03.Community_Winners_Lossers

Angel Rain-Franco

9/8/2022

Contents

1 Setting up the workspace

1.1 Loading Packages

```
rm(list = ls())
library(cowplot)
library(egg)
library(readxl)
library(FactoMineR)
library(factoextra)
library(RColorBrewer)  #Expando color palette
library(dplyr)
library(stringr)  # FOr editing string
library(reshape2)  #For 'melt' function
```

1.2 Loading colorblind palette

```
# Setting the colorblind palette
cbp1 <- c("#999999", "#FFDB6D", "#E69F00", "#56B4E9", "#009E73", "#F0E442",
    "#0072B2", "#D55E00", "#CC79A7", "#293352")
# Expanding the standard pallete Pastel1 for downstream analysis
mycolors <- colorRampPalette(brewer.pal(8, "Pastel1"))(31)
```

1.3 Load OTU table and metadata

```
df1 = data.frame(read.csv("../data/OTU_table_merged200_SILVA_megablast.csv",
    row.names = 1))
tibble(df1)
```

```
## # A tibble: 1,047 x 107
##    Kingdom  Phylum       Class Order Family Genus Specie     C01    C010    C011
##    <chr>    <chr>        <chr> <chr> <chr>  <chr> <chr>    <dbl>   <dbl>   <dbl>
##  1 Bacteria Proteobacte~ Gamm~ Ente~ Alter~ Rhei~ marin~ 0       0       0
##  2 Bacteria Bacteroidota Bact~ Flav~ Flavo~ Flav~ marin~ 0.00893 1.24e-2 0.00206
##  3 Bacteria Chloroflexi  SL56~ mari~ <NA>   o_ma~ <NA>   0       3.44e-4 0
##  4 Bacteria Actinobacte~ Acid~ Micr~ Iluma~ f_Il~ marin~ 0       0       0
##  5 Bacteria Proteobacte~ Gamm~ Burk~ Comam~ f_Co~ marin~ 0       0       0
##  6 Bacteria Proteobacte~ Gamm~ Burk~ Comam~ Pelo~ Pelom~ 0       0       0
##  7 Bacteria Proteobacte~ Gamm~ Burk~ Comam~ f_Co~ <NA>   0       0       0
##  8 Bacteria Proteobacte~ Gamm~ Pseu~ Pseud~ Pseu~ bacte~ 0       0       0
##  9 Bacteria Bacteroidota Bact~ Flav~ Flavo~ Flav~ Flavo~ 0       1.75e-2 0
## 10 Bacteria Proteobacte~ Gamm~ Pseu~ Pseud~ Pseu~ Pseud~ 0       3.44e-4 0
## # i 1,037 more rows
## # i 97 more variables: C012 <dbl>, C013 <dbl>, C014 <dbl>, C015 <dbl>,
## #   C016 <dbl>, C017 <dbl>, C018 <dbl>, C019 <dbl>, C02 <dbl>, C020 <dbl>,
## #   C03 <dbl>, C04 <dbl>, C05 <dbl>, C06 <dbl>, C07 <dbl>, C08 <dbl>,
## #   C09 <dbl>, C11 <dbl>, C110 <dbl>, C111 <dbl>, C112 <dbl>, C113 <dbl>,
## #   C114 <dbl>, C115 <dbl>, C116 <dbl>, C117 <dbl>, C118 <dbl>, C119 <dbl>,
## #   C12 <dbl>, C120 <dbl>, C13 <dbl>, C14 <dbl>, C15 <dbl>, C16 <dbl>, ...
```

```
meta = data.frame(read.csv("../data/metadata_merged200_SILVA_megablast.csv",
    row.names = 1))
tibble(meta)
```

```
## # A tibble: 100 x 3
##    Cycle Replicate M.ID
##    <chr>     <int> <chr>
##  1 C0            1 C01
##  2 C0           10 C010
##  3 C0           11 C011
##  4 CO           12 C012
##  5 CO           13 C013
##  6 CO           14 C014
##  7 CO           15 C015
##  8 CO           16 C016
##  9 CO           17 C017
## 10 CO           18 C018
## # i 90 more rows
```

