Calculation of diversity indices

Luciano L.M. De Benedictis

2026-01-07

```
# setup ------------------------------------------------------------------

library(tidyverse)
```

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.4     v readr     2.1.6
## v forcats   1.0.1     v stringr   1.6.0
## v ggplot2   4.0.1     v tibble    3.3.0
## v lubridate 1.9.4     v tidyr     1.3.2
## v purrr     1.2.0
## -- Conflicts ---------------------------------------- tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
```

```
library(GET)
library(vegan)
```

```
## Loading required package: permute
```

```
resampled <- readRDS("transect_resampled.rds")
speciestraits <- readRDS("traits_imputation.rds")
source("0.functions.R")

indices <- vector(mode = "list")
indexcurves <- vector(mode = "list")

# transform traits ------------------------------------------------------------

speciestraits |>
  select(-n) |>
  summarize(across(where(is.numeric),  ~ boxcox(., check = T))) |>
  pivot_longer(everything(), names_to = "trait", values_to = "lambda") |>
  arrange(lambda, trait)
```

```
## Box-Cox transform with lambda = 0.1
## Box-Cox transform with lambda = -0.5
## Box-Cox transform with lambda = -0.4
## Box-Cox transform with lambda = 0.3
## Box-Cox transform with lambda = 0.2
```

```
## Box-Cox transform with lambda = 0
## Box-Cox transform with lambda = 0.8
## Box-Cox transform with lambda = 1.8
## Box-Cox transform with lambda = -0.2
## Box-Cox transform with lambda = 0

## # A tibble: 10 x 2
##    trait               lambda
##    <chr>                <dbl>
##  1 nmass_mg_g          -0.5
##  2 lma_g_m2            -0.4
##  3 offspring           -0.2
##  4 spread               0
##  5 ssd_combined_mg_mm3  0
##  6 leaf_area_mm2        0.100
##  7 diaspore_mass_mg     0.200
##  8 plant_height_m       0.300
##  9 BBRsize              0.8
## 10 persistence          1.8
```

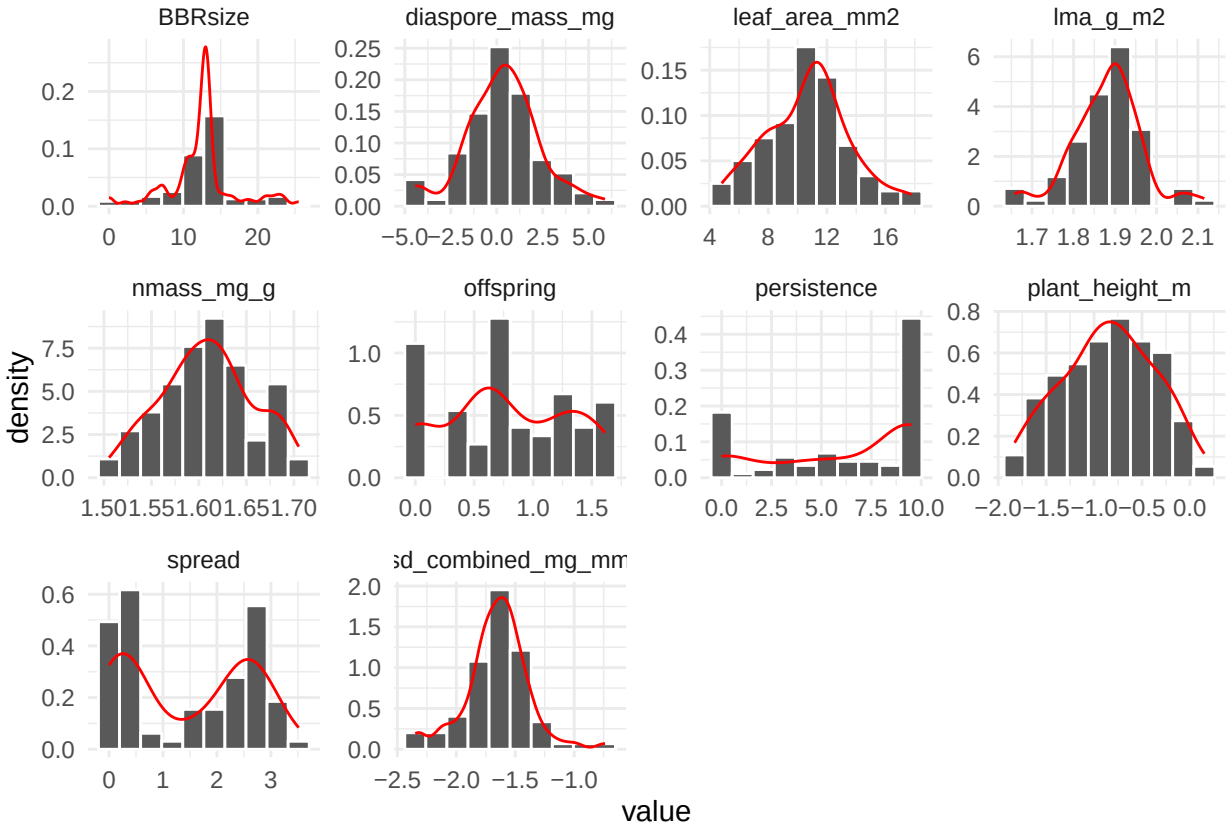
This is a dry run check. The traits will be transformed as such:

- ~ 1/sqrt N mass, LMA, offspring
- ~ log spread, SSD, leaf area, diaspore mass
- ~ sqrt height
- ~ identity BBR size
- ~ square persistence

