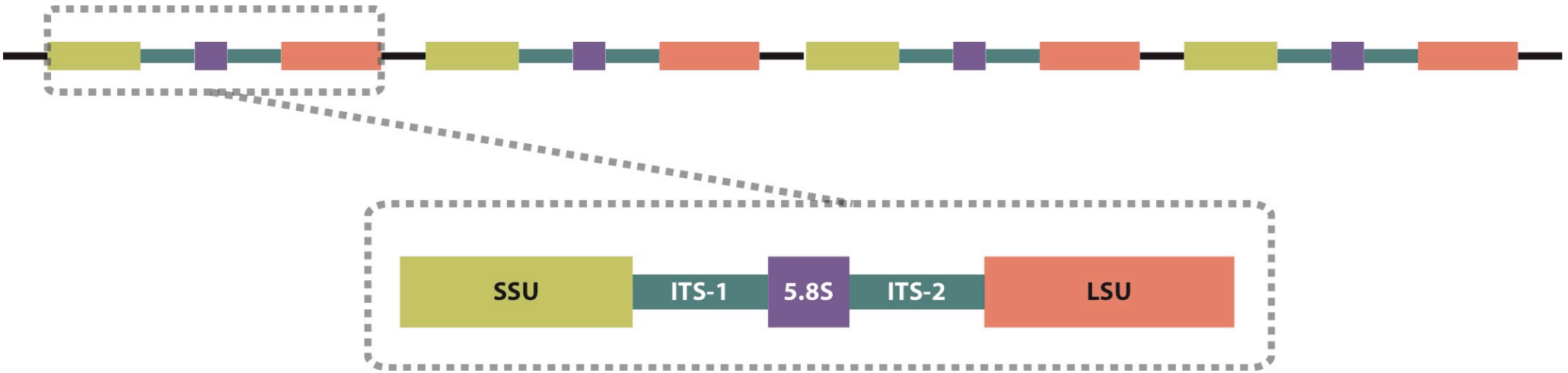
FUNGAL rDNA COPY NUMBER VARIATION

LOTUS LOFGREN

University of California Riverside
Dep. of Microbiology and Plant Pathology Biology

