

Microbiota adaptation review - 2. Analysis

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

01-11-2023

Contents

Review information	1
Data preparation	1
Overall performance score computations	1
Overall visual statistics	2

Review information

Data preparation

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

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

# Groups of criteria
criteria <- colnames(scores)[c(4:13)]
design <- colnames(scores)[c(4:7)]
methods <- colnames(scores)[c(8:12)]
reproducibility <- colnames(scores)[13]

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

Overall performance score computations

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

```
# Generate weight vectors
weight_consensus <- weights$average
names(weight_consensus) <- criteria
```

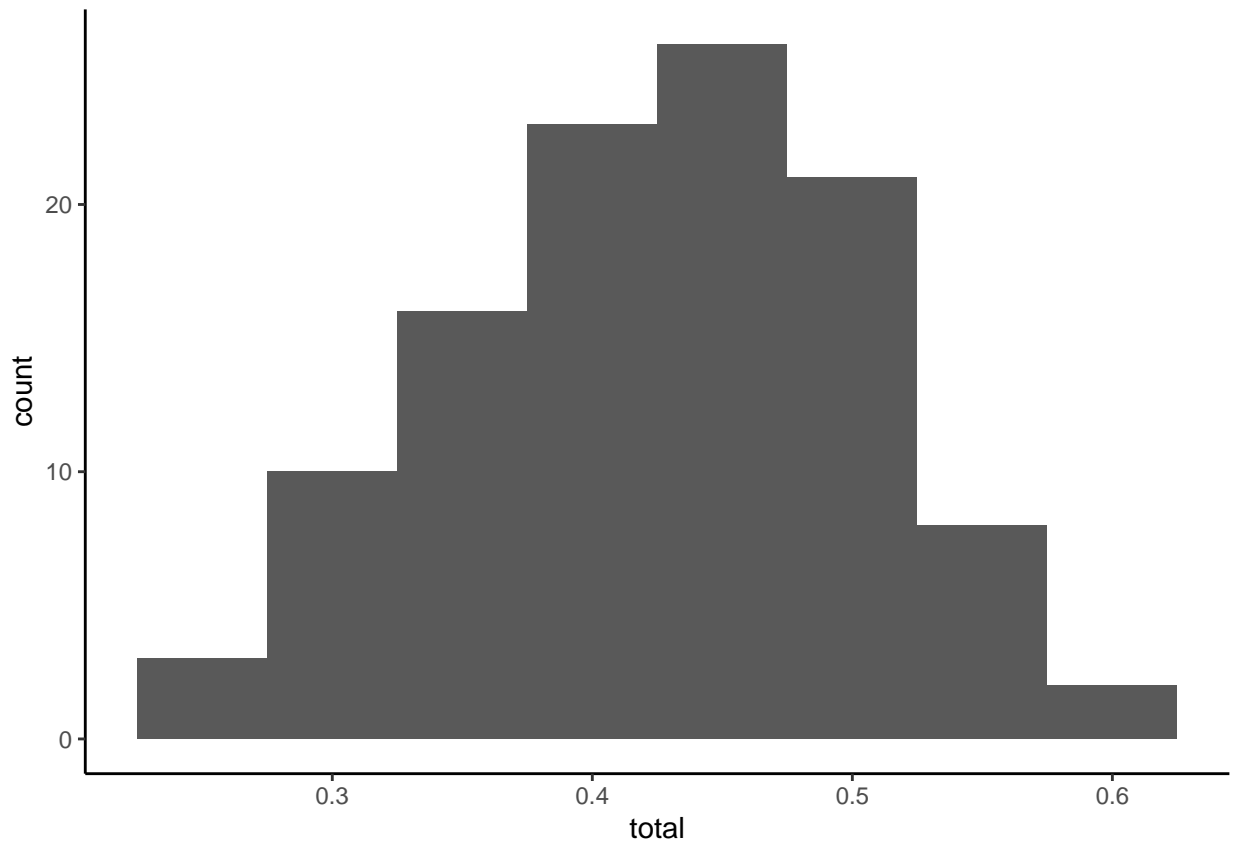
```
weight_consensus_design <- weight_consensus[design]/sum(weight_consensus[design]) # weighed criteria on
weight_consensus_methods <- weight_consensus[methods]/sum(weight_consensus[methods]) # weighed criteria
weight_consensus_reproducibility <- weight_consensus[reproducibility]/sum(weight_consensus[reproducibil

# Calculate weighed averages
scores <- scores %>%
  mutate(total_design = rowSums(across(all_of(design), ~ . * weight_consensus_design[[cur_column()]]))
  mutate(total_methods = rowSums(across(all_of(methods), ~ . * weight_consensus_methods[[cur_column()]]))
  mutate(total_reproducibility = rowSums(across(all_of(reproducibility), ~ . * weight_consensus_reprodu
  mutate(total = rowSums(across(all_of(criteria), ~ . * weight_consensus[[cur_column()]])))
```

