Pipeline to detect the positional 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 closest positional 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 their transcriptional activity. A comprehensive knowledge of structural organization between transposable elements and neighboring genes is important to study TE functional role in gene regulation. We implemented a pipeline, which is capable to reveal the positional 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 1948 [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 [2], and about 85% of the maize genome [3]. There are many different types and structures of TEs [4], 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 transposition and the act of transcription itself, the TEs can regulate gene expression by modifying genes into pseudogenes, by triggering alternative splicing or by modifying epigenetic marks [5], [6], [7], [8]. Experimental findings hint that TE insertions are targeted beyond the apparent mechanistic necessity of open chromatin [9]. Other examples have argued the possibility that more complex molecular mechanisms have evolved and made evident that TEs are an integral part of the regulatory toolkit of the genome [10], [11]. Studying the relationship between TEs and the surrounding genes will help determine the role of TEs in the many layers of gene expression regulation as well as their contribution to the plasticity of eukaryotic genomes and the evolution and adaptation of species. Many existingtools allow the users to predict TE locations in the genome, such as LTRpred [12], PiRATE [13] or REPET [14],[15]. Others tools can reveal the relationship between TEs and host sequences at transcriptome level. LIONS [16] is one of them that detects and quantifies transposable element initiated transcription from RNA-seq. Another software that can provide the screening and selection of potentially important genomic repeats is GREAM (Genomic Repeat Element Analyzer for Mammals) [17]. However, this web-server can offer the TE-gene neighborhood only within mammalian species. There are no tools for revealing positional relationship between TEs and host coding sequences that can be applied for any species. In order to help biologists in studying TE impacts upon gene expression, we developed a pipeline allowing users to report the association between TEs and their closest genes, such as its location and distance to adjacent genes within the genome. Our dedicated command line tool can be used with TE annotation from any prediction tool as well as any genome assembly and annotation. This 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 (*Prunus mandshurica*) and two different TE annotations were used to implement our pipeline: TE annotation from the REPET package and LTR retrotransposon annotation from LTRpred software.

The gene and TE annotation files are firstly optimized for data analysis, by removing duplicate lines and sorting out by chromosome and by start position in ascending order. Then, the files are processed with Python in order to analyze the relationship between nearest genes. Our pipeline could detect and report all scenarios that could occur between TEs and the closest genes in the host genome as shown in the Figure 1. The Python output result is then used in R to create tables and graphs to visualize TE-gene relationships. The whole workflow is descripted in Figure 2.

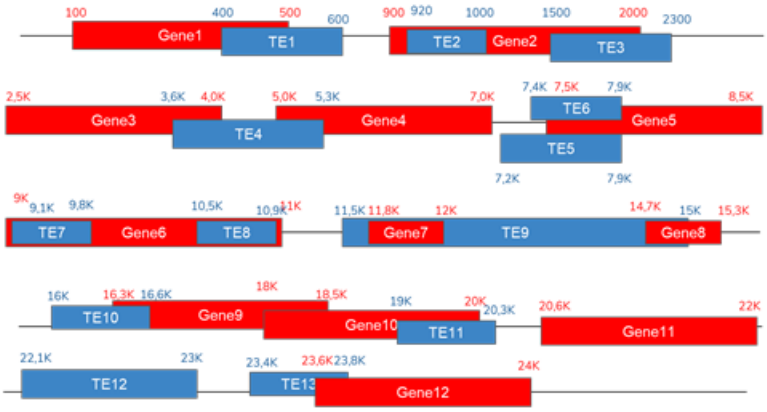

Figure 1: Scenarios of relationship between TEs and the closest genes in the host genome.

