# Measuring Rate of Evolution in Genetic Programming Using Amino Acid to Synonymous Substitution Ratio $k_a/k_s$

Ting Hu
Department of Computer Science
Memorial University of Newfoundland
St. John's, Canada
tingh@cs.mun.ca

Wolfgang Banzhaf
Department of Computer Science
Memorial University of Newfoundland
St. John's, Canada
banzhaf@cs.mun.ca

## ABSTRACT

We define the rate of evolution $R_e$ in a GP system based on the rate of efficient genetic variations being accepted. This definition is motivated by the measurement of "amino acid to synonymous substitution ratio" $k_a/k_s$ in biology. Experimental applications of this rate of evolution measurement show that $R_e$ well reflects how evolution proceeds underneath fitness development and quantifies the rate of innovation through efficient genetic variations.

## Categories and Subject Descriptors

I.2.2 [**Artificial Intelligence**]: Automatic Programming, Program Synthesis

## General Terms

Measurement

## Keywords

Rate of Evolution, Genetic Programming, $k_a/k_s$ Ratio

## 1. INTRODUCTION

Improving capabilities of an artificial evolutionary system has attracted substantial attention recently [1]. Various evolution rates accompany diverse evolution capabilities at different stages of an evolutionary computation process or in different computation models. Measuring the rate of evolution can quantify evolutionary capabilities, and thus can help to accelerate evolution through designing better models. The rate of evolution has not seen an effective formal definition other than measuring fitness progression over generations. At first glance, a definition reflecting how fast an evolutionary population is improving its fitness may seem sufficient. However, evolutionary capabilities cannot be determined by how good population fitness is per se, but should be regarded as a "second-order" effect of fitness improvements. Therefore, we believe that the rate of evolution should be better defined by looking beyond fitness and should be measured by the rate of adaptation being generated and accepted. Biologists use the $k_a/k_s$ ratio in molecular evolution to measure the evolution rate of gene sequences [2]. Such a measurement compares two homologous protein-coding gene sequences from two related species. The $k_a/k_s$ ratio resulting from measuring the number of nonsynonymous (amino acid) substitutions per nonsynonymous site ($k_a$) to the number of synonymous substitutions per synonymous site ($k_s$) characterizes the rate of evolution. In this work, we introduce an analogous measurement of this $k_a/k_s$ ratio to GP.


*GECCO'08,* July 12–16, 2008, Atlanta, Georgia, USA.
ACM 978-1-60558-130-9/08/07.

## 2. MEASURING RATE OF EVOLUTION IN GENETIC PROGRAMMING

We utilize a tree-based GP system to implement this idea because it is the traditional representation of GP and because GP individuals possess similar features to gene sequences. For example, for a GP tree, genetic changes can also be nonsynonymous, leading to produce different functions, or synonymous, which keeps the encoded functions unchanged.

Before establishing a generation $t$, standard mutation and crossover, limited to subtree replacement, are applied to the individual trees in a GP population at generation $t-1$. Truncation tournament selection is then performed on a temporary population including both the parents and offspring to form the next generation $t$. A subtree replacement by mutation or crossover is either nonsynonymous or synonymous. For an individual tree $i$, if a change is silent, the value of nonsynonymous change $m_a^i(t)$ is set to 0 and the value of synonymous change $m_s^i(t)$ is set to 1. In contrast, if a change leads to functional differences, $m_a^i(t)$ is 1 and $m_s^i(t)$ is 0. If tree $i$ is not modified from generation $t-1$ to generation $t$, both $m_a^i(t)$ and $m_s^i(t)$ remain 0. The total number of nonsynonymous substitutions $M_a(t)$ and synonymous substitutions $M_s(t)$ for the entire population at the newly generated generation $t$ can be calculated as

$$M_a(t) = \sum_{i=1}^{S} m_a^i(t), \quad M_s(t) = \sum_{i=1}^{S} m_s^i(t),$$

where $S$ is the population size. Note that $M_a(t)$ and $M_s(t)$ only count those genetic changes accepted into the population, i.e. adaptive substitutions having survived through selection. Next, we adopt the concept of *sensitivity* to describe the *potential* of a GP tree being changed semantically. We define the nonsynonymous sensitivity of a GP tree as the fraction of accumulated nonsynonymous subtree replacements applied to this tree in the total number of subtree replacements. Specifically, for an individual tree $i$, we use $c_a^i(t)$ and $c_s^i(t)$ to denote the accumulated numbers of nonsynonymous and synonymous changes, respectively, obtained by summing up all the previously recorded changes

