BRNO UNIVERSITY OF TECHNOLOGY
FACULTY OF INFORMATION TECHNOLOGY

Excel@FIT 2024
http://excel.fit.vutbr.cz

LTR retrotransposons detection via Probabilistic Finite Automata

Lucie Klímová*

Abstract
LTR retrotransposons are often inserted into one another, which makes them hard to detect. This paper intends to show that it is possible to use Probabilistic Finite Automata (PFA) to accelerate the computation. There are several tools that are supposed to detect these transposable elements, but they widely vary in their runtime, sensitivity, specificity, and capability of detecting nesting. We decided to modify TE-greedy nester [1] because it is able to locate even deeply nested retrotransposons. To localize a transposon, it is necessary to detect its structural components, the sequences of which are available in protein databases. We used the ALERGIA algorithm to learn PFAs representing these databases to efficiently search for these domains.

*xklimo04@vutbr.cz, *Faculty of Information Technology, Brno University of Technology*

1. Introduction

The genetic information of most eukaryotic organisms contains transposable elements (TEs) inserted into the DNA sequence throughout evolution. LTR retrotransposons constitute approximately 8.3% of the human genome, as illustrated in Figure 1. This is a significant part of the genetic information, compared to the 1.5% that is constituted by protein-coding genes [2].

Some retrotransposons may be non-functional or have neutral effects, others have been found to play a crucial role in genome evolution and have regulatory functions, such as controlling gene expression [3]. It is, therefore, essential to localize them and determine the order in which they were nested.

The main task is to create a program that, given a nucleotide sequence as input, is able to find most of the TEs in a reasonable time. The main complications are the frequent nesting of TEs and mutations, which make it impossible to use exact matching algorithms.

Existing tools capable of detecting LTR transposons include, for example, LTR finder [4], which is relatively fast but unable to identify nested TEs. Another tool named RepeatMasker [5] first locates fragments of structural elements that could be part of a transposon and then tries to connect closely located fragments to form a whole TE, thus being able to detect some nesting. The tool we found the most interesting is TE-greedy nester. It is able to detect even deep nesting, but due to the recursive call of the algorithm on the entire input sequence, it appears relatively slow.

Since it was found experimentally that more than 80% of the TE-greedy nester's runtime is taken up by calling a tool named BLAST, we decided to use an alternative algorithm based on Probabilistic Finite Automata that could replace this slow part and thus speed up the whole process. We assume that replacing the BLAST tool with the PTA-based algorithm could significantly speed up the TE-greedy nester and enable the search for transposons even in longer sequences.

2. LTR retrotransposons

LTR retrotransposons are a type of TE that make up a significant part of the genome of many species. They consist of two Long Terminal Repeats (LTRs), typically 250–600 bp in length, at both 5' and 3' ends of the retrotransposon, as shown in Figure 2. Between these two LTRs is a coding region, approximately 5–7 kb long, that contains at least two genes, gag and pol, but the number can vary depending on the type of transposon. These genes encode proteins such as protease (PR) and reverse transcriptase (RT),

which are necessary for the transposon to replicate and move along the host DNA [6].

3. TE-greedy nester

TE-greedy nester is a command line tool that is able to detect even deeply nested LTR retrotransposons. Since older transposons are often fragmented by later inserted transposons, as shown in Figure 3, the program first locates the newest TE, which is then cut out of the original sequence, and the algorithm is repeated until no other transposon is found. This algorithm is described in Figure 4

Since it was found experimentally that more than 80% of the TE-greedy nester's runtime is taken up by calling a tool named BLAST, we decided to use an alternative algorithm based on Probabilistic Finite Automata that could replace this slow part and thus speed up the whole process. We assume that replacing the BLAST tool with the PTA-based algorithm could significantly speed up the TE-greedy nester and enable the search for transposons even in longer sequences.

4. Probabilistic Finite Automata for sequence comparison

Although it is not possible to use exact matching algorithms to determine whether a particular sequence might be the desired protein due to naturally occurring mutations, we can use Probabilistic Finite Automata (PFA) to describe the character of an entire set of sequences and then introduce some randomness by merging similar states. That can be achieved using the ALERGIA algorithm, which uses Inequation 1 to determine whether to merge two states.

4.1 Translation of amino acids into equivalence classes

Amino acids, the basic building blocks of proteins, can be divided into several groups according to their properties, such as polarity or acidity. The substitution of an amino acid with another amino acid from the same group may lead to a conservative replacement. It has been proved that these mutations are much more frequent because they do not cause a significant change in the functionality of the protein [7]. One of the possible divisions into equivalence classes is shown in Table 1.

Because the original ALERGIA algorithm did not prove to be very effective for this application, the same method of transcoding amino acids into these equivalence classes was used as in the master's thesis DNA Sequence Representation by Use of Statistical Finite Automata [7]. Thanks to this modification, we were able to identify approximately 80% of the sequences encoding the gag gene while maintaining a low number of false positives.

4.2 Dealing with non-determinism

Currently, our tool is only able to recognize sequences that exactly correspond to the given protein. In order to identify genes that are part of a longer sequence, the automaton, which is the output of the ALERGIA algorithm, must be modified. The modification is illustrated by Figure 5. It is necessary to add two more states, one at the beginning and the other at the end of the automaton, whereby both will read any symbol from the input with probability one so as not to affect the probability of the given string. The first state will be used to read the characters before the given gene and then non-deterministically go to the original initial state. The last state only serves to read the rest of the given string. Unfortunately, the resulting automaton is no longer deterministic, so a more complex algorithm will have to be used to determine the probability of a given string. It will be necessary to continuously store all possible configurations of the automaton and select those with the highest probability.

5. Conclusions

This paper showed that it may be possible to use the ALERGIA algorithm to learn a PFA representing a protein database and then search for LTR transposon domains in a query sequence using this automaton. To prioritize conservative replacements, which are much more frequent, we encoded the amino acids into classes of equivalents according to their properties. In future work, we would like to focus on the implementation of the extended PFA so that it can also accept sequences that contain the given domain as a substring and determine its exact position. We plan to integrate our code into the TE-greedy nester and test whether there was a significant acceleration of the algorithm.

Acknowledgements

I would like to thank my supervisors doc. Mgr. Lukáš Holík, Ph.D. and Mgr. Juraj Síč for their help.

References

[1] Matej Lexa Pavel Jedlicka Ivan Vanat Michal Cervenansky Eduard Kejnovsky. Te-greedy-nester:

structure-based detection of ltr retrotransposons and their nesting. *Bioinformatics*, 2020.

[2] Richard Cordaux and Mark A. Batzer. The impact of retrotransposons on human genome evolution. *Nature reviews. Genetics*, 2009.

[3] R. A. ELBARBARY. Retrotransposons as regulators of gene expression. *Science*, 2016.

[4] H. XU, Z. WAN. *LTR FINDER User Manual*.

[5] Repeatmasker documentation. Available at: https://www.repeatmasker.org/webrepeatmaskerhelp.html.

[6] L. ZHANG. The structure and retrotransposition mechanism of ltr-retrotransposons in the asexual yeast candida albicans. *Virulence*, August 2014.

[7] A. SHAH. Dna sequence representation by use of statistical finite automata, 2009.