Supporting information

Introduction

Repeat-induced point mutations (RIP) is an adaptive genome defence mechanism unique to fungi that act to prevent further mobilisation and transcirption of transposable elements. This is achieved by targetting repeat sequences for hypermutation; repeat sequences above a length threshold (~400-500bp) that share >80% identity undergo irreversible C-to-T mutations. Thus, a biochemical hallmark of RIP is the accrual of AT-rich regions.

The analyses presented here investigate the composition of three *Epichloe typhina* strains and quantify the proportion AT-rich regions that comprise the genome.

The Data

The data used for these analyses were generated using Occultercut v(v3.6.3)

```
Ecl_gff <- read.table("~/analyses/StartingOut/raw_data/Ecl/Ecl_AT_rich.gff", stringsAsFactors = FALSE, s
knitr::kable(head(Ecl_gff))
```

V1	V2	V3	V4	V5	V6	V7	V8
Ecl_1605_22_1	occult	AT_rich_region	1	151456	.	+	. ID=0.AT_rich.0;
Ecl_1605_22_1	occult	AT_rich_region	157805	162910	.	+	. ID=0.AT_rich.1;
Ecl_1605_22_1	occult	AT_rich_region	167892	203591	.	+	. ID=0.AT_rich.2;
Ecl_1605_22_1	occult	AT_rich_region	238173	438436	.	+	. ID=0.AT_rich.3;
Ecl_1605_22_1	occult	AT_rich_region	449086	485940	.	+	. ID=0.AT_rich.4;
Ecl_1605_22_1	occult	AT_rich_region	490332	492954	.	+	. ID=0.AT_rich.5;

The key column to this analysis is `V4` and `V5` which can be used to calculate the total length of all AT-rich regions in the genome. The following function will obtain the total length and number of AT-rich regions.

```
gff_summary <- function(fname){
  gff <- read.table(fname, stringsAsFactors = FALSE, sep = "\t")
  total <- sum(gff$V5 - gff$V4 + 1)
  n <- nrow(gff)
  list(bp = total, n = n)
}
```

to determine the proportion, the total length of the reference genomes can be retrieved with this function:

```
library(ape)
reference_length <- function(fname){
  ref <- read.dna(fname, format = "fasta")
  chroms <- head(ref, 7)
  chrom_length <- lengths(chroms)
  sum(chrom_length)
}
```

Running the functions

```
Ecl_AT <- gff_summary("~/analyses/StartingOut/raw_data/Ecl/Ecl_AT_rich.gff")
Epo_AT <- gff_summary("~/analyses/StartingOut/raw_data/Epo/Epo_AT_rich.gff")
Ety_AT <- gff_summary("~/analyses/StartingOut/raw_data/Ety/Ety_AT_rich.gff")
```

```
Ecl_ref_len <- reference_length(
  "~/analyses/StartingOut/raw_data/Ecl/Ecl1605_22_Epichloe_clarkii_1605_22_45692596_v2.fna")
Epo_ref_len <- reference_length(
  "~/analyses/StartingOut/raw_data/Epo/Etp76_Epichloe_typhina_var_poae_NFe76_38327242_v1.fna")
Ety_ref_len <- reference_length(
  "~/analyses/StartingOut/raw_data/Ety/Ety1756_Epichloe_typhina_1756_33930528_v4.fna")
```

Finally, to determine the proportion of AT-rich regions vs. non-AT rich region, non-AT-rich regions were calculated as below:

```
Ecl_non_AT <- Ecl_ref_len - Ecl_AT$bp
Epo_non_AT <- Epo_ref_len - Epo_AT$bp
Ety_non_AT <- Ety_ref_len - Ety_AT$bp
```

Generating the dataframe for analyses

```
Ecl <- data.frame(Ecl_ref_len, Ecl_AT$n, Ecl_AT$bp, Ecl_non_AT)
Ecl$Species <- "Ecl"
Epo <- data.frame(Epo_ref_len, Epo_AT$n, Epo_AT$bp, Epo_non_AT)
Epo$Species <- "Epo"
Ety <- data.frame(Ety_ref_len, Ety_AT$n, Ety_AT$bp, Ety_non_AT)
Ety$Species <- "Ety"
names(Ecl)[1:4] <- c("reference_length", "AT_n", "AT_bp", "non_AT")
names(Epo)[1:4] <- c("reference_length", "AT_n", "AT_bp", "non_AT")
names(Ety)[1:4] <- c("reference_length", "AT_n", "AT_bp", "non_AT")
df <- rbind.data.frame(Ecl, Epo, Ety)[,c(5,1,2,3,4)]
knitr::kable(
df
)
```

