Starling-May18
Projects/Katarina Stuart/KStuart.Starling-Aug18/Sv3_Genome/Annotation/2020-04-06.LTR

PDF Version generated by

Katarina Stuart (z5188231@ad.unsw.edu.au)

on

Jun 23, 2022 @04:12 PM NZST

Table of Contents

2020-04-06.LTR

Katarina Stuart (z5188231@ad.unsw.edu.au) - Mar 23, 2022, 8:39 PM NZDT

LTR (long terminal repeat) retrotransposons

https://github.com/xvazquezc/genome_annotation_with_Maker2/blob/master/advanced_repeat_library/Advanced_repeat_lib.md

Set up

```
module add perl/5.28.0
module add repeatmasker/4.0.7
module add genometools/1.5.9
module add muscle/3.8.31
module add blast+/2.6.0
module add repeatmodeler/1.0.11
module add hmmer/3.2.1
```

```
DIR_CRL=/srv/scratch/z5188231/KStuart.Starling-Aug18/Sv3_Genome/Sv3.4_GenomeAnnotation/data_2020/adv_repeats/programs/CRL_Scripts1.0
DIR_PE=/srv/scratch/z5188231/KStuart.Starling-Aug18/Sv3_Genome/Sv3.4_GenomeAnnotation/data_2020/adv_repeats/programs/ProtExcluder-master

BASE_PATH=/srv/scratch/z5188231/KStuart.Starling-Aug18/Sv3_Genome/Sv3.4_GenomeAnnotation/data_2020/
AR_PATH=${BASE_PATH}/adv_repeats
GENOME=/srv/scratch/z5188231/KStuart.Starling-Aug18/Sv3_Genome/Sv3.4_GenomeAnnotation/data_2020/genome_assembly/Sturnus_vulgaris_2.3.1.simp.fasta
INPUT=/srv/scratch/z5188231/KStuart.Starling-Aug18/Sv3_Genome/Sv3.4_GenomeAnnotation/data_2020/genome_assembly/Sturnus_vulgaris_2.3.1.simp.fasta
PREFIX=Starling
CPU=4

EUK_tRNA=/srv/scratch/z5188231/KStuart.Starling-Aug18/Sv3_Genome/Sv3.4_GenomeAnnotation/data_2020/adv_repeats_lib/eukaryotic-tRNAs.fa
TpasesDNA=/srv/scratch/z5188231/KStuart.Starling-Aug18/Sv3_Genome/Sv3.4_GenomeAnnotation/data_2020/adv_repeats_lib/Tpases020812DNA
TpasesPROT=/srv/scratch/z5188231/KStuart.Starling-Aug18/Sv3_Genome/Sv3.4_GenomeAnnotation/data_2020/adv_repeats_lib/Tpases020812
SPROT=/srv/scratch/z5188231/KStuart.Starling-Aug18/Sv3_Genome/Sv3.4_GenomeAnnotation/data_2020/adv_repeats_lib/uniprot_sprot_clean.fasta
```

```
makeblastdb -in ${SPROT} -dbtype prot
makeblastdb -in ${EUK_tRNA} -dbtype nucl
makeblastdb -in ${TpasesDNA} -dbtype prot
makeblastdb -in ${TpasesPROT} -dbtype prot
```

Renaming the genome fasta so contigs have simple names:

```
perl ~/simplifyFastaHeaders.pl ${GENOME} ${PREFIX} ${GENOME%.fasta}.simp.fasta ${GENOME%.fasta}.map
INPUT=${GENOME%.fasta}.simp.fasta
```

Symbolic linking to MITE library produced in 2020-04-06.Mites:

```
ln -s /srv/scratch/z5188231/KStuart.Starling-Aug18/Sv3_Genome/Sv3.4_GenomeAnnotation/programs/MITE-Tracker/results/Sturnus_vulgaris_2.3.1.simp MITE_Tracker
ln -s MITE_Tracker/all.fasta MITE.lib
```

PART 1: LTRs (85%)

Find candidate elements

```
cd ${AR_PATH}
mkdir -p LTR
cd LTR

gt suffixerator -db ${INPUT} -indexname ${PREFIX} -tis -suf -lcp -des -ssp -dna
gt ltrharvest -index ${PREFIX} -out ${PREFIX}.out85 -outinner ${PREFIX}.outinner85 -gff3 ${PREFIX}.gff85 -minlenltr 100 -maxlenltr 6000 -mindistltr 1500 -maxdistltr 25000 -mintsd 5 -maxtsd 5 -vic 10 > ${PREFIX}.result85
```

Find elements with PPT (poly purine tract) or PBS (primer binding site)

```
gt gff3 -sort ${PREFIX}.gff85 > ${PREFIX}.gff85.sort
gt ltrdigest -trnas ${EUK_tRNA} ${PREFIX}.gff85.sort ${PREFIX} > ${PREFIX}.gff85.dgt
perl ${DIR_CRL}/CRL_Step1.pl --gff ${PREFIX}.gff85.dgt
```

