Microbiota adaptation review - 2. Analysis

Garazi Martin Bideguren, garazi.bideguren@sund.ku.dk
Antton Alberdi, antton.alberdi@sund.ku.dk

07-11-2023

Contents

Data preparation

```
library(knitr)
library(dplyr)
library(gridExtra)
library(tidyverse)
library(devtools)
library(ggplot2)
library(vegan)
library(FSA)
```

Prepare the main dataset were all the analysed studies will have their obtained scores for each of the criteria. Subsample the dataset in 4 dataframes: 1)Select the columns corresponding to the 10 analysed criteria, 2)Select the columns corresponding to the experimental design, 3)Select the columns corresponding to the methodological resolution and 4) Select the columns corresponding to the reproducibility. Finally, prepare the file with the weigthed values for each of the analysed criteria.

```
# Dataset
scores <- read.csv("data/scores.csv",header=T)

# Groups of criteria
criteria <- colnames(scores)[c(5:14)]
design <- colnames(scores)[c(5:8)]
methods <- colnames(scores)[c(9:13)]
reproducibility <- colnames(scores)[14]

# Criteria weights
weights <- read.csv("data/weights.csv",header=T,row.names=1) %>%
   rowwise() %>%
   mutate(average=mean(c_across(everything()))) #calculate average weight
```

Performance score statistics

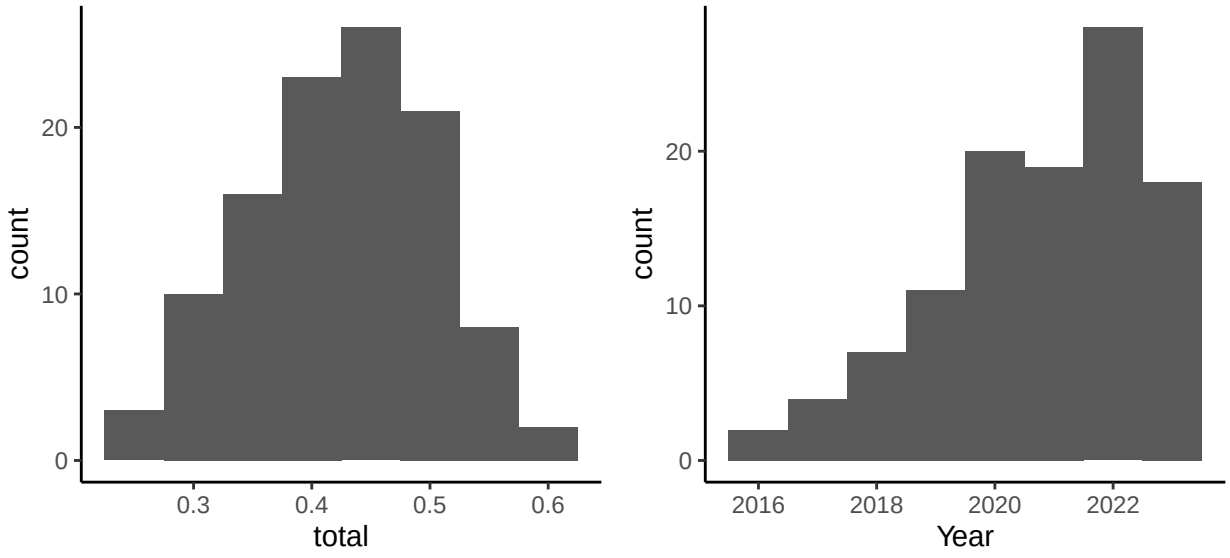
Compute total performance scores given the vector of average criterion weights.

```
# Generate vectors of weights
weight_consensus <- weights$average
names(weight_consensus) <- criteria
# Weighed criteria only for design
weight_consensus_design <-
    weight_consensus[design]/sum(weight_consensus[design])
# Weighed criteria only for methods
weight_consensus_methods <-
    weight_consensus[methods]/sum(weight_consensus[methods])
# Weighed criteria only for reproducibility
weight_consensus_reproducibility <-
    weight_consensus[reproducibility]/sum(weight_consensus[reproducibility])

# Calculate weighed domain-specific and overall performance scores
scores <- scores %>%
  # Total score for design
  mutate(total_design = rowSums(across(all_of(design),
        ~ . * weight_consensus_design[[(cur_column())]]))) %>%
  # Total score for methods
```

```
  mutate(total_methods = rowSums(across(all_of(methods),
        ~ . * weight_consensus_methods[[(cur_column())]]))) %>%
  # Total score for reproducibility
  mutate(total_reproducibility = rowSums(across(all_of(reproducibility),
        ~ . * weight_consensus_reproducibility[[(cur_column())]]))) %>%
  #Overall total score
  mutate(total = rowSums(across(all_of(criteria),
        ~ . * weight_consensus[[(cur_column())]])))
```

Overall visual statistics

```
# Distribution of total scores of the 109 papers analysed
score <- scores %>%
  ggplot(aes(x=total)) +
  geom_histogram(binwidth = 0.05,) +
  theme_classic()

# The 109 papers analysed distributed per year
year <- scores %>%
  ggplot(aes(x=Year)) +
  geom_histogram(binwidth = 1) +
  theme_classic()

  #Composite plot
  grid.arrange(grobs = list(score,year),
        layout_matrix = matrix(1:2, nrow = 1))
```

