

# Annotation of TNAU genomes

---

Repeat masking and annotation of the assemblies generated from TNAU S32, S16, S6 and SY-2 isolates.

## TOC

- [Introduction](#)
- [Repeat Masking](#)
  - [Rename Contigs](#)
  - [Repeat Modeler](#)
  - [RepeatModeler Results Summary](#)
  - [Repeat Masker](#)
- [Annotations](#)
  - [MAKER](#)
    - [EndNotes](#)

## Introduction

---

Collaborators at TNAU aimed to assess the diversity of *Fusarium oxysporum* f. sp. *cubense* (*Focub*) in Tamil Nadu. They conducted a survey in banana-growing areas to identify wilt symptoms in various banana cultivars. Samples with wilt symptoms, including *Focub* TR4-susceptible 'Grande Nain' bananas, were collected, and the pathogen was isolated and identified as TR4.

Three highly virulent isolates (S6, S16, S32) were identified and sequenced (Illumina), though the DNA extraction protocol used remains unspecified due to communication challenges. Raw reads from these isolates were analysed, and the genomes were assembled using SPAdes. For isolates with high contamination levels, only reads mapping to reference genomes were used for *de novo* assembly.

This document follows the repeat masking and annotation of these isolates. Annotations will be generated for these assemblies, however, as assemblies are poor quality and no transcriptome data are available, no further analysis will be conducted on the annotations outputs. These will just be used for genome summary statistics.

## Repeat Masking

---

Followed [this tutorial](#) for repeat modeling and masking. I started with the S32 isolate, but the same approach was taken for all isolates: S32, S16, S6 and SY-2. First, repeats were identified in the assemblies *de novo* using RepeatModeler (version: 2.0.4), followed by repeat masking with RepeatMasker (version: 4.1.5). The RepeatModeler and RepeatMasker install, as well as dependencies can be found here on my Local Machine.

```
/Users/u1983390/Apps/repeat-annotation
```

Note, had install issues with Vettel and it is being decommissioned.

## Rename Contigs

I renamed the contigs to remove the long names generated by SPAdes, where `--nr-width` equals the total number of positions in `nr`. For example, <999 contigs `--nr-width 3`; <9999 contigs `--nr-width 4` etc.

```
#Check contig number
grep -c ">" S32_V4-ContaminantFiltered.fasta
2443

# Rename contigs
seqkit replace -p '.*' -r 'Contig_{nr}_TNAU_Fusarium_Isolate_S32' --nr-
width 4 S32_V4-ContaminantFiltered.fasta -o S32_V5-FS66-
Contiglabelled.fasta
```

## Repeat Modeler

All repeat modeling was run in the following directory:

`[IsolateCode]/RepeatMasking/RepeatModeling`

First, I built a RepeatModeler database.

```
BuildDatabase -name S32_Fusarium_SpNov -engine ncbi S32_V5-FS66-
Contiglabelled.fasta
```

This took only a few seconds and produced the following log:

```
Building database S32_Fusarium_SpNov:
Reading S32_V5-FS66-Contiglabelled.fasta...
Number of sequences (bp) added to database: 2443 ( 40920118 bp )
```

This was followed by RepeatModeler, which used the database I had just built.

```
RepeatModeler -threads 16 -engine ncbi -database S32_Fusarium_SpNov -
LTRStruct 2>&1 | tee 00_repeatmodeler.log
```

The tutorial does not include the `-LTRStruct` flag. However, I included it to increase the number of repetitive elements found, as these isolates are potentially novel species.

RepeatModeler has produced several output dirs and files. Each dir contains output from each round of RepeatModeler. The tutorial explains that the most relevant outputs are in the output FASTA and Stockholm files:

The FASTA library contains important repeat classification information in the header lines, which look something like this: >rnd-1\_family-174#LINE/L1. A unique round and family identifier are included from the internal RepeatModeler run and after the #, repeat classification information is provided.

This took approximately 3 hours for S32, and no clean LTR-RTs were kept by LTR\_retriever.

