

Coexistence crayfish/amphibians #6: Niche comparison

Nadege Belouard*

10/08/2020

Contents

1. Setup	1
2. Compare crayfish and amphibians niches	2
A. Agile frog/All crayfish (n = 9)	2
Prepare the dataset and show mean values	2
Wilcoxon tests	2
Paired permutation tests	3
B. Tree frog/All crayfish (n = 4)	5
Prepare the dataset and show mean values	5
Wilcoxon tests	5
Paired permutation tests	5
C. Palmate newt/All crayfish (n = 5)	6
Prepare the dataset and show mean values	6
Wilcoxon tests	6
Paired permutation tests	6
D. Marbled newt/All crayfish (n = 0)	7

1. Setup

This vignette compares niche metrics between the different study species in ponds We begin by loading the required libraries and datasets.

```
library(tidyr)
library(dplyr)
library(magrittr)
library(reshape2)
library(here)
library(PairedData)
```

*UMR CNRS EcoBio, UMR INRAE DECOD, nadege.belouard@gmail.com

```
Niche_metrics <- read.csv(file.path(here(), "exported_data", "Population_metrics_complete.csv"), h = T)
Mean_position <- read.csv(file.path(here(), "exported_data", "Meanposition_SEAc_complete.csv"), header = T)
```

We merge both datasets (one contains the Layman metrics and the other the SEAc).

```
# Rename a column in one dataset so it is common with the other dataset
Mean_position %<>% rename(Code = Pop)
Pop_metrics <- merge(Niche_metrics, Mean_position, by = c("Pond", "Code"))
```

2. Compare crayfish and amphibians niches

A. Agile frog/All crayfish (n = 9)

Prepare the dataset and show mean values

```
metrics_crayfish <- Pop_metrics %>% filter(TaxaUsed == "Crayfish")
metrics_agile <- Pop_metrics %>% filter(Species == "Agile frog")

metrics <- rbind(metrics_crayfish, metrics_agile)
metrics %<>% filter(Pond %in% c(1016, 111, 112, 66, 72, 888, 89, 91, 949))

metrics %>% group_by(TaxaUsed) %>% summarise(
  meanSEAc = mean(SEAc),
  meanNR = mean(NRcp)
)
```

```
## # A tibble: 2 x 3
##   TaxaUsed   meanSEAc meanNR
##   <chr>       <dbl>   <dbl>
## 1 Agile frog  0.0239  0.281
## 2 Crayfish   0.0535  0.436
```

Wilcoxon tests

```
wilcox.test(SEAc~TaxaUsed,paired=TRUE, data = metrics)
```

```
##
## Wilcoxon signed rank exact test
##
## data: SEAc by TaxaUsed
## V = 1, p-value = 0.007812
## alternative hypothesis: true location shift is not equal to 0
```

```
# V = 1, p-value = 0.007812 **
```

```
wilcox.test(NRcp~TaxaUsed,paired=TRUE, data = metrics)
```

```
##
## Wilcoxon signed rank exact test
##
## data: NRcp by TaxaUsed
## V = 5, p-value = 0.03906
## alternative hypothesis: true location shift is not equal to 0

# V = 5, p-value = 0.03906 *
```

Paired permutation tests

```
metrics %<>% dplyr::select(Pond, NRcp, SEAc, TaxaUsed)
metrics_pivoted <- pivot_wider(metrics, names_from = TaxaUsed, values_from = c(NRcp, SEAc))

# Test par permutation sur la différence de moyenne :
# Calculate the difference between the two species
metrics_pivoted %<>% mutate(
  diffNR = NRcp_Crayfish - `NRcp_Agile frog`,
  diffSEAc = SEAc_Crayfish - `SEAc_Agile frog`
)

# set the number of permutations
sim <- 10^4 #number of permutations

#create an empty vector to store the simulated averages
permNR <- rep(NA, sim)
permSEAc <- rep(NA, sim)

k <- nrow(metrics_pivoted) # number of individuals to sample/permute
set.seed(2022) # make sure the results are going to be the same every time we run the code

# permute data
for (i in 1:sim) { #in each simulation
  # randomly select some pairs to have their difference multiplied by negative to indicate the other sp
  tirage = sample(c(-1,1), size = k, replace = T)
  permSEAc[i] = mean(metrics_pivoted$diffSEAc * tirage)
  # append the average in the vector
  permNR[i] = mean(metrics_pivoted$diffNR * tirage)
}

#add the observed difference as the final item in the vector
permNR[10000] = mean(metrics_pivoted$diffNR)
permSEAc[10000] = mean(metrics_pivoted$diffSEAc)

# Table disjonctive qui compte le nombre de valeurs dans perm qui sont superieures ou egales
# a la difference observee entre les moyennes, puis calcule la valeur de p

tablepropNR = table(abs(permNR) >= abs(mean(metrics_pivoted$diffNR)))
pNR = tablepropNR[2]/10000
pNR #p =0.0419
```

```
## TRUE  
## 0.0419
```

```
tablepropSEAc = table(abs(permSEAc) >= abs(mean(metrics_pivoted$diffSEAc)))  
pSEAc = tablepropSEAc[2]/10000  
pSEAc #  $p = 0.008$ 
```

```
## TRUE  
## 0.008
```

B. Tree frog/All crayfish ($n = 4$)

Prepare the dataset and show mean values

```
## # A tibble: 2 x 3
##   TaxaUsed meanSEAc meanNR
##   <chr>      <dbl> <dbl>
## 1 Crayfish    0.0411  0.369
## 2 Tree frog   0.0400  0.484
```

Wilcoxon tests

```
##
## Wilcoxon signed rank exact test
##
## data: SEAc by TaxaUsed
## V = 4, p-value = 0.875
## alternative hypothesis: true location shift is not equal to 0
```

```
##
## Wilcoxon signed rank exact test
##
## data: NRcp by TaxaUsed
## V = 2, p-value = 0.375
## alternative hypothesis: true location shift is not equal to 0
```

Paired permutation tests

```
## [1] "p-value for NR"
```

```
## TRUE
## 0.3816
```

```
## [1] "p-value for SEAc"
```

```
## <NA>
## NA
```

```
##
## TRUE
## 10000
```

```
## [1] "All simulations are TRUE, so  $p > 0.9999$ , no difference"
```

C. Palmate newt/All crayfish ($n = 5$)

Prepare the dataset and show mean values

```
## # A tibble: 2 x 3
##   TaxaUsed      meanSEAc meanNR
##   <chr>         <dbl>   <dbl>
## 1 Crayfish      0.06    0.52
## 2 Palmate newt  0.03    0.4
```

Wilcoxon tests

```
##
## Wilcoxon signed rank exact test
##
## data: SEAc by TaxaUsed
## V = 15, p-value = 0.0625
## alternative hypothesis: true location shift is not equal to 0
```

```
##
## Wilcoxon signed rank exact test
##
## data: NRcp by TaxaUsed
## V = 10, p-value = 0.625
## alternative hypothesis: true location shift is not equal to 0
```

Paired permutation tests

```
## [1] "p-value for NR"
```

```
## TRUE
## 0.4966
```

```
## [1] "p-value for SEAc"
```

```
## TRUE
## 0.0652
```

D. Marbled newt/All crayfish ($n = 0$)

No stats