Supplementary Note - Manual

This note provides a guide to help users use DeepTE in annotation of transposons in a genome.

Prerequisites

1. **Python**
   Available at https://www.python.org/downloads/. Developed and tested with version 3.7.1.
2. **Python packages**

   Use conda to install required packages (Recommend)
   1) Install conda: https://www.anaconda.com/products/individual
   2) conda create -n py36 python=3.6
   3) conda activate py36
   4) conda install tensorflow-gpu=1.14.0
   5) conda install biopython
   6) conda install keras=2.2.4
   7) conda install numpy=1.16.0

   Use pip to install required packages
   1) biopython: pip install biopython
   2) numpy: pip install numpy==1.16.0 (Developed and tested with version 1.16.0)
   3) tensorflow: pip install tensorflow==1.14.0 (Developed and tested with version 1.14.0)
   4) keras: pip install keras==2.2.4 (Developed and tested with version 2.2.4)

Optional tools

1. **HMMER**
   Available at http://hmmer.org/download.html. Developed and tested with version 3.1b1.

DeepTE Installation

1. **Download DeepTE**
   git clone https://github.com/LiLabAtVT/DeepTE.git.

Example Run

In this example we use maize 'B73' genome (maize.fas) which can be download from https://phytozome.jgi.doe.gov/pz/portal.html#!info?alias=Org_Zmays. We provide a simple example to help users annotate TEs in this maize genome with help of DeepTE.

1. **Create a transposon library from genomes.**
   Users can choose one or more following tools to develop transposon library in the maize genome:
   RepeatModeler (http://www.repeatmasker.org/RepeatModeler/);
   LTR_retriever (https://github.com/oushujun/LTR_retriever);
   LTR_finder (https://code.google.com/archive/p/ltr-finder/downloads);
   LTRharvest (http://genometools.org/);
   miteFinder (https://github.com/screamer/miteFinder)
   Users can also develop the library based on other tools that are not listed as above.

   1) **RepeatModeler**

      Command:

      ```
      RepeatModeler_Path/BuildDatabase -name maize maize.fas
      mv consensi.fa.classified repeatmodeler.lib
      ```

      Note: RepeatModeler will generate **consensi.fa.classified** file that is **maize transposon library**. The '**consensi.fa.classified**' is changed to '**repeatmodeler.lib**' to make the output more clear.

2) **LTR-retriever**

Command:

```
ltr_finder -D 15000 -d 1000 -L 7000 -l 100 -p 20 -C -M 0.9 maize.fas > ltrfinder.scn;
gt suffixerator -db maize.fas -indexname index_maize -tis -suf -lcp -des -ssp -sds -dna;
gt ltrharvest -index index_maize -similar 90 -vic 10 -seed 20 -seqids yes -minlenltr 100 -maxlenltr 7000 -mintsd 4 -maxtsd 6 -motif TGCA -motifmis 1 >  ltrharvest.scn;
LTR_retriever -genome maize.fas -inharvest ltrharvest.scn -infinder ltrfinder.scn
```

Note: LTR_retriever will generate multiple outputs, and ***.LTRlib.fa** is **maize transposon library**.

3) **miteFinder**

Command:

```
miteFinder -input  maize.fas -output maize_miteFinder.lib -pattern_scoring mite_finder2_profile/pattern_scoring.txt -threshold 0.5
```

Note: pattern_scoring.txt is from profile directory of miteFinder downloaded from https://github.com/screamer/miteFinder/tree/master/profile. The output file (**maize_miteFinder.lib**) is **maize transposon library**.

4) **REPET**

TEdenovo included in the REPET (https://urgi.versailles.inra.fr/Tools/REPET) can also help users to construct TE library. TEdenovo contains eight steps in its instruction, and has a clear description for each step. Please follow the instruction in TEdenovo (https://urgi.versailles.inra.fr/Tools/REPET/TEdenovo-tuto). After the step 4 (Build consensus), users will obtain a consensus TE file which can be used as input TE library in the following step.

2. Use DeepTE to classify transposons in the libraries into families.

1) Create working and output directories

Command:

```
mkdir working_dir
mkdir output_dir
```

2) Run DeepTE

a. Run DeepTE without modification.

Command:

```
python DeepTE.py -d working_dir -o output_dir -i repeatmodeler.lib -m P -sp P
```

Note: Users can use transposon libraries generated from the previous step. The libraries can be derived from single tool or multiple tools listed above. The combined library can be generated using ‘cat’. For example: cat repeatmodeler.lib ltr_retriever.lib miteFinder.lib > combine.lib. This example will use library from RepeatModeler. The output_dir will contains opt_DeepTE.fasta file that is the input transposon library in RepeatMasker tool.

b. Run DeepTE with modification.