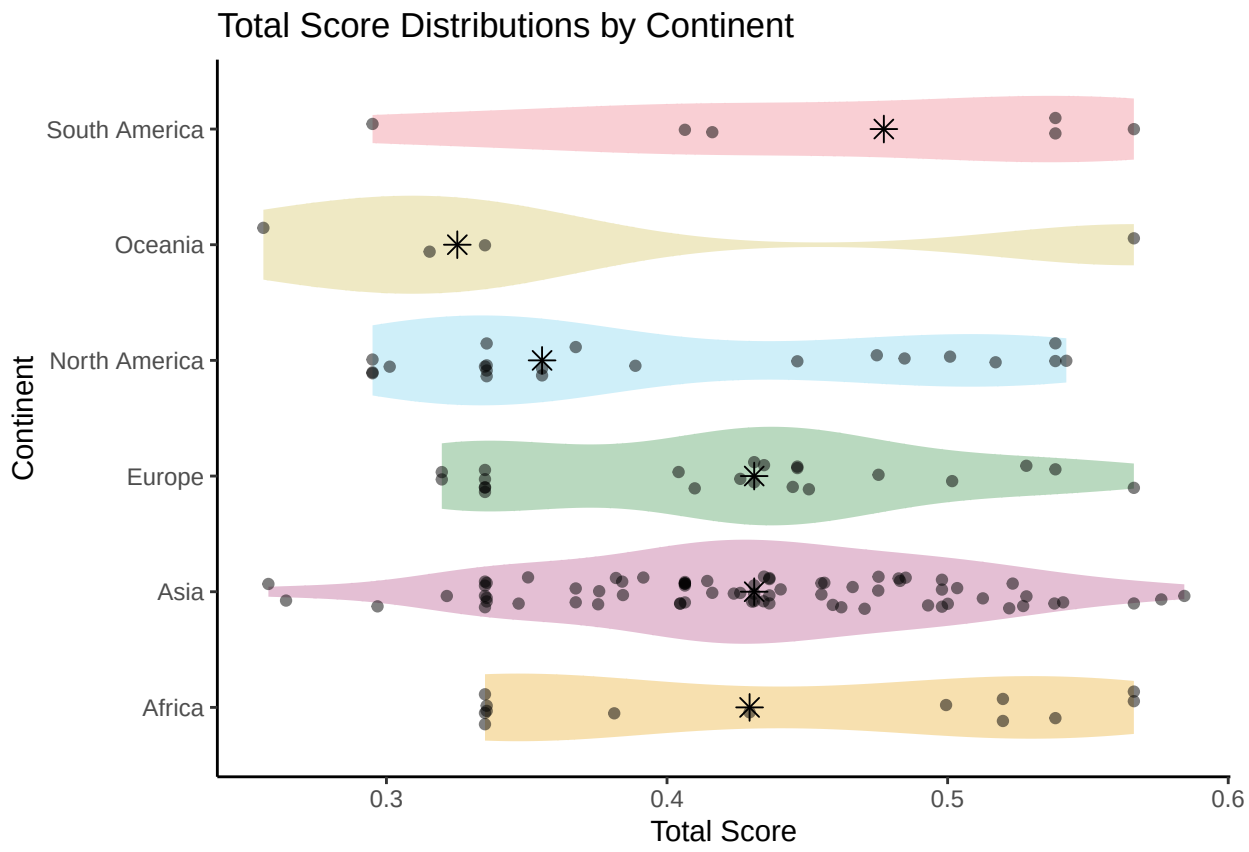


Performance scores per continent

```
scores_continent <- scores %>%
  separate_rows(Continent, sep = ", ")
```

Visualisation (Fig. S2)

```
scores %>%
  mutate(Year = factor(Year)) %>%
  separate_rows(Continent, sep = ", ") %>%
  ggplot(aes(x = Continent, y = total, fill = Continent)) +
  geom_violin(color = NA) +
  geom_point(stat = "summary", fun = "median", shape=8, color = "black", size = 3) +
  geom_jitter(
    width = 0.15,
    height = 0,
    alpha = 0.5,
    size = 1.5) +
  scale_fill_manual(values = c("#E69F0050", "#AA337750", "#22883350",
        "#66CCEE50", "#CCBB4450", "#EE667750", "#4477AA50")) +
  scale_colour_manual(values = c("#E69F00", "#AA3377", "#228833",
        "#66CCEE", "#CCBB44", "#EE6677", "#4477AA"))+
  theme_classic() +
  labs(title = "Total Score Distributions by Continent",
       y = "Total Score",
       x = "Continent") +
  coord_flip() +
  theme(legend.position = "none")
```



Statistical model (Table S2)

Perform a Kruskal-Wallis test to check if there are differences between the total scores obtained and the continents. Also, as an extra step, check if there are any specific pairs of continents that differ significantly from each other, running the Dunn test. Render both analysis into tables to better inspect them.

```
# Kruskal-Wallis test
kruskal <- scores_continent %>%
  kruskal.test(total ~ Continent, .)

# Render results
c(kruskal$statistic,kruskal$parameter,pvalue=kruskal$p.value) %>%
      round(4) %>%
      kable()
```

	x
Kruskal-Wallis chi-squared	5.3242
df	5.0000
pvalue	0.3776

```
# Dunn test for multiple comparisons
dunn <- scores_continent %>%
  mutate(Continent = factor(Continent)) %>%
  dunnTest(total ~ Continent, data=., method="bh")

# Render results
dunn$res %>% kable()
```

