Learn Git and GitHub without any code!

Using the Hello World guide, you'll start a branch, write comments, and open a pull request.

Read the guide

INTABiotechMJ / **MITE-Tracker**

MITE Tracker: An accurate approach to identify miniature inverted-repeat transposable elements in large genomes

8 stars 6 forks

Star Watch

Code Issues 1 Pull requests Actions Projects Wiki Se

master

juan and **juan** add feature to blast2gff ... on 26 May 2019 209

View code

readme.md

About

MITE Tracker: an accurate method for identifying miniature inverted-repeat transposable elements in large genomes.

An efficient and easy to run tool for discovering Miniature Inverted repeats Transposable Elements (MITEs) in genomic sequences. It is written in python 3 and uses ncbi's blast+ for finding inverted repeats and cdhit to do the clustering.

Large genomes can be processed in desktop computers.

Requirements

- tested in macOS 10.13.1, Debian 7.6, Ubuntu 16.04, Windows 7
- ncbi blast+ (Nucleotide-Nucleotide BLAST 2.6.0+)
- python requirements are in requirements.txt file (bipython and pandas)

Installation and running

```
# clone repo
git clone https://github.com/INTABiotechMJ/MITE-Tracker.git
cd MITE-Tracker

# blast
sudo apt-get install ncbi-blast+ virtualenv
# in macOS: brew install ncbi-blast+ virtualenv


#vsearch
wget https://github.com/torognes/vsearch/archive/v2.7.1.tar.gz
tar xzf v2.7.1.tar.gz
cd vsearch-2.7.1
#might need: sudo apt-get install autoconf
sh autogen.sh
./configure
make

#python dependencies
cd ..
virtualenv -p python3 venv
source venv/bin/activate
#might need: sudo apt-get install python3.6-dev
#if pandas failed to install, run: pip3 install cython
pip3 install -r requirements.txt

# running

python3 -m MITETracker -g /path/to/your/genome.fasta -w 3 -j jobname
```

```
# or to run in background

nohup python3 -m MITETracker -g /path/to/your/genome.fasta -w 3 -j jobname
&
```

In order to check the output and progress you can use these command (ctrl+c to exit)

```
#nohup will have the program output as well as the output from cdhit
execution
tail -f nohup.out
#out.log contaings a log file with timing information
tail -f results/[jobname]/out.log
```

Command line options

Argument	Description	Data type	Required or default
-g	Genome file in fasta format	string	required
-j	Jobname. Result files will be created in results/jobname	string	required
-w	Max number of processes to use simultaneously	int	1
-tsd_min_len	TSD min lenght	int	2
-tsd_max_len	TSD max lenght	int	10
-mite_min_len	MITE min lenght	int	50
-mite_max_len	MITE max lenght	int	650
--task	cluster or candidates	string	

Results

All the results are placed in *results/[yourjobname]/*. Here you will find: *families.fasta* all the MITEs sequences divided by families (custom format) *families_nr.fasta* with one MITE per family in fasta format *all.fasta* all MITEs in fasta format *all.gff3* a gff file describing all MITEs found

Troubleshooting

If getting any error while running the BLASTn searches please check you blast+ version

Running large genomes in different computers

This is an example of how we run wheat genome. Each chromosome can be run separately (--task candidates) in a different computers. Results should be merged together using *cat* and then run the cluster command (--task cluster). Files required for clustering are candidates.csv and candidates.fasta.

