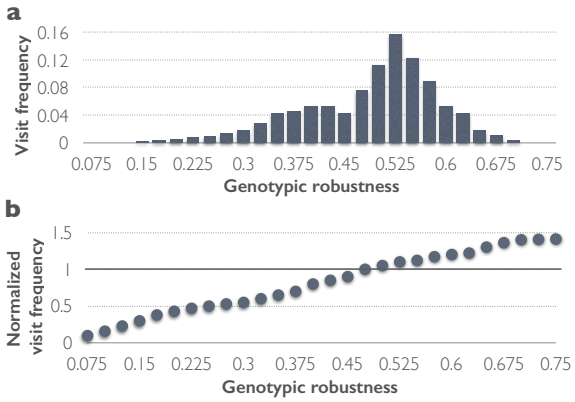

Fig. 8 A random walk in the genotype network of phenotype `x >= y`. a) Distribution of the visit frequency, defined as the proportion of steps in a random walk spent at genotypes of a given robustness value. b) Visit frequency normalized by the frequency with which a given genotypic robustness value is observed in the genotype network, i.e., the distribution in a) divided by the distribution in Figure 6a.

on neutrality and observe its influences on evolution. Each step corresponds to a single-point mutation. We randomly pick a genotype in the phenotype `x >= y` and record all genotypes encountered in a total of four million (an approximation of the total number of genotypes in phenotype `x >= y`) steps. Then we compute the visit frequency of each genotypic robustness value during the entire course. This distribution is shown in Figure 8a.

The visit frequency follows a bi-modal distribution, similar to the distribution of genotypic robustness in the genotype network of `x >= y` (Figure 6a). It is true that the more frequent a robustness value is observed in a genotype network, the more likely a random walker will encounter that robustness value. So we normalize the visit frequency by dividing it by the fraction of a robustness value observed in a genotype network, i.e., by dividing Figure 8a by Figure 6a. The resulting distribution is shown in Figure 8b.

Now we can observe a strong positive correlation of the normalized visit frequency and the genotypic robustness. This suggests that genotypes are not visited uniformly by single-point mutations, but rather in proportion to their robustness. Genotypes of high robustness are visited more often, and genotypes of low robustness are visited less often than would be expected from a random sampling of genotypes from a phenotype.

5.2 Average waiting/adaptation time

We set each of the 16 phenotypes as the target phenotype and one of the other 15 as the starting phenotype. For each of the 16×15 possible combinations of pairs of

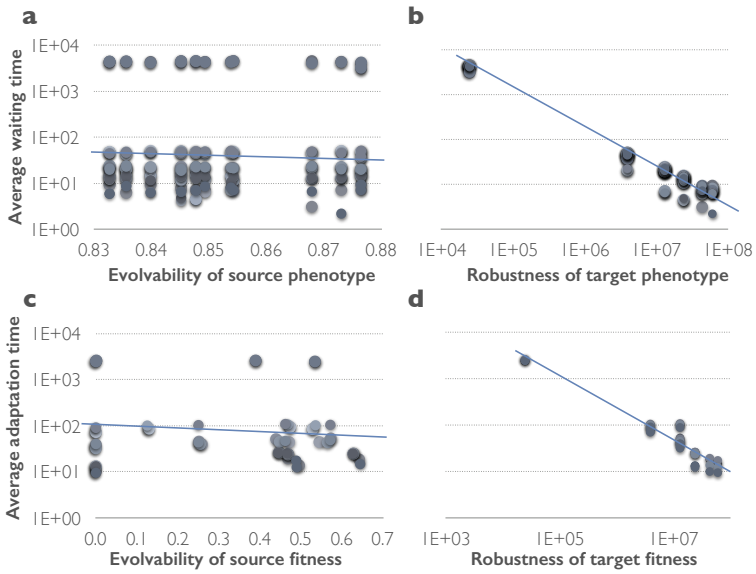

Fig. 9 Average waiting (adaptation) time as a function of the evolvability of the starting phenotype (fitness) and the robustness of the target phenotype (fitness).

unique phenotypes, we then perform 1000 random walks and hill climbs, starting from a designated source phenotype and ending when the random walker or hill climber reaches any genotype in the specified target phenotype. We record the total number of point-mutations/steps required to get from one phenotype to another and calculate the average waiting (adaptation) time across 1000 random walks (hill climbs).