Additional filtering of the candidate elements

```
perl ${DIR_CRL}/CRL_Step2.pl --step1 CRL_Step1_Passed_Elements.txt --repeatfile ${PREFIX}.out85 --resultfile ${PREFIX}.result85 --sequencefile ${INPUT} --removed_repeats CRL_Step2_Passed_Elements.fasta
mkdir fasta_files
mv Repeat_*.fasta fasta_files/
mv CRL_Step2_Passed_Elements.fasta fasta_files/
cd fasta_files/
perl ${DIR_CRL}/CRL_Step3.pl --directory ./ --step2 CRL_Step2_Passed_Elements.fasta --pidentity 60 --seq_c 25
mv CRL_Step3_Passed_Elements.fasta ../
cd ..
```

Identify elements with nested insertions

```
perl ${DIR_CRL}/ltr_library.pl --resultfile ${PREFIX}.result85 --step3 CRL_Step3_Passed_Elements.fasta --sequencefile ${INPUT}
cat lLTR_Only.lib ${AR_PATH}/MITE/MITE.lib > repeats_to_mask_LTR85.fasta
```

Search the repeats (so far) with RepeatMasker in Katana:

```
#!/bin/bash

#PBS -N 2020-04-11.RepeatMasker.pbs
#PBS -l nodes=1:ppn=16
#PBS -l mem=124gb
#PBS -l walltime=12:00:00
#PBS -j oe
#PBS -M katarina.stuart@student.unsw.edu.au
#PBS -m ae

module purge
module load perl/5.28.0
module load repeatmasker/4.0.7

PREFIX=Starling
BASE_PATH=/srv/scratch/z5188231/KStuart.Starling-Aug18/Sv3_Genome/Sv3.4_GenomeAnnotation/data_2020/
```

```
AR_PATH=${BASE_PATH}/adv_repeats
library=${AR_PATH}/LTR/repeats_to_mask_LTR85.fasta
DIR_RM1=/srv/scratch/z5188231/KStuart.Starling-
Aug18/Sv3_Genome/Sv3.4_GenomeAnnotation/data_2020/adv_repeats/programs/repeatmasker/4.0.7/

cd ${AR_PATH}/LTR

${DIR_RM1}/RepeatMasker -pa 16 -lib ${library} -nolow -dir . ${AR_PATH}/LTR/${PREFIX}.outinner85
```

Only ran for 2 mins then finished

```
perl ${DIR_CRL}/cleanRM.pl ${PREFIX}.outinner85.out ${PREFIX}.outinner85.masked > ${PREFIX}.outinner85.unmasked

perl ${DIR_CRL}/rmshortinner.pl ${PREFIX}.outinner85.unmasked 50 > ${PREFIX}.outinner85.clean

blastx -query ${PREFIX}.outinner85.clean -db ${TpasesDNA} -evalue 1e-10 -num_threads ${CPU} -num_descriptions 10 -out
${PREFIX}.outinner85.clean_blastx.out.txt

perl ${DIR_CRL}/outinner_blastx_parse.pl --blastx ${PREFIX}.outinner85.clean_blastx.out.txt --outinner ${PREFIX}.outinner85
```

Building examplars

```
perl ${DIR_CRL}/CRL_Step4.pl --step3 CRL_Step3_Passed_Elements.fasta --resultfile ${PREFIX}.result85 --innerfile
passed_outinner_sequence.fasta --sequencefile ${INPUT}
```

```
makeblastdb -in lLTRs_Seq_For_BLAST.fasta -dbtype nucl
```

```
blastn -query lLTRs_Seq_For_BLAST.fasta -db lLTRs_Seq_For_BLAST.fasta -evalue 1e-10 -num_descriptions 1000 -out
lLTRs_Seq_For_BLAST.fasta.out -num_threads ${CPU}
```

```
makeblastdb -in Inner_Seq_For_BLAST.fasta -dbtype nucl
```

```
blastn -query Inner_Seq_For_BLAST.fasta -db Inner_Seq_For_BLAST.fasta -evalue 1e-10 -num_descriptions 1000 -out
Inner_Seq_For_BLAST.fasta.out -num_threads ${CPU}
```

```
perl ${DIR_CRL}/CRL_Step5.pl --LTR_blast lLTRs_Seq_For_BLAST.fasta.out --inner_blast Inner_Seq_For_BLAST.fasta.out --step3
CRL_Step3_Passed_Elements.fasta --final LTR85.lib --pcoverage 90 --pidentity 80
```

Repetitive elements with RepeatModeler

Merge MITE and LTR libraries:

```
cd ${AR_PATH}
mkdir ADV_REP
cd ADV_REP
cat ../LTR/LTR85.lib ../MITE/MITE.lib > allMITE_LTR.lib
```