21 wheat chromosomes were downloaded in different files.

```
python3 -m MITETracker -g /media/chr1A.fasta -w 2 -j IWGSC_1A --task
candidates
python3 -m MITETracker -g /media/chr1B.fasta -w 2 -j IWGSC_1B --task
candidates
python3 -m MITETracker -g /media/chr1D.fasta -w 2 -j IWGSC_1D --task
candidates
python3 -m MITETracker -g /media/chr2A.fasta -w 2 -j IWGSC_2A --task
candidates
python3 -m MITETracker -g /media/chr2B.fasta -w 2 -j IWGSC_2B --task
candidates
python3 -m MITETracker -g /media/chr2D.fasta -w 2 -j IWGSC_2D --task
candidates
python3 -m MITETracker -g /media/chr3A.fasta -w 2 -j IWGSC_3A --task
candidates
python3 -m MITETracker -g /media/chr3B.fasta -w 2 -j IWGSC_3B --task
candidates
python3 -m MITETracker -g /media/chr3D.fasta -w 2 -j IWGSC_3D --task
candidates
python3 -m MITETracker -g /media/chr4A.fasta -w 2 -j IWGSC_4A --task
candidates
python3 -m MITETracker -g /media/chr4B.fasta -w 2 -j IWGSC_4B --task
candidates
...
mkdir results/IWGSC
cat results/IWGSC_*/candidates.csv > results/IWGSC/candidates.csv
cat results/IWGSC_*/candidates.fasta > results/IWGSC/candidates.fasta
python3 -m MITETracker -g none -w 3 -j IWGSC --task cluster
--min_copy_number 4
```

Publication and citing

https://bmcbioinformatics.biomedcentral.com/articles/10.1186/s12859-018-2376-y

Please cite with:

```
Crescente, Juan Manuel, et al. "MITE Tracker: an accurate approach to identify miniature inverted-repeat transposable elements in large genomes." BMC Bioinformatics 19.1 (2018): 348.
```

Or for bibtex users:

```
@article{crescente2018mite, title={MITE Tracker: an accurate approach to identify miniature inverted-repeat transposable elements in large genomes}, author={Crescente, Juan Manuel and Zavallo, Diego and Helguera, Marcelo and Vanzetti, Leonardo Sebasti{\'a}n}, journal={BMC Bioinformatics}, volume={19}, number={1}, pages={348}, year={2018}, publisher={Springer} }
```

Note:

Due to a problem with additionals files in the publication we have added those files in this repository under supplementary_materials/

rice_mites.fasta: Database of non-redundant MITE family database obtained from the rice genome

wheat_mites.fasta: Database of non-redundant MITE family database obtained from the wheat genome

tools_comparison.csv: Execution summary of MITE Tracker and other tools using several genomes

wheat_genes.csv: Wheat genes containing MITEs within its coding region.

Additional notes

Annotating all arabidopsis MITEs as an example

Clone MITETracker and install dependencies

```
#clone
git clone git@github.com:INTABiotechMJ/MITE-Tracker.git
#enter program directory
cd MITE-Tracker
#create virtual enviornment with python3
```

```
virtualenv -p python3 venv
#activate virtual environment
source venv/bin/activate
#install requirements
pip3 install -r requirements.txt
#run MITE Tracker
python3 MITETracker.py  -g TAIR10_chr_all.fas -j ata
```

With this version of TAIR genome we get a total of 38 distinct MITE families.

I'm gonna use the all.fasta file to map MITEs genome-wide because it contains all found elements.

```
blastn -task blastn -query results/ata/all.fasta  -subject ../data/tair10
/TAIR10_chr_all.fas -outfmt "6 qseqid sseqid qstart qend sstart send score
length mismatch gaps gapopen nident pident evalue qlen slen qcovs" >
results/ata/blast_families_ata.csv
```

Let's run out notebook for filtering blast results, run till the end. This will explain at each step how filtering is done and what are the results.

```
jupyter lab
```

Ultimately, convert the blast filtered output to gff

```
python blast2gff.py -i results/ata/blast_families_ata.filtered.csv  -o
results/ata/mitesInGenome.gff3 -n MITE_TRACKER
```

This is our resulting annotated file

```
results/ata/mitesInGenome.gff3
```

Releases

No releases published

Packages

No packages published

Languages

- **Jupyter Notebook** 86.8%
- **Python** 13.2%