Transposon Annotator "reasonaTE"

Transposon annotation tool for the annotation of transposons, transposon characteristic proteins and structural elements of transposons. *reasonaTE* is part of TransposonUltimate.

- **Input**: Genome assembly (FASTA file).
- **Output**: Lots of transposon annotations (GFF3 file).

Installation

The reasonaTE pipeline comes with two conda environments due to package incompatibilities. For some steps of the environment you will need the first, for others the second conda environment.

- **Note1:** *Please make sure you have "RepeatMasker" and "RepeatModeler" installed on your machine as well if you want the pipeline to consider their annotations as well. As issues with the conda packages of these tools are reported multiple times on the internet and github, we recommend to not use the conda packages of these tools.*
- **Note2:** *For some users the bioconda channel is reported to cause issues with genometools-genometools, therefore you might consider to download it from other channels, e.g. conda-forge: "conda install -y -c bioconda -c conda-forge genometools-genometools".*
- **Note3:** *Some users experience problems with long "environment solving" times of conda. We therefore recommend the use of mamba to accelerate the installation process.*

Installation using conda and mamba (recommended)

```
# Environment 1 - including all annotation tools
conda create -y --name transposon_annotation_tools_env python=2.7
conda activate transposon_annotation_tools_env
conda install -y mamba
#conda install -y -c bioconda repeatmodeler repeatmasker # Recommended not
too install via conda
mamba install -y -c bioconda genometools-genometools # for some users: mamba
install -y -c bioconda -c conda-forge genometools-genometools
mamba install -y -c derkevinriehl transposon_annotation_reasonate
mamba install -y -c derkevinriehl
transposon_annotation_tools_proteinncbicdd1000
conda install -y -c derkevinriehl
transposon_annotation_tools_transposonpsicli
mamba install -y -c derkevinriehl transposon_annotation_tools_mitetracker
mamba install -y -c derkevinriehl transposon_annotation_tools_sinescan=1.1.2
mamba install -y -c derkevinriehl transposon_annotation_tools_helitronscanner
mamba install -y -c derkevinriehl transposon_annotation_tools_mitefinderii
mamba install -y -c derkevinriehl transposon_annotation_tools_mustv2
mamba install -y -c derkevinriehl transposon_annotation_tools_sinefinder
mamba install -y -c anaconda biopython
conda deactivate
# Environment 2 - including CD-Hit and Transposon Classifier RFSB
```

```
conda create -y --name transposon_annotation_reasonaTE
conda activate transposon_annotation_reasonaTE
conda install -y mamba
mamba install -y -c anaconda biopython
mamba install -y -c bioconda cd-hit blast seqkit
mamba install -y -c derkevinriehl transposon_annotation_reasonate
transposon_classifier_rfsb
conda deactivate
```

Installation using yml file (works for Linux64, other OS possible)

```
wget
https://raw.githubusercontent.com/DerKevinRiehl/transposon_annotation_reasona
TE/main/environment_yml/transposon_annotation_tools_env.yml
wget
https://raw.githubusercontent.com/DerKevinRiehl/transposon_annotation_reasona
TE/main/environment_yml/transposon_annotation_reasonaTE.yml
conda env create -f transposon_annotation_tools_env.yml
conda env create -f transposon_annotation_reasonaTE.yml
```

Installation using plain conda (not recommended, can take long time)

```
# Environment 1 - including all annotation tools
conda create -y --name transposon_annotation_tools_env python=2.7
conda activate transposon_annotation_tools_env
#conda install -y -c bioconda repeatmodeler repeatmasker # Recommended not
too install via conda
conda install -y -c bioconda genometools-genometools # for some users: conda
install -y -c bioconda -c conda-forge genometools-genometools
conda install -y -c derkevinriehl transposon_annotation_reasonate
conda install -y -c derkevinriehl
transposon_annotation_tools_proteinncbicdd1000
conda install -y -c derkevinriehl
transposon_annotation_tools_transposonpsicli
conda install -y -c derkevinriehl transposon_annotation_tools_mitetracker
conda install -y -c derkevinriehl transposon_annotation_tools_sinescan=1.1.2
conda install -y -c derkevinriehl transposon_annotation_tools_helitronscanner
conda install -y -c derkevinriehl transposon_annotation_tools_mitefinderii
conda install -y -c derkevinriehl transposon_annotation_tools_mustv2
conda install -y -c derkevinriehl transposon_annotation_tools_sinefinder
conda install -y -c anaconda biopython
conda deactivate
# Environment 2 - including CD-Hit and Transposon Classifier RFSB
conda create -y --name transposon_annotation_reasonaTE
conda activate transposon_annotation_reasonaTE
conda install -y -c anaconda biopython
conda install -y -c bioconda cd-hit blast seqkit
conda install -y -c derkevinriehl transposon_annotation_reasonate
transposon_classifier_rfsb
conda deactivate
```