Overall visual statistics

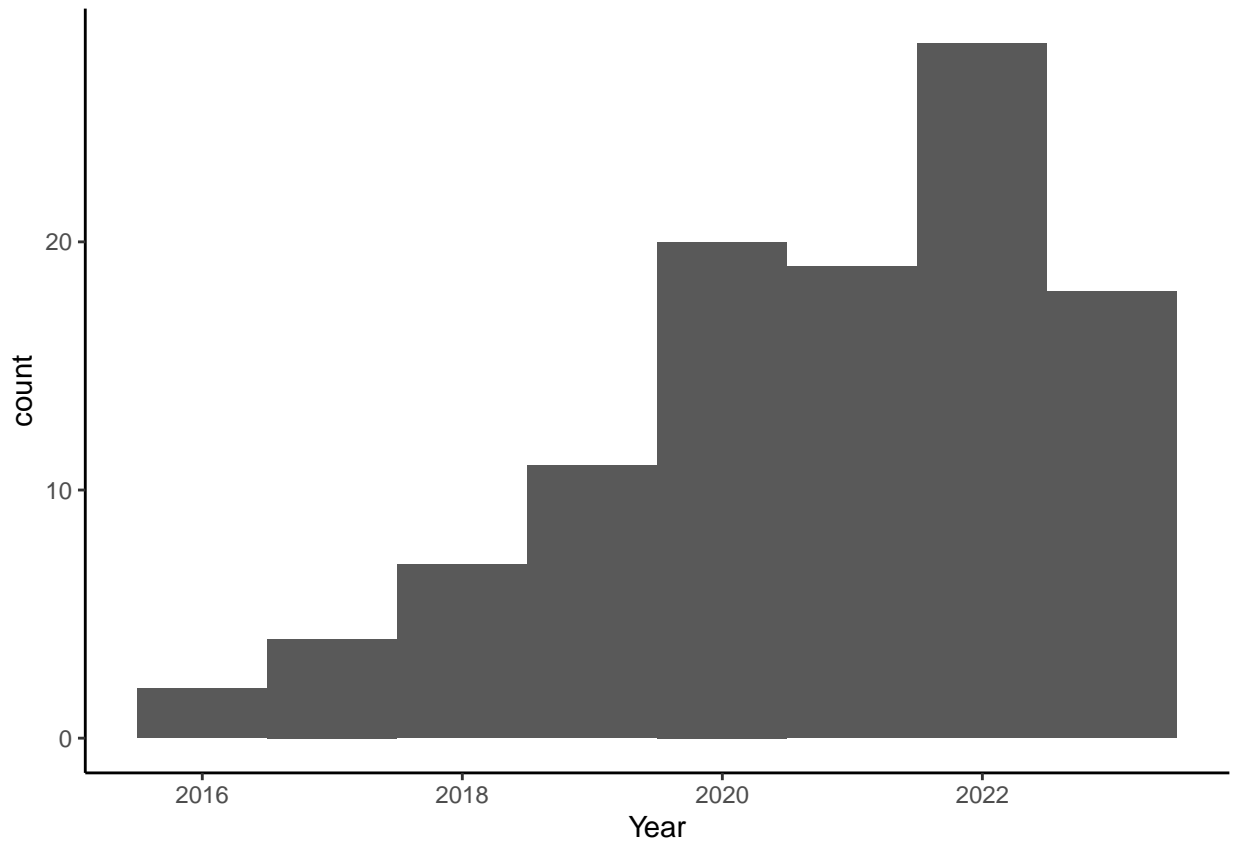
Studies per total score Distribution of total scores of the 109 papers analysed

```
scores %>%
  ggplot(aes(x=total)) +
  geom_histogram(binwidth = 0.05,) +
  theme_classic()
```