RIBOSOMES, rRNA AND rDNA

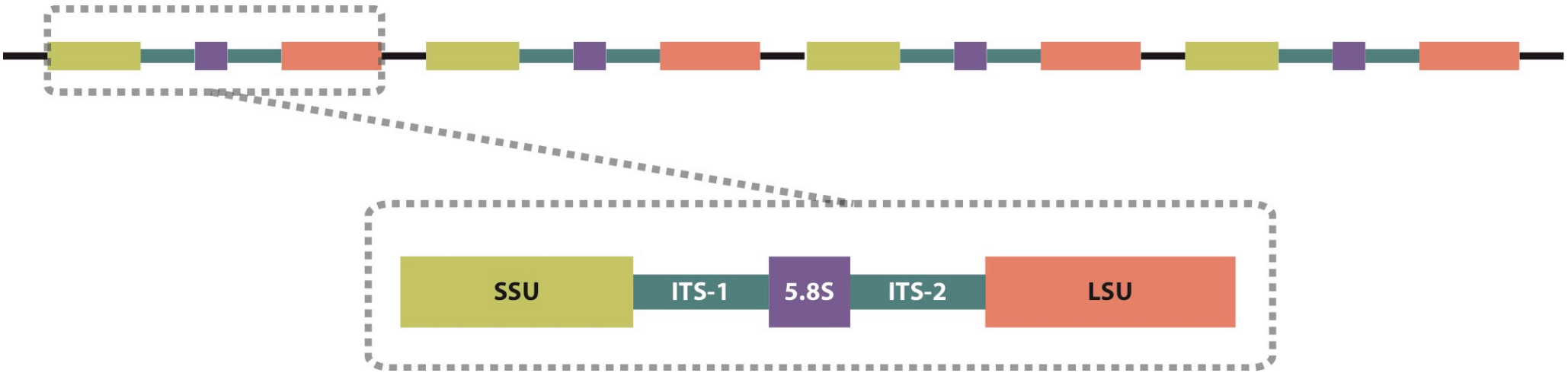
Ribosomes 1:1 proteins, rRNA



RIBOSOMES, rRNA AND rDNA

Ribosomes 1:1 proteins, rRNA

RNA = transcription, but no translation

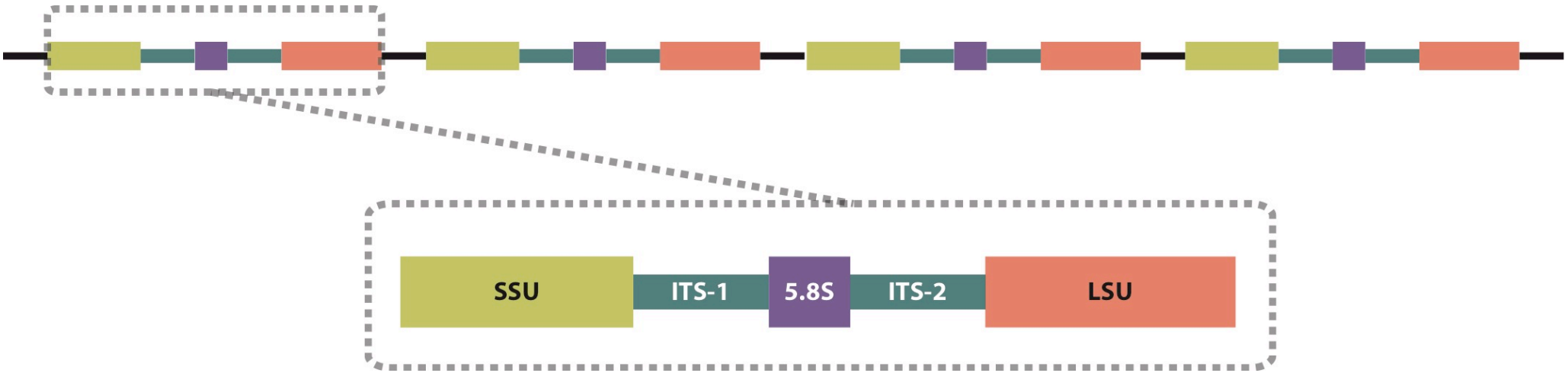


RIBOSOMES, rRNA AND rDNA

Ribosomes 1:1 proteins, rRNA

RNA = transcription, but no translation

Many copies of the gene encoding rRNA



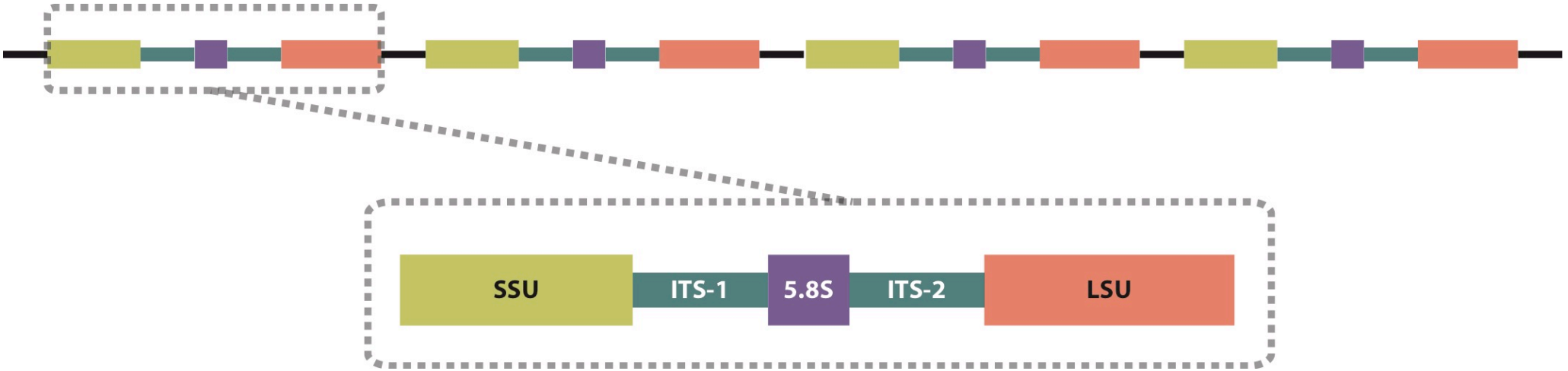
RIBOSOMES, rRNA AND rDNA

Ribosomes 1:1 proteins, rRNA

RNA = transcription, but no translation

Many copies of the gene encoding rRNA