How to use "reasonaTE"

Step 1) Create a project

```
conda activate transposon_annotation_tools_env
mkdir workspace
wget
https://raw.githubusercontent.com/DerKevinRiehl/transposon_annotation_reasona
TE/main/workspace/testProject/sequence.fasta # demo fasta you could use
reasonaTE -mode createProject -projectFolder workspace -projectName
testProject -inputFasta sequence.fasta
```

Step 2) Annotate genome with annotation tools To annotate the genome with different annotation tools, four possible ways exist. We recommend *Option 2* as it allows for parallelization which is vital for reducing processing times for very large genomes.

Option 1: annotate with all tools automatically (this does not include ltrPred). This will annotate the genome with all tools (except for ltrPred) with standard parameters and tool after tool.

```
conda activate transposon_annotation_tools_env
reasonaTE -mode annotate -projectFolder workspace -projectName testProject -
tool all
```

Option 2: annotate with one specific tool (good for parallelization or rerunning, recommended). It is mandatory to run the protein annotation tools *transposonPSI* and *NCBICDD1000* for the next steps.

```
conda activate transposon_annotation_tools_env
reasonaTE -mode annotate -projectFolder workspace -projectName testProject -
tool helitronScanner
reasonaTE -mode annotate -projectFolder workspace -projectName testProject -
tool ltrHarvest
reasonaTE -mode annotate -projectFolder workspace -projectName testProject -
tool mitefind
reasonaTE -mode annotate -projectFolder workspace -projectName testProject -
tool mitetracker
reasonaTE -mode annotate -projectFolder workspace -projectName testProject -
tool must
reasonaTE -mode annotate -projectFolder workspace -projectName testProject -
tool repeatmodel
reasonaTE -mode annotate -projectFolder workspace -projectName testProject -
tool repMasker
reasonaTE -mode annotate -projectFolder workspace -projectName testProject -
tool sinefind
reasonaTE -mode annotate -projectFolder workspace -projectName testProject -
tool sinescan
reasonaTE -mode annotate -projectFolder workspace -projectName testProject -
tool tirvish
reasonaTE -mode annotate -projectFolder workspace -projectName testProject -
tool transposonPSI
reasonaTE -mode annotate -projectFolder workspace -projectName testProject -
tool NCBICDD1000
```

Option 3: run annotation tools with specified parameters (for advanced users) If you want reasonaTE to call annotation tools with specific parameters, but do not want to take care of the locations of input and output files, you can do so as shown in the following example:

```
reasonaTE -mode annotate -projectFolder workspace -projectName testProject -
tool tirvish xxxxx -mintsd 5
```

Additional parameters need to follow after five x symbols "xxxxx". Please note, do only set parameters that are not related to locations of input and output files. If you want total control please have a look at Option 3.

Option 4: run annotation tools completely with user specified parameters (for expert users) For this purpose, we provide conda packages of all transposon_annotation_tools except for ltrPred. Please use the fasta file with renamed sequence names of the workspace project folder. (e.g. *workspace/testProject/sequence.fasta*) Please note, as some tools (HelitronScanner, MiteFinderII, MITE-Tracker, SINE-Finder, TIRvish) do not annotate on both strands, we recommend to run these on the reverse complementary as well (e.g. *workspace/testProejct/sequence_rc.fasta*). Once you annotated the genomes with your own specified parameter settings, please copy the result files into the workspace's project Folder as shown in the example project (e.g. results of HelitronScanner to copy into *workspace/testProject/helitronScanner*) and rename the files accordingly. Please note, it is mandatory to run the protein annotation tools *transposonPSI* and *NCBICDD1000* for the next steps using the commands of *option 2*.