Comparison	Z	P.unadj	P.adj
Africa - Asia	0.1602402	0.8726919	0.8726919
Africa - Europe	0.5917461	0.5540206	0.6392546
Asia - Europe	0.6515432	0.5146959	0.7018580
Africa - North America	1.1354864	0.2561716	0.5489391
Asia - North America	1.4210462	0.1553033	0.4659100
Europe - North America	0.6349538	0.5254585	0.6568232
Africa - Oceania	1.4675488	0.1422268	0.7111341
Asia - Oceania	1.5394450	0.1236957	0.9277176
Europe - Oceania	1.1628865	0.2448755	0.6121889
North America - Oceania	0.8035711	0.4216447	0.7027412
Africa - South America	-0.5400536	0.5891601	0.6312430
Asia - South America	-0.7409209	0.4587414	0.6881121
Europe - South America	-1.0281911	0.3038599	0.5697374
North America - South America	-1.4414497	0.1494577	0.5604663
Oceania - South America	-1.7128587	0.0867385	1.0000000

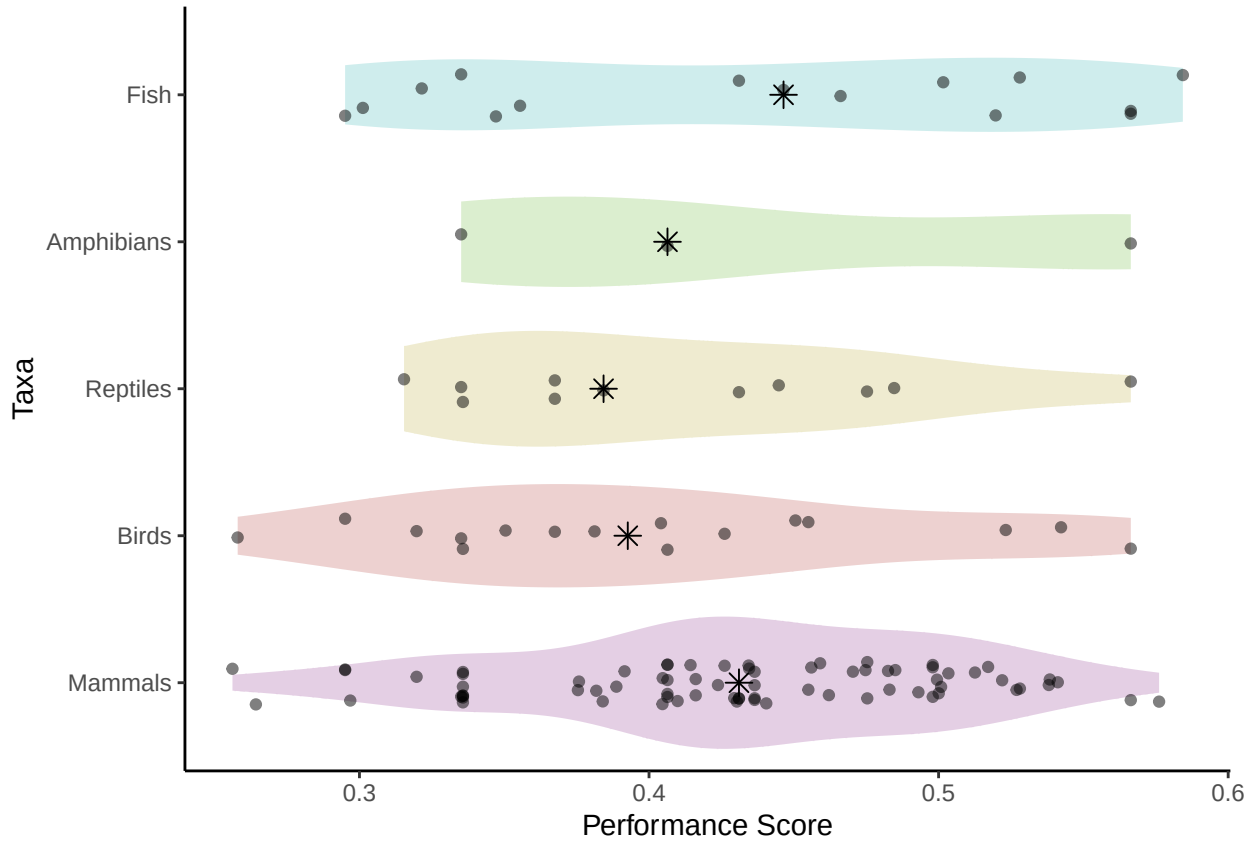
Performance scores per animal taxa

```
scores_taxa <- scores %>%
  separate_rows(Taxa, sep = ", ") %>%
  mutate(Taxa = ifelse(Taxa == "Mammalia", "Mammals", Taxa)) %>%
  mutate(Taxa = ifelse(Taxa == "Aves", "Birds", Taxa)) %>%
```

```
  mutate(Taxa = ifelse(Taxa == "Reptilia", "Reptiles", Taxa)) %>%
  mutate(Taxa = ifelse(Taxa == "Amphibia", "Amphibians", Taxa)) %>%
  mutate(Taxa = ifelse(Taxa == "Dipnoi", "Fish",
      ifelse(Taxa == "Actinopterygii", "Fish", Taxa))) %>%
  mutate(Taxa = factor(Taxa,
      levels = c("Mammals","Birds","Reptiles","Amphibians","Fish")))
```

Visualisation (Fig. 2c)

```
scores_taxa %>%
  ggplot(aes(x = Taxa, y = total, fill = Taxa)) +
  geom_violin(color = NA) +
  geom_point(stat = "summary", fun = "median", shape=8, color = "black", size = 3) +
  geom_jitter(
    width = 0.15,
    height = 0,
    alpha = 0.5,
    size = 1.5) +
  scale_fill_manual(values = c("#AB67AA50", "#C66E6A50", "#CCC26A50",
      "#8FC96750", "#68C8C650")) +
  scale_colour_manual(values = c("#AB67AA", "#C66E6A", "#CCC26A",
      "#8FC967", "#68C8C6"))+
  theme_classic() +
  labs(y = "Performance Score") +
  coord_flip() +
  theme(legend.position = "none")
```



