clemgoub / **dnaPipeTE**

Branch: master ▾ dnaPipeTE / README.md Find file Copy path

clemgoub Update README.md d3ee573 on Dec 7, 2017

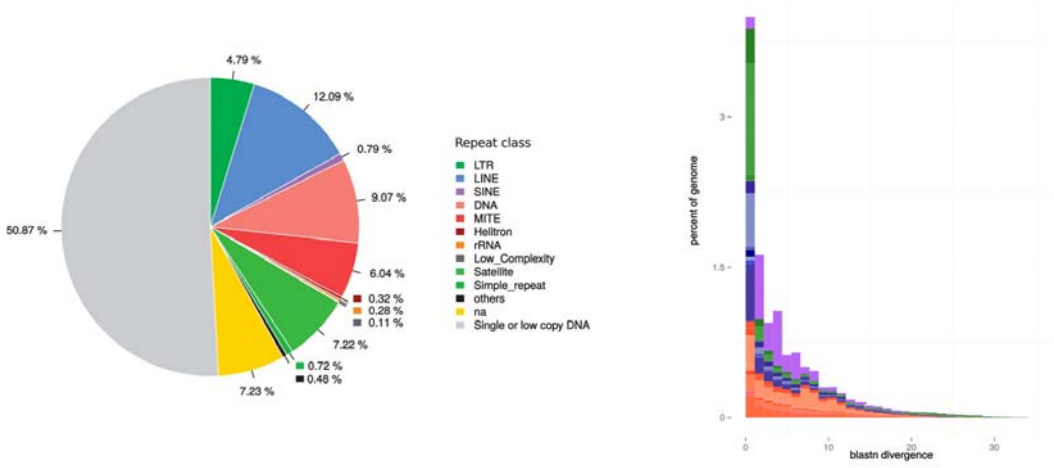
1 contributor

345 lines (244 sloc) 17.7 KB

dnaPipeTE 1.3 - README

dnaPipeTE (for **de**-**n**ovo **a**ssembly & annotation **Pipe**line for **T**ransposable **E**lements), is a pipeline designed to find, annotate and quantify Transposable Elements in small samples of NGS datasets. It is very useful to quantify the proportion of TEs in newly sequenced genomes since it **does not require genome assembly** and works on **small datasets (< 1X per run)**.

- dnaPipeTE is developped by Clément Goubert, Laurent Modolo and the TREEP team of the LBBE: http://lbbe.univ-lyon1.fr/-Equipe-Elements-transposables-.html?lang=en
- You can find the original publication in GBE here: https://academic.oup.com/gbe/article/7/4/1192/533768



output examples of quantification and TE landscape (relative age) produced by dnaPipeTE

Changelog v1.3.1** 07.Dec.2017

- Fixed missing class column for some Academ families causing errors with landscape graphs (thanks @rotifergirl for reporting!)

Changelog v1.3** 01.Dec.2017

- Updated Trinity with latest version (v2.5.1)
- Updated RepeatMasker with latest version (version Open 4.0.7)
- Compatible with latest Repbase (RepeatMasker compatible) libraries (20170127)
- fix bug for the blast sample wich turned out to be a reclycling of the sample 1 instead of a new independant sample. However, test showed this had no striking influence on the results (actual sampling variation between runs is more likely to create variation between outputs).

- remove most of the files from the `bin` folder and replace it with the `init.sh` script so that user can make their own installation.
- Landscape graph are now expressed relative to genome %
- Clean git repository of larges files

You can download previous version on the github repository, clicking on "branch" menu and selecting the desired version

Changelog v1.2**

- Estimation of repeat content is now performed on the ratio of aligned bases (bp) on repeat contig over the total number of base sampled, instead of the number of reads mapping / total of read sampled; this produces a better estimate of the repeat content and reduces potential overestimations. In addition, it allows more accurate estimates if the size of reads used as input is variable.
- If different part of one same read match different repeats contigs (e.g. in case adjacent TEs or TE in TE), all bases are retained instead only the one of the best hit.
- New graph "Bases per component" replaces "reads per component"; is very similar to reads per component graph but represent the total amount of bases aligned over the dnaPipeTE contigs.
- Bug fix: in last version, repbase library was not merged to annotated dnaPipeTE contigs for repeat estimates, now it is (as presented in the pipeline cartoon)
- New option: "-Trin_glue" to specify a minimum number of reads supporting the joining of kmer contigs during assembly (Chrysalis step in trinity)
- New option: "-contig_length" to set a minimum size (in bp) to report a contig (default is 200 bp)

1 - INSTALLATION

If you encounter some issues during installation, do not hesitate to ask for help !