1.4 Subsetting dataset

```
# Overview by hierarchical level (Order, Family or Genus) aggregate
# counts by Genus
df2 <- aggregate(. ~ Genus, data = df1[, c(6, 8:107)], sum, na.rm = TRUE)
colSums(df2[, 2:101])
```

```
##  C01 C010 C011 C012 C013 C014 C015 C016 C017 C018 C019  C02 C020  C03  C04  C05
##    1    1    1    1    1    1    1    1    1    1    1    1    1    1    1    1
##  C06  C07  C08  C09  C11 C110 C111 C112 C113 C114 C115 C116 C117 C118 C119  C12
##    1    1    1    1    1    1    1    1    1    1    1    1    1    1    1    1
## C120  C13  C14  C15  C16  C17  C18  C19  C41 C410 C411 C412 C413 C414 C415 C416
##    1    1    1    1    1    1    1    1    1    1    1    1    1    1    1    1
## C417 C418 C419  C42 C420  C43  C44  C45  C46  C47  C48  C49  C61 C610 C611 C612
##    1    1    1    1    1    1    1    1    1    1    1    1    1    1    1    1
## C613 C614 C615 C616 C617 C618 C619  C62 C620  C63  C64  C65  C66  C67  C68  C69
##    1    1    1    1    1    1    1    1    1    1    1    1    1    1    1    1
##  C71 C710 C711 C712 C713 C714 C715 C716 C717 C718 C719  C72 C720  C73  C74  C75
##    1    1    1    1    1    1    1    1    1    1    1    1    1    1    1    1
##  C76  C77  C78  C79
##    1    1    1    1
```

```
order <- df2[, 1]  #vector with orders
rownames(df2) <- df2[, 1]  #rownames
df2 <- df2[, -1]  #remove column with orders and keep only abundance data
colSums(df2)  #test colSums to see if values close to 1 are reached
```

```
##  C01 C010 C011 C012 C013 C014 C015 C016 C017 C018 C019  C02 C020  C03  C04  C05
##    1    1    1    1    1    1    1    1    1    1    1    1    1    1    1    1
##  C06  C07  C08  C09  C11 C110 C111 C112 C113 C114 C115 C116 C117 C118 C119  C12
##    1    1    1    1    1    1    1    1    1    1    1    1    1    1    1    1
## C120  C13  C14  C15  C16  C17  C18  C19  C41 C410 C411 C412 C413 C414 C415 C416
##    1    1    1    1    1    1    1    1    1    1    1    1    1    1    1    1
## C417 C418 C419  C42 C420  C43  C44  C45  C46  C47  C48  C49  C61 C610 C611 C612
##    1    1    1    1    1    1    1    1    1    1    1    1    1    1    1    1
## C613 C614 C615 C616 C617 C618 C619  C62 C620  C63  C64  C65  C66  C67  C68  C69
##    1    1    1    1    1    1    1    1    1    1    1    1    1    1    1    1
##  C71 C710 C711 C712 C713 C714 C715 C716 C717 C718 C719  C72 C720  C73  C74  C75
##    1    1    1    1    1    1    1    1    1    1    1    1    1    1    1    1
##  C76  C77  C78  C79
##    1    1    1    1
```

1.5 Use Genus level for downstream analysis

```
### NEW STEP ## USE GENUS INSTEAD OF OTU LEVEL TAXONOMY ##
df2$Genus = row.names(df2)
tax.genus = unique(df1[, 1:6])
df1 = merge(tax.genus, df2, by = "Genus", all.x = TRUE)
row.names(df1) = df1$Genus
```

2 Elo-rating

2.1 Setting up dataset

```
meta$M.ID = as.factor(meta$M.ID)
meta$Replicate = as.factor(meta$Replicate)

# For microcosm
df.C0.C6 = df1[, c(7:106)]
dim(df.C0.C6)
```

```
## [1] 118 100
```

```
# Only those that represent 0.1% (comprising 41 Genera)
df.C0.C6 = df.C0.C6[which(rowMeans(df.C0.C6) > 0.001), ]
dim(df.C0.C6)
```

```
## [1]  41 100
```

```
# Get ocurrence at C0 and C6
df.tmp.occurrence <- data.frame(C0 = rowSums(df.C0.C6[, c(1:20)] > 0),
    C6 = rowSums(df.C0.C6[, c(61:80)] > 0))
```