Statistical model (Table S2)

Perform a Kruskal-Wallis test to check if there are differences between the total scores obtained and the animal taxas. Also, as an extra step, check if there are any specific pairs of taxas that differ significantly from each other, running the Dunn test. Render both analysis into tables to better inspect them.

```
# Kruskal-Wallis test
kruskal <- scores_taxa %>%
  kruskal.test(total ~ Taxa, .)

# Render results
c(kruskal$statistic,kruskal$parameter,pvalue=kruskal$p.value) %>%
  round(4) %>%
  kable()
```

	x
Kruskal-Wallis chi-squared	2.8549
df	4.0000
pvalue	0.5824

```
# Dunn test for multiple comparisons
dunn <- scores_taxa %>%
  dunnTest(total ~ Taxa, data=., method="bh")

# Render results
```

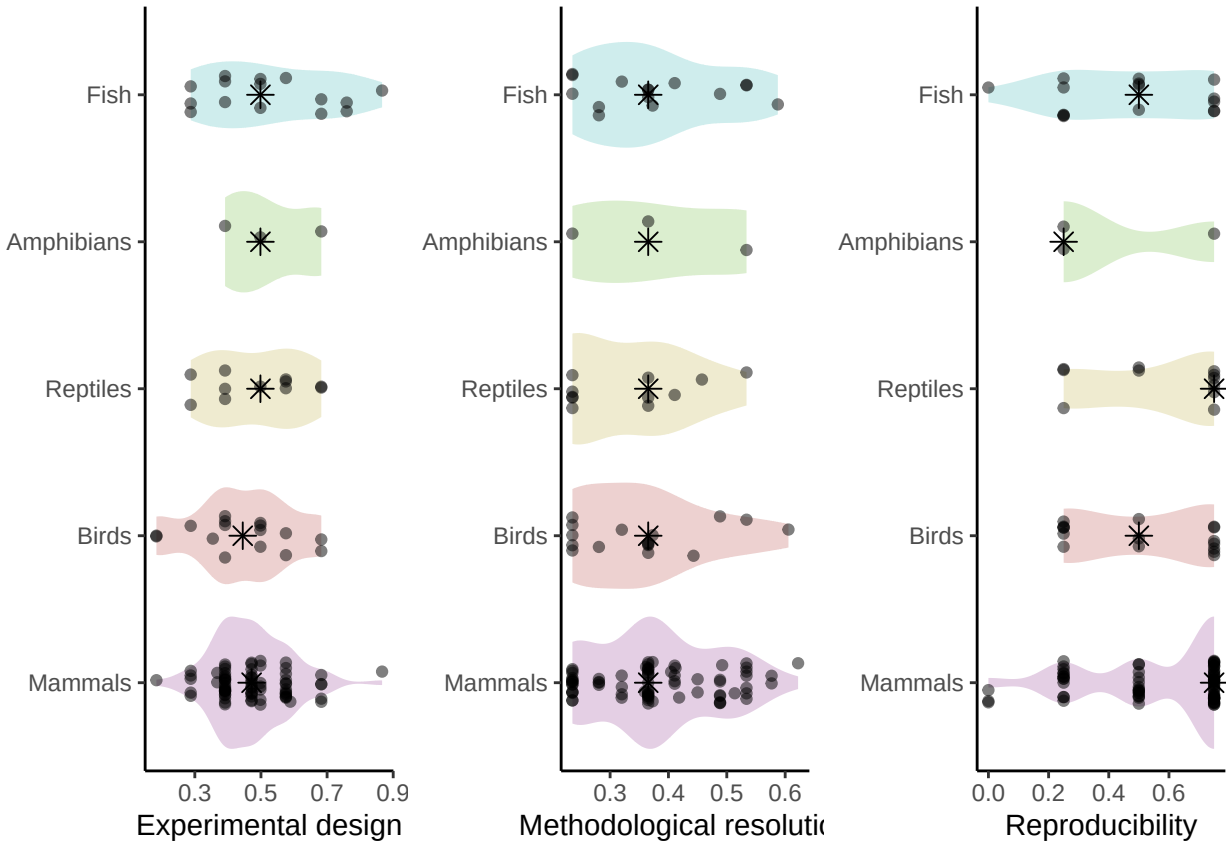
```
dunn$res %>% kable()
```

Comparison	Z	P.unadj	P.adj
Amphibians - Birds	0.4886043	0.6251219	1.0000000
Amphibians - Fish	-0.2389318	0.8111585	1.0000000
Birds - Fish	-1.2758029	0.2020252	1.0000000
Amphibians - Mammals	-0.1427120	0.8865177	0.8865177
Birds - Mammals	-1.4181864	0.1561363	1.0000000
Fish - Mammals	0.2364933	0.8130499	0.9033888
Amphibians - Reptiles	0.3258953	0.7445036	1.0000000
Birds - Reptiles	-0.2429018	0.8080814	1.0000000
Fish - Reptiles	0.9154165	0.3599730	0.8999325
Mammals - Reptiles	0.9162354	0.3595434	1.0000000