In fungi, (usually) a single set of tandem repeats = rDNA cassettes



RIBOSOMES, rRNA AND rDNA

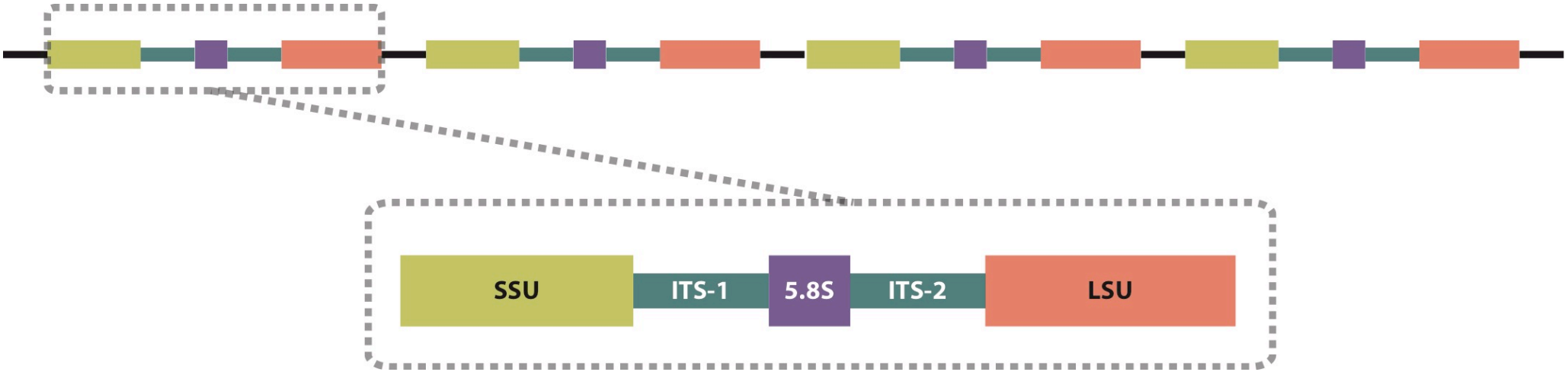
Ribosomes 1:1 proteins, rRNA

RNA = transcription, but no translation

Many copies of the gene encoding rRNA

In fungi, (usually) a single set of tandem repeats = rDNA cassettes

Number of cassettes varies



rDNA: FUNCTIONS AND CORRELATIONS

- More copies than necessary for protein production

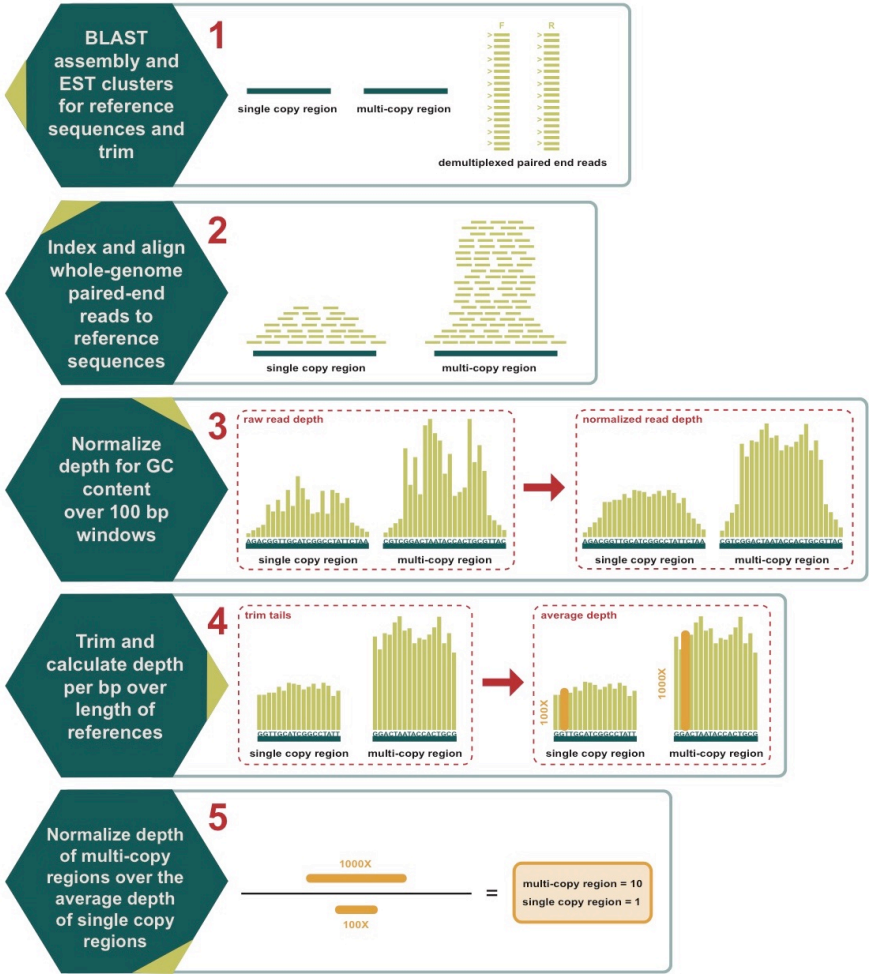
rDNA: FUNCTIONS AND CORRELATIONS

- More copies than necessary for protein production
- In bacteria, rDNA CN is related to lifestyle