2.2 Distribution of the mean rankings

2.2.1 Screen distribution of the ranking during the cycles (C0 to C6)

```
ave.genus.cycles <- data.frame(Mean.abundance = rowMeans(df.C0.C6))  # Estimate average relative abunda

# Create dataframe for model fitting
M <- data.frame(Mean.abundance = ave.genus.cycles$Mean.abundance[order(-ave.genus.cycles$Mean.abundance
    PA = 1:dim(ave.genus.cycles)[1])

# Fitting model estimation
results.nls <- nls(Mean.abundance ~ ((alpha^(dim(ave.genus.cycles)[1] -
    PA)) - 1)/sum((alpha^(dim(ave.genus.cycles)[1] - PA)) - 1), data = M,
    start = list(alpha = 0.1))
summary(results.nls)
```

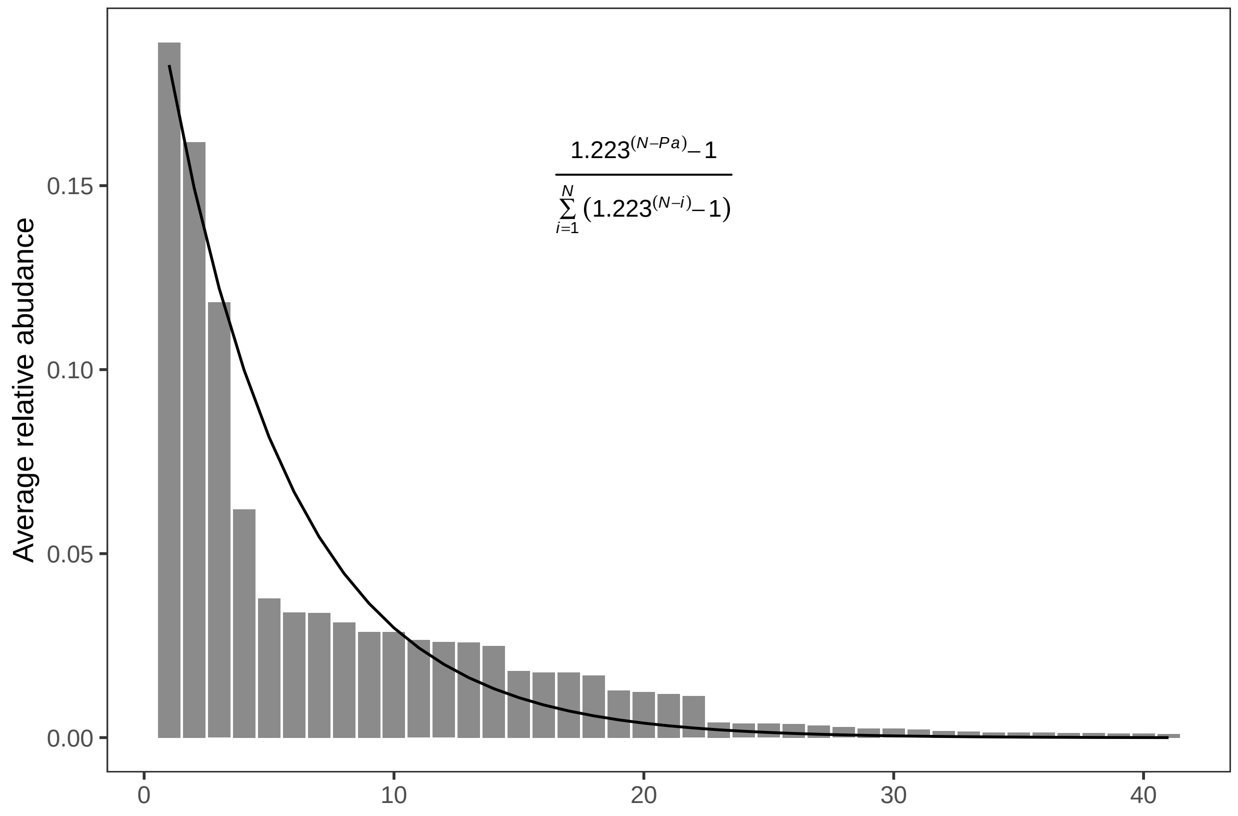
```
##
## Formula: Mean.abundance ~ ((alpha^(dim(ave.genus.cycles)[1] - PA)) - 1)/sum((alpha^(dim(ave.genus.cy
##     PA)) - 1)
##
## Parameters:
##       Estimate Std. Error t value Pr(>|t|)
## alpha  1.22303    0.01409   86.81   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.01252 on 40 degrees of freedom
##
## Number of iterations to convergence: 10
## Achieved convergence tolerance: 3.057e-06
```

```
# Save results from nls model
M$Expected <- predict(results.nls)

# Get Pseudo-R2-square
print(1 - (deviance(results.nls)/sum((M$Mean.abundance - mean(M$Mean.abundance))^2)))
```

```
## [1] 0.9062178
```

```
# plot figure
plot.rank <- M %>%
    ggplot(aes(PA, Mean.abundance)) + geom_col(alpha = 0.7) + theme_bw() +
    theme(panel.grid.major = element_blank(), panel.grid.minor = element_blank()) +
    # geom_point(aes(PA,Expected),size=0.5)+
geom_line(aes(PA, Expected)) + labs(y = "Average relative abudance", x = "") +
    geom_label(x = 20, y = 0.15, label = expression(frac("1.223"^{
        (italic(N) - italic(Pa))
    } * -1, sum(, italic(i) == 1, italic(N))("1.223"^{
        (italic(N) - italic(i))
    } * -1))), size = 3, label.size = NA)
plot.rank
```

```
## Warning in is.na(x): is.na() applied to non-(list or vector) of type
## 'expression'
```



