Simulation-based estimation of branching models for LTR retrotransposons

Serge Moulin, Nicolas Seux, Stéphane Chrétien, Christophe Guyeux, Emmanuelle Lerat

Bioinformatics
doi.10.1093/bioinformatics/xxxxxx
Advance Access Publication Date: Day Month Year
Manuscript Category

OXFORD

Genome analysis

Simulation-based estimation of branching models for LTR retrotransposons

Serge Moulin [1]*, Nicolas Seux [2], Stéphane Chrétien [3], Christophe Guyeux [1], and Emmanuelle Lerat [4,]

[1]Département d'Informatique des Systèmes Complexes, FEMTO-ST Institute, UMR 6174 CNRS, Université de Bourgogne Franche-Comté, 15 Bis Avenue des Montboucons, 25030 Besançon, France
[2]Laboratoire de Mathématiques de Besançon, UMR 6623 CNRS, Université de Bourgogne Franche-Comté, 16 route de Gray, 25030 Besançon, France
[3]National Physical Laboratory, Hampton Road, Teddington, United Kingdom and
[4]Laboratoire Biometrie et Biologie Evolutive, Universite Claude Bernard - Lyon 1, UMR 5558 CNRS, 43 boulevard du 11 novembre 1918, 69622 Villeurbanne, France

Associate Editor: XXXXXXX

Received on XXXXX; revised on XXXXX; accepted on XXXXX

Abstract

Motivation: LTR retrotransposons are mobile elements that are able, like retroviruses, to copy and move inside eukaryotic genomes. In the present work, we propose a branching model for studying the propagation of LTR retrotransposons in these genomes. This model allows us to take into account both the positions and the degradation level of LTR retrotransposons copies. In our model, the duplication rate is also allowed to vary with the degradation level.
Results: Various functions have been implemented in order to simulate their spread and visualization tools are proposed. Based on these simulation tools, we have developed a first method to evaluate the parameters of this propagation model. We applied this method to the study of the spread of the transposable elements ROO, GYPSY, and DM412 on a chromosome of *Drosophila melanogaster*.
Availability: Our proposal has been implemented using Python software. Source code is freely available on the web at https://github.com/SergeMOULIN/retrotransposons-spread.
Contact: serge.moulin@univ-fcomte.fr
Supplementary information: Supplementary data are available at *Bioinformatics* online.

1 Introduction

A transposable element (TE) is a DNA sequence able to move from one location to another inside a genome. These sequences, discovered during the 50's by Barbara McClintock (**?**) exist in almost all living organisms and are the source of a huge number of mutations. They are considered as a major cause of genetic disease in human (Belancio *et al.*, 2008) or in *Drosophila* where they are responsible for more than 80% of the spontaneous mutations (Green, 1988). DNA sequences derived from these TEs can represent a large part of a genome. For example, they represent about 45% of the human genome (Lander *et al.*, 2001) and over 70% of the corn genome (Sanmiguel and Bennetzen, 1998). Fortunately, most of these sequences correspond to fragments or "dead" elements that have lost their ability to move in the genome due to several lethal mutations or are controlled, especially via epigenetic mechanisms.

TEs have two possible ways to move in a genome, according to their type (**?**) (Wicker *et al.*, 2007). The first class of mobile elements are cut from their original place to move to another one, and are called "DNA transposons" or "Class II transposable elements". The other class of mobile elements, called "retrotransposons" or "Class I transposable elements", use an RNA intermediate to duplicate themselves, the new copy being inserted into another location of the genome. Two orders are identified among the retrotransposons according to the presence or absence of Long Terminal Repeat (LTR) sequences at their extremities. The LTR retrotransposons are similar in structure to retroviruses such as HIV. In both classes, TEs can be classified as either "autonomous", if they encode the enzymes that will allow them to move, or "non autonomous" if they use the enzymes produced by other elements. In an assembled genome,

the various sequences corresponding to TE insertions can be found using different bioinformatic approaches (see (Lerat *et al.*, 2011) for a review), which allow us to determine the exact number and positions of each TE insertion. In this article, we focused on the important problem of inferring the history of the spreading of LTR retrotransposons. For this purpose, we modeled the evolution using a branching process where each element (*i.e.*, a copy of a given TE) can randomly evolve via duplication or mutation.

Instances of branching processes have already been proposed in the literature, as putative models for the propagation of TEs. However, most of these studies focus on the evolution of the host population, and not on the propagation of the TEs in the host. The "subject" of these branching models (*i.e.*, the entity able to change or duplicate) is generally the host, while in our case it is the TE itself. For instance, Michael E. Moody (Moody, 1988) has used a branching model, where the studied variable was the number of individuals owning i copies of a given TE. Sawyer *et al.* (Sawyer *et al.*, 1987) produced almost the same model, in order to study the distribution and abundance of insertion sequences.

Kaplan *et al.* proposed a model where TEs can be either of wild type (*i.e.*, non mutated) or of mutant one, which is a little closer to our proposal. When a host gives birth to its child, wild copies can mutate or be deleted, whereas mutant ones can only be removed. New copies can be additionally created. This number of new created copies is supposed to decrease with the proportion of mutants. More recently, interesting models have been proposed that take into account the location of TEs. For instance, Drakos and Wahl (Drakos and Wahl, 2015) suggested a model of mobile promoter evolution, where the probabilities for promoters to duplicate inside or outside their region is potentially not the same.

