Transposable Elements in the Era of Data Science

Hand-On session/Demo: LTR retrotransposon detector and classificator using Deep Learning

Romain Guyot (IRD), Simon Orozco-Arias (UAM)

May 13th

Introduction

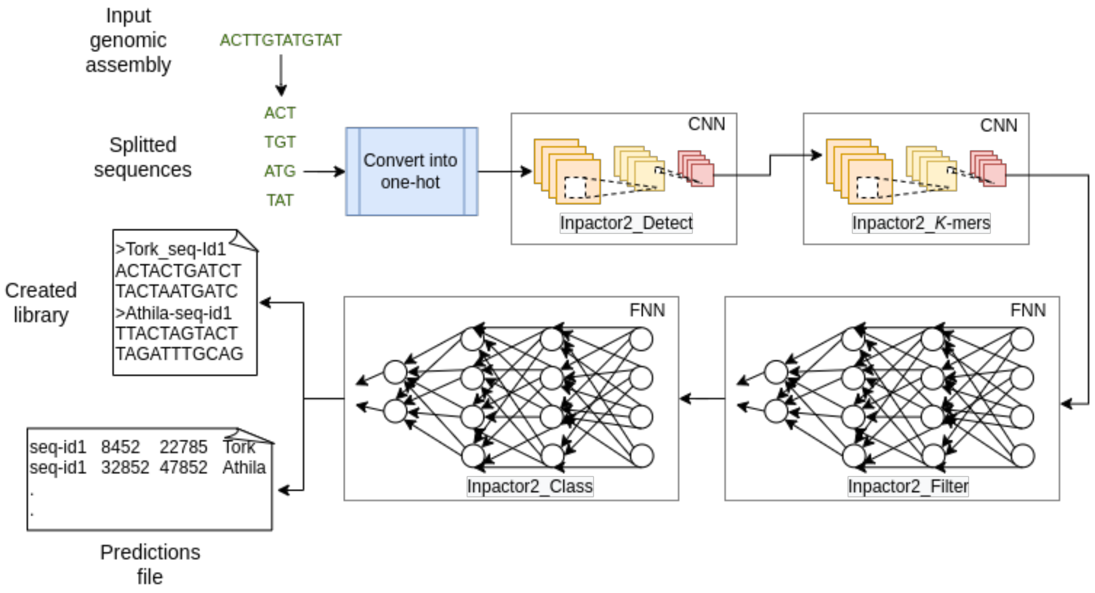
What is Inpactor2 ?

https://github.com/simonorozcoarias/Inpactor2/tree/main

Inpactor2 was designed and developed to detect reference **LTR retrotransposons** (LTR-RTs), filtering out those elements that correspond to fragments or have nested insertions. In addition, Inpactor2 classifies LTR-RTs down to the **family** level.

Optionally, this tool annotates the elements discovered through RepeatMasker.

Inpactor2 uses neural networks to perform its tasks accurately while maintaining high sensitivity and a low false positive rate.



```
>Tork_seq-Id1
ACTACTGATCT
TACTAATGATC
>Athila-seq-id1
TTACTAGTACT
TAGATTTGCAG
```

```
seq-id1   8452    22785   Tork
seq-id1   32852   47852   Athila
.
.
```

Introduction

What is Inpactor2 ?

https://github.com/simonorozcoarias/Inpactor2/tree/main

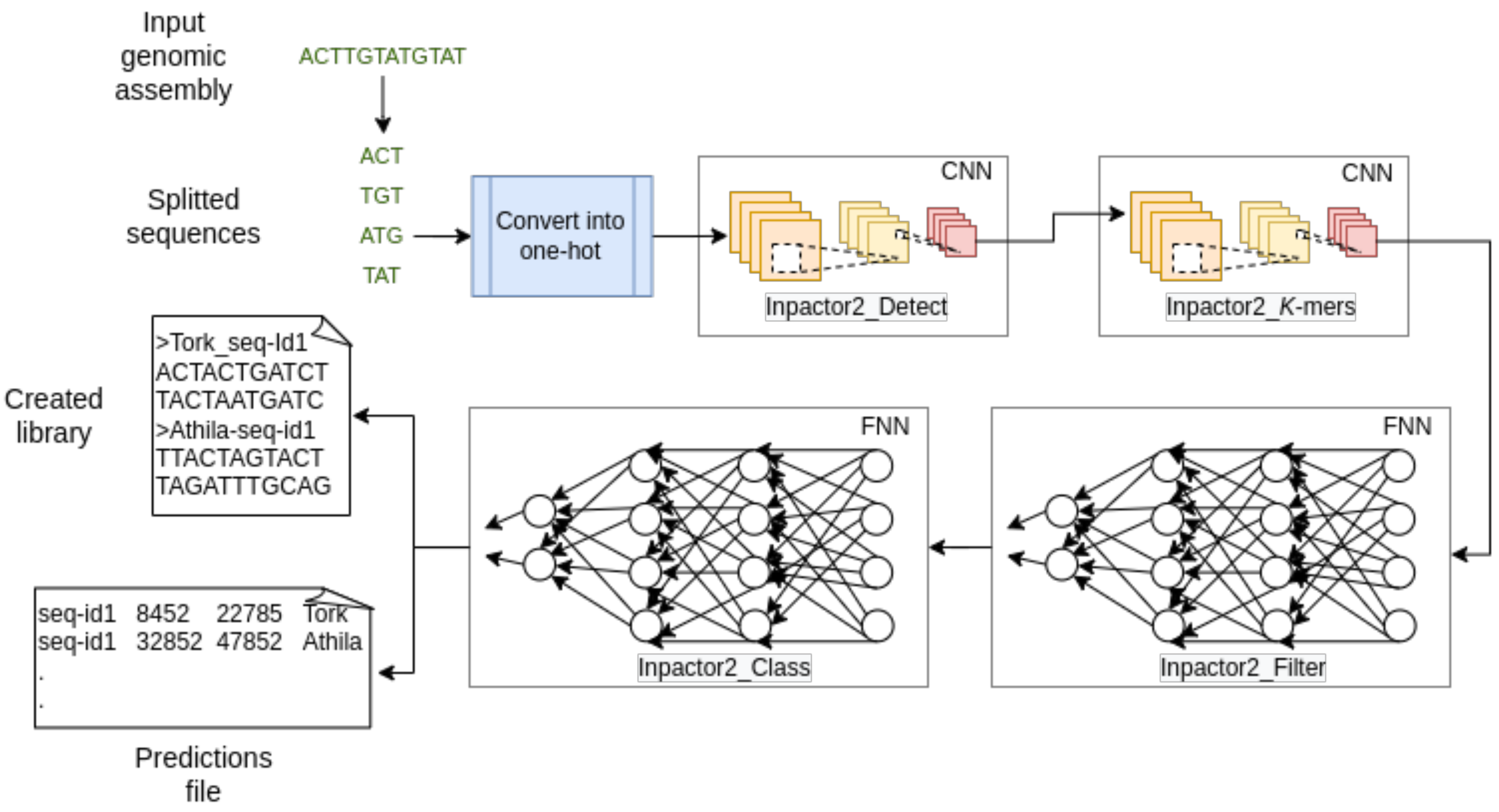
Other tools of TE detection and classification using DL or ML:

Tools	DL	Detection	Classification	Target	Links
Inpactor2	CNN	yes	yes	LTR retrotransposon superfamily/family in plant	https://github.com/simonorozcoarias/Inpactor2
DeepTE	CNN	no	yes	ALL TE/organism types	https://github.com/LiLabAtVT/DeepTE
TERL	CNN	no	yes	ALL TE type	https://github.com/muriloHoracio/TERL
TEclass	no/svm	no	yes	ALL TE type	https://www.compgen.uni-muenster.de/tools/teclass/index.hbi?
TEclass2	Transformers based DL	no	yes	ALL TE type	https://github.com/IOB-Muenster/TEclass2?tab=readme-ov-file

Introduction

How it works ?

https://github.com/simonorozcoarias/Inpactor2/tree/main



```
>Tork_seq-Id1
ACTACTGATCT
TACTAATGATC
>Athila-seq-id1
TTACTAGTACT
TAGATTTGCAG
```

```
seq-id1  8452   22785  Tork
seq-id1  32852  47852  Athila
.
.
```

Hand-on session

Installation of Inpactor2

Preparation

```
git clone https://github.com/simonorozcoarias/Inpactor2.git
```

For linux users:

```
conda env create -f Inpactor2/Inpactor2.yml

conda activate Inpactor2
```

Inpactor2.yml ***not working actually on Mac Intel/M cpu***

For Mac users:

only intel cpu, use `Inpactor2_mac.yml` (not working with ARM)

Download the *Utricularia gibba* genome (https://doi.org/10.1038/nature12132) **or** the genome of your choice !

```
wget https://ftp.ncbi.nlm.nih.gov/genomes/all/GCA/002/189/035/
GCA_002189035.1_U_gibba_v2/GCA_002189035.1_U_gibba_v2_genomic.fna.gz
```

Hand-on session

Running Inpactor2

Running impactor2 without annotation by Repeat Masker (26 min with one CPU)

```
python3 ./Inpactor2.py -f GCA_002189035.1_U_gibba_v2_genomic.fna -o .
```

Results

```
Inpactor2_library.fasta
Inpactor2_predictions.tab
```

Post analysis of Inpactor2_library.fasta:

```
python Inpactor2_dereplicate.py
```

—>Remove redundancy of Inpactor2_library.fasta using mmseq (recommended before RepeatMasker)

```
python Inpactor2_utils_stats.py
```

—>give the sequence number per family and length (with outliers)

```
python Inpactor2_utils_stats_clean.py
```

—> remove outliers from the library if necessary

Hand-on session

Masking the genome with Inpactor2 library

Using Alignment-free algorithm to make the *utricularia* genome.

```
git clone https://github.com/NGSEP/NGSEPcore.git
```

Running TransposonsFinder (with 1 CPU=7m8s)

```
java -jar NGSEPcore/olderVersions/NGSEPcore_5.0.0.jar TransposonsFinder -i
GCA_002189035.1_U_gibba_v2_genomic.fna -d
GCA_002189035.1_U_gibba_v2_genomic.fna_RESULTS/Inpactor2_library.fasta -o out.gff
```

Running RepeatMasker (with 1 CPU=35m21s)

```
RepeatMasker -e ncbi -pa 1 -u -nolow -lib Inpactor2_library.fasta
GCA_002189035.1_U_gibba_v2_genomic.fna
```

Compare TF and RP results