#Load Elo-rating from iterative calculations Elo-rating calculated values from 1000 iterations we calculated in Python3 according https://github.com/djcunningham0/multielo were retrieve for downstream analysis in R.

```
Elo.0 = read.csv("../data/EloRating_C0_results1000iter.csv", header = T)[,
    -1]
Elo.1 = read.csv("../data/EloRating_C1_results1000iter.csv", header = T)[,
    -1]
Elo.4 = read.csv("../data/EloRating_C4_results1000iter.csv", header = T)[,
    -1]
Elo.6 = read.csv("../data/EloRating_C6_results1000iter.csv", header = T)[,
    -1]
```

2.3 Getting mean value from the 1000 iterations

```
sum.elo.0 = aggregate(. ~ player_id + n_games, data = Elo.0[, -1], mean)
sum.elo.0$Cycle = "C00"
head(sum.elo.0)
```

```
##            player_id n_games    rating Cycle
## 1      Bradyrhizobium       1 1000.1723   C00
## 2           Emticicia       1  990.3320   C00
## 3       Brevundimonas       2  995.2917   C00
## 4 f_Caulobacteraceae       2 1064.3713   C00
## 5        Caulobacter       3 1065.1167   C00
## 6      Chitinibacter       3  990.1293   C00
```

```
sum.elo.1 = aggregate(. ~ player_id + n_games, data = Elo.1[, -1], mean)
sum.elo.1$Cycle = "C01"
head(sum.elo.1)
```

```
##            player_id n_games   rating Cycle
## 1     Bradyrhizobium       1 1024.644   C01
## 2          Emticicia       1 1015.626   C01
## 3      Brevundimonas       2 1040.649   C01
## 4 f_Caulobacteraceae       2 1062.287   C01
## 5        Caulobacter       3 1102.510   C01
## 6      Chitinibacter       3 1023.327   C01
```

```
sum.elo.4 = aggregate(. ~ player_id + n_games, data = Elo.4[, -1], mean)
sum.elo.4$Cycle = "C04"
head(sum.elo.4)
```

```
##            player_id n_games   rating Cycle
## 1     Bradyrhizobium       1 1050.645   C04
## 2          Emticicia       1 1053.112   C04
## 3      Brevundimonas       2 1037.692   C04
## 4 f_Caulobacteraceae       2 1070.247   C04
## 5        Caulobacter       3 1063.884   C04
## 6      Chitinibacter       3 1006.275   C04
```

```
sum.elo.6 = aggregate(rating ~ player_id + n_games, data = Elo.6[, -1],
    mean)
sum.elo.6$Cycle = "C06"
head(sum.elo.6)
```

```
##            player_id n_games    rating Cycle
## 1     Bradyrhizobium       1 1049.6802   C06
## 2          Emticicia       1 1070.9531   C06
## 3      Brevundimonas       2 1040.2766   C06
## 4 f_Caulobacteraceae       2 1066.6892   C06
## 5        Caulobacter       3 1070.2905   C06
## 6      Chitinibacter       3  984.0774   C06
```

```
sum.elo = rbind(sum.elo.0, sum.elo.1, sum.elo.4, sum.elo.6)
sum.elo$Cycle.n = as.numeric(str_remove(sum.elo$Cycle, "C"))
sum.elo$Cycle.label = sum.elo$Cycle.n + 2
```