rDNA: FUNCTIONS AND CORRELATIONS

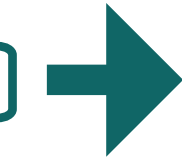
- More copies than necessary for protein production
- In bacteria, rDNA CN is related to lifestyle
- In plants and animals, rDNA CN is positively related to genome size

THE rDNA CN PIPELINE



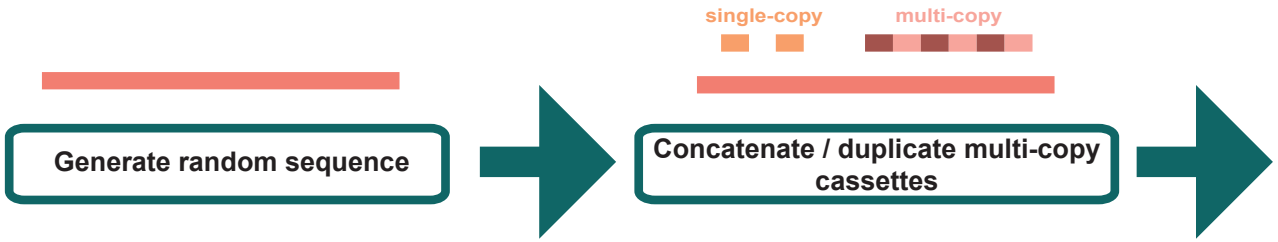
IN-SILICO CONFIRMATION

Generate random sequence



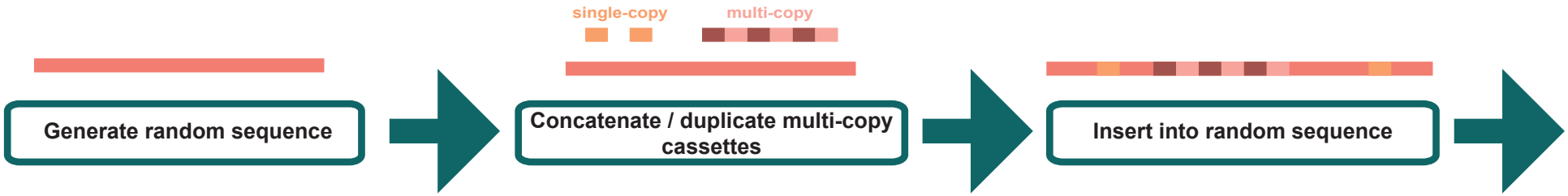
```
#set path
.libPaths(new='~/R/x86_64-pc-linux-gnu-library/3.2')
#load libraries
library(readr)

#initiate genome
MakeGenome <- function(genomesize = 1){
  # this function returns a simulated DNA sequence of genomesize bases
  bps <- sample(size = genomesize, x = c("A","C","G","T"), replace = TRUE, prob = c(0.4,0.1,0.1,0.4))
  genomeout <- paste(bps, collapse = "")
  return(genomeout)
}

Synth_genome <- MakeGenome(genomesize = 52000000)
```