Running ltrPred: If you want to include ltrPred annotations into the pipeline as well, install and run ltrPred. Later on, copy the result files into the project folder (*workspace/testProject/ltrPred*) and rename the files accordingly. Please find our tutorial for manually running LTRpred even without docker using the conda package udocker. Based on our experience, ltrPred contributed valuable annotations including transposons and structure features. However, we were not able to create a conda package for easy and automated use, and it takes manual efforts to run it.

Check status of annotation tools: If you are running multiple annotation tools in parallel, or run the manually, copied and renamed the result files into the workspace folder, you can check the status of the annotation files by:

```
conda activate transposon_annotation_tools_env
reasonaTE -mode checkAnnotations -projectFolder workspace -projectName
testProject
>Checking helitronScanner        ... completed
>Checking ltrHarvest     ... completed
>Checking ltrPred        ... completed
>Checking mitefind       ... completed
>Checking mitetracker    ... completed
>Checking must   ... completed
>Checking repeatmodel    ... completed
>Checking repMasker      ... completed
>Checking sinefind       ... completed
>Checking sinescan       ... completed
>Checking tirvish        ... completed
>Checking transposonPSI  ... completed
>Checking NCBICDD1000    ... completed
```

All files that are reported as "completed" will be considered by **reasonaTE** in the next steps.

Step 3) Parse annotations Each of the tools will produce different output file formats. **reasonaTE** therefore provides a parser module that will unify different output files to one standardized format (GFF3). The parser module will automatically detect annotations that are available as a result from step 2, and only the available files will be considered in the next steps by the pipeline.

```
conda activate transposon_annotation_tools_env
reasonaTE -mode parseAnnotations -projectFolder workspace -projectName
testProject
```

If you are unsure about the status of the parsing, you can run following command:

```
conda activate transposon_annotation_tools_env
reasonaTE -mode checkParsed -projectFolder workspace -projectName testProject
```

Step 4) Run the pipeline on the genome annotations

```
conda activate transposon_annotation_reasonaTE
reasonaTE -mode pipeline -projectFolder workspace -projectName testProject
```

Step 5) Calculate final statistics Once all results are calculated, summarizing statistics can be generated using:

```
conda activate transposon_annotation_reasonaTE
reasonaTE -mode statistics -projectFolder workspace -projectName testProject
```

The results will be print to console and stored to the statistics files (see section "Documentation of output files" below). The results consist of three tables, presenting the number of transposons, the number of base pairs included by the transposon annotations and the number of base pairs annotated by the transposon mask annotation. The numbers are present by transposon class (horizontaly) and sequence (verticaly, using the renamed sequence names and the original sequence names) for the first two mentioned numbers, and just by sequences for the last mentioned number. All reported values are separated by tabulator. The three tables are separated by two empty lines.

```
SeqID   SeqName #Num_transposons by classes
SeqID   SeqName all     1       1/1     1/1/1   1/1/2   1/1/3   1/2     1/2/1
        1/2/2   2       2/1     2/1/1   2/1/2   2/1/3   2/1/4   2/1/5   2/1/6
        2/2     2/3
all     all     28750   1290    1267    279     977     11      23      15      8
        27460   27000   2276    4292    2952    1716    15674   90      77
        383
seq1    chrI    3707    220     218     43      173     2       2       1       1
        3487    3430    265     566     343     313     1935    8       10
        47
seq2    chrII   5041    194     187     29      157     1       7       2       5
        4847    4760    366     625     605     240     2908    16      7
        80
...
```

```
SeqID   SeqName #BP_transposons by classes
SeqID   SeqName all     1       1/1     1/1/1   1/1/2   1/1/3   1/2     1/2/1
        1/2/2   2       2/1     2/1/1   2/1/2   2/1/3   2/1/4   2/1/5   2/1/6
        2/2     2/3
all     all     30914788        5533640 5410363 951426  4438898 20039   123277
        120973  2304    25381148        23998275        2331383 7151643 6435895
        853227  7149583 76544   1097361 285512
seq1    chrI    4950902 1017833 1013925 204630  806578  2717    3908    3660
        248     3933069 3356902 264572  1364087 688504  155671  876952  7116
        520558  55609
seq2    chrII   4934076 583591  574266  102006  471723  537     9325    7763
        1562    4350485 4192651 283217  1297024 901063  138632  1553229 19486
        69580   88254
...


SeqID   SeqName #BP_transposons
all     all23412418
seq1    chrI    3875312
seq2    chrII   3492273
...
```