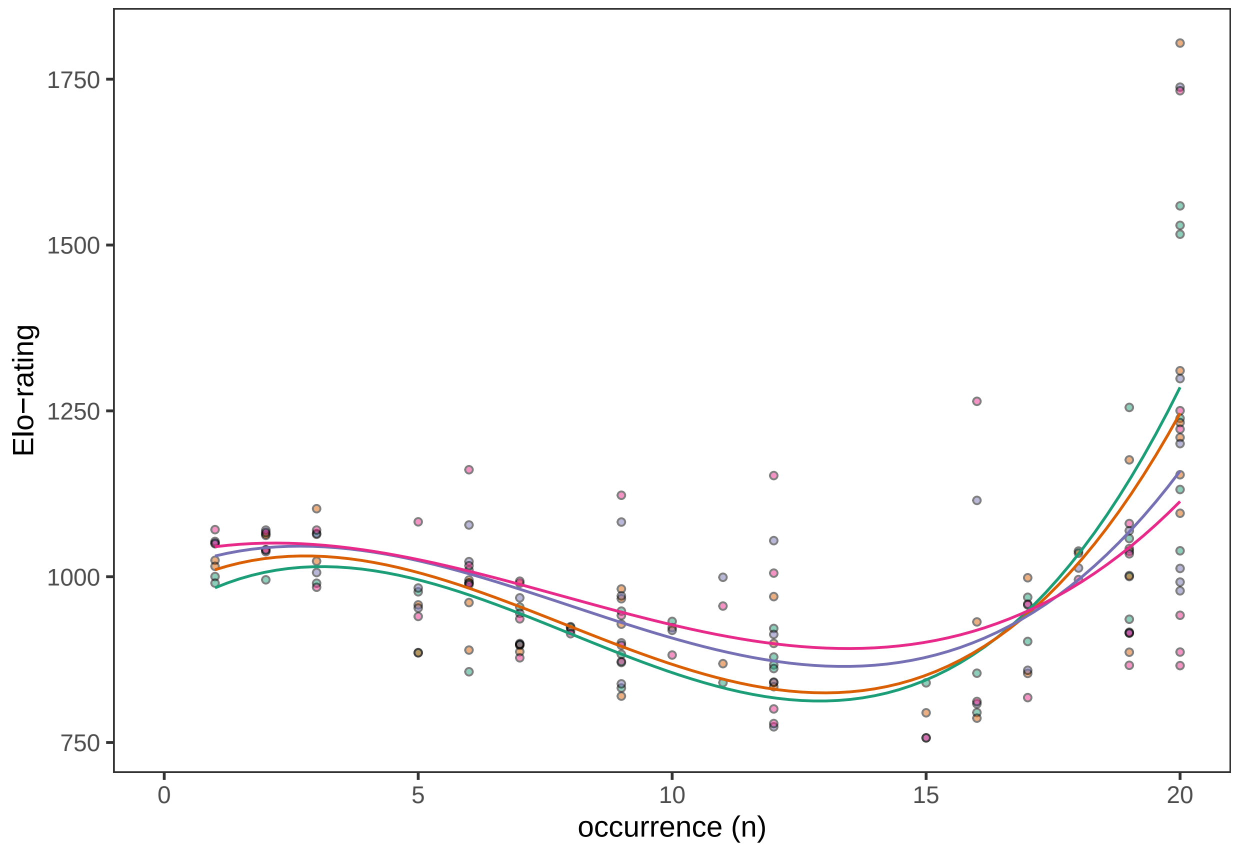
2.4 Explore relationship from Elo-rating and Abundance and ocurrence (Figure S1)

3 Occurrence vs Elo

```
plot.Elo.ocurrence<-sum.elo%>%
ggplot(aes(x=n_games,y=(rating),fill=Cycle,colour=Cycle))+
  geom_point(alpha=0.5,shape=21,size=1,color="black")+
  theme_bw()+labs(y="Elo-rating",x="occurrence (n)")+
  scale_color_brewer(palette = "Dark2")+scale_fill_brewer(palette = "Dark2")+#ylim(0,11)+
    theme(legend.position="none", panel.grid.minor = element_blank(),panel.grid.major = element_blank()
  geom_smooth(method='lm', formula = y~poly(x,3),se=F,size=0.5)
```

```
## Warning: Using `size` aesthetic for lines was deprecated in ggplot2 3.4.0.
## i Please use `linewidth` instead.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.
```

```
plot.Elo.ocurrence
```