Species	reference_length	AT_n	AT_bp	non_AT
Ecl	45619412	648	22156257	23463155
Epo	38251297	538	15464574	22786723
Ety	33832076	362	10667734	23164342

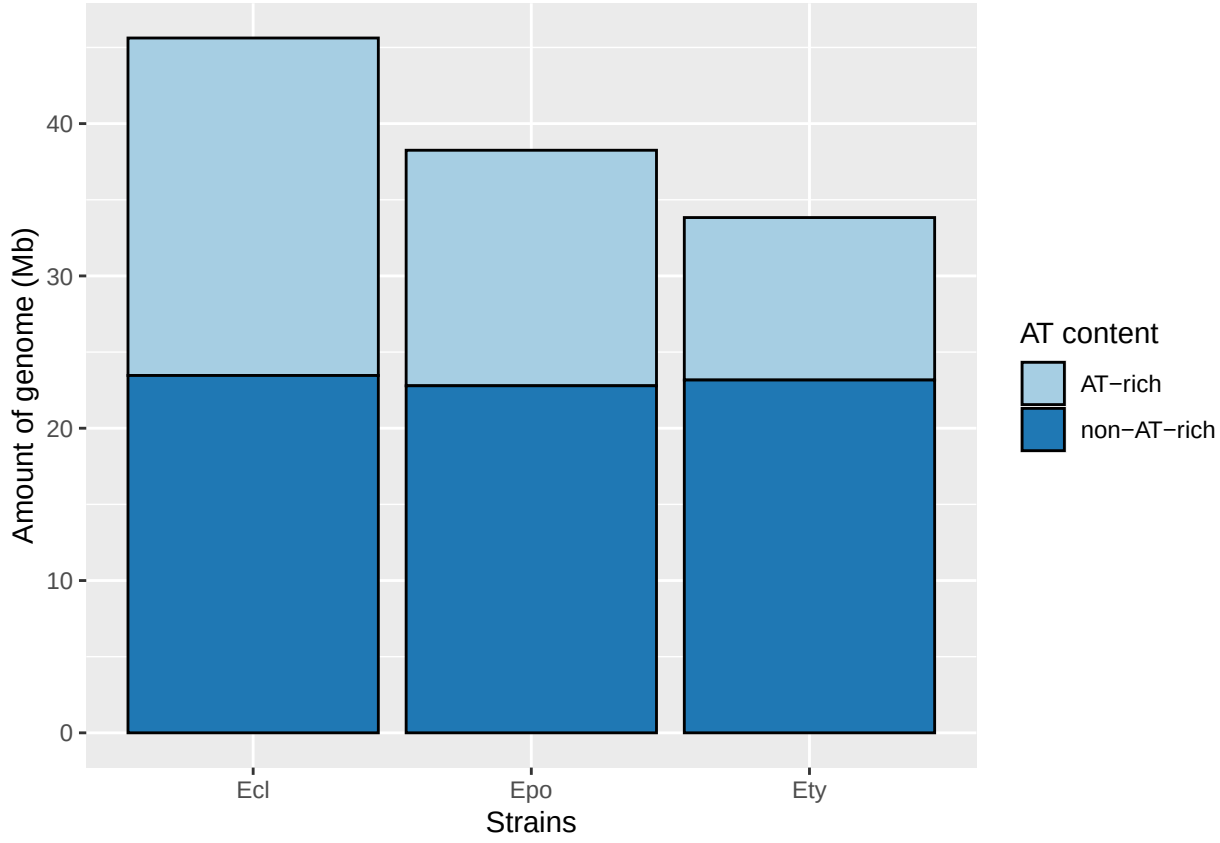
Data Visualisation

```
library(reshape)
library(tidyverse)
```

```
## -- Attaching packages ---------------------------------------------------------------------------------------
## v ggplot2 3.3.2     v purrr   0.3.4
## v tibble  3.0.3     v dplyr   1.0.0
## v tidyr   1.1.0     v stringr 1.4.0
## v readr   1.3.1     v forcats 0.5.0
```

```
## -- Conflicts ------------------------------------------------------------------------------------
## x tidyr::expand() masks reshape::expand()
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## x dplyr::rename() masks reshape::rename()
```

```
molten_df <- melt(data = select(df, Species, AT_bp, non_AT), id.vars = "Species")
library(ggplot2)
ggplot(molten_df, aes(x = Species, y = value, fill = variable)) + geom_col(color = "black") + scale_fil
```