IN-SILICO CONFIRMATION



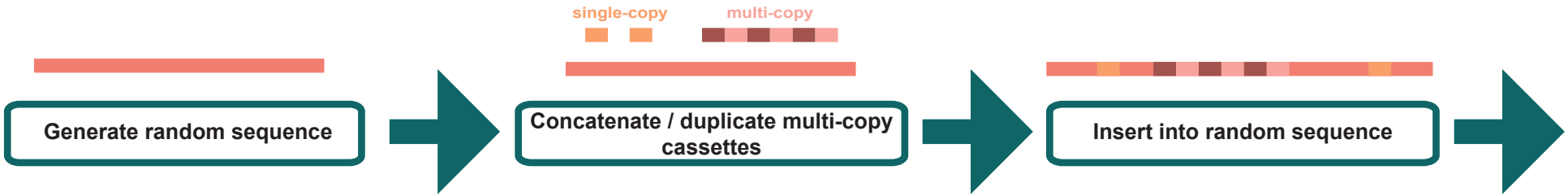

```
#make multi-copy regions, multi-copy and merge into cassette.
S_brevipes_multi_IR_60 <- rep(c(S_brevipes_ITS_IR, S_brevipes_LSU_IR), 60)
S_brevipes_multi_IR_60_reps <- paste(S_brevipes_multi_IR_60, collapse = "")

#get length to 5000000bp buffer of each gene or cat
len_ITS_LSU <- nchar(S_brevipes_multi_IR_60_reps) + 5000000
len_RPB2 <- nchar(S_brevipes_RPB2_IR) + 5000000
len_GAPDH <- nchar(S_brevipes_GAPDH_IR) + 5000000
len_RPB1 <- nchar(S_brevipes_RPB1_IR) + 5000000
len_RPB1 <- nchar(S_brevipes_RPB1_IR) + 5000000
len_ELF1 <- nchar(S_brevipes_ELF1_IR) + 5000000
len_GH63 <- nchar(S_brevipes_GH63_IR) + 5000000
len_MCM7 <- nchar(S_brevipes_MCM7_IR) + 5000000
len_G6PDH <- nchar(S_brevipes_G6PDH_IR) + 5000000
len_MLS <- nchar(S_brevipes_MLS_IR) + 5000000
len_LYS2 <- nchar(S_brevipes_LYS2_IR) + 5000000
```

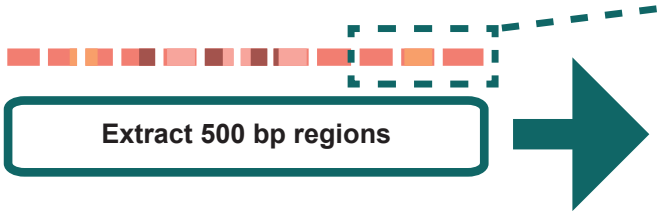
IN-SILICO CONFIRMATION




```
#insert multi-copy cassette into synthetic genome
Synth_genome_w_multi <- paste((substr(Synth_genome,0,1000)), S_brevipes_multi_IR_60_reps, (substr(Synth_genome, 1001,

#insert single-copy genes into synthetic genome
Synth_genome_w_multi_RPB2 <- paste((substr(Synth_genome_w_multi,0,(len_ITS_LSU)+1000)), S_brevipes_RPB2_IR, (substr(S
Synth_genome_w_multi_RPB2_GAPDH <- paste((substr(Synth_genome_w_multi_RPB2,0,(len_ITS_LSU + len_RPB2)+1000)), S_brevi
Synth_genome_w_multi_GAPDH_RPB1 <- paste((substr(Synth_genome_w_multi_RPB2_GAPDH,0,(len_ITS_LSU + len_RPB2 + len_GAPD
Synth_genome_w_multi_RPB1_ELF1 <- paste((substr(Synth_genome_w_multi_GAPDH_RPB1,0,(len_ITS_LSU + len_RPB2 + len_GAPDH
Synth_genome_w_multi_ELF1_GH63 <- paste((substr(Synth_genome_w_multi_RPB1_ELF1,0,(len_ITS_LSU + len_RPB2 + len_GAPDH
Synth_genome_w_multi_GH63_MCM7 <- paste((substr(Synth_genome_w_multi_ELF1_GH63,0,(len_ITS_LSU + len_RPB2 + len_GAPDH
Synth_genome_w_multi_MCM7_G6PDH <- paste((substr(Synth_genome_w_multi_GH63_MCM7,0,(len_ITS_LSU + len_RPB2 + len_GAPDH
Synth_genome_w_multi_G6PDH_MLS <- paste((substr(Synth_genome_w_multi_MCM7_G6PDH,0,(len_ITS_LSU + len_RPB2 + len_GAPDH
Synth_genome_w_multi_MLS_LYS2 <- paste((substr(Synth_genome_w_multi_G6PDH_MLS,0,(len_ITS_LSU + len_RPB2 + len_GAPDH +
Synth_genome_w_multi_LYS2_TOP2 <- paste((substr(Synth_genome_w_multi_MLS_LYS2,0,(len_ITS_LSU + len_RPB2 + len_GAPDH +
```