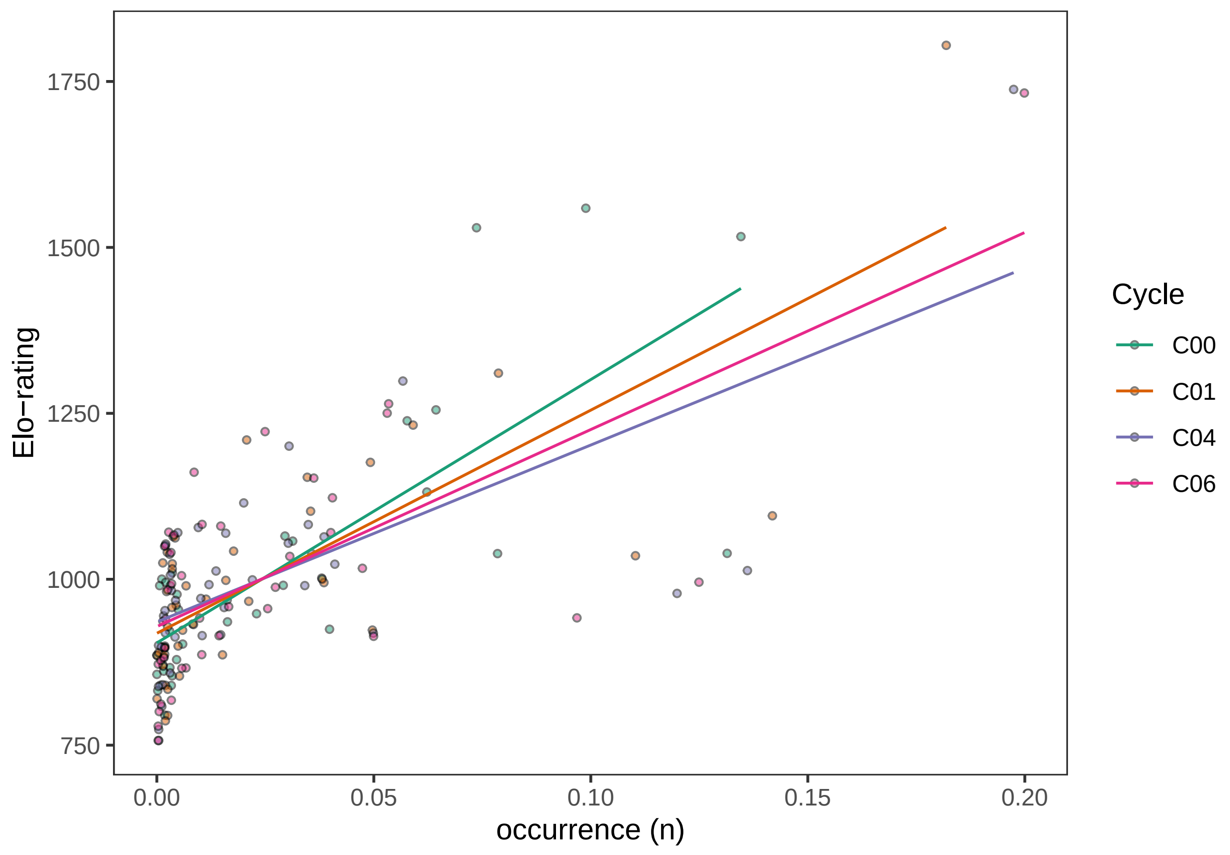


## Abundance vs Elo-rating

```
df.comparison <- data.frame(Cycle = rep(c("C00", "C01", "C04", "C06"),
    each = 41), Genus = row.names(df.C0.C6), m.rel.abundance = c(rowMeans(df.C0.C6[,
    c(1:20)]), rowMeans(df.C0.C6[, c(21:40)]), rowMeans(df.C0.C6[, c(41:60)]),
    rowMeans(df.C0.C6[, c(61:80)])))

elo.temporal <- sum.elo
elo.temporal$index <- paste0(elo.temporal$Cycle, ".", elo.temporal$player_id)
```

```
df.comparison$index <- paste0(df.comparison$Cycle, ".", df.comparison$Genus)
df.comparison <- merge(df.comparison, elo.temporal, by = "index")

plot.elo.abundance <- df.comparison %>%
    ggplot(aes(x = (m.rel.abundance), y = (rating), fill = Cycle.x, colour = Cycle.x)) +
    geom_point(alpha = 0.5, shape = 21, size = 1, color = "black") + theme_bw() +
    labs(y = "Elo-rating", x = "occurrence (n)") + scale_color_brewer(name = "Cycle",
    palette = "Dark2") + scale_fill_brewer(name = "Cycle", palette = "Dark2") +
    theme(legend.position = "right", panel.grid.minor = element_blank(),
        panel.grid.major = element_blank()) + geom_smooth(method = "lm",
    se = F, size = 0.5)

plot.elo.abundance
```

```
## 'geom_smooth()' using formula = 'y ~ x'
```



```
jpeg("../Figures/Elo_panel_supplementary.jpg", width = 21, height = 7,
    units = "cm", res = 300)
plot_grid(plot.rank, plot.Elo.ocurrence, plot.elo.abundance, rel_widths = c(0.8,
    0.78, 1), nrow = 1, labels = c("a)", "b)", "c)"))
```

```
## Warning in is.na(x): is.na() applied to non-(list or vector) of type
## 'expression'
## `geom_smooth()` using formula = 'y ~ x'
```

```
dev.off()
```

```
## pdf
##   2
```

#Include quantile

```
q = quantile(sum.elo$rating[sum.elo$Cycle == "C00"])
res.C00 = aggregate(rating ~ player_id, sum.elo[sum.elo$Cycle == "C00",
    ], mean)
res.C00$quantileC0 = "q1"
res.C00$quantileC0[which(res.C00$rating > q[2])] = "q2"
res.C00$quantileC0[which(res.C00$rating > q[3])] = "q3"
res.C00$quantileC0[which(res.C00$rating > q[4])] = "q4"

# Merging quantile and dataset
sum.elo = merge(sum.elo, res.C00, by = "player_id")
```