```
data <- speciestraits |>
  select(species, where(is.numeric)) |>
  select(-n) |>
  mutate(across(where(is.numeric),  ~ boxcox(.))) |>
  as.data.frame()
```

```
## Box-Cox transform with lambda = 0.1

## Box-Cox transform with lambda = -0.5

## Box-Cox transform with lambda = -0.4

## Box-Cox transform with lambda = 0.3

## Box-Cox transform with lambda = 0.2

## Box-Cox transform with lambda = 0

## Box-Cox transform with lambda = 0.8

## Box-Cox transform with lambda = 1.8

## Box-Cox transform with lambda = -0.2

## Box-Cox transform with lambda = 0
```

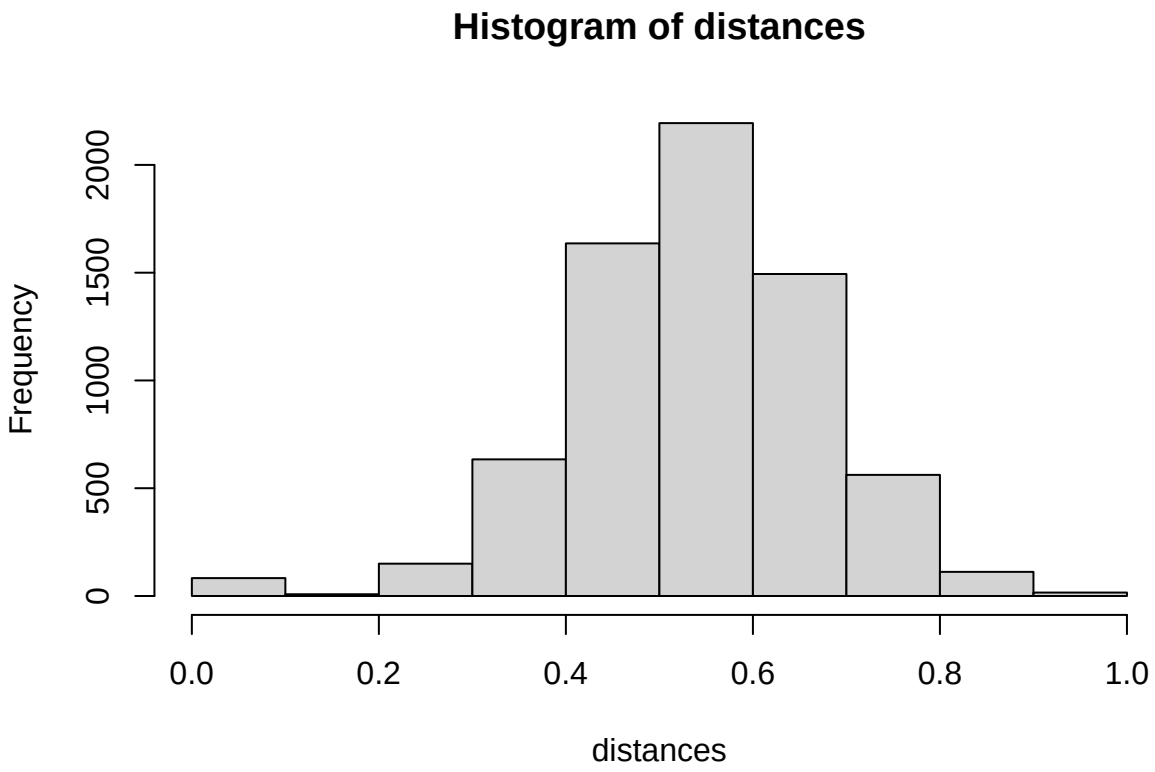
```
# check trait distributions

data |>
  pivot_longer(-species, names_to = "trait", values_to = "value") |>
  histdensity(value)+
  facet_wrap(~ trait, scales = "free")+
  theme_minimal()
```




```
# overall ----------------------------------------------------------------

## calculate distances ----------------------------------------------------

distances <- vegdist(data[-1], method = "mahalanobis", diag = T, upper = T) |>
  as.matrix()

#scaling to 0-1

distances <- distances / max(distances)