Studies per year The 109 papers analysed distributed per year

```
scores %>%
  ggplot(aes(x=Year)) +
  geom_histogram(binwidth = 1) +
  theme_classic()
```



```
scores %>%
  mutate(Year = factor(Year)) %>%
  separate_rows(Continent, sep = ", ") %>%
  ggplot(aes(x = Continent, y = Total, fill = Continent)) +
  geom_violin() +
  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("#E69F00", "#AA3377", "#228833", "#66CCEE", "#CCBB44", "#EE6677", "#4477AA", "#444444", "#CCBB44", "#EE6677", "#4477AA", "#444444")) +
  scale_colour_manual(values = c("#E69F00", "#AA3377", "#228833", "#66CCEE", "#CCBB44", "#EE6677", "#4477AA", "#444444", "#CCBB44", "#EE6677", "#4477AA", "#444444")) +
  theme_classic() +
  labs(title = "Total Score Distributions by Continent",
       y = "Total Score",
       x = "Continent") +
  coord_flip()
```

Total scores distribution in each continent (CONTINENT DATA ARE MISSING!)

```
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")))
```

Performance scores per animal taxa

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

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

Statistical model

	x
Kruskal-Wallis chi-squared	2.8548905
df	4.0000000
pvalue	0.5823953

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

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

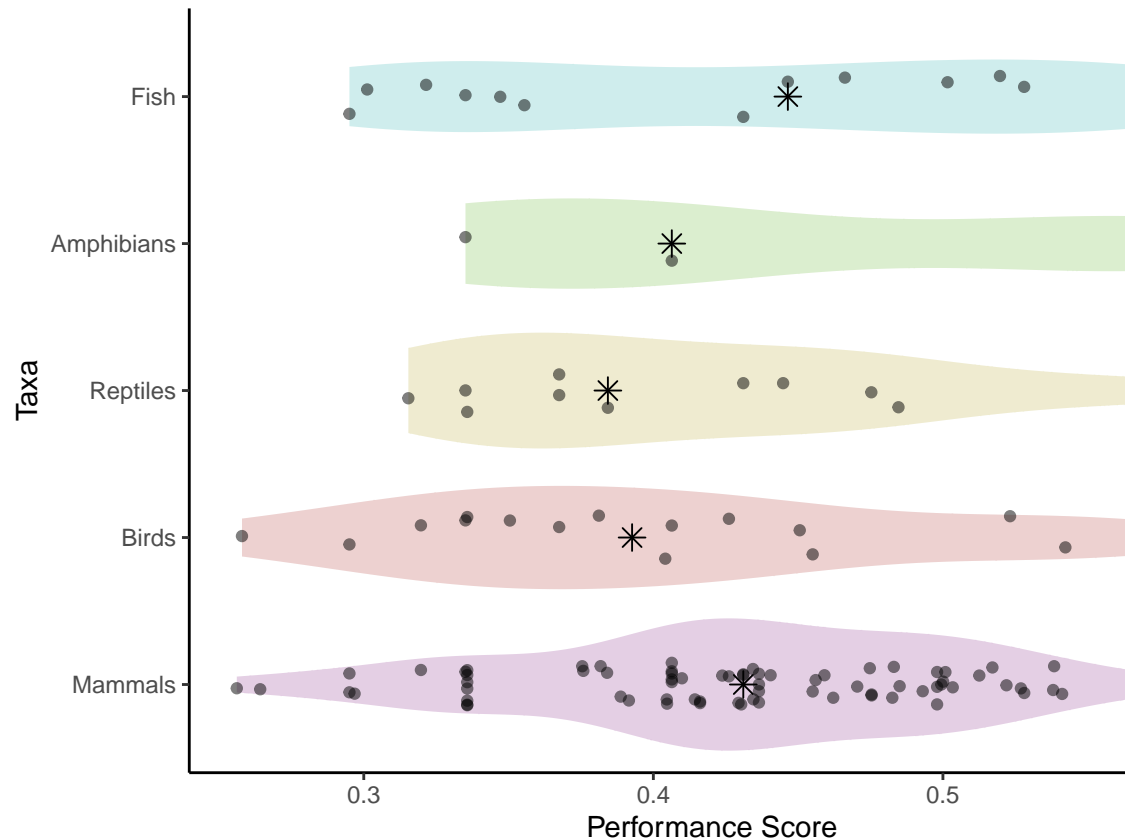
Comparison	Z	P.unadj	P.adj
Amphibians - Birds	0.4886043	0.6251219	1.0000000
Amphibians - Fish	-0.2389318	0.8111585	1.0000000
Birds - Fish	-1.2758029	0.2020252	1.0000000
Amphibians - Mammals	-0.1427120	0.8865177	0.8865177
Birds - Mammals	-1.4181864	0.1561363	1.0000000
Fish - Mammals	0.2364933	0.8130499	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

```
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")