3.1 Fig Mean abundance C0

```
tmp.C0 <- data.frame(C0 = rowMeans(df.C0.C6[, c(1:20)]))
tmp.C0$player_id <- as.factor(rownames(tmp.C0))
tmp.C0$occ <- rowSums(df.C0.C6[, c(1:20)] != 0)
tmp.C0$player_id <- factor(tmp.C0$player_id, unique(sum.elo$player_id[order(sum.elo$rating.y)]))

# Add pseudo-count
tmp.C0$C0[tmp.C0$C0 <= 0] = 1e-05


sum.elo <- merge(sum.elo, tmp.C0, by = "player_id")

# Apply linear regression to each dataset
library(correlation)
df_lm <- sum.elo[, c(1, 3, 5)] %>%
    group_by(player_id) %>%
    correlation(method = "spearman", p_adjust = "BH")
## Merging regression results with dataset
sum.elo = merge(sum.elo, df_lm, by.x = "player_id", by.y = "Group")
tibble(sum.elo)
```

```
## # A tibble: 164 x 20
##    player_id      n_games rating.x Cycle Cycle.n Cycle.label rating.y quantileC0
##    <chr>            <int>    <dbl> <chr>   <dbl>       <dbl>    <dbl> <chr>
##  1 Acidovorax          20    1733. C06         6           8    1516. q4
##  2 Acidovorax          20    1738. C04         4           6    1516. q4
##  3 Acidovorax          20    1805. C01         1           3    1516. q4
##  4 Acidovorax          20    1516. C00         0           2    1516. q4
##  5 Aeromonas           20    1039. C00         0           2    1039. q4
##   6 Aeromonas           20     942. C06          6           8    1039. q4
##   7 Aeromonas           20     979. C04          4           6    1039. q4
##   8 Aeromonas           20    1096. C01          1           3    1039. q4
##   9 Allorhizobium~       5     886. C00          0           2     886. q2
##  10 Allorhizobium~       5    1083. C06          6           8     886. q2
## # i 154 more rows
## # i 12 more variables: C0 <dbl>, occ <dbl>, Parameter1 <chr>, Parameter2 <chr>,
## #   rho <dbl>, CI <dbl>, CI_low <dbl>, CI_high <dbl>, S <dbl>, p <dbl>,
## #   Method <chr>, n_Obs <int>
```

```
## include values from correlation
sum.elo$sig = NA
sum.elo$sig = ifelse(sum.elo$rho > 0, "Increased", "Decreased")

## Preparing figure
sum.elo$player_id[sum.elo$player_id == "Allorhizobium-Neorhizobium-Pararhizobium-Rhizobium"] <- "ANPR"
sum.elo$player_id <- str_replace(sum.elo$player_id, "f_", "uncl_")


plot.elo.T0 <- sum.elo[sum.elo$Cycle == "C00", ] %>%
    ggplot(aes(x = reorder(player_id, rating.y), y = (rating.y), colour = sig)) +
    geom_hline(yintercept = 1000, color = "grey") + geom_point(aes(colour = ifelse(p <
    0.05, sig, "No change")), size = 1.5) + scale_color_manual(values = c("red3",
    "steelblue1", "black"), name = NULL) + theme(legend.position = "none") +
    labs(x = NULL, y = bquote(Elo - rating[C0]), title = "") + geom_segment(aes(y = 1000,