Figure 9 shows the average waiting (adaptation) time as a function of the evolvability of the source phenotype (fitness) and the robustness of the target phenotype (fitness). It is speculated that if a random walker or hill climber starts from a more evolvable phenotype, it may find a target phenotype faster. However, the evolvability of the source phenotype fails to make a prediction on the waiting time (Figure 9a, $R^2 = 0.001$, $p = 0.62$), neither does the evolvability of the source fitness on the adaptation time (Figure 9c, $R^2 = 0.02$, $p = 0.32$). This observation puts into question the currently available quantification of evolvability. Recall that the evolvability of a phenotype/fitness measures the mutational potential to reach other phenotypes/fitness. It only captures the very first step leaving a phenotype/fitness, but fails to provide further insights on the long-term trajectory of the evolutionary process.

The robustness of the target phenotype (fitness), on the other hand, shows strong predictive power for the average waiting (adaptation) time. In Figure 9 b and d, a strong and negative correlation is observed between average waiting time and robustness of the target phenotype ($R^2 = 0.95$, $p \ll 0.001$), as well as between average adaptation time and robustness of the target fitness ($R^2 = 0.88$, $p \ll 0.001$).

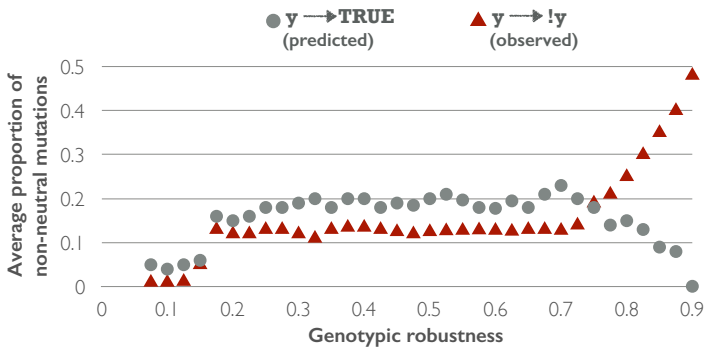

Fig. 10 Comparison of mutational transitions from phenotype `y` to `TRUE` and to `!y`. Filled circles are genotypes of `y` that have non-neutral mutational connections to genotypes of phenotype `TRUE`, and triangles are genotypes of `y` that have non-neutral mutational connections to phenotype `!y`.

These results are intuitive and suggest that more robust phenotypes (fitness) are easier to reach from other phenotypes via single-point mutations since they are over-represented by more genotypes, and random mutations will more likely lead to more robust phenotypes.

The probabilistic nature of random walks and hill climbing can be captured by Markov Chain analysis, meaning that the average waiting (adaptation) time could be predicted analytically rather than through empirical simulations. By considering each phenotype as a state, and the mutational connections between phenotype i and phenotype j (f_{ij} in Equation 1) as their transition probability, we can apply Markov Chain analysis to determine the expected waiting (adaptation) time for moving from one phenotype to another. We find a strong correlation between the analytical prediction and the empirical observation, yet also large relative residuals, i.e., 126 steps comparing to 112 steps in the average waiting and adaptation times, respectively.

This discrepancy between the analytical prediction and empirical observations suggests that the mutational connections between phenotypes might not serve as the most accurate estimate of transitional probabilities from one phenotype to another. Let us take a close look at an example for moving from source phenotype `y` to target phenotype `!x AND y`: The predicted most likely path for this transition is `y –> TRUE –> !x AND y`, but the observed most frequent path is `y –> !y –> TRUE –> !x AND y`, despite the fact that the observed path is longer than the predicted path and `y` has more mutational connections to `TRUE` than `!x AND y` (i.e., $f_{\mathtt{y},\mathtt{TRUE}} = 0.19$ and $f_{\mathtt{y},\mathtt{!y}} = 0.18$)!

The transitional probabilities are measured at the phenotypic level, but mutations occur at the genotypic level. Therefore, if the mutational connections between phenotypes do not provide the most accurate estimation of the transition likelihoods, a mutational bias must be introduced at the genotypic level. We take phenotypes `y`, `TRUE`, and `!y` as examples and look into the genotypes that allow a transfer from `y` to `TRUE` and to `!y`. Figure 10 shows the comparison of the transitions between `y` and `TRUE` and between `y` and `!y`. The non-neutral mutations connect `y` to `TRUE`

(filled circles) through more genotypes with low robustness but less genotypes with high robustness, whereas the non-neutral mutations connect `y` to `!y` (filled triangles) through less genotypes with low robustness but more genotypes with high robustness. Recall that more robust genotypes are visited more frequently (Figure 8b). This is the source of the bias required and explains why mutations to genotypes of `y` more likely lead to phenotype `!y` than to `TRUE`, despite the fact that the total amount of non-neutral mutations between `y` and `TRUE` is greater than that between `y` and `!y`.