```



Visualisation (Fig. 2c)

```

# 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())

```

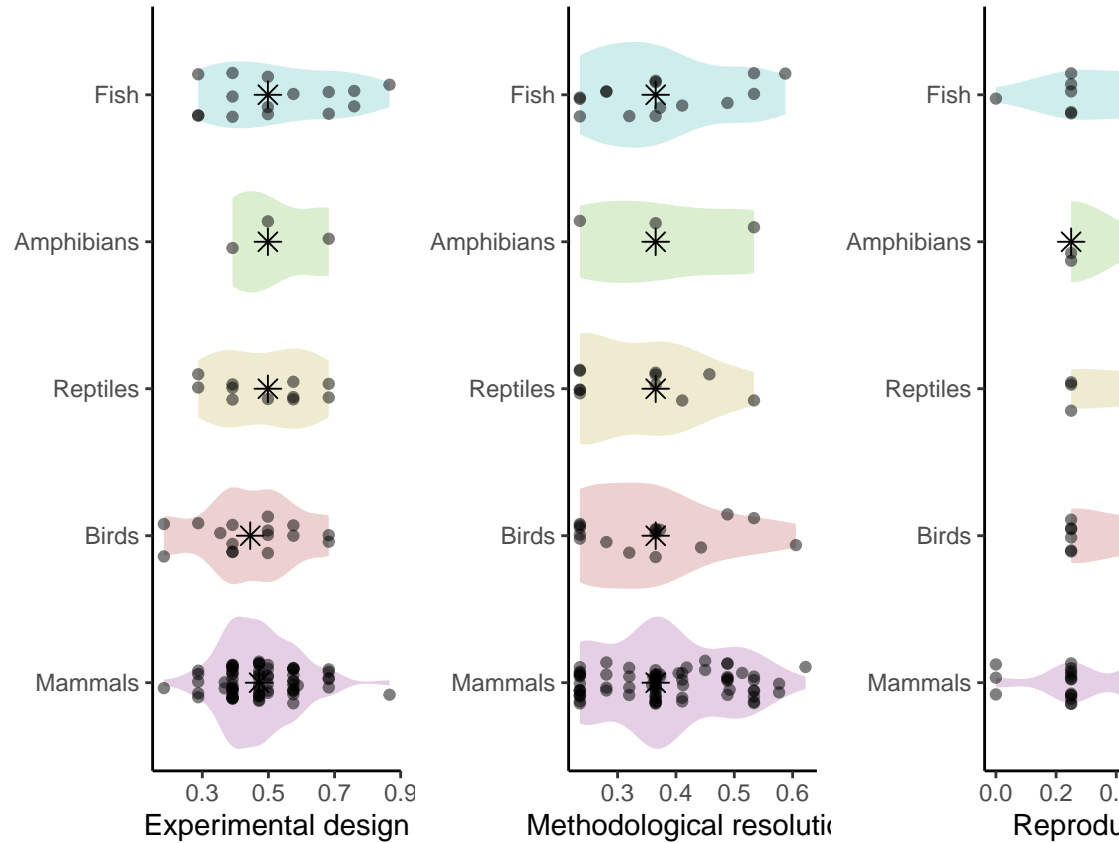
```

# 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() +
  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))