System requirement

dnaPipeTE only runs on Linux x64 environments (tested on ubuntu 14.04 PC, Debian 3.2.57-3 x86_64 cluster and CentOS 7.3.1611).

However, Trinity (used for assembly) can use a lot of RAM ! Here are some examples of RAM usages :

- 100 000 reads ~10 Go RAM (two Trinity iterations)
- 3 000 000 reads ~40 Go RAM (two Trinity iterations)

Thus we recommend to use it on assembly-dedicated servers but it could work (if RAM is sufficient) on a PC.

Dependencies

We provide some of the dependencies dnaPipeTE needs to run, however the pipeline requiers the following programs to be already installed:

- **Python 3**, including *argparse*, *configparser*, *os*, *re*, *subprocess*, *time*, *sys*, *random*, *ntpath*
- **Perl 5**
- **R** version 3.0.2 or later (not tested below) including *ggplot2* package.

The following dependancies are provided in the package or will be automatically installed with dnaPipeTE (see INSTALLATION):

- **Java 1.8**
- **TRF** (Tandem Repeat Finder)
- **GNU Parallel** version 3.
- **Trinity** (RNAseq assembly) vers. 2.5.1
- **RepeatMasker**, including **RMblastn**
- **blastn** (from blast+ suite)

Installation

- Download and unzip the dnaPipeTE package in the folder of your choice
- or clone the github repository

```
git clone https://github.com/clemgoub/dnaPipeTE
```

It will create a new directory called dnaPipeTE with all the necessary files. Do not move or modify any of those file.

- Move to the dnaPipeTE directory and open the **init.sh** file.

```
cd dnaPipeTE
nano init.sh
```

- You need to provide your GIRI (Repbase) login and password at the top of the "init.sh" script located in the dnaPipeTE folder. If you don't have a (free) GIRI account, you can request one online at http://www.girinst.org/. Simply replace *username* and *password* with yours, then save and close the file. From nano: Ctrl-X, type "Y" and then Enter to save and exit.

```
#set your GIRI username and password
GIRINST_USERNAME='username'
GIRINST_PASSWORD='password'
```

- You can now execute the "init.sh" script which will download and install most of the dependencies:

```
./init.sh
```

- Now your need to execute the RepeatMasker ./configure script, that will build the libraries. Provide the following informations when asked:

```
cd bin/RepeatMasker
./configure


Enter path [ /usr/local/bin/perl ]:
```

Just press enter. The path should be found automatically.

```
Enter path [ /yourpath/dnaPipeTE/bin/RepeatMasker ]:
```

Same thing here, the path to RepeatMasker should be found automatically.

```
Enter path [ ]: </yourpath>/dnaPipeTE/bin/trf
```

Here, replace by the path to the dnaPipeTE folder. The default prompt should be empty, unless you already have trf in your path.

```
Add a Search Engine:
   1. CrossMatch: [ Un-configured ]
   2. RMBlast - NCBI Blast with RepeatMasker extensions: [ Un-configured ]
   3. WUBlast/ABBlast (required by DupMasker): [ Un-configured ]
   4. HMMER3.1 & DFAM: [ Un-configured ]

   5. Done


Enter Selection: 2
```

Choose 2 and press enter

```
**RMBlast (rmblastn) INSTALLATION PATH**

  This is the path to the location where
  the rmblastn and makeblastdb programs can be found.



Enter path [  ]: </yourpath>/dnaPipeTE/bin/ncbi-blast-2.2.28+/bin
```

Same thing here, replace with the path to your local installation of dnaPipeTE. Then press enter. Be carefull to not add a final / after the 'bin', for some reason it is very sensitive!

```
Do you want RMBlast to be your default
search engine for Repeatmasker? (Y/N)  [ Y ]:Y
```

Choose Y and press enter. Yes!!!

```
Add a Search Engine:
   1. CrossMatch: [ Un-configured ]
```

```
    2. RMBlast - NCBI Blast with RepeatMasker extensions: [ Configured, Default ]
    3. WUBlast/ABBlast (required by DupMasker): [ Un-configured ]
    4. HMMER3.1 & DFAM: [ Un-configured ]

    5. Done


Enter Selection: 5
```

Choose 5 here and press enter, we are done! Your are close!!!