Visualisation (Fig. S2) Were do we place this in the paper??

```
# Experimental design
total_design_taxa_plot <- scores_taxa %>%
  ggplot(aes(x = Taxa, y = total_design, fill = Taxa)) +
  geom_violin(color = NA) +
  geom_point(stat = "summary", fun = "median", shape=8, color = "black", size = 3) +
  geom_jitter(
    width = 0.15,
    height = 0,
    alpha = 0.5,
    size = 1.5) +
  scale_fill_manual(values = c("#AB67AA50", "#C66E6A50", "#CCC26A50",
        "#8FC96750", "#68C8C650")) +
  scale_colour_manual(values = c("#AB67AA", "#C66E6A", "#CCC26A",
        "#8FC967", "#68C8C6"))+
  theme_classic() +
  labs(y = "Experimental design") +
  coord_flip() +
  theme(legend.position = "none", axis.title.y = element_blank())

# Methodological resolution
total_methods_taxa_plot <- scores_taxa %>%
  ggplot(aes(x = Taxa, y = total_methods, fill = Taxa)) +
  geom_violin(color = NA) +
  geom_point(stat = "summary", fun = "median", shape=8, color = "black", size = 3) +
  geom_jitter(
    width = 0.15,
    height = 0,
    alpha = 0.5,
    size = 1.5) +
  scale_fill_manual(values = c("#AB67AA50", "#C66E6A50", "#CCC26A50",
        "#8FC96750", "#68C8C650")) +
  scale_colour_manual(values = c("#AB67AA", "#C66E6A", "#CCC26A",
        "#8FC967", "#68C8C6"))+
  theme_classic() +
  labs(y = "Methodological resolution") +
  coord_flip() +
```

```r
  theme(legend.position = "none", axis.title.y = element_blank())

# Reproducibility
total_reproducibility_taxa_plot <- scores_taxa %>%
  ggplot(aes(x = Taxa, y = total_reproducibility, fill = Taxa)) +
  geom_violin(color = NA) +
  geom_point(stat = "summary", fun = "median", shape=8, color = "black", size = 3) +
  geom_jitter(
    width = 0.15,
    height = 0,
    alpha = 0.5,
    size = 1.5) +
  scale_fill_manual(values = c("#AB67AA50", "#C66E6A50", "#CCC26A50",
        "#8FC96750", "#68C8C650")) +
  scale_colour_manual(values = c("#AB67AA", "#C66E6A", "#CCC26A",
        "#8FC967", "#68C8C6"))+
  theme_classic() +
  labs(y = "Reproducibility") +
  coord_flip() +
  theme(legend.position = "none", axis.title.y = element_blank())

#Composite plot
grid.arrange(grobs = list(
        total_design_taxa_plot,
        total_methods_taxa_plot,
        total_reproducibility_taxa_plot),
        layout_matrix = matrix(1:3, nrow = 1))
```



Performance scores across years

Linear regression model between the performance score and the publication year.

Statistical model (Table S1)

```
total_performance_year_model <- lm(total ~ Year, data = scores) %>% summary()
total_performance_year_model_table <-
      total_performance_year_model$coefficients[2,c(1,3,4)]

# Render table
total_performance_year_model_table %>% round(4) %>% kable()
```

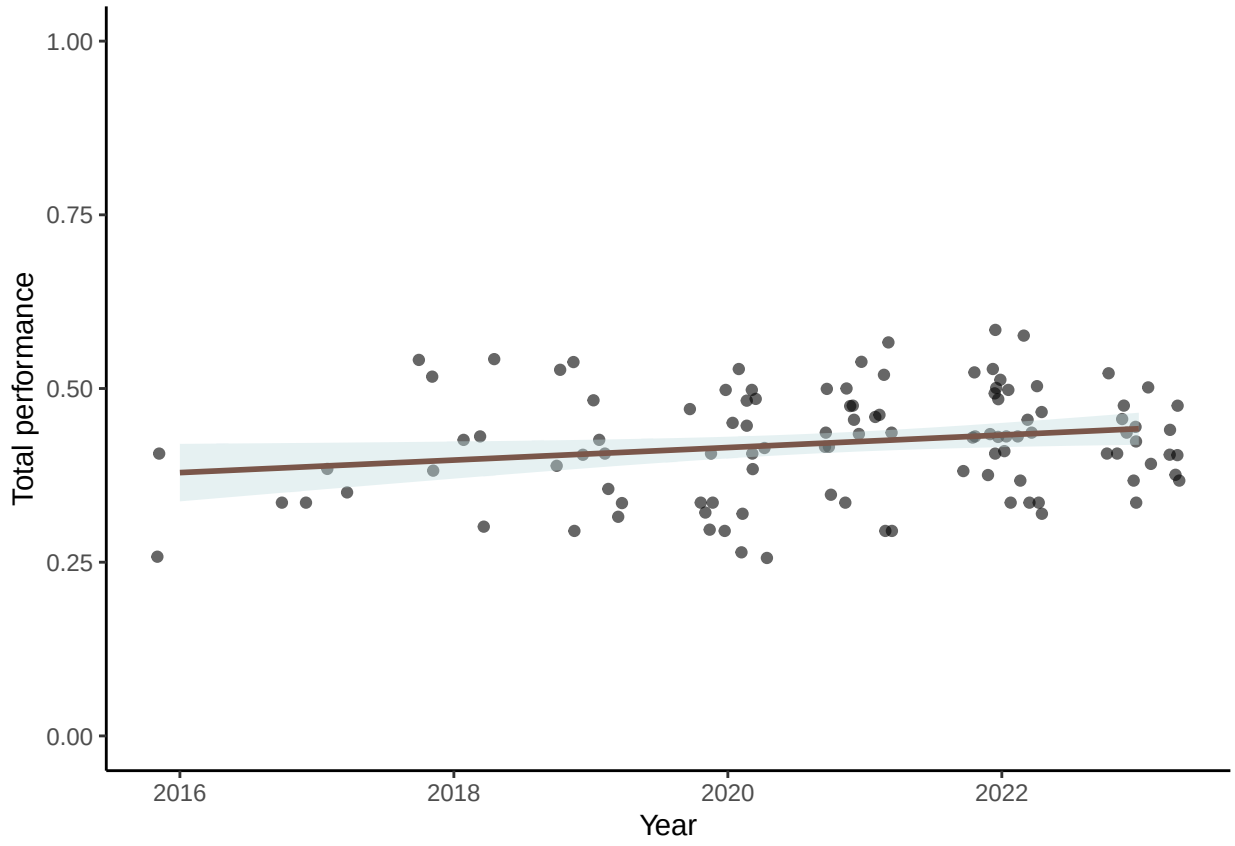
	x
Estimate	0.0090
t value	2.1959
Pr(>|t|)	0.0303