Proportion of AT-richness for Ecl, Epo, and Ety, respectively

```
prop <- c(df[1,4]/df[1,2]*100, df[2,4]/df[2,2]*100, df[3,4]/df[3,2]*100)
prop
```

```
## [1] 48.56761 40.42889 31.53142
```

Scripts - Python

To calculate RIP index of AT-rich regions

```
from Bio import SeqIO
from collections import Counter
import argparse

parser = argparse.ArgumentParser(
    description = "makes two files with info on dinucs and a file with RIP indices"
    )
```

```
parser.add_argument("--seqs", dest = "seqs", required = True)
parser.add_argument("--outstem", dest = "outstem")

def count_dinucs(record):
    """count dinucleotides"""
    all_dinucs = []
    for i in range (len(record.seq)-1):
        all_dinucs.append(str(record.seq[i:i+2]))
    dinuc_counted = Counter(all_dinucs)
    return(dinuc_counted)

def calc_RIP_idx(dinuc_counted):
    try:
        RIP_idx_1 = round(dinuc_counted['TA']/dinuc_counted['AT'],4)
    except ZeroDivisionError:
        RIP_idx_1 = "NaN"
    try:
        RIP_idx_2 = round((dinuc_counted['CA']+dinuc_counted['TG'])/
        (dinuc_counted['AC']+dinuc_counted['GT']), 4)
    except ZeroDivisionError:
        RIP_idx_2 = "NaN"
    return([RIP_idx_1, RIP_idx_2])


if __name__ == "__main__":
    args = parser.parse_args()
    seqs = SeqIO.parse(args.seqs, "fasta")
    R_idx_handle = open(args.outstem + "_indices.tsv", "w")
    R_idx_handle.write("RIP_idx_1\tRIP_idx_2\n")

    with open(args.outstem + ".tsv", "w") as out:

        for rec in seqs:
            dn = count_dinucs(rec)

            idx_1, idx_2 = calc_RIP_idx(dn)

            R_idx_handle.write("{}\t{}\n".format(idx_1, idx_2))
            biglist = ['AA','AT','AG','AC','TA','TT','TG','TC','GA',
            'GT','GG','GC','CA','CT','CG','CC']
            denom = sum([n for (d, n) in dn.items() if "N" not in d])
            for d in biglist:
                percentage = dn[d]/denom * 100
                out.write("{}\t{}\t{:.4f}\n".format(d, dn[d], percentage))
```

Mean RIP of AT-rich regions

```
Ecl_idx <- read.table("../Ecl_AT_getfasta.out_RIP_indices.tsv", header = TRUE)
Ety_idx <- read.table("../Ety_AT_getfasta.out_RIP_indices.tsv", header = TRUE)
Epo_idx <- read.table("../Epo_AT_getfasta.out_RIP_indices.tsv", header = TRUE)

mean(Epo_idx$RIP_idx_1)
```

```
## [1] 1.682533
```

```
mean(Epo_idx$RIP_idx_2)
```

```
## [1] 0.2857494
```

```
mean(Ety_idx$RIP_idx_1)
```

```
## [1] 1.645638
```

```
mean(Ety_idx$RIP_idx_2)
```

```
## [1] 0.2597483
```

```
mean(Ecl_idx$RIP_idx_1)
```

```
## [1] 1.641003
```

```
mean(Ecl_idx$RIP_idx_2)
```

```
## [1] 0.3658724
```