[9/11/2023] Again, no clean LTR-RTs were kept by LTR\_retriever for S16 - Wondering if the install has not worked for LTR\_retriever. However, I think I can continue using this approach and if install has failed, explain it.

```
# S32
grep -c ">" S32_Fusarium_SpNov-families.fa
39
# S16
grep -c ">" S16_Fusarium_SpNov-families.fa
57
```



Following the tutorial, I added the species code to the head of each element, and split the library into classified and unclassified elements.

```
#Add the prefix:
cat S32_Fusarium_SpNov-families.fa | seqkit fx2tab | awk '{ print
"FusSpNov-S32-"$0 }' | seqkit tab2fx > S32_Fusarium_SpNov-
families.prefix.fa

#Split the library: Classified
cat S32_Fusarium_SpNov-families.prefix.fa | seqkit fx2tab | grep -v
"Unknown" | seqkit tab2fx > S32_Fusarium_SpNov-
families.prefix.fa.classified

#Split the library: Unclassified
fx2tab | grep "Unknown" | seqkit tab2fx > S32_Fusarium_SpNov-
families.prefix.fa.unclassified
```

## RepeatModeler Results Summary

- For S32, this resulted in 5 classified families and 34 unclassified.
- For S16, this resulted in 8 classified families and 49 unclassified.
-  classified families and [X] unclassified.
-  classified families and [X] unclassified.

These unclassified families could be classified further with repeat classifier and an additional TE database, such as Repbase. However, we do not have a RepBase subscription, and curating a further database is time consuming and challenging, considering we are potentially working with a novel species and the assemblies are of poor quality, I do not think it is worth the additional work. If we receive better quality raw data, then we can focus on repeat modeling and masking.

## Repeat Masker

RepeatMasker will be run repeatedly, with each round using different repeat libraries to annotate the genomes. This is done by taking the masked output from one round of RepeatMasker, and feeding it into the subsequent round of repeat annotation and masking.

The rounds will progress in the following order:

1. Simple repeats
2. Curated repeats from a library[^1]
3. Known species-specific repeats from RepeatModeler
4. Unknown species-specific repeats from RepeatModeler

First, we make directories for the output of each run

```
mkdir -p logs 01_Simple_out 02_Fusarium_out 03_Classified_out  
04_Unclassified_out
```

Then we begin the first round of simple annotations, where `-noint` and `-xsmall` only annotate and soft masks simple repeats.

```
# round 1:  
RepeatMasker -pa 16 -a -e ncbi -dir 01_Simple_out -noint -xsmall S32_V5-  
FS66-Contiglabelled.fasta 2>&1 | tee logs/01_SimpleMask.log  
# I renamed the output file using "SimpleMask" to keep track, e.g.  
mv 01_Simple_out/S32_V5-FS66-Contiglabelled.fasta.masked  
01_Simple_out/S32_V5-FS66-Contiglabelled.SimpleMask.fasta
```

Next, I masked Fusarium repeats from the Dfam database. I used the `-nolow` to annotate/mask only complex, interspersed repeats (e.g., transposons) and used the `-species Fusarium` option to identify repeat consensus sequences from Fusarium in the Dfam (version 3.7) database. Theoretically, this would also include RepBase, but we do not have a subscription.

```
# round 2:  
RepeatMasker -pa 16 -a -e ncbi -dir 02_Fusarium_out -nolow -species  
fusarium 01_Simple_out/S32_V5-FS66-Contiglabelled.SimpleMask.fasta 2>&1 |  
tee logs/02_FusariumMask.log  
# I renamed the output file using "FusariumMask" to keep track, e.g.  
mv 02_Fusarium_out/S32_V5-FS66-Contiglabelled.SimpleMask.fasta.masked  
02_Fusarium_out/S32_V5-FS66-Contiglabelled.FusariumMask.fasta
```

This masked only 50bp of S32, I think probably due to the fragmented nature of the genome and because I was using only the Dfam database.

No repetitive sequences were detected in S16, either, so I had to use 01\_SimpleMask fasta as input for S16 repeat masking round 3.

The third round of masking makes use of the output from RepeatModeler, first masking the classified elements I identified earlier. Again, I used the `-nolow` to annotate/mask only complex, interspersed repeats (e.g., transposons) and the `-lib` option to define the library ()

```
# round 3:
RepeatMasker -pa 16 -a -e ncbi -dir 03_Classified_out -nolow -lib
RepeatModeling/S32_Fusarium_SpNov-families.prefix.fa.classified
02_Fusarium_out/S32_V5-FS66-Contiglabelled.FusariumMask.fasta 2>&1 | tee
logs/03_ClassifiedMask.log

# I renamed the output file using "ClassifiedMask" to keep track, e.g.
mv 03_Classified_out/S32_V5-FS66-Contiglabelled.FusariumMask.fasta.masked
03_Classified_out/S32_V5-FS66-Contiglabelled.ClassifiedMask.fasta
```