Visualisation (Fig. 2b)

```
scores %>%
  mutate(Year = as.numeric(Year)) %>%
  ggplot(aes(x=Year,y=total)) +
```

```
  geom_jitter(alpha=0.6, width=0.3) +
  ylim(0,1) +
  geom_smooth(method=lm, colour="#7A564A", fill = "#C1DDE040") +
  theme_classic() +
  labs(y = "Total performance", x = "Year")
```

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



Temporal trend of performance domains

Linear regression between the performance domain scores and the publication year.

Statistical model (Table S1)

```
# Experimental design
total_design_year_model <-
      lm(total_design ~ Year, data = scores) %>% summary()

# Methodological resolution
total_methods_year_model <-
      lm(total_methods ~ Year, data = scores) %>% summary()

# Reproducibility
total_reproducibility_year_model <-
      lm(total_reproducibility ~ Year, data = scores) %>% summary()
```

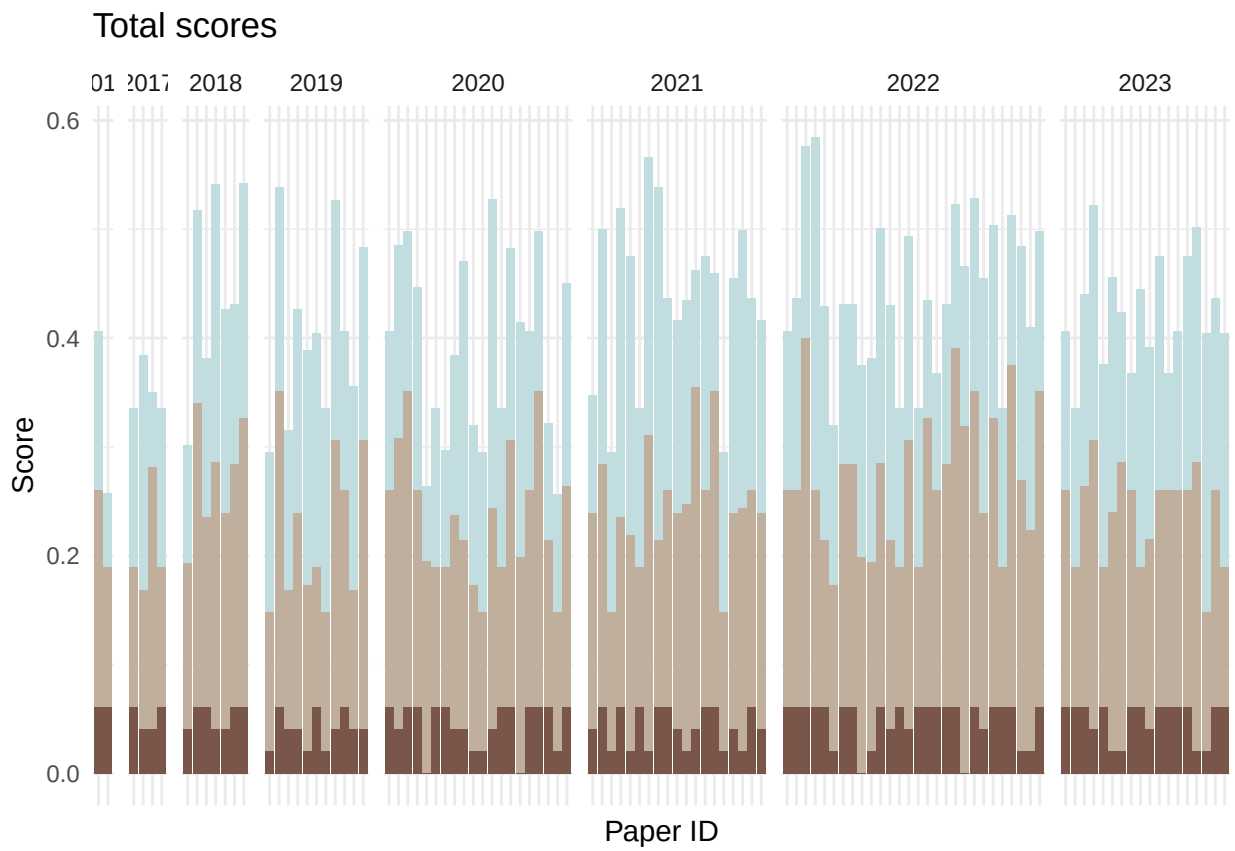
```r
# Composite result table
performance_domains_year_models_table<- rbind(
      total_design_year_model$coefficients[2,c(1,3,4)],
      total_methods_year_model$coefficients[2,c(1,3,4)],
      total_reproducibility_year_model$coefficients[2,c(1,3,4)]
    )
rownames(performance_domains_year_models_table) <-
      c("Experimental design","Methodological resolution","Reproducibility")

# Render table
performance_domains_year_models_table %>% round(4) %>% kable()
```