In the present work, the objective is to propose a new approach for the propagation of LTR retrotransposons that combines a location-dependent model with the fact that LTR retrotransposons can face degradation (*i.e.*, mutations, recombination, etc.), which may decrease their duplication rate, that is, their potentiality to copy and insert elsewhere in the genome. Then, we have developed a first method to evaluate the model parameters: average distance traveled by the TEs before insertion, location of the original copy, average time between two degradations (mutations, recombination, etc.), average impact of a degradation, and the impact of degradations undergone by copies on their duplication speed. This method requires to define a distance between the results of the simulations and the observed chromosome, which is based on the Hungarian method (Kuhn, 1955; Munkres, 1957). This method has been applied to the spreading of the LTR retrotransposons ROO, DM412, and GYPSY on the chromosome 3L of *Drosophila melanogaster*. The parameters associated to each TE are computed and a branching tree is proposed in each case. Our results show that, according to our model and method, the roots of ROO, DM412, and GYSPSY on the chromosome 3L could correspond to the annotated copies FBti0059644, FBti0061034, and FBti0062705 respectively.

2 System and methods

2.1 The branching model

2.1.1 The branching tree

An example of branching tree is shown in Figure 1.

This branching tree represents the spread of the LTR retrotransposon "ROO" between times 0 and 5.601. At time 0 there is only one copy, called the "root" in this article. At time 1.488, the root duplicates itself to give birth to its first "child".

Finally, the process ends at time T_{obs} = 5.601 with 32 copies. The branching tree represents only the duplications, but copies are also subject to degradations as explained in Section 2.1.2. The state of the tree (*i.e.*, number of TE copies, copy positions, degradations undergone by the copies) at time T_{obs} = 5.601 is named "final state" of this tree. The

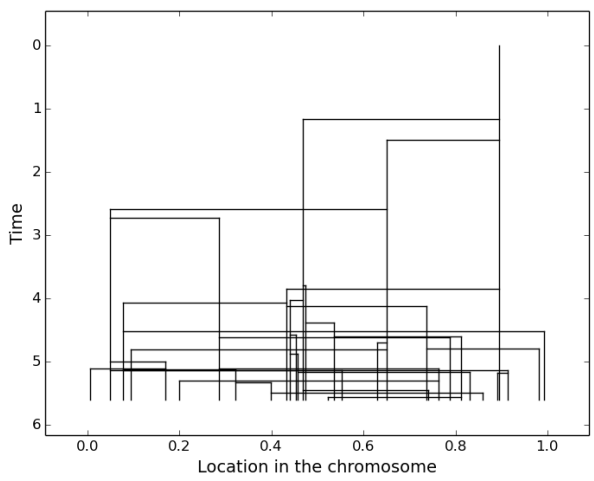

Fig. 1. ROO spread

working principle of our estimation method is to simulate trees, in order to determine in which conditions the final states of simulated trees match well with the observed chromosome. To compare the final state of a simulated tree with the observed chromosome, a distance was build, as detailed in Section 2.2.2.

2.1.2 The general model

The branching model is constructed as follows.

1. The spread starts with only one copy, called the root, at time zero in a location X_0 to be determined.
2. Each copy can either be duplicated or undergo degradations (*i.e.*, mutations, recombinations, etc.) at any time.
3. The number of copies increases due to duplications. When a new copy is created, it receives an index equal to the number of existing copies at the time of its birth, including itself. In the remainder of this article, let T_i be the birth date of the i^{th} copy and let $\tau_{i,k}$ be the time when the i^{th} copy faces its k^{th} degradation.
4. The time interval $\tau_{i,k+1} - \tau_{i,k}$ between two degradations is supposed to be independent and identically distributed (i.i.d.) with an exponential distribution $\mathcal{E}(\frac{1}{\mu})$, where μ (*i.e.*, average time between two degradations) must be determined.
5. Each copy is also associated to its Needleman-Wunsch (Needleman and Wunsch, 1970) similarity to the original state of the root. This similarity is a value between 0 and 1. Let us denote $S_i(t)$ the similarity between the original state of the root and the state of the i^{th} copy at time t. This similarity decreases as a function of time, due to degradation effects. In addition, we defined the *state of deterioration* by $D_i(t) = 1 - S_i(t)$.
6. At each degradation, the similarity to the root is divided by $1+E(\frac{1}{\beta})$, where β has to be determined. In other terms, $S_i(\tau_{i,k+1}) = \frac{S_i(\tau_{i,k})}{1+E(\frac{1}{\beta})}$.
7. At time t, for the i^{th} copy, conditionally on $D_i(t)$, we assume that the time before the next duplication follows a distribution $\mathcal{E}(\frac{1}{1+p\times D_i(t)})$, where $p > 0$ is a parameter to be determined. In other words, the time before the next duplication is longer when the copy is far from the original state of the root (in terms of Needleman-Wunsch distance). Note that, in this article, "duplication rate" is just

the inverse of the "average time before the next duplication". In other words, duplication rate = $1 + p \times D_i(t)$.

