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

Table of contents:

Parallan:

Parallel Analyzer and Classifier of LTR Retransposons for Large Genomes.

Date: May 31 2017.

Document: User Manual.

General description:

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

As previous requirement Parallan needs the output of LTR_STRUC or repet (TEdenovo package), but also Parallan can be used with a fasta file which contained contigs, or the whole genome (view software requirement section).

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: Parallan 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 otput (TEdenovo Package) in fasta format: Parallan 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: Parallan can analyze fasta files with contigs and also with whole genome. In this case Parallan 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 Parallan 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, Parallan 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.

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:

Parallan run over linux environments, the software was tested in Centos 6,7. then we show you a list of the prerequisites previous to Parallan 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 prerequistes, you must clone the repository of the current version of Parallan using:

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

#cd Parallan

mpicc parallan.c -o parallan

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

Setting up the process of analysis:

Parallan 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 Parallan can find the data entry and data output.

##########################Configuration file ####################

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

#result_directory is the path of the output of Parallan.
result_directory=/home/user/Parallan_output

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

```
#with this option Parallan 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/Parallan_output/step1/all_tabfiles.tab

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

####################### 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/Parallan_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

#this tabfile is not necessary if classification is true
tabfileS4=/home/user/Parallan_output/step2/all_tabfiles.tab_ALL.R
LC_RLG.TAB


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

Executing analysis with Parallan:

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 parallan's executable file generated in installation section. Is

highly recommended if you have less sequences or elements in your input than CPUs in your system, to use the same number of processes than sequences or elements in your input, but if you have more sequences or elements than CPUs, is better to use the same number of processes than CPUs in your system.

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

Output files of Parallan:

Parallan 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.families.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.families	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
	clasifiedTE.fa	Final fasta file with sequences of all classified LTR-RT elements
Domain Extraction	Genome.all_ALL.RLC_RLG.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:

Parallan 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

Bioinformatics and computational biology center of Colombia: (www.bios.co)

- Simón Orozco Arias
- Reinel Tabares Soto
- Diego Hernando Ceballos López

For more information please write to: tecnologia@bios.co - contacto@bios.co