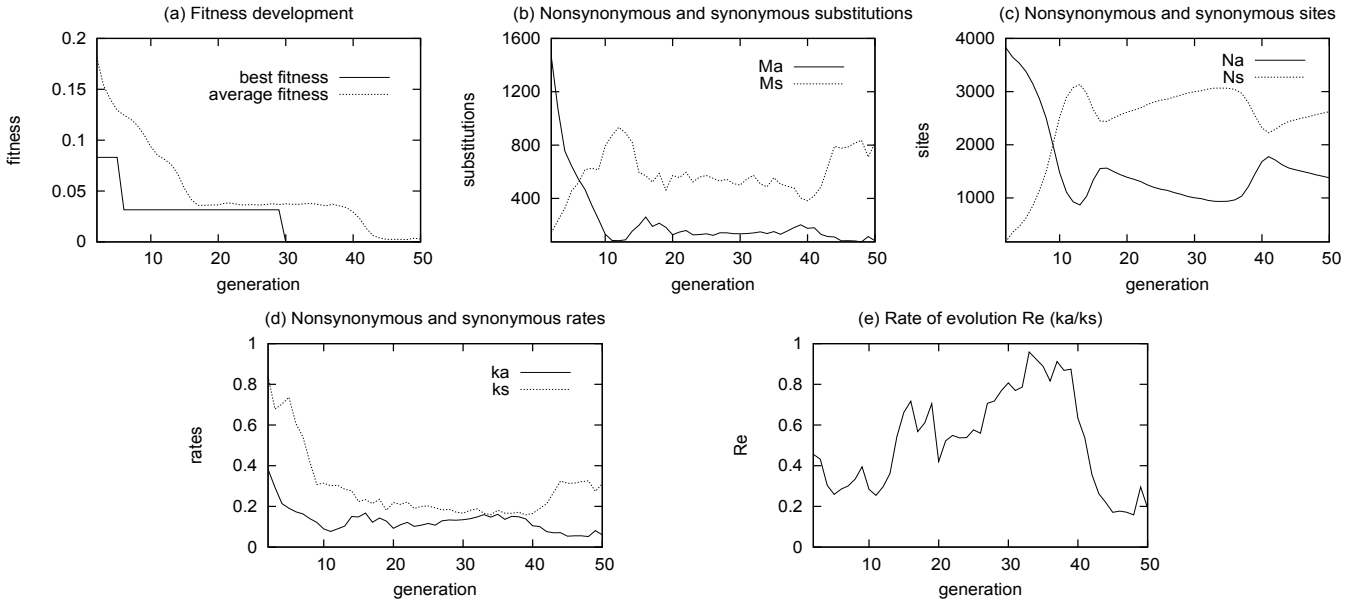


**Figure 1: Rate of evolution measurement in a GP evolutionary process**

that have happened to this tree, accepted or not,

$$c_a^i(t) = c_a^i(t-1) + m_a^i(t), \quad c_s^i(t) = c_s^i(t-1) + m_s^i(t),$$

with

$$c_a^i(0) = c_s^i(0) = 0.$$

Therefore, the nonsynonymous and synonymous sensitivities of tree $i$ at generation $t$ can be obtained from the fraction of each type of changes, and these metrics indicate the degree of tree $i$ being changed nonsynonymously or synonymously,

$$n_a^i(t) = \frac{c_a^i(t)}{c_a^i(t) + c_s^i(t)}, \quad n_s^i(t) = \frac{c_s^i(t)}{c_a^i(t) + c_s^i(t)}.$$

We add up the sensitivities of all individuals in the population to obtain the total nonsynonymous and synonymous sensitivities at the current generation,

$$N_a(t) = \sum_{i=1}^{S} n_a^i(t), \quad N_s(t) = \sum_{i=1}^{S} n_s^i(t).$$

Last, we define the nonsynonymous and synonymous substitution rates $k_a$ and $k_s$ at generation $t$ as

$$k_a(t) = \frac{M_a(t)}{N_a(t)}, \quad k_s(t) = \frac{M_s(t)}{N_s(t)}.$$

The rate $k_a(t)$ measures the rate of generating nonsynonymous adaptive changes. The rate $k_s(t)$ describes the rate of producing neutral changes in an evolutionary process. Without changes at the functional level, these neutral changes will not experience pressure in evolution. Thus, $k_s(t)$ practically provides "clock ticks" for the acceptance of mutation or crossover changes in a GP system. Since $k_a(t)$ measures the rate of accepted effective changes, the ratio $k_a(t)/k_s(t)$ represents the "evolutionary distance" in relation to the "evolutionary time", therefore, the rate of effective adaptation at generation $t$. Thus, we propose the rate of evolution $R_e$ in the GP tree population at generation $t$ to be

$$R_e(t) = \frac{k_a(t)}{k_s(t)}.$$

To verify the effectiveness of this measurement, we calculate $R_e$ using GP to solve a benchmark quintic polynomial symbolic regression problem $x^5 - 2x^3 + x$ (*ramped half-and-half* initialization, population size 4000, tournament size 4, crossover rate 0.9, and mutation rate 0.1). In this GP evolutionary process, plotting the above metrics shows the adaptive substitutions and indicates the rate at which the evolutionary search proceeds (Figure 1).

# 3. FUTURE WORK

The $R_e$ measurement can be extended in different ways. First, this measurement can be used to investigate the effectiveness of different evolutionary operators and parameters. Second, applications of this measurement to various methods in evolutionary computation need to be thoroughly investigated. Specific variants of the definition may be needed for different methods. Third, we propose to use this measurement for research on quantification of evolvability since it can reflect the evolutionary capabilities of a system.

# 4. ACKNOWLEDGEMENTS

The authors acknowledge support from NSERC Discovery Grant under RGPIN 283304-07.

# 5. REFERENCES

[1] W. Banzhaf, G. Beslon, S. Christensen, J. A. Foster, F. Kepes, V. Lefort, J. F. Miller, M. Radman, and J. J. Ramsden. From artificial evolution to computational evolution: A research agenda. *Nature Reviews Genetics*, 7(9):729–735, September 2006.

[2] Z. Yang and J. P. Bielawski. Statistical methods for detecting molecular adaptation. *Trends in Ecology and Evolution*, 15(12):496–503, December 2000.