

This manual covers details about the features and function of Inpactor Version 1.0 final.

## **Table of contents:**

<b>Inpactor:</b>	<b>1</b>
<b>General description:</b>	<b>2</b>
<b>Prerequisites:</b>	<b>3</b>
<b>Installation:</b>	<b>4</b>
<b>Setting up the process of analysis:</b>	<b>4</b>
<b>Executing analysis with Inpactor:</b>	<b>7</b>
<b>License:</b>	<b>7</b>
<b>Authors:</b>	<b>7</b>

## **Inpactor:**

Integrated and Parallel Analyzer and Classifier of LTR Retransposons for Large Genomes.

Date: 15 May 2018.

Document: User Manual.

---

## General description:

Inpactor was developed using MPI standard, in C language. It is composed by 4 modules.

As previous requirement Inpactor needs the output of LTR\_STRUC or repet (TEdenovo package)(view software requirement section). Also Inpactor can process contigs or scaffolds in a fasta file.

In a configuration file is possible to define general information such as input information (folder with LTR\_STRUC output, repet output file or fasta file), result directory, verbose mode and clean mode at the end of the execution. In addition each module requires that different parameters must be indicated in the configuration file. Preprocessing, Classification and domain extraction modules can run independently, in contrast Tree creation and insertion time Module needs to be executed with Domain extraction module.

The first module executed in the process of analysis is the preprocessing. The objective here is to group together all information from the input information into one tabular text file, to organize the information, that can be:

- 1) LTR\_STRUC output: Inpactor uses two LTR\_STRUC files: (i) in report file we got features such as LTR Identity, primer binding site (PBS), PolyPurine Tract (PPT), length, Active size, Longest Open Reading Frame (ORF), Target Site Duplication (TSD), Long Terminal Repeat (LTR) A length, LTR B length, and strand; (ii) in another file, we used Fasta file to extract important sequences like LTR A and B using Seqret and Extractseq tools from Emboss and the sequence of the full element.
- 2) repet output (TEdenovo Package) in fasta format: Inpactor extracts information such as element length, LTR Identity, Long Terminal Repeat (LTR) A length, LTR A sequence, LTR B length, LTR B sequence and information about domains found.
- 3) fasta file: Inpactor can analyze fasta files with contigs and also with whole genome. In this case Inpactor executes LTR-FINDER to find completed elements inside the input file,

looking for features such as element length, LTR Identity, PPT, Longest ORF, Long Terminal Repeat (LTR) A length, LTR A sequence, LTR B length, LTR B sequence, Strand and information about domains found.

The second step is call the classification module. Using the result file from previous module, a classification was performed as follow: (i) if the element carried at least one principal domain (RT, INT, and RNaseH) with keywords RLC or RLG, the LTR-RT was classified as complete-family element (Copia or Gypsy); (ii) if the element didn't carry any domain, it was classified as non-autonomous element; (iii) if the element had only a GAG domain or GAG and AP domains, the element was classified as TR-GAG elements.

The third module of Inpactor is the Domain extraction module, in this part of the process, we were interested into the extraction of RT domain sequences from each complete-family element, because this domain is the most conserved and appropriated for phylogenetic analysis. Other domains from the LTR-RT polyprotein might be used alternatively.

Lastly, Inpactor execute the fourth module, composed by two steps:

Analyzing LTR retrotransposon insertion times.

The insertion times of full-length copies, as defined by a minimum of 80% of nucleotide identity over 100% of the reference element length, were dated using a substitution rate that is provided by the user (we used 0.000000013).

Phylogenetic tree creation.

Using the protein Fasta file from RT domain extraction module, a multiple alignment was performed using Mafft with -thread option to indicate the number of cores.

## **Prerequisites:**

Inpactor run over linux environments, the software was tested in

Centos 6,7. then we show you a list of the prerequisites previous to Inpactor installation:

- NCBI-Blast version 2.5.0  
(<ftp://ftp.ncbi.nlm.nih.gov/blast/executables/blast+/2.5.0/>)
- Emboss version 6.6.0  
(<ftp://emboss.open-bio.org/pub/EMBOSS/>)
- Wise2 version 2.4.0  
(<http://www.ebi.ac.uk/~birney/wise2/>)
- OpenMPI version 1.8.8  
(<https://www.open-mpi.org/software/ompi/v1.8/>)
- Censor version 4.2.29  
(<http://www.girinst.org/downloads/software/censor/>)
- Mafft version 7.305  
(<http://mafft.cbrc.jp/alignment/software/>)
- LTR-FINDER version 1.0.5  
(<http://mafft.cbrc.jp/alignment/software/>)

### **Installation:**

After install all prerequisites, you must clone the repository of the current version of Inpactor using:

```
# git clone https://github.com/simonorozcoarias/Inpactor.git,  
then you might run as following:
```

```
#cd Inpactor
```

```
# mpicc Inpactor.c -o Inpactor
```

This step produces an executable, which will be used in next

sections.