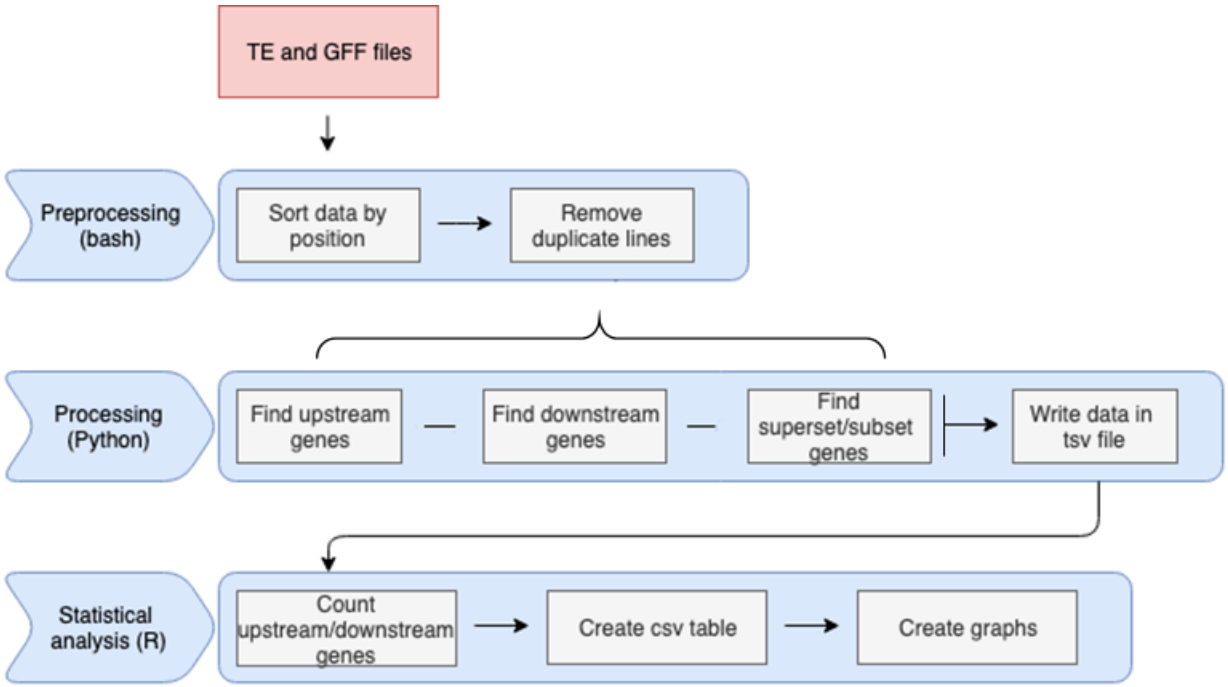

Figure 2: The workflow of the pipeline comprises three different steps: preprocessing, processing and statistical analysis.

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. Examples from Figure 1: Gene1-TE1; Gene4-TE5/TE6.
- check_downstream_gene: this function returns the closest gene to the TE positioned in a downstream location.

Examples from Figure 1: Gene9-TE10; Gene11-TE11.

- check_upstream_overlap_gene: this function returns genes with at least one base overlapping the upstream part of the TE. Example from Figure 1: Gene1-TE1.
- check_downstream_overlap_gene: this function returns genes with at least one base overlapping the downstream part of the TE. Example from Figure 1: Gene9-TE10.
- check_subset_superset_gene: this function searches for genes, which are either a subset (Gene7-TE9) or a superset (Gene6-TE7/TE8) of the TE.
- calculate_distance: this function allows to calculate the distance between a 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 python output is a TSV file containing the TE-gene relationship information based on the physical structure of the genome as descripted above. 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 types of TEs 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 TEs with an upstream or downstream gene within 0-500 bp, 500-1000 bp, 1000-2000 bp and more than 2000 bp intervals.

The genes reported as upstream or downstream of TEs in R scripts were defined based on the TEs' strand as described in Figure 3.

Figure 3: Representation of downstream and upstream genes based on autonomous TEs' strand.

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 runby 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 LTR retrotransposons as shown in Figure 4, or distance between LTR retrotransposons and genes as shown in Figure 5, in table format as well as graph visualization.

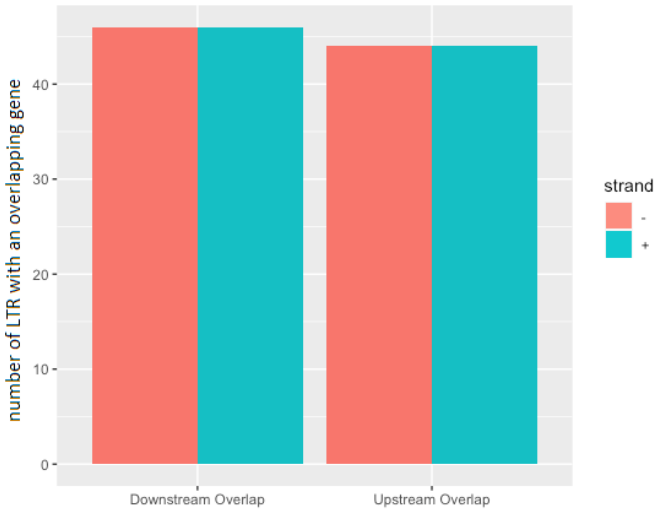

Figure 4: Number of sense LTR retransposons (strand+) and antisense LTR retransposons (strand-) with a downstream-overlap gene and upstream-overlap gene in *Prunus mandshurica*.

