Pipeline to detect the relationship between transposable elements and adjacent genes in host genome

Caroline Meguerditchian[1], Ayse Ergun[1], Veronique Decroocq[1], Marie Lefebvre*[1], and Quynh-Trang Bui*[1]

[1]UMR 1332 Biologie du Fruit et Pathologie, INRAE, University of Bordeaux, UMR BFP, Villenave-d'Ornon, France

1 Abstract

Understanding the relationship between transposable elements (TEs) and their associated genes in the host genome is a key point to explore their potential role in genome evolution. Transposable elements can regulate and affect gene expression not only because of their mobility within the genome but also because of its transcriptional activity. Gene expression can be suppressed, decreased or increased and cellular signalling pathways can be activated through the act of the nearby TE expression itself or subsequent TE replication intermediates. We implemented a pipeline which is capable to reveal the relationship between TEs and adjacent gene distribution in the host genome. Our tool is freely available here : https://github.com/marieBvr/TEs_genes_relationship_pipeline

2 Introduction

Transposable elements (TEs) were first discovered in maize by Barbara McClintock in 1944 [1]. They are mobile repetitive DNA sequences, and correspond to a significant part of eukaryotic genomes. These elements make up nearly half of the human genome, and about 85% of the maize genome. There are many different types and structures of TEs [2], but they can be divided into two major classes: Class I contains elements called *retrotransposons* and Class II groups DNA transposon elements. These two classes are distinguished by their transposition mechanisms. The retrotransposons are firstly transcribed into RNA then retrotranscribed into double-stranded sequences to integrate into the genome, whereas the DNA transposons achieve their mobility based on an enzyme called transposase, which helps to catalyze DNA elements from its original location to insert into a new site within the genome. Due to their role in transposition and the act of transcription itself, the TEs can regulate gene expression by modifying the closest reading frames into pseudogenes, by triggering alternative splicing or by modifying epigenetic marks [3], [4], [5], [6]. Studying the relationship between TEs and the surrounding genes will help determine the important role of TEs as well as their contribution to the plasticity of eukaryotic genomes and the evolution and adaptation of species. Many existing tools allow the users to predict TE locations in the genome, such as LTRpred [7], PiRATE [8] [9] or REPET [10], but none of them can reveal the relationship between TEs and host coding sequences. In order to help biologists in studying TE impacts upon gene expression, we developed a pipeline allowing users to report the relationship of TEs with their closest genes, such as its location and distance to adjacent genes within the genome. Our pipeline also provides a graph visualization in R.

3 Materials and methods

3.1 General workflow

Two input files are needed, a GFF file annotation of the host genome and a TSV file containing information about TE annotation. The Apricot dataset and two different TE annotations were used to implement our pipeline: TE annotation from the REPET package and LTR annotation from LTRpred software. The gene and TE annotation files are firstly optimized for data analysis, by removing duplicate lines and sorting out TEs in order to increase information retrieved from their position in the genome. Then, the files are processed with Python in order to analyse the relationship between nearest genes and TEs, and detect for each TE its upstream, downstream, subset and superset genes. The