IN-SILICO CONFIRMATION



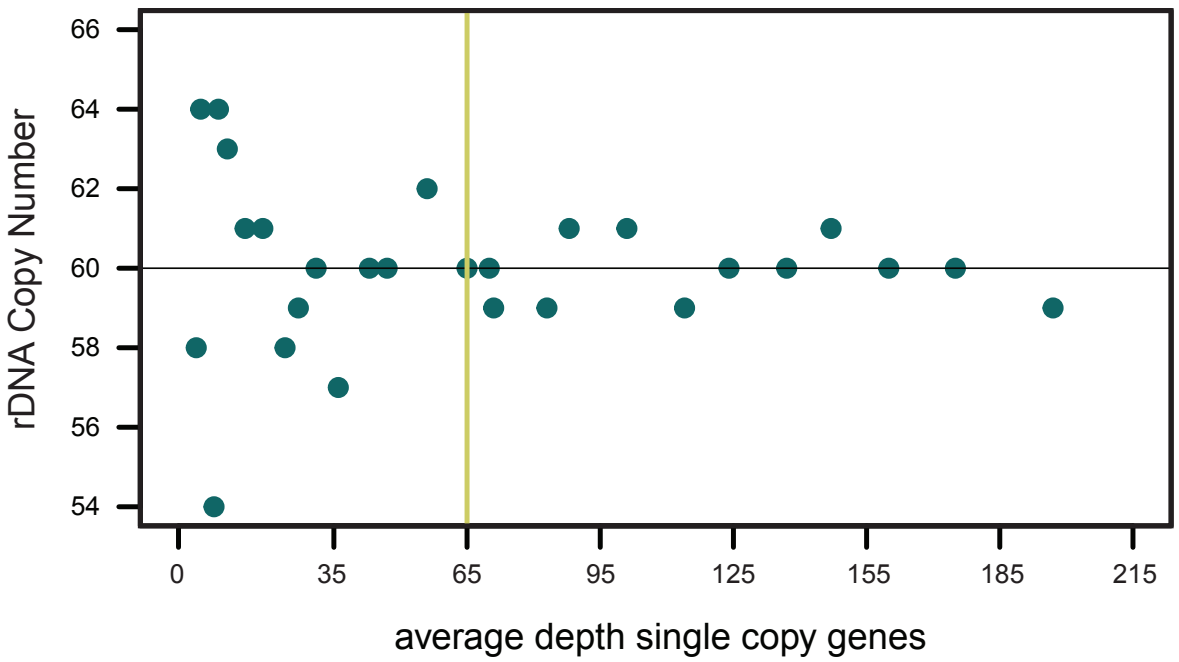



```
#function to take n 500 bp draws based on input n
Fragment = function(string) {
  nStart <- sample(1:(nchar(string) -500), 1)
  samp <- substr(string, nStart, nStart + 499)
}
```