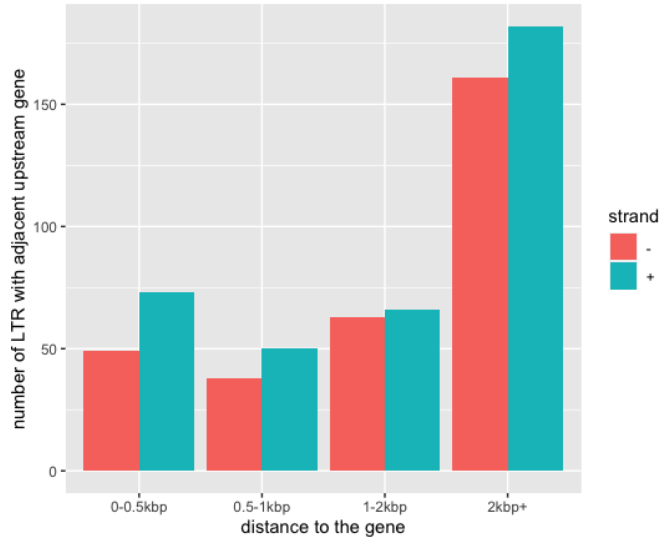

Figure 5: Number of sense LTR retransposons (strand+) and antisense LTR retransposons (strand-) with an adjacent upstream gene in *Prunus mandshurica.*

5 Conclusion

Transposable elements are repetitive DNA sequences that have the ability to move within the genome. These mobile elements can 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 running on two different TE annotation softwares. This pipeline could be useful to subsequently analyze potential effects of TEs on gene expression as well as on specific gene function.

6 Software and data 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.
The raw data of the Apricot genome, *Prunus mandshurica*, are available on the European Nucleotide Archive (ENA) under the project name PRJEB42606: https://www.ebi.ac.uk/ena/browser/view/PRJEB42606 and the assembled genome is available at https://www.rosaceae.org/node/10811682.

7 Competing interests

No competing interests were disclosed.

8 Grant information

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

9 References

[1] B. McClintock. “Mutable loci in maize”. In: *Carnegie Institution of Washington Yearbook* 47 (1948), pp. 155-169.

[2] A. P. Jason de Koning et al. “Repetitive Elements May Comprise Over Two-Thirds of the Human Genome”. In: *PLOS Genetics* 7.12 (2011), pp. 1–12. DOI: 10.1371/journal.pgen.1002384.

[3] Sarah N. Anderson et al. “Transposable elements contribute to dynamic genome content in maize”. In: *The Plant Journal* 100.5 (2019), pp. 1052–1065. DOI: https://doi.org/10.1111/tpj.14489.

[4] 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.

[5] QT. Bui and M. A. Grandbastien. “LTR retrotransposons as controlling elements of genome response to stress?”. In: *Plant Transposable Elements*. Springer (2012), pp. 273–296.

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

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

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

[9] T. Mourier et al. “Transposable elements in cancer as a by-product of stress-induced evolvability”. In: *Front. Genet. 5* (2014). DOI: 10.3389/fgene.2014.00156.

[10] R.K. Slotkin and R. Martienssen. “Transposable elements and the epigenetic regulation of the genome”. In: *Nat. Rev. Genet.* 8 (2007), pp. 272–285. DOI: 10.1038/nrg2072.

[11] D. Drongitis et al. “Roles of Transposable Elements in the Different Layers of Gene Expression Regulation”. In: *Int J Mol Sci 20* (2019). DOI: 10.3390/ijms20225755.

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

[13] J. Berthelier et al. “PiRATE: a Pipeline to Retrieve and Annotate Transposable Elements”. In: *BMC Genomics* (2018). DOI: https://doi.org/10.17882/51795.

[14] 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.

[15] 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.

[16] A. Babaian et al. “LIONS: analysis suite for detecting and quantifying transposable element initiated transcription from RNA-seq”. In: *Bioinformatics* 35 (19) (2019), pp. 3839–3841. DOI: https://doi.org/10.1093/bioinformatics/btz130.

[17] D.S. Chandrashekar et al. “Web Server to Short-List Potentially Important Genomic Repeat Elements Based on Over-/Under-Representation in Specific Chromosomal Locations, Such as the Gene Neighborhoods, within or across 17 Mammalian Species”. In: *PLOS ONE* 10.7 (2015), pp. 1–17 DOI: https://doi.org/10.1371/journal.pone.0133647.