	Estimate	t value	Pr(>|t|)
Experimental design	0.0117	1.6407	0.1038
Methodological resolution	0.0084	1.4946	0.1380
Reproducibility	0.0004	0.0344	0.9726

Visualisation (Fig. 2a)

Total performance scores of each study with the relative contribution of each domain indicated by a different colour.

```r
scores %>%
  mutate(design = rowSums(across(all_of(design),
        ~ . * weight_consensus[[(cur_column())]]))) %>%
  mutate(methods = rowSums(across(all_of(methods),
        ~ . * weight_consensus[[(cur_column())]]))) %>%
  mutate(reproducibility = rowSums(across(all_of(reproducibility),
        ~ . * weight_consensus[[(cur_column())]]))) %>%
  select(DOI, Year, design, methods, reproducibility) %>%
  pivot_longer(cols = c(design, methods, reproducibility),
        names_to = "Category",
        values_to = "Score") %>%
  mutate(DOI = factor(DOI, levels = unique(DOI))) %>%
  mutate(Year = factor(Year, levels = sort(unique(Year)))) %>%
  ggplot(., aes(x = DOI, y = Score, fill = Category)) +
      geom_bar(stat = "identity") +
      labs(title = "Total scores",
           x = "Paper ID",
           y = "Score") +
      theme_minimal() +
      facet_grid(~ Year, scales = "free", space = "free")+
      theme(axis.text.x = element_blank(),axis.ticks.x=element_blank())+
      scale_fill_manual(values = c(
            "design" = "#C1DDE0",
            "methods" = "#C0AF9D",
            "reproducibility" = "#7A564A")) +
      theme(legend.position = "none")
```

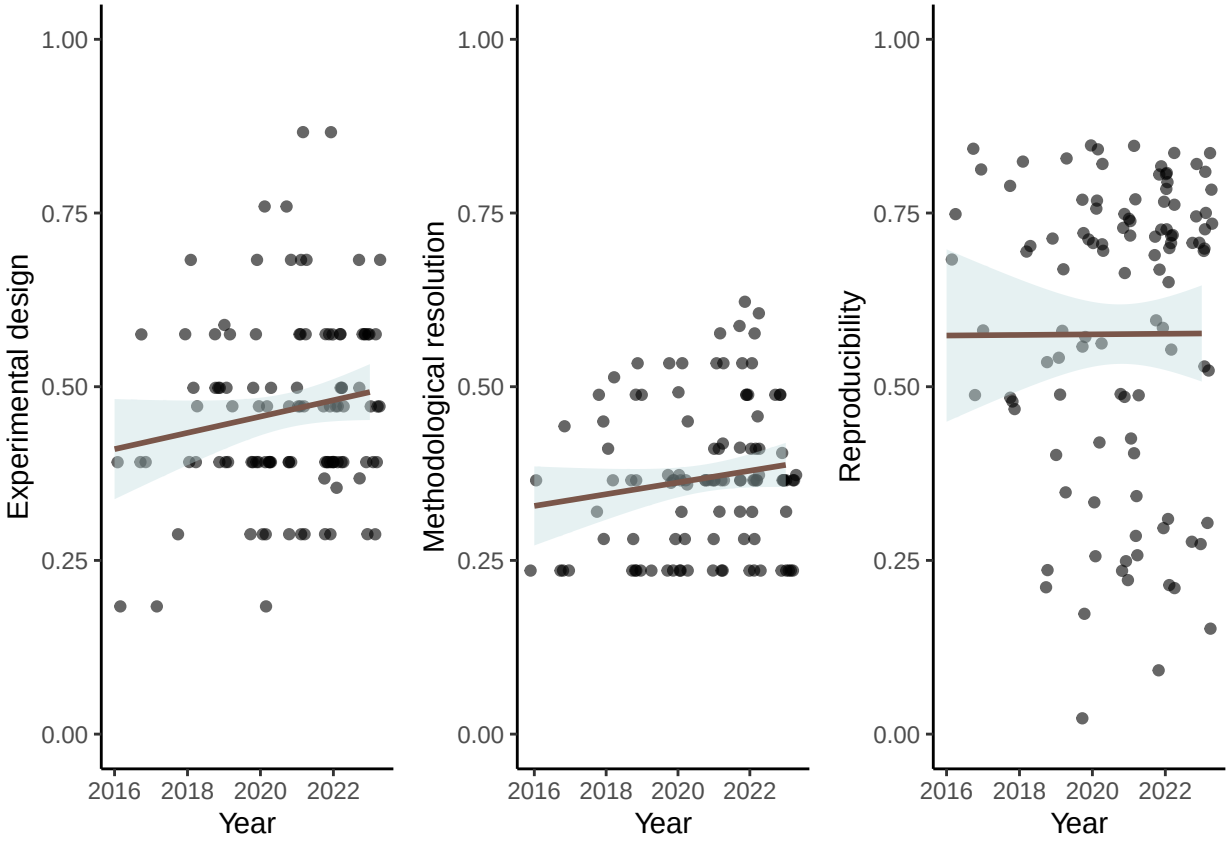


Visualisation (Fig. S3)