Usage Parameter Summary

ModeNr	Mode	Parameter	Mandatory	Description
1	"createProject"	projectFolder	(mandatory)	Directory to create annotation projects in (=annotation workspace)
		projectName	(mandatory)	Desired name of the annotation project
		inputFasta	(mandatory)	Genome file (FASTA) that should be annotated for transposons
2	"annotate"	projectFolder	(mandatory)	Directory with annotation projects (=annotation workspace)
		projectName	(mandatory)	Name of the annotation project
		tool	(mandatory)	Annotation tool that should be used. Possible options: "helitronScanner", "ltrHarvest", "mitefind", "mitetracker", "must", "repeatmodel", "repMasker", "sinefind", "sinescan", "tirvish", "transposonPSI", "NCBICDD1000", "all"
3	"checkAnnotations"	projectFolder	(mandatory)	Directory with annotation projects (=annotation workspace)
		projectName	(mandatory)	Name of the annotation project
4	"parseAnnotations"	projectFolder	(mandatory)	Directory with annotation projects (=annotation workspace)
		projectName	(mandatory)	Name of the annotation project
5	"checkParsed"	projectFolder	(mandatory)	Directory with annotation projects (=annotation workspace)

ModeNr	Mode	Parameter	Mandatory	Description
		projectName	(mandatory)	Name of the annotation project
6	"pipeline"	projectFolder	(mandatory)	Directory with annotation projects (=annotation workspace)
		projectName	(mandatory)	Name of the annotation project
7	"statistics"	projectFolder	(mandatory)	Directory with annotation projects (=annotation workspace)
		projectName	(mandatory)	Name of the annotation project
8	"sequenceRenamer"	seqNames	(mandatory)	sequence_heads.txt file location with original and new sequence names
		inputGFF	(mandatory)	Input GFF file
		outputGFF	(mandatory)	Target location of GFF file with renamed (=original) sequences