8. Moreover, each copy is also associated to its position in the chromosome. This position is denoted by X_i for the i^{th} copy. This position is constant with time. We assume that each child j of a copy i satisfies $X_j = X_i + \chi_{i,j}$, where $\chi_{i,j}$ follows a distribution $\mathcal{U}\{-1, 1\} \times \mathcal{E}(\frac{1}{L})$, in which $\mathcal{U}$ represents the uniform law (*i.e.*, the probability to choose -1 or 1 is the same) and L is a parameter to be determined.
9. We also take into account the host structure (*i.e.*, position of host genes) to insert or not the child in the chromosome. Concretely, we calculate

$$\text{density} = \frac{\frac{\text{Number of TEs in genes}}{\text{Surface occupied by genes}}}{\frac{\text{Number of TEs out of genes}}{\text{Unoccupied surface}}}$$

on the real chromosome. During simulations, when the child moves into a gene, it can be inserted with a probability equal to "density", otherwise the child position is recomputed (if "density" > 1, the child can always insert itself). Furthermore, the child position is also relaunched when it goes outside the chromosome (*cf.* Part 3.1.3).

Our goal is thus to estimate the parameters of this model, *i.e.*, $X_0, \mu, \beta, p,\ L$ and T_{obs}. Note that the duplication speed of the non-degraded root is set to 1 and it does not need to be determined. Indeed, this duplication speed is redundant with μ and p. In addition, note that the parameter L is not really informative about the mean distance traveled by the child before its insertion, due to putative relaunching processes. This is why we also provide the mean traveled distance in simulations, that is, the mean jump, which is denoted by J. See Part 3.1 for an example of its computation.

2.2 The estimation method

As explained in Section 2.1.1, the working principle of our estimation method is to simulate trees in order to determine in which conditions the final states of simulated trees match well with the observed chromosome.

Trees are simulated according to the model defined in Section 2.1.2. The stopping criterion of these simulated trees depends on the number of copies in the observed chromosome. Actually, the simulation was constrained to stop at the birth date of the $n + 1^{th}$ copy, where n is the number of copies in the observed chromosome, see Section 3.1.2 for further details.

In order to improve computation speed, parameters estimation has been split in three parts, i) we estimate μ, β, and L (*cf.* Section 2.2.1); ii) we estimate X_0 and L (*cf.* Section 2.2.2), which requires to define a distance between simulated trees and the observed chromosome; iii) we estimate J and T_{obs} (*cf.* Section 2.2.3).

2.2.1 Estimation of μ, β, and p

The objective here is to estimate the parameters μ, β, and p. We can note that these parameters only affect the distribution of deterioration states. They have no direct influence on copy positions. Thus, the goal at this step is to minimize the differences between the distribution of states of deterioration in the simulated trees, and the states of deterioration distribution in the observed chromosome. More precisely, we want to minimize $D1(Tr, C) = \sum_{i=1}^{n}(D_{i,Tr} - D_{i,C})^2$ where $D_{1,Tr}...D_{n,Tr}$ is the sorted distribution of the states of deterioration for the simulated tree, $D_{1,C}...D_{n,C}$ is the sorted distribution of the states of deterioration for the observed chromosome, and n is the number of TEs in the observed chromosome.

For this purpose, a 3-dimensional grid has been constructed, where each point of this grid represents a triplet (μ, β, p), while X_0 and L are set to predefined values (they do not matter at this stage). A score $S_1 = \sum_{i=1}^{N1} D1(Tr_i, C)$ has been associated to each of these triplets. In this formula, Tr_i is the i^{th} tree simulated with the parameter set, C represents the observed chromosome, and $N1$ is a parameter chosen by the user of the optimization method (for instance, $N1$ = 10,000 in the case study of Section 4). The best of these points is selected, and a smaller grid is constructed around it. This iterative process is continued until the precision chosen by the user of the optimization method has been obtained.

2.2.2 Distance between trees, estimation of X_0 and L

The first step, in the estimation of X_0 and L, is to define a distance between the final state of a simulated tree and the observed chromosome. For this purpose, we first need to design a distance between a copy of the simulated tree and a copy of the observed chromosome. Let us name R_i the i^{th} copy of the simulated tree, X_i its position, and $D_i(T_{obs})$ its state of deterioration at the end of the process. Similarly, $\widetilde{R}_j$ is the j^{th} copy of the observed chromosome, $\widetilde{X}_j$ its position, and $\widetilde{D}_j(T_{obs})$ its state of deterioration. The distance between two copies has been designed as follows:

$$D2(R_i, \widetilde{R}_j) = \frac{(X_i - \widetilde{X}_j)^2}{w_1} + \frac{(D_i(T_{obs}) - \widetilde{D}_j(T_{obs}))^2}{w_2}$$

where $w_1 = \sum_{i=1...n, j=1...n} (\widetilde{X}_i - \widetilde{X}_j)^2$ and $w_2 = \sum_{i=1...n, j=1...n} (\widetilde{D}_i(T_{obs}) - \widetilde{D}_j(T_{obs}))^2$. This weighting by w_1 and w_2 allows us to give the same weight to positions and states of deterioration.

From these distances between two copies, we can now create a matrix of distances W, verifying $W_{ij} = D2(R_i, \widetilde{R}_j)$. Then, the distance between final states of two trees has been defined as the best possible adjustment between copies, using the so-called Kuhn-Munkres algorithm, also named the Hungarian method (Kuhn, 1955; Munkres, 1957). The Hungarian method is an algorithm that allows us to minimize the sum of n elements of a $n \times n$ matrix, under the condition that there is only one element by row and only one element by column. In our case, the Hungarian method allows us to assign exactly one copy of the simulated tree to each copy of the real chromosome while minimizing the sum of distances between paired copies. Let us name $D3$ this distance created by this way.