Users need to download and install hmmscan tool (http://hmmer.org/).

Command:

```
mkdir working_dir_md
mkdir output_dir_md
python DeepTE_domain.py -d working_dir_md -o output_dir_md -i repeatmodeler.lib -s DeepTE/supfile_dir --hmmscan hmmscan
python DeepTE.py -d working_dir -o output_dir -i repeatmodeler.lib -m P -sp P -modify
opt_te_domain_pattern.txt
```

Note: If users detect domains of TEs to improve performance of DeepTE, they need to use DeepTE_domain.py with help of hmmscan tool. The opt_te_domain_pattern.txt can be found in the output_dir_md. The output_dir will contains opt_DeepTE.fasta file that is the input transposon library in RepeatMasker tool. Users can easily track which TE is corrected in the working directory (working_dir/store_temp_opt_dir/All_results.txt and working_dir/store_temp_opt_dir/ClassI_results.txt).

Caution: This correction step ignores nested TEs. For example, DeepTE assigns an LTR/TE into LTR class, but body region of this LTR/TE is inserted by another nLTR/TE. The correction process will move this LTR/TE to nLTR class since it identifies a 'EN' domain in-side this LTR/TE. In this case, the LTR/TE is a nested TE which com-bines a single LTR/TE and a single nLTR/TE, but the correction step falsely classifies the LTR/TE to nLTR/TE. However, this correction step provides a clue that the corrective TE may be a nested TE.

3. **Use RepeatMasker to mask individual transposons in the maize genome**

Users need to download and install RepeatMasker (http://www.repeatmasker.org/RMDownload.html).

Command:

```
mkdir repeatmasker_opt_dir
RepeatMasker maize.fas -lib opt_DeepTE.fasta -gff -dir repeatmasker_opt_dir
```

Note: The output file *.out.gff is the annotation information of transposons.

An overview of all parameters

Table 1. All parameters for DeepTE.py

Required parameters	
-d [working directory]	Working directory stores intermediate files of each step.
-o [output directory]	Output directory stores the output files.
-i [input fasta file]	Input sequences that are unknown transposons or DNA sequences. If -UNS is not initiated, the input sequences are unknown transposons, otherwise, are DNA sequences.
-m [model name]	Model name: '-m P' or '-m M' or '-m F' or '-m O' or '-m U'. The model directory will be automatically downloaded if users initiate this argument. P or M or F or O. P: Plants, M: Metazoans, F: Fungi, and O: Others. If '-m U' is initiated, please set '-UNS yes' to classify unknown sequences.
-m_dir [model directory]	Model directory that can be downloaded from links in https://github.com/LiLabAtVT/DeepTE. If users set -UNS yes, please provide UNS_model directory that can be downloaded in the above link.
-sp [species type]	Specify which species the input sequences belong to. P or M or F or O. P: Plants, M: Metazoans, F: Fungi, and O: Others.
Optional parameters	
-fam [transposon family name]	Provide TE family name for the input transposon sequence. ClassI: the input sequence is Class I TEs. ClassII: the input sequence is Class II subclass1 TEs. LTR: the input sequence is LTR TEs. nLTR: the input sequence is nLTR TEs. LINE: the input sequence is LINE TEs. SINE: the input sequence is SINE TEs. Domain: the input sequence is Class II subclass1 TEs with specified super families. If users do not initiate '-fam', DeepTE will regard your input sequences are unknown TEs.

-modify [domain information file]	Domain file is generated from script: DeepTE_domain.py in https://github.com/LiLabAtVT/DeepTE. This modification argument helps to increase performance of DeepTE tool.
-UNS [Yes]	If set this argument, users need change the input fasta file to the DNA sequences instead of unknown transposons; This argument will classify the sequences into transposons (TEs), coding sequences (CDS), or Intergenic sequences (INS). Also, -sp and -fam do not need to provide. Note: this model is used for plants rather than metazoans and fungi.

Table 2. All parameters for DeepTE_domain.py

Required parameters	
-d [working directory]	Working directory stores intermediate files of each step.
-o [output directory]	Output directory stores the output files.
-i [input fasta file]	Input sequences that are unknown transposons.
-s [supplementary directory]	Specify supplementary directory that contains required files to generate domain information of the transposons. It is from supfile_dir from https://github.com/LiLabAtVT/DeepTE.
--hmmscan [hmmscan tool]	Specify file path to hmmscan executable. For example, /usr/bin/hmmscan.