IN-SILICO CONFIRMATION

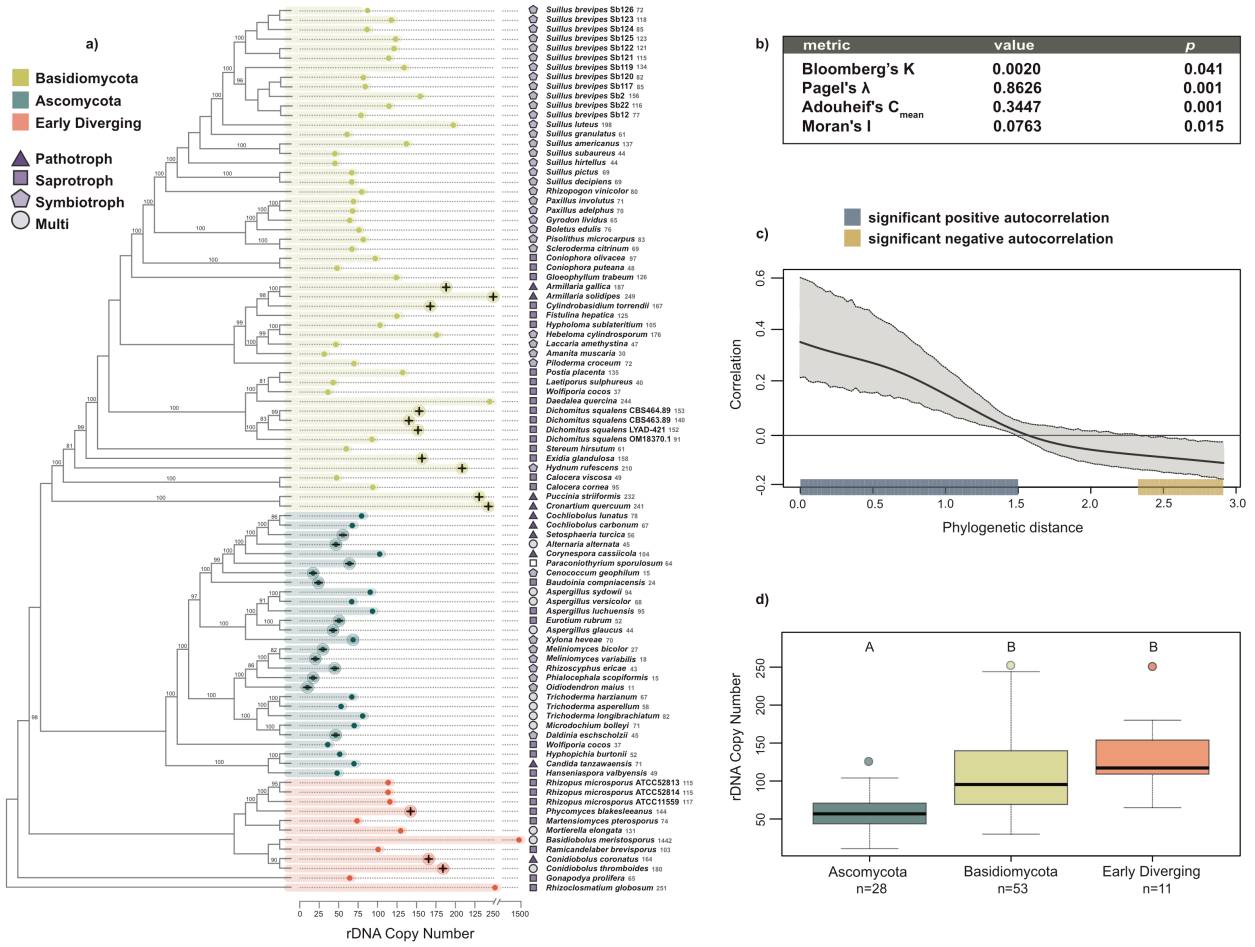
```
#function to make unique id line
Header_ID = function(){
  header = c("@HISEQ15:100:AXXXXXXXX:1:1000:", sample(1000:2000, 1), ":", sample(1000:2000, 1), " ", "1:N:0:XXXXXX")
  paste(header, collapse="")
}

#function to assemble fastq files
make_fastq_ill <- function(df) {
  forward.read  <- NULL
  reverse.read  <- NULL
  fragment <- seq(from=1, to=(nrow(df)))
  first.101<- seq(from=1, to=101)
  for(i in 1:nrow(df)){
    header <- Header_ID()
    fragment[i] <- df[i,1]
    first.101 <- substring(fragment,1,101)
    last.101 <- substring(fragment,400,501)
    q_line <- rep("~",101)
    q_line2 <- paste(q_line, collapse="")
    x <- ("+")
    x2 <- paste(x, collapse="")
    complement <- chartr("ATGC","TACG", last.101)
    reverse <- stri_reverse(complement)
    reverse_header <- gsub("1:N:0", "2:N:0", header)
    forward.read[i] <- cat(header, first.101[i], x2, q_line2, sep = "\n")
    reverse.read[i] <- cat(reverse_header, reverse[i], x2, q_line2, sep = "\n")
  }}
```