Once this distance between trees has been defined, we use it to estimate X_0 and L with the same type of process as in the previous step. In other words, a 2-dimensional grid has been constructed, where each point of this grid represents a couple of parameters (X_0, L) while μ, β, and p are fixed to the values found in the previous stage. The score $S_2 = \sum_{i=1}^{N1} D3(Tr_i, C)$ is then computed for each of these points, and a smaller grid is recursively built around the best point.

2.2.3 Estimation of J and T_{obs}

Conversely to these μ, β, p, X_0, and L, which are inputted in our simulation algorithm, J (mean jump) and T_{obs} are outputs. It is thus easier to estimate them. In this step, we run $N2$ simulations where μ, β, p, X_0, and L are set to the values found in the two previous steps. The estimations of J and T_{obs} are then the mean results of the output J and T_{obs} of these $N2$ simulations (for instance $N2$ = 20,000 in the case study of Section 4).

Table 1. Example of the output T

[[	0.5	0.	0.]
[	0.19031606	1.83699228	0.]
[	0.18321005	11.25706728	0.]
[	0.66442132	17.61532334	2.]
[	0.48479738	25.45993783	1.]
[	0.13876928	28.11662473	1.]]

3 Algorithm

Our proposal has been implemented using Python[1]. A short application programming interface is detailed thereafter.

3.1 TreeBuild

This main function is used to build branching trees following the model defined in Section 2.1.2. Its halt condition is the targeted number of copies. Its prototype meets the following canvas:

$$(S, T, T_{obs}) = TreeBuild(X, \mu, \beta, p, L, n, genes, density),$$

where n is the desired number of copies, while X_0, μ, β, p, and L are the model parameters as defined in Section 2.1.2. Moreover $genes$ is the positions of each gene in the observed chromosome, and $density$ is the value defined in Part 2.1.2. Concerning the outputs, S is a $n \times 1$ vector representing states of deterioration while T_{obs} is the propagation time. Finally T is a $n \times 3$ matrix containing, for each copy: its position, its birth date, and the row of its mother, like in Table 1.

In this example, the mother of the copy located in 0.1832 is the root. The mother of the copy located in 0.6644 is the copy located in 0.1832. Other details regarding this main function are provided thereafter. The mean jump J is not a direct output of TreeBuild, but it can be easily computed with T. In this example, $J = \frac{|0.19-0.5|+|0.18-0.5|+|0.66-0.18|+|0.48-0.19|+|0.14-0.19|}{5}$.

3.1.1 Multiple clocks management

The working principle of TreeBuild can be summarized as follows: it determines the next event (deterioration or duplication) and executes it until the stopping criterion is satisfied. To determine the next event means to know its nature (deterioration or duplication), its time, and in which of the available copies it happens. Let j be the number of available copies at time t_1. The easiest way to determine the next event is to simulate $2 \times j$ exponential laws, one for each possible deterioration or duplication. The minimum of these $2 \times j$ simulations can thus provide the time, the nature, and the copy related to the next event.

Actually, TreeBuild does not really simulate $2 \times j$ exponential laws, as two properties of this law have been used to shorten computations. Indeed, $\forall (p_1, ..., p_{2j}) \in \mathbb{R}^{2j}, \forall (Y_1, ..., Y_{2j}) \sim (\mathcal{E}(p_1), ..., \mathcal{E}(p_{2j}))$, we have:

1. $min\,(Y_1, ..., Y_{2j}) \sim \mathcal{E}\left(\sum_{i=1}^{2j} p_i\right)$,
2. $\forall i \in 1...2j, P\,(Y_i = min\,(Y_1, ..., Y_{2j})) = \frac{p_i}{\sum_{k=1}^{2j} p_k}$.

Hence, due to the first property, the time of the next event can be simulated by a single exponential law. The second property, for its part, allows us to determine the nature and the copy affected by the next event using a single uniform law.

[1] Available at **https://github.com/SergeMOULIN/retrotransposons-spread**

3.1.2 Stopping criterion

As stated before, the stopping criterion of TreeBuild is related to n (the number of copies of the observed chromosome). But when a chromosome is observed, there is no way to detect that a new duplication has just occurred. Thus, the program cannot stop exactly at the birth of the n^{th} copy. Actually, TreeBuild must run until the T_{n+1} iteration (the birth date of the $n + 1^{th}$ copy), and then the propagation time T_{obs} can be determined by:

$T_{obs} = \frac{T_n + T_{n+1}}{2}$

Furthermore, each value taken by S and T between T_n and T_{n+1} is kept in memory. Thereby, the values of T and S returned by TreeBuild are values of T and S at time T_{obs}.

3.1.3 The management of copy locations

Copy positions in the chromosome are in the interval [0,1]. The distance traveled by a TE before insertion is assumed to follow an exponential law, but this latter can send the new copy outside the interval [0,1]. The solution chosen in this case is to launch again the computation of the new copy position.

In addition, the copy position is also relaunched with a probability "density" if its position falls into a host gene, as explained in Part 2.1.2. In other words:

$$\text{while } (X_{child} \notin [0,1] \text{ or } (X_{child} \in \textit{gene and } U1 < \text{density})) : \\ X_{child} = X_{mother} + U2 \times Y$$

where $U \sim U\{-1, 1\}$, $U2 \sim U\{-1, 1\}$, and $Y \sim \mathcal{E}(\frac{1}{L})$.

3.1.4 Critical situations

When TreeBuild is launched for each point of the grid of parameters, some critical situations can happen, which may induce a significant slowdown of the program. In particular, when μ is small and β is large, the probability for an event to be a duplication rather than a deterioration becomes very low. Thus, TreeBuild executes an inordinate number of deteriorations before reaching the desired number of copies. To solve this issue, we have decided that when the similarity to the root becomes lower than 0.03, then this copy cannot be degraded anymore.