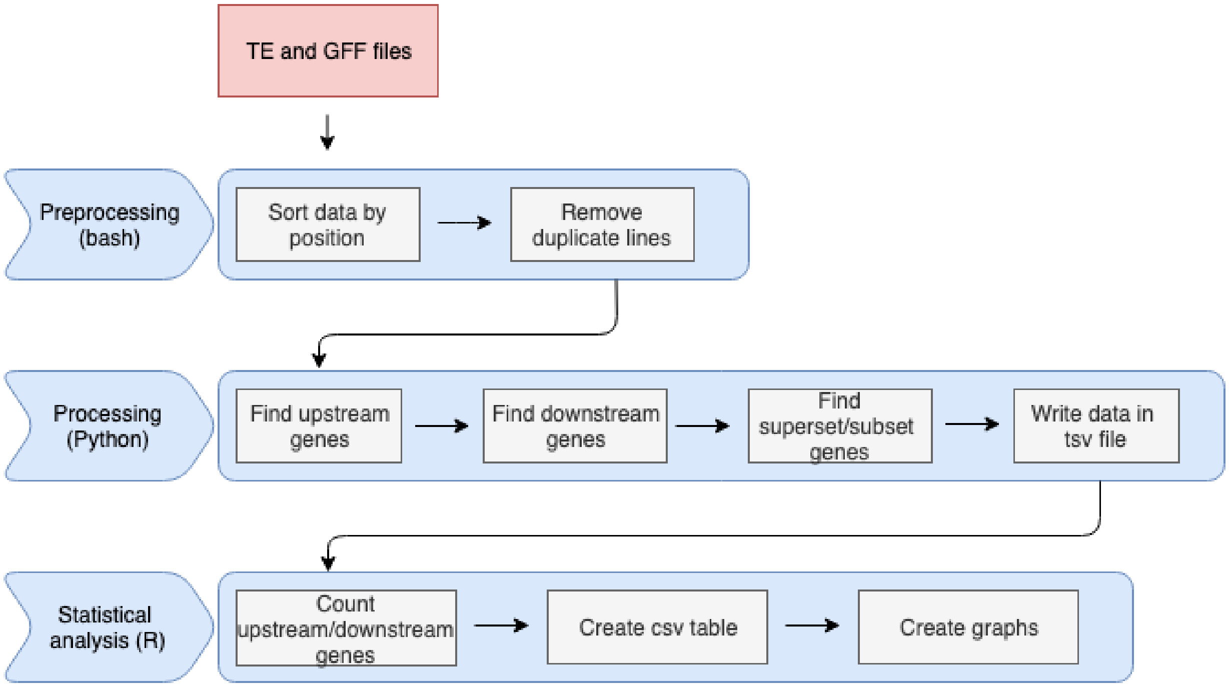

Figure 1: The workflow of the pipeline comprises three different steps: preprocessing, processing and statistical analysis.

Python output result is then used in R to create tables and graphs to visualize TE-coding sequence relationships as shown on Figure 1.

3.2 Implementation

The first part of the algorithm was written using Python 3.7. The program contains six functions aiming to detect genes surrounding any TE, as well as other functions allowing to read the input data and to write the output in a TSV file. Here is a summary of those functions:

- check_upstream_gene: this function returns the closest gene to the TE positioned in an upstream location.
- check_downstream_gene: this function returns the closest gene to the TE positioned in an downstream location.
- check_upstream_overlap_gene: this function returns gene with at least one base overlapping the upstream part of the TE.
- check_downstream_overlap_gene: this function returns gene with at least one base overlapping the the downstream part of the TE.
- check_subset_superset_gene: this function searches for gene, which is either a subset or a superset of the TE.
- calculate_distance: this function allows to calculate the distance between TE and its closest gene.
- write_data: this function writes an output file containing all the information about the genes detected by the previous functions.

The output file is a TSV file containing the following information: upstream genes, downstream genes, genes as superset of TEs, genes as subset of TEs. For each case, the start position, end position, ID and strand information are recorded. The TE ID, type, strand and position are also reported.

Graphical visualization of the data is obtained by R scripts with the 4.0.2 version. Using the output file from the Python script, descriptive statistics of the relation between TEs and genes in the genome are presented. The output of the R scripts is graphs and CSV files containing the values counted by the algorithm. There are three R scripts allowing users to report three different statistics:

- TE statistics, which show how many TEs and what type of TE are present in the file and the distribution of TEs between those types.

- Overlap statistics, which show how many TEs have an overlap with genes, both upstream and downstream.
- Distance statistics, which show the number of TE with an upstream or downstream gene within 0-500 bp, 500-1000 bp, 1000-2000 bp and more than 2000 bp intervals.

3.3 Operation

Our workflow requires R 4.0.2 or upper, Python 3.7 and can be run on any operating system with common specifications (1Go disk space, 4Go RAM, multicore CPU is recommended).

4 Use case

In order to explain how the program works, we will use two example files named gene_testing_data and transposon_testing_data, which gather all possible scenarios of the position of TEs relative to their closest genes. These two input files can be downloaded from the github page. The following command must be run in order to execute the program :

```
python3 Multiprocessing/Create_Data_multipro.py \
        -g data/Gene_testing_data.tsv \
        -te data/Transposon_testing_data.tsv \
        -o result/output_TE.tsv
```

The -g argument is used to provide the gene file, the -te is used to come up with the TE file and the -o to supply the name of the result file. The output file is in TSV format. Once this file has been obtained, the R analysis can be run by adapting the following script line :

```
Rscript Rscript/number_te.r \
        -f result/output_TE.tsv \
        -o result/count_TE_transposons.pdf

Rscript Rscript/Overlap_counting.r \
        -f result/output_TE.tsv \
        -p result/overlap_TE_results.pdf \
        -o result/overlap_TE_results.csv

Rscript Rscript/Distance_counting.r \
        -f result/output_TE.tsv \
        -p result/distance_TE_results.pdf \
        -o result/distance_TE_results.csv
```

The R scripts will provide output files with the statistics regarding overlapping genes and TEs as shown on Figure 2, or distance between LTRs and genes as shown on Figure 3, in table format as well as graph visualization.

5 Conclusion

Transposable elements are repetitive DNA sequences that have the ability to change their position within the genome. These mobile elements play an important role in gene regulation and have a large impact on genome evolution. We have developed the first pipeline which can report directly the relationship between TEs and its nearest genes among the genome. The accuracy of the tool has been verified by using a test dataset and also running on two different TE annotation software. This pipeline could be useful to reveal potential effects of TEs on gene expression as well as on the study of specific gene function.