6 Conclusion

Neutrality is commonly observed in evolutionary algorithms where mutations may not alter the phenotypic outcome. Neutrality is a result of the redundant genotype-to-phenotype mapping and debates have raged on whether neutrality is beneficial for the search ability of an evolutionary algorithm.

The effects of neutrality on its two observed properties, robustness and evolvability, can be studied quantitatively. Both robustness and evolvability capture how an evolutionary system responds to genetic changes. Robustness refers to the resilience to retain phenotypic traits in face of mutational perturbations, whereas evolvability characterizes the capability of using random mutations to generate novel and adaptive phenotypes. The relationship of robustness and evolvability may seem antagonistic, but is in fact highly collaborative.

Studying the relationship of robustness and evolvability helps to better understand the fundamental mechanisms of evolution. The framework of genotype networks has been used to quantitatively measure robustness and evolvability and to analyze their relationship. Moreover, the relationship should be better studied at the genotypic, phenotypic, and fitness levels since robustness and evolvability can take different qualifications and correlate differently at those levels.

In this book chapter, we reported on the quantitative analysis of robustness and evolvability at the genotypic, phenotypic, and fitness levels. A small-scale Linear GP system was adopted as our test system, which provides multiple advantages for our purposes. The Linear GP algorithm has a compact presentation which allows exhaustive enumeration of all possible genotypes and phenotypes. Thus the entire genotype and phenotype spaces can be characterized.

We followed evolutionary biological studies on robustness and evolvability in RNA networks and defined quantitative measures of robustness and evolvability at the genotypic, phenotypic, and fitness levels. We showed that robustness and evolvability correlate differently at those levels. At the genotypic level, a more robust genotype is less evolvable. At the phenotypic level, the correlation of robustness and evolvability is non-monotonic with the least robust and the most robust phenotypes having the highest evolvability. However, no correlation was observed at the fitness level. This finding calls for more advanced fitness evaluation methods in

the future that incorporate mutational connections at the genotypic and phenotypic levels rather than simply the similarity between phenotypes.

Using an ensemble of random walkers and hill climbers, we showed how the structure of genotype, phenotype, and fitness networks can influence the evolutionary search. We found that more robust phenotypes are more accessible from other phenotypes via random mutations, however starting from a more evolvable phenotype does not guarantee a more efficient search for novel phenotypes. This is due to the limitations of evolvability measures currently available and calls for further studies.

We also found that robust genotypes play a crucial role in the evolutionary search process. More robust genotypes are visited more often than would be expected in a random sampling of genotypes, i.e., random mutations are biased leading to more robust genotypes. Therefore, robust genotypes can influence the evolutionary search by guiding it to their adjacent phenotypes. This finding is of particular interest since it may inspire mechanisms of evolutionary search that utilize robust genotypes.

Acknowledgements TH is supported by the Ignite R&D fund of Research and Development Corporation of Newfoundland and Labrador and the Canadian Natural Sciences and Engineering Research Council (NSERC) Discovery grant RGPIN-04699-2016. WB acknowledges the support from the Canadian Natural Sciences and Engineering Research Council (NSERC) Discovery grant RGPIN-283304-2012.

References

1. Banzhaf W (1994) Genotype-phenotype mapping and neutral variation - a case study in genetic programming. In: Davidor Y, Schwefel HP, Manner R (eds) Parallel Problem Solving from Nature, Springer, Berlin, Lecture Notes in Computer Science, vol 866, pp 322–332
2. Banzhaf W, Leier A (2006) Evolution on neutral networks in genetic programming. In: Yu T, Riolo R, Worzel B (eds) Genetic Programming Theory and Practice III, Springer, chap 14, pp 207–221
3. Banzhaf W, Beslon G, Christensen S, Foster JA, Kepes F, Lefort V, Miller JF, Radman M, Ramsden JJ (2006) From artificial evolution to computational evolution: a research agenda. Nature Review Genetics 7:729–735
4. Brameier MF, Banzhaf W (2001) A comparison of linear genetic programming and neural networks in medical data mining. IEEE Transactions on Evolutionary Computation 5(1):17–26
5. Brameier MF, Banzhaf W (2007) Linear Genetic Programming. Springer
6. Ciliberti S, Martin OC, Wagner A (2007) Innovation and robustness in complex regulatory gene networks. Proceedings of the National Academy of Sciences 104(34):13,591–13,596
7. Collins M (2006) Finding needles in haystacks is harder with neutrality. Genetic Programming and Evolvable Machines 7:131–144
8. Cowperthwaite MC, Economo EP, Harcombe WR, Miller EL, Meyers LA (2008) The ascent of the abundant: How mutational networks constrain evolution. PLoS Computational Biology 4(7):e1000,110
9. Draghi JA, Parsons TL, Wagner GP, Plotkin JB (2010) Mutational robustness can facilitate adaptation. Nature 463:353–355
10. Ferrada E, Wagner A (2008) Protein robustness promotes evolutionary innovations on large evolutionary time-scales. Proceedings of The Royal Society B 275:1595–1602