3.2 Estimation method

In the available package, the estimation of the branching model parameters is realized by the *Optim* function. Its prototype is as follows:

$$(Best, Score) = Optim(Grid, Case, n1, N1, N2, genes).$$

Here, $Grid$ is a 5×4 matrix of settings defined exactly as in Section 2. $Case$, for its part, is a $2 \times n$ matrix containing locations and state of deterioration for each copy of the observed chromosome. $N1$ and $N2$ are settings defined in Sections 2.2.1 and 2.2.3, while $n1$ indicates how the grid is shrunk at each step after obtaining the best point (*cf.* the following section). Finally, $genes$ are the positions of each gene in the observed chromosome. The output $Best$ is the parameter set ($X_0, \mu, \beta, p, L, MJ$, T_{obs}) returned by the *Optim* function, while $Score$ is the sum of $N2$ differences between simulations and the observed chromosome (this $Score$ is useful if we relaunch Optim several times).

3.2.1 Interval reduction

As explained in Sections 2.2.1 and 2.2.2, the estimation method works with a grid where each point represents a parameter set. When the best point of the grid is found, a new grid is constructed around this point. Note that the new grid is not necessarily included in the previous one, in order to provide a larger degree of freedom of the parameters (in particular, when the latter are close to zero). For instance, in the case of parameter

L, the minimum of the new interval is $min\left(\frac{L_{min}}{2}, L_{best} - \frac{L_{delta}}{2\times n1}\right)$, where L_{min} and L_{max} are the minimum and maximum of the previous interval, $L_{delta} = L_{max} - L_{min}$, L_{best} is the L coordinate of the best parameter set, and $n1$ is the reduction parameter selected by the user. Thus, the minimum value of the test interval is divided by two at each time the best point of the grid is close enough to zero. The maximum value of the new interval is simply $L_{best} - \frac{L_{delta}}{2\times n1}$. These formulas, written for L, are also valid for β, μ, and p.

3.2.2 Location in the chromosome

Unlike the other parameters for which we scan a continuous interval, in the case of X_0, we only consider the positions of the TE copies on the observed chromosome. Thus we scan the interval of integers $1, .., n$. However, we process in the same way as with the other parameters (excepted that we use rounded values), and we never get out of the original interval $1, .., n$ in this case.

For instance, if we choose to test this interval of integers in four points (as we do in our case studies): at the first step, X_0 is tested in the first, $round(1 + \frac{n-1}{3})$-th, $round(1 + \frac{2\times(n-1)}{3})$-th, and n-th position of TE copies in the observed chromosome. In the next step, the new test interval thus becomes $max(1, round(X_{best} - \frac{X_{delta}}{2\times n1})...min(n, round(X_{best} - \frac{X_{delta}}{2\times n1}))$ where X_{delta} is $n - 1$ here.

3.3 Module and package dependencies

The Hungarian method has been applied using the "munkres" module, implemented in 2008 by Brian M. Clapper (Brian M. Clapper, munkres 1.0.7 for Python, https://pypi.python.org/pypi/munkres/).

4 Results and Discussion

4.1 The data

This proposal has been applied to the spread of the LTR retrotransposons ROO, DM412, and GYPSY on the euchromatin part of the chromosome 3L of the *Drosophila melanogaster* genome. This sequence corresponds to the left arm of the chromosome 3, which is the largest autosomal chromosome of *D. melanogaster*.

This is also the most prolific chromosome for each of the LTR retrotransposons we considered, this is why it has been chosen for this case study. ROO corresponds to the LTR retrotransposon in *Drosophila melanogaster* with the largest number of copies (Kaminker *et al.*, 2002; Lerat *et al.*, 2003; De la Chaux and Wagner, 2009). DM412 is supposed to have been recently acquired by *D. melanogaster* through horizontal transfer from a close relative species (Bartolomé *et al.*, 2009; Lerat *et al.*, 2011; Modolo *et al.*, 2014). Finally, GYPSY is an older and likely well regulated LTR retrotransposon (Lerat *et al.*, 2011).

Chromosome 3L contains 32 copies of ROO (with a mean nucleotide identity of 68.82%), 16 copies of DM412 (mean nucleotide identity of 60.24%), and six copies of GYPSY (mean nucleotide identity of 13.6%).

Three databases have been used during the experiments. The first one contains positions and nucleotide sequences for each TE copy annotated in *D. melanogaster* (flybase website[2] version number 5.51 of the *D. melanogaster* genome (Adams *et al.*, 2000; Smith *et al.*, 2007)). The second database has been downloaded from the RepBase website[3] and contains the consensus sequences for each TE corresponding to reference elements. The Needleman-Wunsch distance between each TE copy (from the first database) and its reference (from the second database) has been calculated, in order to obtain the deterioration states.

Finally, the third database comes from flybase too. This is the position of all the annotated genes in the euchromatin part of chromosome 3L, in version number 5.51 of the *D. melanogaster* genome.

In this case study, the estimation method described in Section 2.2 has actually been applied not only once but 40 times in each situation, in order to check the consistency of the obtained parameter sets. The best parameter set of each case study, considering the output "score" (*cf.* Part 3.2), is presented in Section 4.3. The whole obtained parameter sets are presented in supplementary data with their descriptive statistics. Some indications about consistency of these results are provided in Section 4.4.