    colour = ifelse(p < 0.05, sig, "No change"), x = player_id, yend = (rating.y),
    xend = player_id)) + geom_vline(xintercept = 11, linetype = "dashed",
    linewidth = 0.3) + geom_label(label = "25th Percentile", x = 10, y = 1600,
    color = "black", size = 2, fontface = "italic", label.size = NA) +
    geom_vline(xintercept = 31.5, linetype = "dashed", linewidth = 0.3) +
    geom_label(label = "75th Percentile", x = 33, y = 1600, color = "black",
        size = 2, fontface = "italic", label.size = NA) + geom_vline(xintercept = 21,
    linetype = "dashed", linewidth = 0.3) + geom_label(label = "Median",
    x = 23, y = 1600, color = "black", size = 2, fontface = "italic", label.size = NA) +
    theme_bw() + coord_flip() + theme(text = element_text(size = 8), legend.position = c(0.65,
    0.1), legend.key.height = unit(0, "cm"), legend.margin = margin(0,
    0, 0, "cm")) + ylim(750, 1850) + theme(panel.grid.minor = element_blank(),
    panel.grid.major = element_blank())
```

```
## Warning: A numeric ‘legend.position‘ argument in ‘theme()‘ was deprecated in ggplot2
## 3.5.0.
## i Please use the ‘legend.position.inside‘ argument of ‘theme()‘ instead.
## This warning is displayed once every 8 hours.
## Call ‘lifecycle::last_lifecycle_warnings()‘ to see where this warning was
## generated.
```

```
## Fig Elo rating at C6
plot.elo.T6 <- sum.elo[sum.elo$Cycle == "C06", ] %>%
    ggplot(aes(x = reorder(player_id, rating.y), y = (rating.x), colour = sig)) +
    geom_hline(yintercept = 1000, color = "grey") + geom_point(aes(colour = ifelse(p <
    0.05, sig, "No change")), size = 1.5) + scale_color_manual(values = c("red3",
    "steelblue1", "black"), name = NULL) + theme(legend.position = "none") +
    labs(x = NULL, y = bquote(Elo - rating[C6]), title = "") + geom_segment(aes(y = 1000,
    colour = ifelse(p < 0.05, sig, "No change"), x = player_id, yend = (rating.x),
    xend = player_id)) + geom_vline(xintercept = 11.5, linetype = "dashed",
    linewidth = 0.3) + geom_vline(xintercept = 31.5, linetype = "dashed",
    linewidth = 0.3) + geom_vline(xintercept = 21, linetype = "dashed",
    linewidth = 0.3) + theme_bw() + coord_flip() + theme(text = element_text(size = 8),
    legend.position = "none", axis.text.y = element_blank()) + ylim(750,
    1850) + theme(panel.grid.minor = element_blank(), panel.grid.major = element_blank())
```

4 Panel ELo feactures

```
# Export only relative abundance plots
pdf("../Figures/Fig5.pdf", width = 4.33071, height = 4)

plot_grid(plot.elo.T0, NULL, plot.elo.T6, rel_widths = c(0.25, -0.005,
    0.147), nrow = 1, labels = c("a)", "", "b)"), vjust = 1.3, label_size = 14,
    align = "h")

dev.off()
```

```
## pdf
##   2
```