```



Visualisation (Fig. Sx)

Temporal trend of total performance scores

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

```
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)] %>% round(2)

# Render table
kable(total_performance_year_model_table)
```

Statistical model

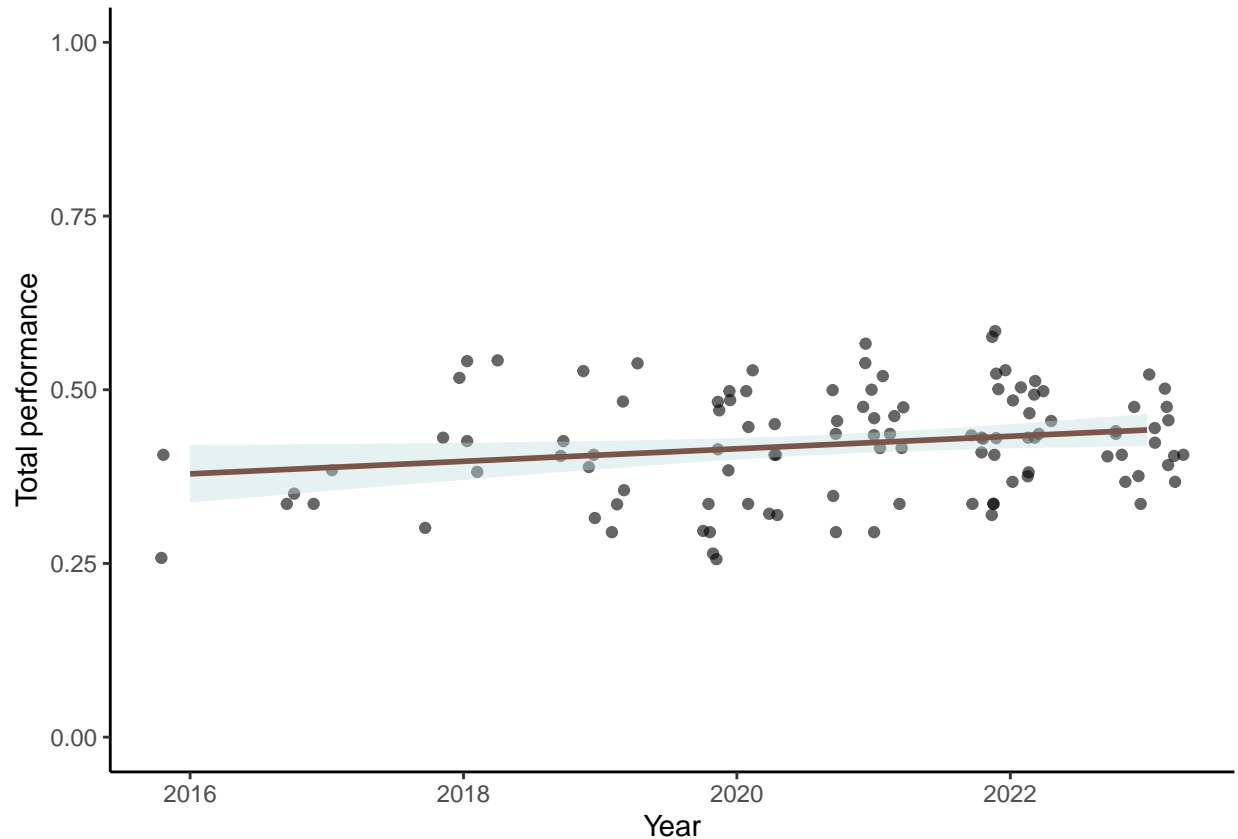
	x
Estimate	0.0090
t value	2.1959
Pr(> t)	0.0303