Table 2. Setting table

parameter	starting point	end point	interval division	desired accuracy
X_0	1	n	3	10^{-3}
μ	0.1	10	3	10^{-2}
β	0.01	1	3	10^{-3}
p	0.1	100	3	10^{-1}
L	0.01	1	3	10^{-3}

4.2 Settings

Let us first recall that X_0, which represents the root position in the chromosome, is inside the interval [0 , 1]. In other words, copy positions have been divided by the chromosome size. For the euchromatic part of chromosome 3L, this size has been set at 24,543,557 base pairs (bp) in the version 5.51. (Smith *et al.*, 2007).

In Table 2, each row represents the beginning and the end of the test interval, the number by which the test interval has been divided, and the final desired accuracy regarding the parameter. In particular, in the third column, the value associated to each is 3. This latter means that these parameters have been tested at the beginning, in the first third, in the second one, and at the end of the test interval.

Finally, at each iteration, the grid used for μ, β, and p estimations contains $4^3 = 64$ points while, at each iteration, the grid used to estimate X_0 and L contains $4^2 = 16$ points.

The other parameters are:

- $n1$ = 1.5: at each step, after the best point has been found, the grid's dimensions have been divided by 1.5.
- $N1$ = 10,000: each point has been tested 10,000 times during estimation of X_0, μ, β, p, and L.
- $N2$ = 20,000: J and T_{obs} estimations are the average values of 20,000 simulations. The output "score" is computed based of these simulations.

Furthermore, we can notice that in the case of ROO, 18 of the 32 copies are located inside genes while genes hold 72.7% of the studied part (euchromatin). Thus, density $= \frac{\frac{18}{72,7}}{\frac{14}{27,2}} = 0.482$. In the cases of DM412 and GYPSY, the densities are respectively equal to 0.625 ($\frac{10}{16}$) and 0.187 ($\frac{2}{6}$).

4.3 Results

The obtained parameters are summarized in Table 3.

If we consider for instance the spread of ROO, the obtained parameters can be interpreted as follows:

[2] **http://flybase.org/**

[3] **http://www.girinst.org/repbase/**

Table 3. Results and consistency

Best parameter sets			
parameter	**ROO**	**DM412**	**GYPSY**
X_0	29	14	5
μ	2.396	1.135	0.050
β	0.351	0.235	0.093
p	0.051	0.001	0.007
L	1.331	0.336	0.016
J	0.300	0.216	0.013
T_{obs}	4.070	3.353	2.379
Consistency indicators			
parameter	**ROO**	**DM412**	**GYPSY**
X_0	0.104	0.096	0.392
μ	0.020	0.017	0.002
β	0.040	0.064	0.044
p	0.001	0.000	0.000
L	0.262	0.036	0.033
J	0.010	0.013	0.024
T_{obs}	0.042	0.014	0.031

- $X_0 = 29$. The root is on the 29^{th} position in chromosome 3L. This is the copy FBti0059644, which is located between the $21,954,331^{th}$ and the $21,954,698^{th}$ nucleotide.
- $\mu = 2.396$. The average time between two degradations is 2.396, where 1 is the average time before duplication of the root. Degradations are thus less frequent than duplications. Please note that this estimation of μ is without time unit: it is related to the duplication speed of the root. It allows us to estimate duplication speed when the deterioration speed is known, and *vice versa*.
- $\beta = 0.351$. Each degradation causes a division by $1 + E(\frac{1}{0.351})$ of the similarity. Thus the similarity is divided by 1.351 on average at each degradation.
- $p = 0.051$. p allows us to determine how many degradations led to a decrease in the duplication speed. For example, in this case, if the identity between a copy and the reference is 0.75 (*i.e.*, state of deterioration = 0.25), then the duplication speed of this copy is reduced by 1.275% (indeed $0.25 \times 0.051 = 0.01275$). This looks like a low effect.
- L = 1.331. The distance traveled by the TE before insertion follows a distribution $E(\frac{1}{1.331})$, with the relaunching process: (1) when the position of the child is out of the chromosome (*cf.* Section 3.1.3) and (2) possibly when the position of the child is inside a gene. As explained in Part 2.1.2, the parameter J is better to represent the average distance before insertion. However, we can notice that L = 1.331 is a pretty large number (larger than 1). This value tends to suggest that the child inserts itself more or less anywhere in the case of ROO (compared to DM412 or GYPSY).
- J = 0.300. The mean distance traveled by the TE before insertion is 0.300.
- T_{obs} = 4.070. The time of ROO propagation is 4.070 larger than the non-deteriorated root. In addition, 4.070/2.396 = 1.699, thus the root has faced 1.699 degradations on average. From a global perspective, we can note that in each case, the root corresponds to the border elements. They are in position 29^{th} over 32, 14^{th} over 16, and 5^{th} position over 6 respectively for ROO, DM412, and GYPSY. The latter correspond to positions 0.894, 0.965, and 0.969 when the chromosome is considered as a [0,1] interval. This can be due to a larger density of TEs in this area. In addition, we can notice that p is really close to 0 in each case. This should imply no (linear) effect of the degradation on the duplication rate. This is an unexpected result. Indeed, the fact that the degradation undergone by the copies reduces their ability to duplicate sounds natural. This is why the parameter p was added in the model. Finally, we can notice that in the case of ROO, the results suggest a few big degradations, while in the case of GYPSY, they suggest a lot of little ones.