Testing dnaPipeTE installation

Test scripts and java fix

- To test the installation, go back to the main dnaPipeTE folder and run the bash script ./test_config.sh

```
cd ../../
./test_config.sh
```

- This script will test if your version of Java is up to date (version 1.8) and if the RepeatMasker libraries are correclty installed for dnaPipeTE. If your Java version is outdated, you can either update it (needs sudo rights) or use the provided version. For that you will need to execute the folling script: ./fixjava.sh

```
./fixjava.sh
source ~/.bashrc
```

To be sure that the fix worked, run again the script ./test_config.sh if you encounter difficulties with the RepeatMasker libraries, leave a message on the github forum

Test run

Once everything is OK, you are ready for a test run!!! A sample file is available either in .fastq or .fastq.gz (test_dataset.fastq[.gz]). This file is only provided to test if all the differents components of dnaPipeTE works well together.

```
cd ~yourpath/dnaPipeTE
python3 ./dnaPipeTE.py -input ./test/test_dataset.fastq -output ~/path/to/the/output_folder -genome_size 2000000
-genome_coverage 0.5 -sample_number 2
```

If the pipeline worked properly you should be able to see the 3 output graphs (piechart, bases_per_component and landscapes) with annotations similar to those provided in the folder `dnaPipeTE/test/test_dataset_example_output` . Since this is a very small dataset used with at low coverage, it is normal that running this test several times won't produce exactly the same estimate of the total amount of repeats.

> **Note.** The following warnings/errors might appear and are expected during the test run:
>
> ```
> awk: fatal: cannot open file \`test1/Trinity_run0/chrysalis/readsToComponents.out.sort' for reading (No such
> file or directory)
> ```
>
> *This is just a bug: since there is no iteration 0 of Trinity, there is no run 0 to look for. Yes, we need to fix this...*
>
> ```
> Parsing blast3 output...
>
> sort: cannot read: test1/blast_out/reads_vs_unannoted.blast.out: No such file or directory
>
> rm: cannot remove 'test1/blast_out/int.reads_vs_annoted.blast.out': No such file or directory
> ```
>
> *The toy dataset is from D. melanogaster, where most repeats are known and annotated. The few contigs created are likely to be annotated, thus, they might not be unnanoted contigs in the test run leading to these errors.*

2 - RUNNING dnaPipeTE

Input File

The input file must be a **single-end FASTQ or FASTQ.GZ** file of NGS reads. dnaPipeTE do not handle paired-end (we found chimerism issues in PE assembly of TEs). Typically, your input file is your cleaned sequencing output. dnaPipeTE will sample it (so you can put deep-sequencing data in input) to produce "low coverage" samples for the run (see in the next section for sample size). Using .fasta file as input works, however, since dnaPipeTE expects .fastq (4 lines per sequence), only half of the file will be considered (2 lines / 4 wont be read in the sampling step).

> **IMPORTANT: We recommend to remove from your reads mitochondrial and other none nucleic DNA, such as known symbionts or guts' bacterias. We found that if mitochondrial reads are left in the samples, RepeatMasker will annotate the corresponding conting to "Gypsy-12_DVir-I LTR/Gypsy" with however a weak score.**

Run dnaPipeTE

Move into dnaPipeTE folder and type:

```
cd ~yourdirectory/dnaPipeTE
python3 ./dnaPipeTE.py -input ~/path/to/your/input.fastq[.gz] -output ~/path/to/the/output_folder -cpu N
-genome_size N -genome_coverage N -sample_number N [...]
```

> **/!\ VERY IMPORTANT:** run dnaPipeTE from its installation folder, otherwise the dependant scripts won't run. This advice is important especially if you run it into a computing cluster/server, ask first your job to move to the install folder before executing the python command /!\

dnaPipeTE arguments:

Argument	Type	Description
-input	INPUT_FILE_PATH	input fastq or fastq.gz files (single end only). It will be sampled
-output	OUTPUT_FOLDER	complete path with name for the outputs
-cpu	INTEGER	maximum number of cpu to use
-sample_number	INTEGER	number of trinity iterations
-genome_size	INTEGER	size of the genome [use it with -genome_coverage; if used, do not use -sample_size] Ex. 175000000 for 175Mb
-genome_coverage	FLOAT	coverage of the genome for each sample [use it with -genome_size; if used, do not use -sample_size] Ex: 0.1 for 0.1X coverage per sample
-sample_size	INTEGER	number of reads to sample [use without -genome_size and -genome_coverage]
-RM_lib	PATH_TO_FILE.fasta	path to custom repeat library for RepeatMasker. The format myst be a valid .fasta file with for each repeat the following name: >Repeat_name#CLASS/Subclass with CLASS in "DNA, LINE, LTR, SINE, MITE, Helitron, Simple Repeat, Satellite" (if not set, default is to use RepeatMasker database)
-species	STRING	RepeatMasker library to use. Must be a valid NCBI for species or clade ex: homo,drosophila, "ciona savignyi". By default "All" is used. Do not used with -RM_lib
-RM_t	FLOAT	Annotation threshold: minimal percentage of the query (dnaPipeTE contig) aligned on the repeat to keep the anotation from RepeatMasker. Ex: 0.2 for 20% of query in db
-keep_Trinity_output		Keep Trinity output files at the end of the run. Default files are removed (large and numerous).
-Trin_glue	INTEGER	number of overlapping reads to join Inchworm (k-mer) contigs in Trinity (default 1)
-contig_length	INTEGER	minimum size of a repeat contig to be retained (default 200bp)

Continuing a crashed run:

> dnaPipeTE is capable to skip some steps if a run crashes after a checkpoint. For example, if it crashes during the Trinity assembly, the sampling won't be performed again if you launch the run again in the same output folder. The checkpoints are for now 1-sampling of Trinity inputs; 2- Trinity assembly. More to follow...

3 - dnaPipeTE OUTPUTS

dnaPipeTE produces a lot of outputs, some of them are very interesting.

The outfolder is divided into the following parts:

- **main folder (output name):**

important files:

File	Description
"Trinity.fasta"	this file contains the dnaPipeTE contigs, this is the last assembly performed with Trinity
"reads_per_component_and_annotation"	table with the count of reads and bp aligned per dnaPipeTE contigs (from blastn 1), as well as its best RepeatMasker annotation. Col1: counts (#reads);

File	Description
	Col2: aligned bases; Col3 dnaPipeTE contig name; col4 RepeatMakser annotation; col5 proportion of the dnaPipeTE contig that received the RM hit
"pieChart.pdf/png"	graph with the relative proportion of the main repeat classes, informs about the estimated proportion of repeats in the genome (from blastn 2 and 3)
"Bases_per_component.pdf/png"	graph with the number of base-pairs aligned on each dnaPipeTE contig (from blast 1), ordered by genome proportion of the dnaPipeTE contig.
"landscapes.pdf"	TEs landscape graphs (TE age distribution). Plot the blastn divergence distribution between reads and the contigs on which they map.

less important files you may like:

File	Description
"Trinity.fasta.out"	raw RepeatMasker output (not sorted) of Trinity.fasta on repbase libraries.
"Counts.txt"	count of bp of the sample aligned for each TE class (used for the pieChart)
"Reads_to_components_Rtable.txt"	input file to compute the reads and bp per contig (one line per reads)
"reads_landscape"	reads used for the landscape graph, including the blastn divergence from one reads to the contig on which it maps.

- **"Annotation" folder:**

important files:

File	Description
"one_RM_hit_per_Trinity_contigs"	sorted RepeatMasker output containing the best hit on repbase for each of the dnaPipeTE contigs (Trinity.fasta)
"Best_RM_annot_80_80"	subset of the previous table, including contigs for which at least 80% of the sequence is mapping to at least 80% percent of the target sequence.
"Best_RM_annot_partial"	same but for contigs for which at least 80% of the sequence is mapping to less than 80% percent of the target sequence
"[repeat-class].fasta"	subsets of the Trinity.fasta file for each repeat type detected by RepeatMasker
"unannotated.fasta"	subsets of the Trinity.fasta for contigs that didn't find any match...

- **"blast_out" folder:**

important files:

File	Description
"sorted.reads_vs_Trinity.fasta.blast.out"	best hit per reads from blastn 1
"sorted.reads_vs_annotated.blast.out"	best hit per reads from blastn 2
"sorted.reads_vs_unannotated.blast.out"	best hit per reads from blastn 3

less important files you may like:

File	Description
"reads_vs_[anything]"	raw blast out from previous files

- **Trinity_runX** Those files contains the raw Trinity outputs and intermediates files produced during assembly steps. For futher detail see the Trinity documentation (http://trinityrnaseq.sourceforge.net/)