11. Fontana W, Schuster P (1998) Continuity in evolution: On the nature of transitions. Science 280:1451–1455
12. Galvan-Lopez E, Poli R (2006) An empirical investigation of how and why neutrality affects evolutionary search. In: Cattolico M (ed) Proceedings of the Genetic and Evolutionary Computation Conference, pp 1149–1156
13. Guven A (2009) Linear genetic programming for time-series modeling of daily flow rate. Journal of Earth System Science 118(2):137–146
14. Hu T, Banzhaf W (2009) Neutrality and variability: Two sides of evolvability in linear genetic programming. In: Rothlauf F (ed) Proceedings of the Genetic and Evolutionary Computation Conference, pp 963–970
15. Hu T, Banzhaf W (2010) Evolvability and speed of evolutionary algorithms in light of recent developments in biology. Journal of Artificial Evolution and Applications 2010:568,375
16. Hu T, Payne JL, Banzhaf W, Moore JH (2011) Robustness, evolvability, and accessibility in linear genetic programming. In: Silva S, Foster JA, Nicolau M, Machado P, Giacobini M (eds) Proceedings of the European Conference on Genetic Programming, Lecture Notes in Computer Science, vol 6621, pp 13–24
17. Hu T, Payne JL, Banzhaf W, Moore JH (2012) Evolutionary dynamics on multiple scales: A quantitative analysis of the interplay between genotype, phenotype, and fitness in linear genetic programming. Genetic Programming and Evolvable Machines 13:305–337
18. Hu T, Moore JH, Banzhaf W (2014) The effects of recombination on phenotypic exploration and robustness in evolution. Artificial Life 20(4):457–470
19. Landry CR, Lemos B, Rifkin SA, Dickinson WJ, Hartl DL (2007) Genetic properties influcing the evolvability of gene expression. Science 317:118–121
20. Masel J, Trotter MV (2010) Robustness and evolvability. Trends in Genetics 26:406–414
21. McBride RC, Ogbunugafor CB, Turner PE (2008) Robustness promotes evolvability of thermotolerance in an RNA virus. BMC Evolutionary Biology 8:231
22. Reidys C, Stadler PF, Schuster P (1997) Generic properties of combinatory maps: neutral networks of RNA secondary structures. Bulletin of Mathematical Biology 59(2):339–397
23. Reisinger J, Miikkulainen R (2007) Acquring evolvability through adaptive representation. In: Proceedings of the Genetic and Evolutionary Computation Conference, pp 1045–1052
24. Rodrigues J, Wagner A (2011) Genotype networks, innovation, and robustness in sulfur metabolism. BMC Systems Biology 5:39
25. Rothlauf F, Goldberg DE (2003) Redundant representations in evolutionary computation. Evolutionary Computation 11(4):381–415
26. Schuster P, Fontana W, Stadler PF, Hofacker IL (1994) From sequences to shapes and back: A case study in RNA secondary structures. Proceedings of The Royal Society B 255:279–284
27. Smith T, Husbands P, O'Shea M (2001) Neutral networks and evolvability with complex genotype-phenotype mapping. In: Kelemen J, Sosik P (eds) Proceedings of the European Conference on Artificial Life, Springer-Verlag, Lecture Notes in Artificial Intelligence, vol 2159, pp 272–281
28. Song D, Heywood MI, Zincir-Heywod A (2001) A linear genetic programming approch to intrusion detection. In: Proceedings of the Genetic and Evolutionary Computation Conference, Lecture Notes in Computer Science, vol 2724, Springer-Verlag, pp 2325–2336
29. Soule T (2006) Resilient individuals improve evolutionary search. Artificial Life 12:17–34
30. Wagner A (2005) Robustness, evolvability, and neutrality. Federation of European Biochemical Societies Letters 579(8):1772–1778
31. Wagner A (2008) Robustness and evolvability: A paradox resolved. Proceedings of The Royal Society B 275(1630):91–100
32. Yu T, Miller JF (2006) Through the interaction of neutral and adaptive mutations, evolutionary search finds a way. Artificial Life 12:525–551