Visualisation (Fig. 2b) Linear regression between the performance score and the publication year.

```
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.

```
# 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()

# 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)]
) %>% round(.,4)
```



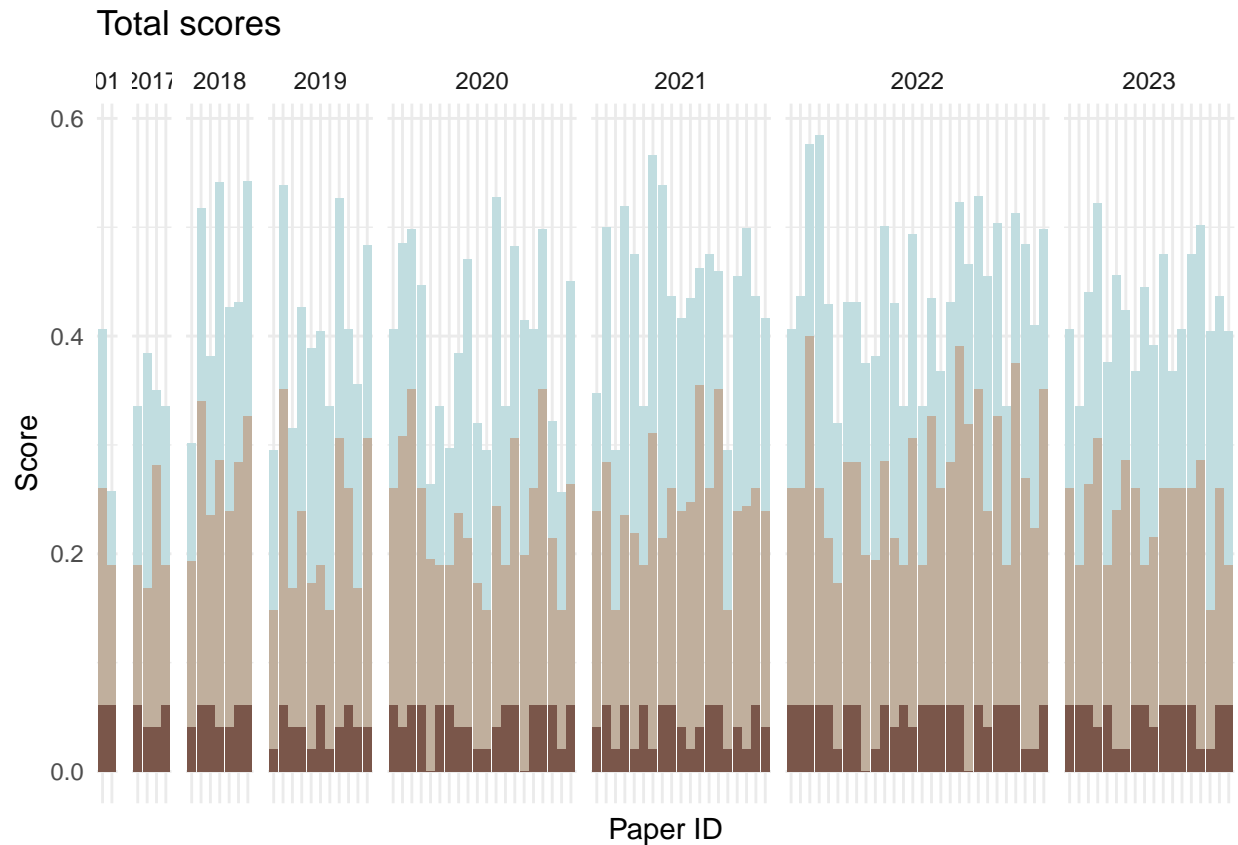
```
rownames(performance_domains_year_models_table) <- c("Experimental design", "Methodological resolution",
# Render table
kable(performance_domains_year_models_table)
```

Statistical model

	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.

