Starling-May18
Projects/Katarina
Stuart/KStuart.Starling-Aug18/At1_Genome/Annotation/2023-01-31.SpeciesSpecificRepeatLib

PDF Version generated by

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

on

Aug 23, 2023 @09:36 AM NZST

Table of Contents

Katarina Stuart (z5188231@ad.unsw.edu.au) - Feb 17, 2023, 10:54 PM GMT+13

SpeciesSpecificRepeatLib

genome_annotation_with_Maker2/Advanced_repeat_lib.md at master · xvazquezc/genome_annotation_with_Maker2 (github.com)

Filter transposases

Completed previously, code here just for reference.

# Running blast+ and BUSCO config

```
module add perl/5.20.1
module add maker/2.31.9
module add blast+/2.2.31
module add snap/2013-11-29
module add repeatmasker/4.0.7
module add exonerate/2.2.0
module add python/3.5.2
module add hmmer/3.1b2
module add augustus/3.2.2
module add emboss/6.5.7
module add busco/3.0.2b
module add R/3.3.2-bioconductor-3.5
module add samtools/1.6
```

# Transposase-protein-database
# Searches the SwissProt transposases.
# Gets the list of SwissProt proteins with matches.
# Generate a list of SwissProt proteins to keep.
# Generate a SwissProt-filtered fasta file.

```
TpasesPROT=Tpases020812
UniprotSprot=uniprot_sprot.fasta
makeblastdb -in $TpasesPROT -input_type fasta -dbtype prot -out TpasesPROT
blastp -query $UniprotSprot -db TpasesPROT -evalue 1e-10 -max_hsps 1 -max_target_seqs 1 -num_threads 1 -outfmt 6 -out sprot_tpasesprot.tab
cut -f 1 sprot_tpasesprot.tab > sprot_tpaseprot.txt
grep ">" ../TPASES.2018-08-21/uniprot_sprot.fasta | grep -v -f sprot_tpaseprot.txt | sed 's/^>//g' | sed 's/[ ].*//g' > sprot_notpasesprot.txt
xargs samtools faidx $UniprotSprot < sprot_notpasesprot.txt > uniprot_sprot_notpasesprot.fasta
```

# Transposase-DNA-database
# Now time to do the same with the Transposase DNA database:

```
TpasesDNA=Tpases020812DNA
makeblastdb -in $TpasesDNA -input_type fasta -dbtype prot -out TpasesDNA
blastp -query uniprot_sprot_notpasesprot.fasta -db TpasesDNA -evalue 1e-10 -max_hsps 1 -max_target_seqs 1 -num_threads 4 -outfmt 6 -out sprot_tpasesdna.tab
cut -f 1 sprot_tpasesdna.tab > sprot_tpasedna.txt
grep ">" uniprot_sprot_notpasesprot.fasta | grep -v -f sprot_tpasedna.txt | sed 's/^>//g' | sed 's/[ ].*//g' > sprot_clean.txt
xargs samtools faidx uniprot_sprot_notpasesprot.fasta < sprot_clean.txt > uniprot_sprot_clean.fasta
```

Filter transposases from SwissProt
You can skip this if you already have a curated SwissProt database free of transposases.

Have already done so in 2018-09-26.Species specific repeat library

Available on Nesi at:

```
ls -lh /nesi/nobackup/uoa02613/kstuart_projects/At1_MynaGenome/annotation/species_repeats/transposases
```

```
uniprot_sprot_clean.fasta
uniprot_sprot_notpasesprot.fasta
```

MITES-minature inverted-repeat transposable elements

https://github.com/INTABiotechMJ/MITE-Tracker

```
cd /nesi/nobackup/uoa02613/kstuart_projects/programs

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

VSearch:

```
cd /nesi/nobackup/uoa02613/kstuart_projects/programs/MITE-Tracker

wget https://github.com/torognes/vsearch/archive/v2.7.1.tar.gz
tar xzf v2.7.1.tar.gz
cd vsearch-2.7.1
sh autogen.sh
./configure
make
```

Running MITE-tracker from within program directory

```
#!/bin/bash -e

#SBATCH --job-name=2023_02_01.mites.sl
#SBATCH --account=uoa02613
#SBATCH --time=00-100:00:00
#SBATCH --mem=5GB
#SBATCH --output=%x_%j.errout
#SBATCH --mail-user=katarina.stuart@auckland.ac.nz
#SBATCH --mail-type=ALL
#SBATCH --nodes=1
#SBATCH --ntasks=1
#SBATCH --cpus-per-task=16
#SBATCH --profile task

module load BLAST/2.13.0-GCC-11.3.0
module load Python/3.7.3-gimkl-2018b

cd /nesi/nobackup/uoa02613/kstuart_projects/programs/MITE-Tracker

GENOMEDIR=/nesi/nobackup/uoa02613/kstuart_projects/At1_MynaGenome/analysis/curation/step4_scaffolding/ragtag_atris_synteny/renamed/
GENBASE=AcTris_vAus2.0
GENOME=${GENOMEDIR}/${GENBASE}.fasta
python3 -m MITETracker -g $GENOME -w 16 -j $GENBASE
```