6 Software availability

Up-to-date source code, and tutorials are available at: https://github.com/marieBvr/TEs_genes_relationship_pipeline
Archived source code as at time of publication are available from: https://doi.org/10.5281/zenodo.4442377

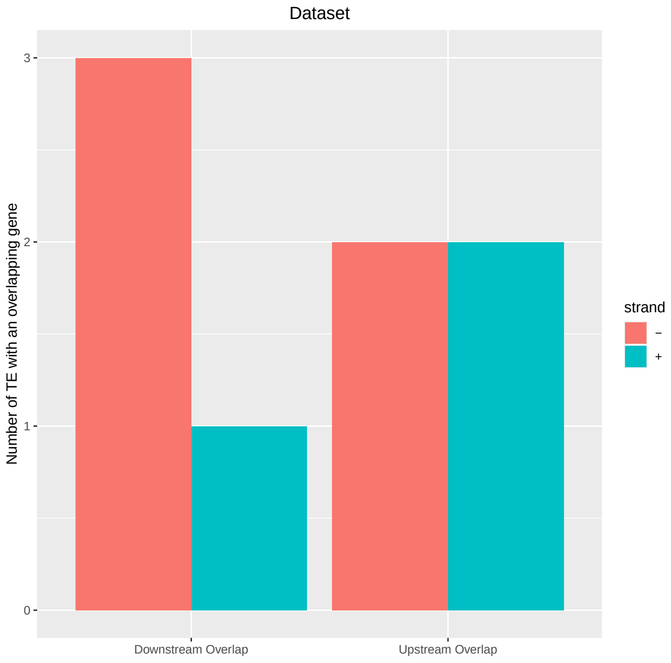

Figure 2: Number of TEs with a downstream overlapping gene and upstream overlapping gene.

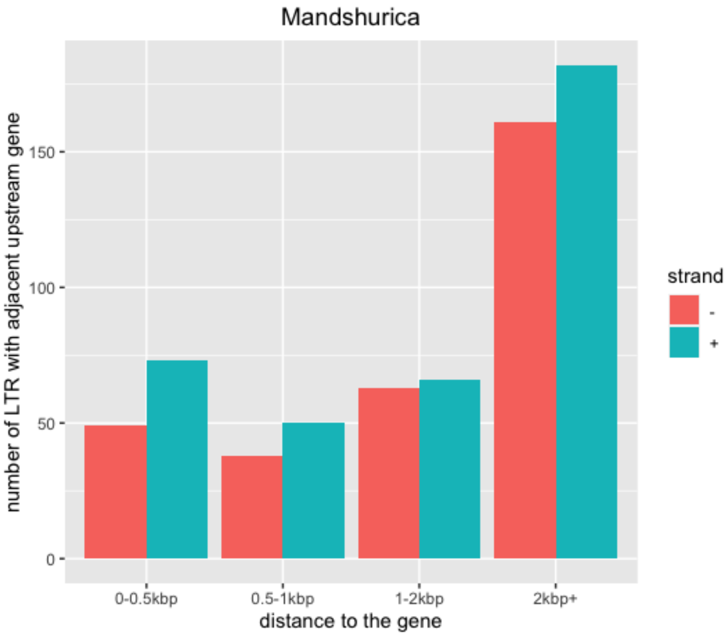

Figure 3: Number of LTRs with an adjacent upstream gene for the Prunus specie Mandshurica

Competing interests

No competing interests were disclosed.

Grant information

The authors declared that no grants were involved in supporting this work.

References

[1] M. Barbara. "The Origin and Behavior of Mutable Loci in Maize". In: *Proceedings of the National Academy of Sciences of the United States of America* 36 (1950), pp. 344–355.

[2] M. Tollis and S. Boissinot. "The evolutionary dynamics of transposable elements in eukaryote genomes". In: *Genome Dynamics* 7 (2012). DOI: `https://doi.org/10.1159/000337126`.

[3] Q. Bui and M. A. Grandbastien. *Plant Transposable Elements*. Springer, 2012, pp. 273–296.

[4] M. Percharde et al. "A LINE1-Nucleolin Partnership Regulates Early Development and ESC Identity". In: *Cell* 174 (2018), Issue 2, 391–405.e19.

[5] J. Cho and J. Paszkowski. "Regulation of rice root development by a retrotransposon acting as a microRNA sponge". In: *eLife* 6 (2017), p. 796.

[6] R. Petri et al. "LINE-2 transposable elements are a source of functional human microRNAs and target sites". In: *PLOS Genetics* 15.3 (Mar. 2019), pp. 1–18. DOI: `https://doi.org/10.1371/journal.pgen.1008036`.

[7] H. G. Drost. "LTRpred: de novo annotation of intact retrotransposons". In: *Journal of Open Source Software* 5 (2020).

[8] J. Berthelier et al. *PiRATE: a Pipeline to Retrieve and Annotate Transposable Elements*. 2018. DOI: `https://doi.org/10.17882/51795`. URL: `https://www.seanoe.org/data/00406/51795/`.

[9] H. Quesneville et al. "Combined Evidence Annotation of Transposable Elements in Genome Sequences". In: *PLOS Computational Biology* 1 (2005). DOI: `https://doi.org/10.1371/journal.pcbi.0010022`.

[10] T. Flutre et al. "Considering Transposable Element Diversification in De Novo Annotation Approaches". In: *PLOS One* 6 (2011). DOI: `https://doi.org/10.1371/journal.pone.0016526`.