The fact that some of the obtained parameters are outside the test interval chosen at the beginning of the program (for instance, L = 1.331 in the case of ROO or μ = 0.050 in the case of GYSPY) is a desired effect, to let a larger freedom to the parameters. In particular, the aim was to let parameters to be as close as possible to zero if required (*cf.* Section 3).

In each of these three cases, one billion trees have been simulated with the obtained parameter set. The best of these trees is shown in Figure 1 for ROO and in supplementary data for DM412 and GYPSY.

4.3.1 Focusing on the roots

According to these models and methods, the root of ROO could correspond to the FBti0059644 copy. This is an incomplete copy (368 bp, compared to the reference which has 9,112 bp) that corresponds to a solo-LTR, a remnant from a LTR-LTR recombination. This copy is thus no longer active, but it is quite recent since its divergence to the reference is rather low (95.71% of identity on the aligned part of the sequences).

The root of DM412 could correspond to the FBti0061034 copy. This is a very degraded copy that has 88 bp in length, corresponding to an internal portion of the reference element, whose length is 7440 bp. The copy is old since it is very divergent compared to the reference (80.90% of identity).

Finally, the root of GYPSY could correspond to the FBti0062705 copy. This copy is an incomplete and very degraded element of 1,282 bp length (7,471 bp for the reference) with a very high divergence to the reference (70.59% identity). This copy corresponds to a piece of the inner part of the gypsy element, and this is a very old and degraded copy that is not currently active.

4.4 Consistency of results

As explained previously, the optimization method has been actually applied 40 times for each TE. The descriptive statistics for these three cases are summarized in the supplementary data. In addition, for each parameter that has been estimated by scanning an interval (*i.e.*, X_0, μ, β, p, and L), quotients $\frac{\text{Standard deviation of the results}}{\text{Test interval}}$ have been computed in each case, in order to assess the consistency of the results. These quotients are reproduced in Table 3. Standard deviations of the output J and T_{obs} are also set in the same table.

Several versions of the code have been implemented in this project in order to increase the consistency of these results. In a previous version, X_0, μ, β, p, and L were estimated all together (*i.e.*, steps 2.2.1 and 2.2.2 were merged). This approach implied to work in a $4^5 = 1024$ points grid, thus, it provided a lower number of trials by point, which reduced consistency. In our very first version, the duplication rate was a parameter to be estimated while $T_{obs} = 1$ was the stopping criterion of TreeBuild. The difference between n and the number of copies at the end of the simulation (which was not necessarily n in this setup) was penalized. However, in this setup, the duplication rate was the only parameter to be consistent.

Finally, in the setup presented here, the consistency looks acceptable in most of the cases (*i.e.*, excepted for L in the case of ROO or for μ in the case of GYPSY). Nevertheless, this requires a large number of simulations implying that the program runs several days.

4.5 Conclusion and future perspectives

In this article, a model has been proposed for the propagation of LTR retrotransposons in a genome. Various functions have been implemented to simulate this spread as well as graphic representations. Finally, a first method for estimating the parameters of this propagation model has been proposed and applied to the spread of TEs corresponding to the ROO, GYPSY, and DM412 elements in a chromosome of *Drosophila melanogaster*. However, this work can be improved in various directions, some of them being listed below.

The first point is that the model of propagation should be applied to the full genome instead of a single chromosome of *Drosophila melanogaster*. Indeed, a copy inserted in a given chromosome can produce a child that will not necessarily inserts itself in the same chromosome. An idea to extend the model to the full genome could be to let the position of the child copy following a $\delta(a)\mathcal{U} + \delta(1-a)\mathcal{E}(\frac{1}{L})$ law. In other words, the child copy inserts itself anywhere in the genome (uniform distribution), with a probability a, or it inserts itself on the same chromosome than the mother copy (with the distance to the mother following an exponential law) with probability $1-a$, where a is a new parameter to determine. Nevertheless, this approach raises various questions that must be answered in a further study. Firstly, should $\mathcal{E}(\frac{1}{L})$ represent a number of nucleotides or a proportion of the considered chromosome? How to redefine the distance built in Part 2.2.2? etc.

Secondly, as explained in Part 4.4, the estimation method needs a large compilation time to give acceptably consistent results. This compilation time increases quickly when the sample size (number of TEs) increases or if we want to estimate more parameters. Therefore, we will test other ways (likelihood maximization, neighbor joining, or Bayesian estimation) to improve our proposal. These types of methods would also allow us to produce some confidence intervals for the estimated parameters. Our long-term objective is to create a useful tool for estimating consistently both parameters and the branching process itself. In other word, our goal is to produce a tool close to phylogenetic tree estimation but adapted to TE constraints. Another possibility of improvement could be to consider the possibility of several roots. For instance, a method of unsupervised classification like Gaussian Mixture model could be applied in order to detect the number of clusters.

In this project, we used RepBase consensus sequences, based on a lot of TE sequences as root sequence. Another possibility could also be to produce an ancestral reconstruction based only on the sequences of the case study. In this way, it would not be necessary to search the data in two different databases. We can also note that in this work, we consider all modifications (*i.e.*, mutations, LTR recombination, and so on) as one single and global deterioration effect. It could be interesting to try to distinguish each effect. Finally the effect of an epigenetic regulation that can affect TE behaviour even if they do not face sequence degradation could be taken into account.

Funding

Computations have been performed on the supercomputer facilities of the Mésocentre de calcul de Franche-Comté. This work has been supported by the Transposable Elements project of the Franche-Comté region. The work of S.C. was also sponsored by NMS/IRD project 117480 at NPL.