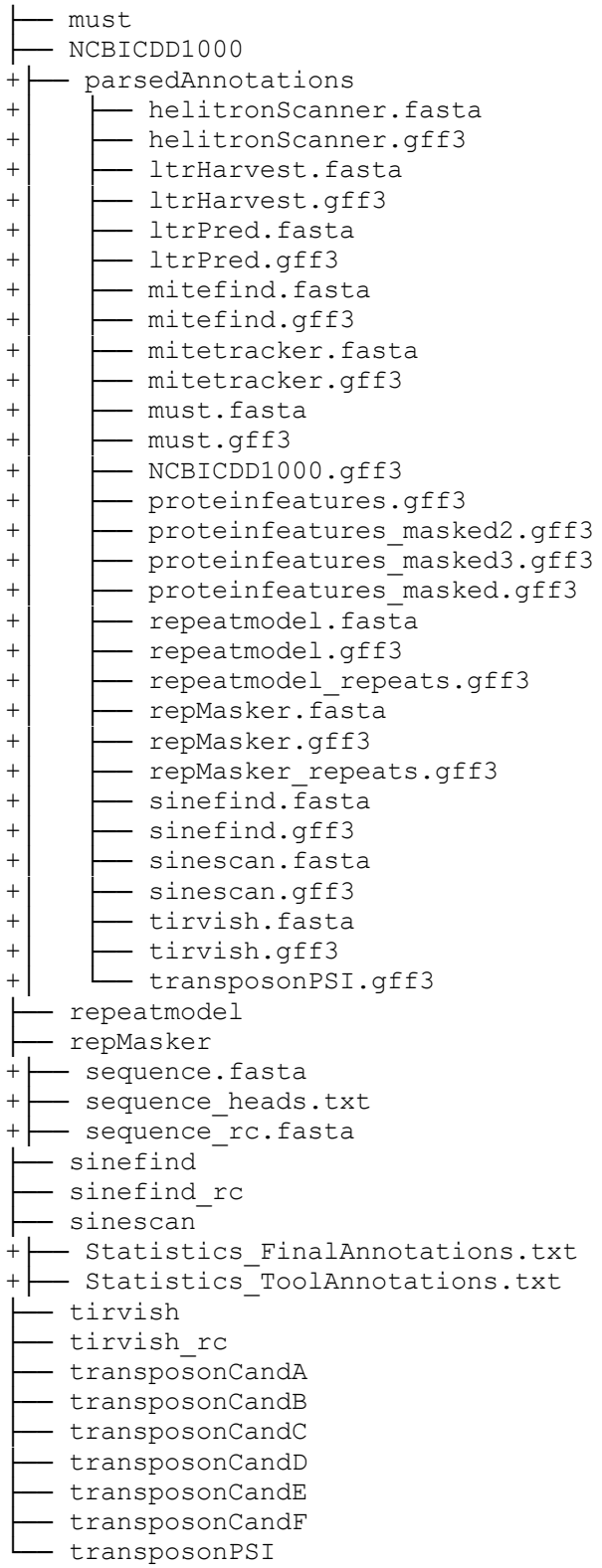
Documentation of output files

Introduction The outputs of the pipeline consist of mainly two parts:

- Tool Annotations = merging the annotations by annotation software tools
- Pipeline Annotations = Tool annotations + additional copies found in the genome

Project folder structure Inside a project's folder (e.g. *testProject*) there are multiple output folders, that are presented in the following. The collapsed folders and marked files (by the + symbol in green) represent the relevant output files:

```
+├── finalResults
+│   ├── FinalAnnotations_ProteinFeatures.gff3
+│   ├── FinalAnnotations_StructuralFeatures.gff3
+│   ├── FinalAnnotations_TransposonMask.gff3
+│   ├── FinalAnnotations_TransposonSequences.fasta
+│   ├── FinalAnnotations_Transposons.gff3
+│   ├── PipelineAnnotations_ProteinFeatures.gff3
+│   ├── PipelineAnnotations_TransposonMask.gff3
+│   ├── PipelineAnnotations_TransposonSequences.fasta
+│   ├── PipelineAnnotations_Transposons.gff3
+│   ├── ToolAnnotations_ProteinFeatures.gff3
+│   ├── ToolAnnotations_StructuralFeatures.gff3
+│   ├── ToolAnnotations_TransposonMask.gff3
+│   ├── ToolAnnotations_TransposonSequences.fasta
+│   └── ToolAnnotations_Transposons.gff3
├── helitronScanner
├── helitronScanner_rc
├── ltrHarvest
├── ltrPred
├── mitefind
├── mitefind_rc
├── mitetracker
├── mitetracker_rc
```



First of all, the fasta file used for the creation of the project was copied to *sequence.fasta*. The sequences in the fasta file were renamed, a matching can be found in *sequence_heads.txt*. Also, the reverse complement sequence was copied to *sequence_rc.fasta* for all softwares that annotate

a single strand only. If you would like to use the original sequence names, you can do so using Mode 8 of reasonaTE (see table before).

Moreover, *Statistics_FinalAnnotations.txt* and *Statistics_ToolAnnotations.txt* contain the statistics produced by the statistics mode for the two outputs of **reasonaTE**.

The folder *parsedAnnotations* includes the parsed transposon annotations, structural feature annotations and transposon characteristic protein annotations by the different software tools in GFF3 format, as well as extracted sequences for each annotation in a FASTA file.

The folder *finalResults* includes all results - including the tool and pipeline annotations. The *ToolAnnotations_* files contain the tool annotations, the *PipelineAnnotations_* files contain the additional copies found and the *FinalAnnotations_* include both of the prior merged into one file. There are files of the annotated transposons, transposon characteristic proteins, structural features, the mask of transposon regions and the extracted and classified sequences as FASTA file. As transposon annotations are not intersection free and can include nested or overlapping transposon annotations, the basepairs annotated in the mask represent all base pairs that are annotated by one or more transposons of the transposon annotations.

Citations

Please cite our paper if you find TransposonUltimate useful:

Kevin Riehl, Cristian Riccio, Eric A Miska, Martin Hemberg, TransposonUltimate: software for transposon classification, annotation and detection, Nucleic Acids Research, 2022; gkac136, https://doi.org/10.1093/nar/gkac136

```
@article{riehl2022transposonultimate,
  title={TransposonUltimate: software for transposon classification,
annotation and detection},
  author={Riehl, Kevin and Riccio, Cristian and Miska, Eric and Hemberg,
Martin},
  journal={Nucleic Acids Research},
  year={2022}
}
```