```
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" = "#COAF9D", "reproducibility" = "#7
    theme(legend.position = "none")
```



```
# 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))

```

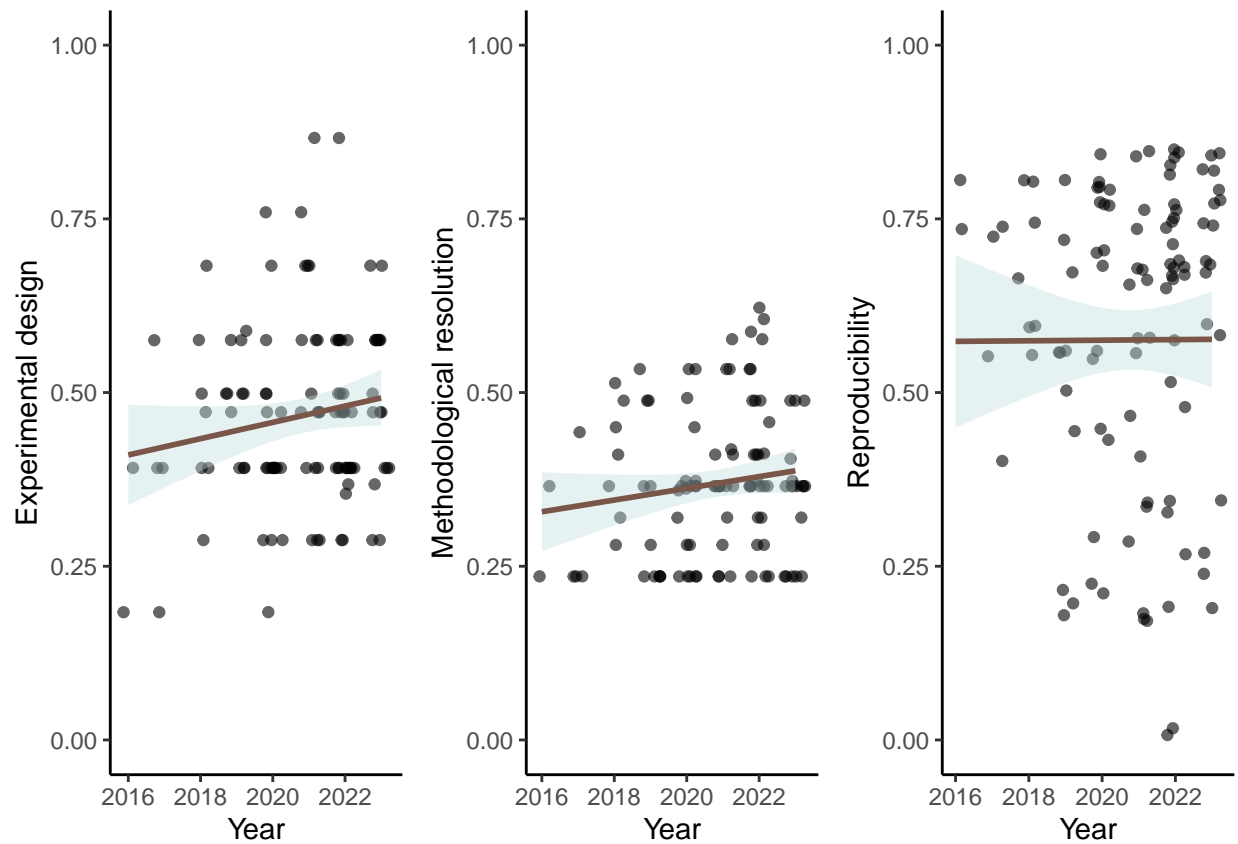
Visualisation (Fig. Sx)