### Setting up the process of analysis:

Inpactor need a configuration file for define the parameters which will be use in the execution of the analysis. Below is explained the sections of this file:

```
#The first section of this file specify mainly the directories  
#were Inpactor can find the data entry and data output.
```

```
#####Configuration file#####
```

```
#directory indicates the path of the output folder of LTR_STRUC  
or #repet output which was executed previously  
input=/home/user/LTR_STRUC_output_folder_or_repet_output_file
```

```
#result_directory is the path of the output of Inpactor.  
result_directory=/home/user/Inpactor_output
```

```
#indicates if Parralan write all the actions during the analysis  
#process through standard output  
verbose=true
```

```
#with this option Inpactor will erase all the temporally files  
#used in the analysis process, some of these files can be  
#relevant for the user to self determination.  
clean=true  
#Input type can be LTR_STRUC, repet or fasta  
input_type=LTR_STRUC
```

```
##### Preprocessing #####
```

```
#This line in the configuration file confirms the execution of  
#this step. In specific cases is possible that previously you  
#have run this line and do not need run it again.  
preprocessing=true
```

```
#this path indicates the location of databases that contains the  
#domains (genes)  
database=/home/user/cores-database-wickercode.Lineage_Bianca.fa
```

##### Classification #####

#This line in the configuration file confirms the execution of  
#this step. In specific cases is possible that previously you  
#have run this line and do not need run it again.  
classification=true

#  
#this tabfile is not necessary if preprocessing is true  
tabfile=/home/user/Inpactor\_output/step1/all\_tabfiles.tab

# This line allow reclassify unclassified LTR-RT previously.  
80-80-80-rule=true

##### Domain Extraction #####

#This line in the configuration file confirms the execution of  
#this step. In specific cases is possible that previously you  
#have run this line and do not need run it again.  
extraction=true

#This line specify the database of interest for the domain.  
RTdatabase=/home/user/RTcores-database-wickercode.Lineage\_Bianca.  
fa

#This line specify the list of already classified LTR-RTs, if you  
have, for the #classification like references.  
references=/home/user/references.fasta

#this fasta file is not necessary if classification is true  
fastafile=/home/user/Inpactor\_output/step2/all\_tabfiles.tab\_ALL.R  
LC\_RLG.FA

#filter RT size. For a lot of RT, is better RTlength; 200,  
#for less RTlength; 180 or 150  
RTlength=200

#Blast\_evalue is the evalue to be used for de analysis  
Blast\_evalue=1e-4

#### Insertion Time analysis and Phylogenetic tree creation ####

#This line in the configuration file confirms the execution of  
#this step. In specific cases is possible that previously you

#have run this line and do not need run it again. Is important  
#consider that this step require compulsory the previous step.

```
insertion=true
substitution_rate=0.000000013
#this tabfile is not necessary if classification is true
tabfileS4=/home/user/Inpactor_output/step2/all_tabfiles.tab_ALL.R
LC_RLG.TAB
```

##### End of configuration file #####

### Executing analysis with Inpactor:

This command execute the process of analysis. Is very important to consider that all the software listed in the prerequisites section must be load in the path of the system. This step uses Inpactor's executable file generated in installation section.

```
mpirun -np "number of process (depend of the number of cores
available in your system" Inpactor "configuration file"
```

### Output files of Inpactor:

Inpactor creates some files in each module:

Module	File	Description
Preprocessing	all_tabfiles.tab	This file contains all information from LTR_STRUC output files, repet output file in fasta format or fasta file with contigs
Classification	all_tabfiles.tab_ALL.RLC_RLG.FA	Fasta file with sequences of LTR-RT elements classified as autonomous (Copia and Gypsy)

	all_tabfiles.tab_ALL .RLC_RLG.TAB	tabular file with relevant information of LTR-RT elements classified as autonomous (Copia and Gypsy)
	all_tabfiles.tab_ALL .RLC_RLG.TAB.familie s.FA	Fasta file with sequences of LTR-RT elements classified as autonomous, indicating the subfamily (Copia and Gypsy)
	all_tabfiles.tab_ALL .RLC_RLG.TAB.familie s	tabular file with relevant information of LTR-RT elements classified as autonomous, indicating the subfamily (Copia and Gypsy)
	all_tabfiles.tab_ALL .RXX.FA	Fasta file with sequences of LTR-RT elements classified as non-autonomous (TRIM and LARD)
	all_tabfiles.tab_ALL .RXX.TAB	Tabular file with relevant information of LTR-RT elements classified as non-autonomous (TRIM and LARD)
	all_tabfiles.tab_ALL .TR_GAG.FA	Fasta file with sequences of LTR-RT elements classified as TR-GAG
	all_tabfiles.tab_ALL .TR_GAG.TAB	Tabular file with relevant information



		of LTR-RT elements classified as TR-GAG
	all_tabfiles.tab.NOC .FA	Fasta file with sequences of unclassified LTR-RT elements
	all_tabfiles.tab.NOC .TAB	Tabular file with relevant information of unclassified LTR-RT elements
	classifiedTE.fa	Final fasta file with sequences of all classified LTR-RT elements
Domain Extraction	Genome.all_ALL.RLC_R LG.fa.rt	Fasta file with sequences of domains found in autonomous LTR-RT elements
Insertion Time analysis and Phylogenetic tree creation	Distribution-in-age. tab	This file contains the quantity of autonomous LTR-RT elements divided by periods of insertion time.
	Final.insertion-time .tab	This file contains all autonomous LTR-RT, divergence per site and estimation of insertion time
	multiple_align	Multiple alignment done with all domains found in previous step.
	multiple_align.tree	Distance tree created with

		previous multiple alignment
--	--	-----------------------------

### License:

Inpactor is licensed under GNU GLP v3

(<https://www.gnu.org/licenses/gpl-3.0.en.html>)

### Authors:

IRD France: (<http://www.ird.fr/>)

- Romain Guyot

Autonomous University of Manizales:  
(<https://www.autonoma.edu.co/>)

- Simón Orozco Arias
- Reinel Tabares Soto

University of Caldas: (<http://www.ucaldas.edu.co/>)

- Diego Hernando Ceballos López
- Andrea Garavito

For more information please write to:  
simon.orozco.arias@gmail.com