Plot the three linear regressions of the domains against the overall performance of the studies.

```
# Experimental design
total_design_year_plot <- scores %>%
  mutate(Year = as.numeric(Year)) %>%
  ggplot(aes(x=Year,y=total_design)) +
  geom_jitter(alpha=0.6, width=0.3) +
  ylim(0,1) +
  geom_smooth(method=lm, colour="#7A564A", fill = "#C1DDE040") +
  theme_classic() +
  labs(y = "Experimental design", x = "Year")

# Methodological resolution
total_methods_year_plot <- scores %>%
  mutate(Year = as.numeric(Year)) %>%
  ggplot(aes(x=Year,y=total_methods)) +
  geom_jitter(alpha=0.6, width=0.3) +
  ylim(0,1) +
  geom_smooth(method=lm, colour="#7A564A", fill = "#C1DDE040") +
  theme_classic() +
  labs(y = "Methodological resolution", x = "Year")

# Reproducibility
total_reproducibility_year_plot <- scores %>%
```

```
  mutate(Year = as.numeric(Year)) %>%
  ggplot(aes(x=Year,y=total_reproducibility)) +
  geom_jitter(alpha=0.6, width=0.3) +
  ylim(0,1) +
  geom_smooth(method=lm, colour="#7A564A", fill = "#C1DDE040") +
  theme_classic() +
  labs(y = "Reproducibility", x = "Year")

#Composite plot
grid.arrange(grobs = list(
      total_design_year_plot,
      total_methods_year_plot,
      total_reproducibility_year_plot),
      layout_matrix = matrix(1:3, nrow = 1))
```



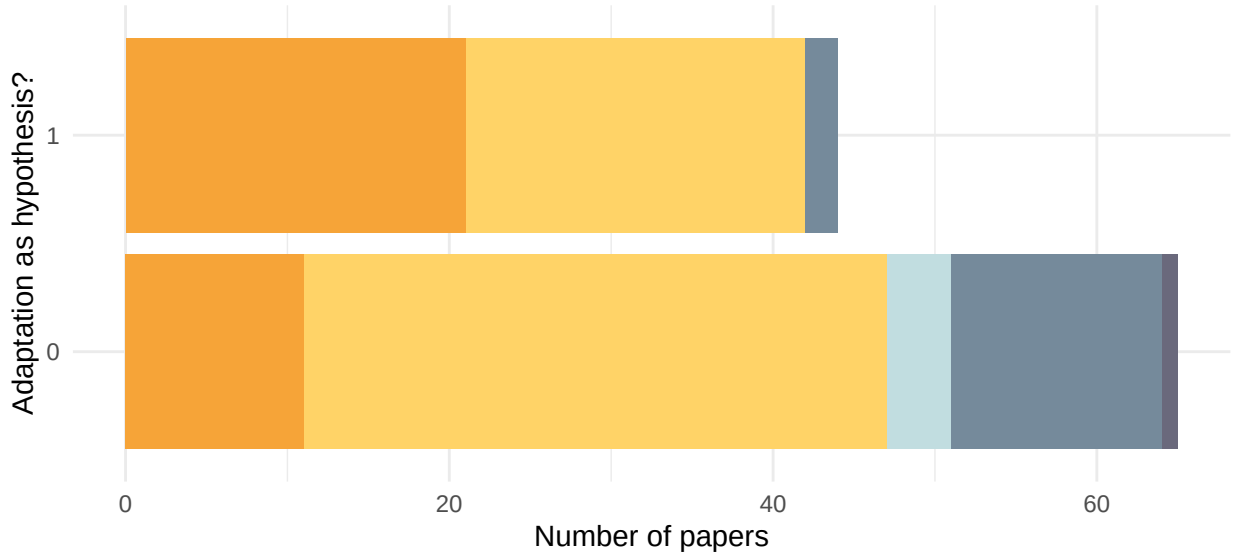
Performance scores vs. conclusiveness level

Conclusiveness scores distribution (Fig. 2d)

Distribution of the papers depending on the conclusiveness level, that is, the interpretation of the results obtained from their results in regards of adaptation microbe-driven adaptation, split between studies that explicitly addressed the increased adaptability provided by microbiomes and those that did not.

```
scores %>%
  select(DOI,Adaptation.means,Conclusiveness) %>%
  mutate(Conclusiveness = factor(Conclusiveness, levels = c(0,1,2,3,4))) %>%
```

```
  mutate(Adaptation.means = factor(Adaptation.means)) %>%
  ggplot(., aes(x = Adaptation.means, fill=Conclusiveness)) +
    geom_bar(position='stack', stat='count') +
    labs(x = "Adaptation as hypothesis?", y = "Number of papers") +
    scale_fill_manual(values = c("#6A697C", "#758A9B", "#C1DDE0", "#FFD367", "#F6A438")) +
    coord_flip() +
    theme_minimal() +
    theme(legend.position = "botton")
```



0. Adaptation only mentioned in the discussion
1. Results not interpreted in an adaptation framework
2. Results interpreted in an adaptation framework
3. Results interpreted as evidence for potential adaptation
4. Results interpreted as evidence for adaptation

Statistical model

```
total_performance_conclusiveness_model <-
        lm(total ~ Conclusiveness, data = scores) %>% summary()
total_performance_conclusiveness_model_table <-
        total_performance_conclusiveness_model$coefficients[2,c(1,3,4)]

# Render table
total_performance_conclusiveness_model_table %>% round(4) %>% kable()
```

	x
Estimate	0.0294
t value	4.2323
Pr(>|t|)	0.0000

Visualisation (Fig. 2e)

```
scores %>%
  mutate(Conclusiveness = as.numeric(Conclusiveness)) %>%
```

```r
ggplot(aes(x=Conclusiveness,y=total)) +
geom_jitter(alpha=0.6, width=0.3) +
ylim(0,1) +
geom_smooth(method=lm, colour="#7A564A", fill = "#C1DDE040") +
theme_classic() +
labs(y = "Performance score", x = "Conclusiveness")
```