A Bioinformatics Approach to Detect Plagiarism in Source Code

Kaio P. Gomes[1], **Simone N. Matos**[2]

[1]Department of Informatics
Federal University of Technology - Parana (UTFPR) – Ponta Grossa, PR – Brazil

[2]Department of Informatics
Federal University of Technology - Parana (UTFPR) – Ponta Grossa, PR – Brazil

kgomes@alunos.utfpr.edu.br, snasser@utfpr.edu.br

Palavras-chave: Programming plagiarism; Bioinformatics method, Source code

Abstract. The concern about plagiarism is growing in the programming scope, and new solutions are being proposed to handle the detection of plagiarized source code. Such an issue affects not only in the education field but also in the industries dealing with software development. For instance, at the academic level, the students can act with dishonesty on introductory programming course assignments, and at the industry level, the software reuse can be common practice even though using someone else work or ideas without authorization. Nowadays, several higher education institution has been adopting the usage of automatic tools for detecting academic dishonesty on its programming and related courses, and the outcomes reveal that the incident of plagiarism chopped considerably. Thus, New approaches based on different methods and techniques are being developed to address the programming plagiarism problem for the diverse emerging demand. According to the systematic mapping review in the present research, there are different approaches to detect plagiarism in source code. For instance, methods based on token, N-Gram, birthmark, Fuzzy, Bytecode, metric, graph, tree and others. The main solutions try to contemplate the most diverse types of programming plagiarism as well as improving the time complexity of its algorithms. This research proposes a novel approach based on a bioinformatics inspired method to improve plagiarism detection performance and accuracy. The main goal of this proposal is dealing with different types of modifications on plagiarized source code following the classification of programming plagiarism elaborated by [Faidhi and Robinson 1987]. The proposed approach works with the premise of modeling a source code into a synthetic DNA sequence and executing alignment among these types of sequence to identify similarities. The identified similarities correspond to a percentage rate, which indicates how similar source codes can be. This approach seeks to obtain a less computational cost in terms of time complexity comparing to consolidated tools available for this field such as JPLAG. The evaluation of this study considers test scenarios experimentations, which include 223 source codes collected from programming course assignments and others 30 created manually to insert different specific types of plagiarism.