```

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

## Warning: Removed 2 rows containing missing values (`geom_point()`).

```



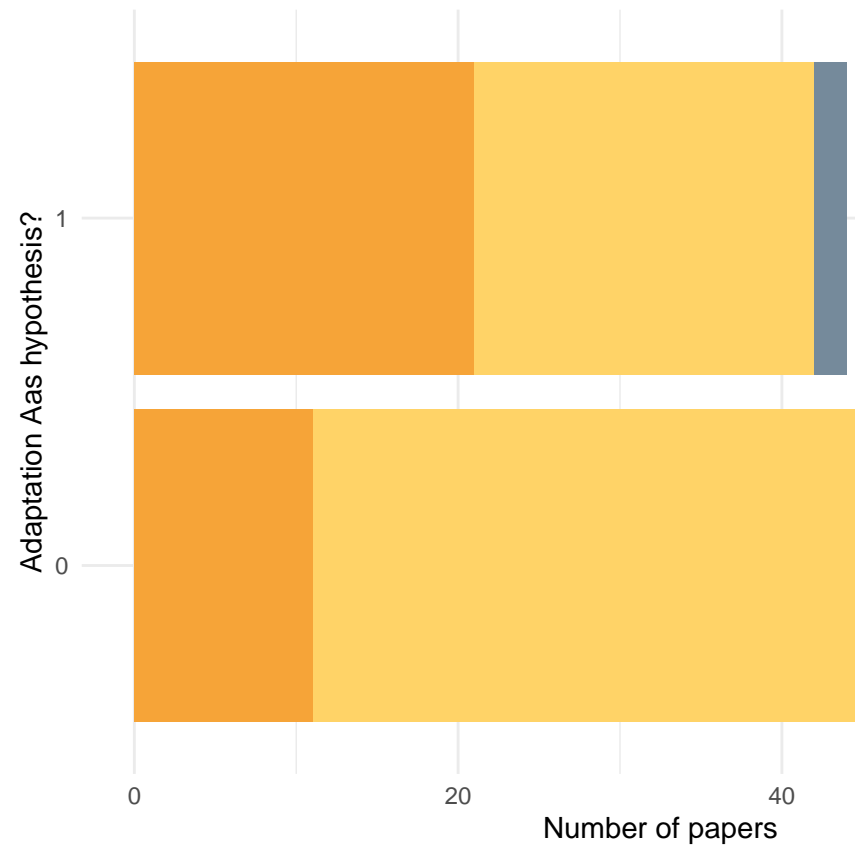
Comparison between performance and conclusiveness level

```

scores %>%
  select(DOI,Adaptation.means,Conclusiveness) %>%
  mutate(Conclusiveness = factor(Conclusiveness, levels = c(0,1,2,3,4))) %>%
  mutate(Adaptation.means = factor(Adaptation.means)) %>%
  #pivot_longer(cols = c(Adaptation.means,Conclusiveness), names_to = "Category", values_to = "Score")
  ggplot(., aes(x = Adaptation.means, fill=Conclusiveness)) +
  geom_bar(position='stack', stat='count') +

```

```
labs(x = "Adaptation Aas hypothesis?", y = "Number of papers") +
scale_fill_manual(values = c("#6A697C", "#758A9B", "#C1DDE0", "#FFD367", "#F6A438")) +
coord_flip() +
theme_minimal() +
theme(legend.position = "none")
```



Conclusiveness scores distribution (Fig. 2d)

```
total_performance_conclusiveness_model <- lm(total ~ Conclusiveness, data = scores) %>% summary()
total_performance_conclusiveness_model_table <- total_performance_conclusiveness_model$coefficients[2,c
# Render table
kable(total_performance_conclusiveness_model_table)
```

Statistical model

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

```
scores %>%
mutate(Conclusiveness = as.numeric(Conclusiveness)) %>%
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 = "Reproducibility", x = "Conclusiveness")
```

Visualisation (Fig. 2d)

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