```
#!/bin/bash

#PBS -N 2023-02-03.MITE-tracker.pbs
#PBS -l nodes=1:ppn=16
#PBS -l mem=24gb
#PBS -l walltime=48:00:00
#PBS -j oe
#PBS -M katarina.stuart@unsw.edu.au
#PBS -m ae
```

```
module load blast+/2.6.0
module load python/3.6.5

cd /srv/scratch/z5188231/KStuart.Starling-Aug18/Sv3_Genome/Sv3.4_GenomeAnnotation/programs/MITE-Tracker

GENOMEDIR=/srv/scratch/z5188231/KStuart.Starling-Aug18/At1_MynaGenome/data/genome
GENBASE=AcTris_vAus2.0
GENOME=${GENOMEDIR}/${GENBASE}.fasta

python3 -m MITETracker -g $GENOME -w 16 -j $GENBASE
```

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

module load perl/5.36.0
module load repeatmasker/4.1.4
module load repeatscout/1.0.5
module load trf/4.09.1
module load repeatmasker/4.1.4
cd ${AR_PATH}/ltr
```

```
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
AR_PATH=/srv/scratch/z5188231/KStuart.Starling-Aug18/At1_MynaGenome/annotation/species_repeats
GENOME=/srv/scratch/z5188231/KStuart.Starling-Aug18/At1_MynaGenome/data/genome/AcTris_vAus2.0.fasta
INPUT=${GENOME%.fasta}.fasta
PREFIX=AcTris
CPU=8

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 above:

```
ln -s /srv/scratch/z5188231/KStuart.Starling-Aug18/Sv3_Genome/Sv3.4_GenomeAnnotation/programs/MITE-Tracker/results/AcTris_vAus2.0 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:

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

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/

${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.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/
library=${AR_PATH}/ADV_REP/allMITE_LTR.lib

cd ${AR_PATH}/ltr

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

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

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

Now run RepeatModeler on Katana: the below took about 3 days :(NESI

```
#!/bin/bash -e

#SBATCH --job-name=2023_02_14.annotation_repeatmodeler.sl
#SBATCH --account=uoa02613
#SBATCH --time=00-24:00:00
#SBATCH --mem=30GB
#SBATCH --output=%x_%j.errout
#SBATCH --mail-user=katarina.stuart@auckland.ac.nz
#SBATCH --mail-type=ALL
#SBATCH --nodes=1
#SBATCH --ntasks=1
#SBATCH --cpus-per-task=16
#SBATCH --profile task
```

```
module purge
module load RepeatModeler/2.0.2a-gimkl-2020a

AR_PATH=/nesi/nobackup/uoa02613/kstuart_projects/At1_MynaGenome/annotation/species_repeats
GENOME=/nesi/nobackup/uoa02613/kstuart_projects/At1_MynaGenome/analysis/curation/step4_scaffolding/ragtag_atris_synteny/renamed/AcTris_vAus2.0.fasta
INPUT=${GENOME%.fasta}.fasta
PREFIX=AcTris

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: NESI --up to here

```
cd /nesi/nobackup/uoa02613/kstuart_projects/At1_MynaGenome/annotation/species_repeats/ltr/RM_66979.TueFeb141757362023
DIR_CRL=/nesi/nobackup/uoa02613/kstuart_projects/At1_MynaGenome/annotation/species_repeats/programs
module load Perl/5.24.1-gimkl-2017a-mt #needs this version, or at least not 5.28 +
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: KATANA**

```
module load blast-legacy/2.2.26
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 #non identified
```

```
cd ${AR_PATH}/ltr
mkdir final_libs
cp RM_66979.TueFeb141757362023/consensi.fa.classified final_libs/ #from NESI
cp ${AR_PATH}/MITE/MITE.lib final_libs/ #from Katana
cp LTR85.lib final_libs/ #from Katana
cd final_libs
cat all.MITE.fasta consensi.fa.classified LTR85.lib > allLTR_rename.lib
```

Excluding gene fragments

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

for lib in allLTR_rename.lib; do
blastx -query allLTR_rename.lib -db ${SPROT} -evalue 1e-10 -num_descriptions 10 -num_threads ${CPU} -out allLTR_rename.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


blastx -query allLTR_rename.lib -db ${SPROT} -evalue 1e-10 -num_descriptions 10 -num_threads ${CPU} -out allLTR_rename.lib _blast_results.txt
    perl ProtExcluder.pl ../allLTR_rename.lib_blast_results.txt ../allLTR_rename.lib
```

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
```