References

Adams, M. D., Celniker, S. E., Holt, R. A., Evans, C. A., Gocayne, J. D., Amanatides, P. G., Scherer, S. E., Li, P. W., Hoskins, R. A., Galle, R. F., *et al.* (2000). The genome sequence of drosophila melanogaster. *Science*, **287**(5461), 2185–2195.

Bartolomé, C., Bello, X., and Maside, X. (2009). Widespread evidence for horizontal transfer of transposable elements across drosophila genomes. *Genome Biol*, **10**(2), R22.

Belancio, V. P., Hedges, D. J., and Deininger, P. (2008). Mammalian non-ltr retrotransposons: for better or worse, in sickness and in health. *Genome research*, **18**(3), 343–358.

Clapper, B. (2008). munkres 1.0.7 for python. https://pypi.python.org/pypi/munkres/.

Cleveland, W. S. and Devlin, S. J. (1988). Locally weighted regression: an approach to regression analysis by local fitting. *Journal of the American Statistical Association*, **83**(403), 596–610.

De la Chaux, N. and Wagner, A. (2009). Evolutionary dynamics of the ltr retrotransposons roo and rooa inferred from twelve complete drosophila genomes. *BMC evolutionary biology*, **9**(1), 205.

Drakos, N. E. and Wahl, L. M. (2015). Extinction probabilities and stationary distributions of mobile genetic elements in prokaryotes: The birth–death-diversification model. *Theoretical population biology*, **106**, 22–31.

Egner, C. and Berg, D. E. (1981). Excision of transposon tn5 is dependent on the inverted repeats but not on the transposase function of tn5. *Proceedings of the National Academy of Sciences*, **78**(1), 459–463.

Epanechnikov, V. A. (1969). Non-parametric estimation of a multivariate probability density. *Theory of Probability & Its Applications*, **14**(1), 153–158.

Foster, T. J., Lundblad, V., Hanley-Way, S., Halling, S. M., and Kleckner, N. (1981). Three tn10-associated excision events: relationship to transposition and role of direct and inverted repeats. *Cell*, **23**(1), 215–227.

Green, M. M. (1988). Mobile dna elements and spontaneous gene mutation. *Banbury Rep*, **30**, 41–50.

Kaminker, J. S., Bergman, C. M., Kronmiller, B., Carlson, J., Svirskas, R., Patel, S., Frise, E., Wheeler, D. A., Lewis, S. E., Rubin, G. M., *et al.* (2002). The transposable elements of the drosophila melanogaster euchromatin: a genomics perspective. *Genome Biol*, **3**(12), RESEARCH0084.

Kaplan, N., Darden, T., and Langley, C. H. (1985). Evolution and extinction of transposable elements in mendelian populations. *Genetics*, **109**(2), 459–480.

Kuhn, H. W. (1955). The hungarian method for the assignment problem. *Naval research logistics quarterly*, **2**(1-2), 83–97.

Lander, E. S., Linton, L. M., Birren, B., Nusbaum, C., Zody, M. C., Baldwin, J., Devon, K., Dewar, K., Doyle, M., FitzHugh, W., *et al.* (2001). Initial sequencing and analysis of the human genome. *Nature*, **409**(6822), 860–921.

Lerat, E., Rizzon, C., and Biémont, C. (2003). Sequence divergence within transposable element families in the drosophila melanogaster genome. *Genome research*, **13**(8), 1889–1896.

Lerat, E., Burlet, N., Biémont, C., and Vieira, C. (2011). Comparative analysis of transposable elements in the melanogaster subgroup sequenced genomes. *Gene*, **473**(2), 100–109.

McClintock, B. (1987). *The discovery of characterization of transposable elements: the collected papers of Barbara McClintock.*

Modolo, L., Picard, F., and Lerat, E. (2014). A new genome-wide method to track horizontally transferred sequences: application to drosophila. *Genome biology and evolution*, **6**(2), 416–432.

Moody, M. E. (1988). A branching process model for the evolution of transposable elements. *Journal of mathematical biology*, **26**(3), 347–357.

Munkres, J. (1957). Algorithms for the assignment and transportation problems. *Journal of the Society for Industrial and Applied Mathematics*, **5**(1), 32–38.

Needleman, S. B. and Wunsch, C. D. (1970). A general method applicable to the search for similarities in the amino acid sequence of two proteins. *Journal of molecular biology*, **48**(3), 443–453.

Sanmiguel, P. and Bennetzen, J. L. (1998). Evidence that a recent increase in maize genome size was caused by the massive amplification of intergene retrotransposons. *Annals of Botany*, **82**(suppl 1), 37–44.

Sawyer, S. A., Dykhuizen, D. E., DuBose, R. F., Green, L., Mutangadura-Mhlanga, T., Wolczyk, D. F., and Hartl, D. L. (1987). Distribution and abundance of insertion sequences among natural isolates of escherichia coli. *Genetics*, **115**(1), 51–63.

Smith, C. D., Shu, S., Mungall, C. J., and Karpen, G. H. (2007). The release 5.1 annotation of drosophila melanogaster heterochromatin. *Science*, **316**(5831), 1586–1591.

Wicker, T., Sabot, F., Hua-Van, A., Bennetzen, J. L., Capy, P., Chalhoub, B., Flavell, A., Leroy, P., Morgante, M., Panaud, O., *et al.* (2007). A unified classification system for eukaryotic transposable elements. *Nature Reviews Genetics*, **8**(12), 973–982.