RETURNS +- 1 COPY

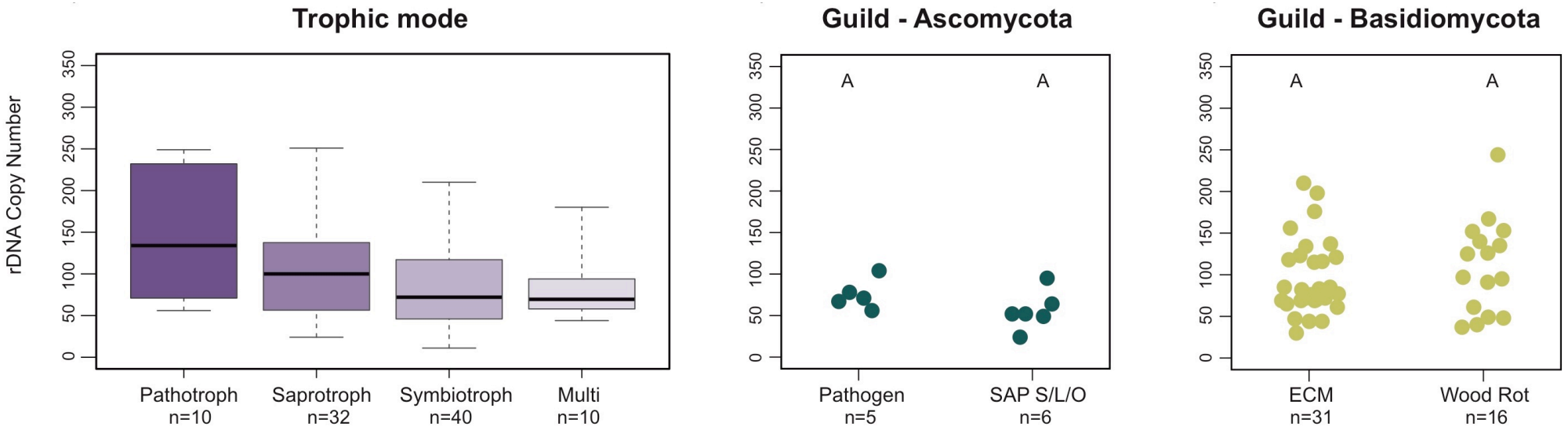


Pipeline is consistent after a depth >65X

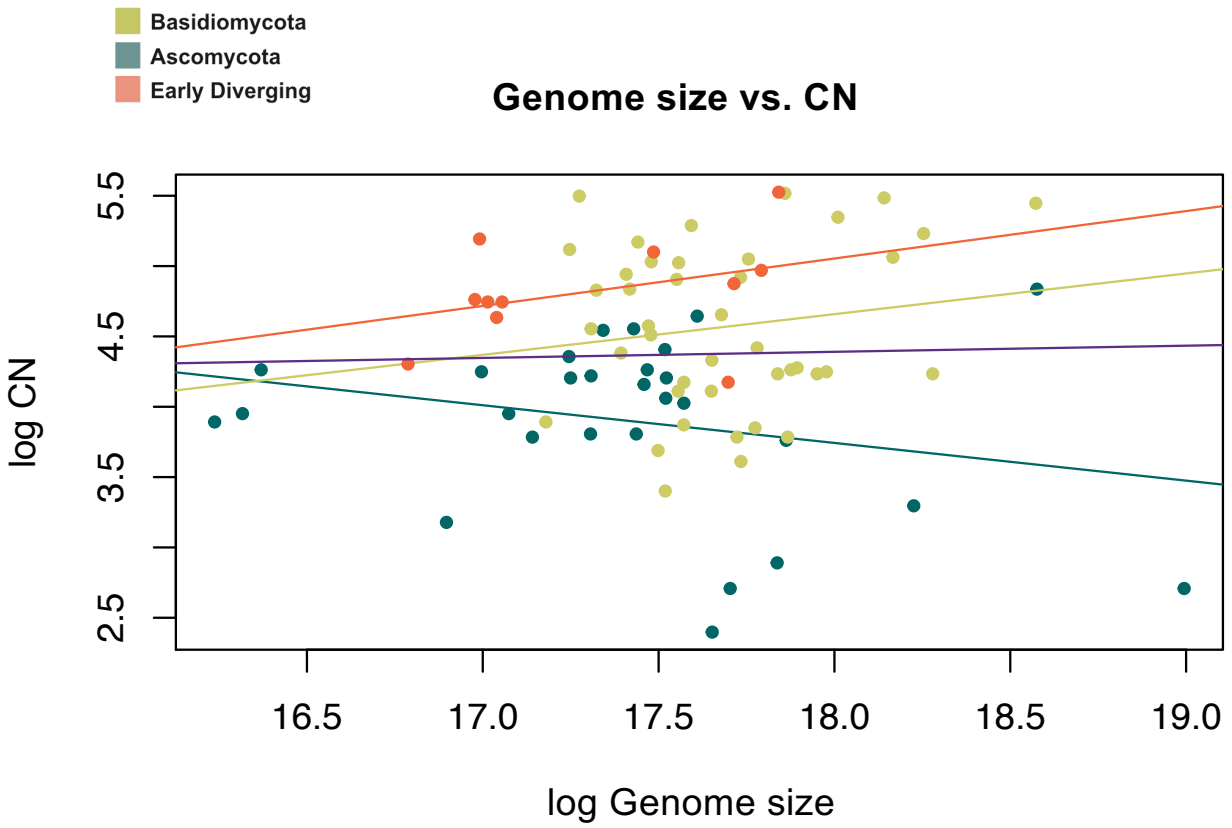
INVERSELY RELATED TO PHYLO. DISTANCE



NOT CONSERVED BY TROPHIC MODE/GUILD



CN IS NOT RELATED TO GENOME SIZE



Multiple hypothesis testing using both parametric and non-parametric tests, did not support this relationship