Mask the genome:

```
DIR_RM1=/srv/scratch/z5188231/KStuart.Starling-
Aug18/Sv3_Genome/Sv3.4_GenomeAnnotation/data_2020/adv_repeats/programs/repeatmasker/4.0.7/
BASE_PATH=/srv/scratch/z5188231/KStuart.Starling-Aug18/Sv3_Genome/Sv3.4_GenomeAnnotation/data_2020/
AR_PATH=${BASE_PATH}/adv_repeats
PREFIX=Starling
library=${AR_PATH}/ADV_REP/allMITE_LTR.lib
INPUT=/srv/scratch/z5188231/KStuart.Starling-
```

```
Aug18/Sv3_Genome/Sv3.4_GenomeAnnotation/data_2020/genome_assembly/Sturnus_vulgaris_2.3.1.simp.fasta

cd ${AR_PATH}/LTR

${DIR_RM1}/RepeatMasker -pa 16 -lib ${library} -dir . ${INPUT}
```

This removes the masked elements (no need to predict them again) --up to here

```
cd ${AR_PATH}/LTR
perl ${DIR_CRL}/rmaskedpart.pl ${INPUT##*/}.masked 50 > um_${INPUT##*/}
```

Now run RepeatModeler on Katana: the below took about 16 hrs.

```
#!/bin/bash

#PBS -N 2020-05-14.RepeatModeler.pbs
#PBS -l nodes=1:ppn=16
#PBS -l mem=124gb
#PBS -l walltime=24:00:00
#PBS -j oe
#PBS -M katarina.stuart@student.unsw.edu.au
#PBS -m ae

module purge
module load perl/5.28.0
module load recon/1.08
module load repeatscout/1.0.5
module load trf/4.09
module load rmblast/2.6.0
module load repeatmasker/4.0.7
module load repeatmodeler/1.0.11

PREFIX=Starling
BASE_PATH=/srv/scratch/z5188231/KStuart.Starling-Aug18/Sv3_Genome/Sv3.4_GenomeAnnotation/data_2020/
AR_PATH=${BASE_PATH}/adv_repeats

GENOME=/srv/scratch/z5188231/KStuart.Starling-
Aug18/Sv3_Genome/Sv3.4_GenomeAnnotation/data_2020/genome_assembly/Sturnus_vulgaris_2.3.1.fasta
INPUT=/srv/scratch/z5188231/KStuart.Starling-
Aug18/Sv3_Genome/Sv3.4_GenomeAnnotation/data_2020/genome_assembly/Sturnus_vulgaris_2.3.1.simp.fasta

cd ${AR_PATH}/LTR

BuildDatabase -name um_${INPUT##*/}db -engine ncbi um_${INPUT##*/}

nohup RepeatModeler -pa 16 -database um_${INPUT##*/}db >& um_${PREFIX}.out
```

RepeatModeler is able to identify some repeats but not other. Let's separate them and keep processing the unknowns:

```
cd /srv/scratch/z5188231/KStuart.Starling-
Aug18/Sv3_Genome/Sv3.4_GenomeAnnotation/data_2020/adv_repeats/LTR/RM_32721.TueMay191046472020
perl ${DIR_CRL}/repeatmodeler_parse.pl --fastafile consensi.fa.classified --unknowns repeatmodeler_unknowns.fasta --identities
repeatmodeler_identities.fasta
```

repeatmodeler_unknowns.fasta **are searched against the transposase database and the matching sequences are classified as such:**

```
blastx -query repeatmodeler_unknowns.fasta -db ${TpasesPROT} -evalue 1e-10 -num_descriptions 10 -out modelerunknown_blast_results.txt -num_threads 16

perl ${DIR_CRL}/transposon_blast_parse.pl --blastx modelerunknown_blast_results.txt --modelerunknown repeatmodeler_unknowns.fasta
```

The completely unknown elements are renamed and all the identified ones (from RepeatModeler and Blast) merged:

```
mv unknown_elements.txt ModelerUnknown.lib
cat identified_elements.txt repeatmodeler_identities.fasta > ModelerID.lib
```

```
cd ${AR_PATH}/LTR
mkdir final_libs
cp RM_32721.TueMay191046472020/ModelerID.lib final_libs/
cp ${AR_PATH}/MITE/MITE.lib final_libs/
cp LTR85.lib final_libs/
cp RM_32721.TueMay191046472020/ModelerUnknown.lib final_libs/
cd final_libs
sed 's/(//g' LTR85.lib | sed 's/)//g' > allLTR_rename.lib
```

Excluding gene fragments

```
module load hmmer/3.3
module load protexcluder/20190924

for lib in ModelerID.lib allLTR_rename.lib MITE.lib ModelerUnknown.lib; do
    blastx -query ${lib} -db ${SPROT} -evalue 1e-10 -num_descriptions 10 -num_threads ${CPU} -out ${lib}_blast_results.txt
    perl ProtExcluder.pl ${lib}_blast_results.txt ${lib}
    echo -e "${lib}\tbefore\t$(grep -c ">" ${lib})\tafter\t$(grep -c ">" ${lib}noProtFinal)"
done
```

The final (wanted) output will be the ${lib}noProtFinal files.

All filtered known repeats are merged:

```
cat MITE.libnoProtFinal allLTR_rename.libnoProtFinal ModelerID.libnoProtFinal > KnownRepeats.lib
```

And finally, we create the final repeat library:

```
cat KnownRepeats.lib ModelerUnknown.libnoProtFinal > allRepeats.lib
```