The same steps taken in round 3 were taken for round 4, but the unclassified library was used instead, and annotations were added to the output of round 3.

```
# round 4:
RepeatMasker -pa 16 -a -e ncbi -dir 04_Unclassified_out -nolow -lib
RepeatModeling/S32_Fusarium_SpNov-families.prefix.fa.unclassified
03_Classified_out/S32_V5-FS66-Contiglabelled.ClassifiedMask.fasta 2>&1 |
tee logs/04_UnclassifiedMask.log

# I renamed the output file using "ClassifiedMask" to keep track, e.g.
mv 04_Unclassified_out/S32_V5-FS66-
Contiglabelled.ClassifiedMask.fasta.masked 04_Unclassified_out/S32_V5-
FS66-Contiglabelled.UnclassifiedMask.fasta
```

Now all of the outputs are combined to generate the final masked assembly.

```
# Make the directory for the final, combined mask results to sit.
mkdir 05_FullMask_out

# combine full RepeatMasker result files - .cat.gz
cat 01_Simple_out/S32_V5-FS66-Contiglabelled.SimpleMask.fasta.cat.gz
02_Fusarium_out/S32_V5-FS66-Contiglabelled.FusariumMask.fasta.cat.gz
03_Classified_out/S32_V5-FS66-Contiglabelled.ClassifiedMask.fasta.cat.gz
04_Unclassified_out/S32_V5-FS66-
Contiglabelled.UnclassifiedMask.fasta.cat.gz > 05_FullMask_out/S32_V5-
FS66-Contiglabelled.FullMask.cat.gz

# combine RepeatMasker tabular files for all repeats - .out
cat 01_Simple_out/S32_V5-FS66-Contiglabelled.SimpleMask.fasta.out \
<(cat 02_Fusarium_out/S32_V5-FS66-Contiglabelled.FusariumMask.fasta.out |
tail -n +4) \
```

```

<(cat 03_Classified_out/S32_V5-FS66-
Contiglabelled.ClassifiedMask.fasta.out | tail -n +4) \
<(cat 04_Unclassified_out/S32_V5-FS66-
Contiglabelled.UnclassifiedMask.fasta.out | tail -n +4) \
> 05_FullMask_out/S32_V5-FS66-Contiglabelled.FullMask.out

# copy RepeatMasker tabular files for simple repeats - .out
cat 01_Simple_out/S32_V5-FS66-Contiglabelled.SimpleMask.fasta.out >
05_FullMask_out/S32_V5-FS66-Contiglabelled.SimpleMask.out

# combine RepeatMasker tabular files for complex, interspersed repeats -
.out
cat 02_Fusarium_out/S32_V5-FS66-Contiglabelled.FusariumMask.fasta.out \
<(cat 03_Classified_out/S32_V5-FS66-
Contiglabelled.ClassifiedMask.fasta.out | tail -n +4) \
<(cat 04_Unclassified_out/S32_V5-FS66-
Contiglabelled.UnclassifiedMask.fasta.out | tail -n +4) \
> 05_FullMask_out/S32_V5-FS66-Contiglabelled.ComplexMask.out

# combine RepeatMasker repeat alignments for all repeats - .align
cat 01_Simple_out/S32_V5-FS66-Contiglabelled.SimpleMask.fasta.align \
02_Fusarium_out/S32_V5-FS66-Contiglabelled.FusariumMask.fasta.align \
03_Classified_out/S32_V5-FS66-Contiglabelled.ClassifiedMask.fasta.align \
04_Unclassified_out/S32_V5-FS66-
Contiglabelled.UnclassifiedMask.fasta.align \
> 05_FullMask_out/S32_V5-FS66-Contiglabelled.FullMask.align

```

Now I have generated combined files for the `.out`, `.align`, and `.cat.gz` outputs, I will generate a new, combined `.tbl` file.

```

# summarize repeat compositions from combined analysis of all RepeatMasker
rounds
ProcessRepeats -a -species Fusarium 05_FullMask_out/S32_V5-FS66-
Contiglabelled.FullMask.cat.gz 2>&1 | tee logs/05_fullmask.log

```

Next, we can create GFF3 files for these repeat annotations using a script distributed with RepeatMasker; `rmOutToGFF3.pl`

```

# Note; this was performed in the 05_FullMask_out directory.
for i in *.out ; do rmOutToGFF3.pl ${i} > ${i}.gff ; done

```

## Annotations

For annotations, the MAKER pipeline (version XXX) was used. The Repeat Masking step was skipped, as we have already modeled and masked repeats in these assemblies.

We do not have transcriptome data for these isolates and, as we are not clear on their species, have used a *Fusarium* wide reference proteome for annotation.

---

## MAKER

### EndNotes

[^1]: I used the default Dfam library I had installed with RepeatMasker. Using Master RepeatMasker Database: /Users/u1983390/Apps/repeat-annotation/RepeatMasker/Libraries/RepeatMaskerLib.h5 | Dfam Version : 3.7 | Date : 2023-01-11 | Families : 19,768.