#double checking the squared euclidean property

ade4::is.euclid(as.dist(sqrt(distances)))
```

```
## [1] TRUE
```

```r
#check distance distribution

hist(distances)
```




```r
#pairs with distance 0

as.data.frame(as.table(distances)) |>
  filter(Freq == 0 & Var1 != Var2)
```

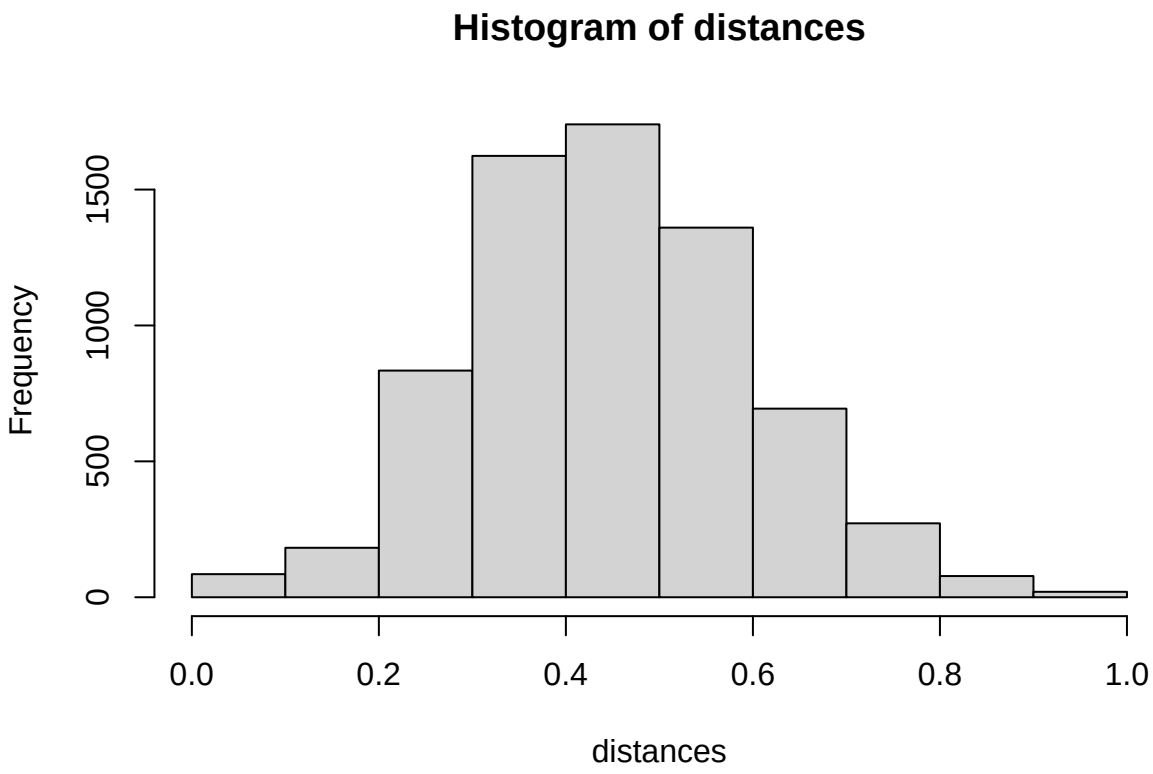
```
## [1] Var1 Var2 Freq
## <0 rows> (or 0-length row.names)
```

```r
## calculate indices ----------------------------------------------------

species <- data$species

raotrait <- lapply(resampled, function (x)
  lapply(x, function (y) qdecomp(y, distances, species)) |>
    bind_rows(.id = "step") |>
    mutate(step = as.numeric(step))
) |>
  bind_rows(.id = "site")

#scale steps in units of area
```

```r
raotrait <- raotrait |>
  mutate(step = step / 100)

#add to list
indices[[1]] <- raotrait
names(indices)[1] <- "overall"


## create curve set ------------------------------------------------------

curvesets <- vector(mode = "list", length = sum(map_lgl(raotrait, is.numeric))-1)

for (i in seq_along(curvesets)){
  column <- colnames(raotrait)[i+2]
  curves <- raotrait |>
    select(site, step, {{column}}) |>
    pivot_wider(names_from = site, values_from = 3) |>
    filter(if_all(where(is.numeric), ~ !is.na(.)))
  curvesets[[i]] <- curve_set(obs = as.data.frame(curves[-1]), r = curves[[1]])
  names(curvesets)[i] <- column
}

indexcurves[[1]] <- curvesets
names(indexcurves)[1] <- "overall"


# aboveground ----------------------------------------------------------


## calculate distances ---------------------------------------------------

distances <- vegdist(data[2:7], method = "mahalanobis", diag = T, upper = T) |>
  as.matrix()

#scaling to 0-1

distances <- distances / max(distances)

#double checking the squared euclidean property

ade4::is.euclid(as.dist(sqrt(distances)))
```

```
## [1] TRUE
```

```r
#check distance distribution

hist(distances)
```




```
#pairs with distance 0

as.data.frame(as.table(distances)) |>
  filter(Freq == 0 & Var1 != Var2)
```

```
## [1] Var1 Var2 Freq
## <0 rows> (or 0-length row.names)
```

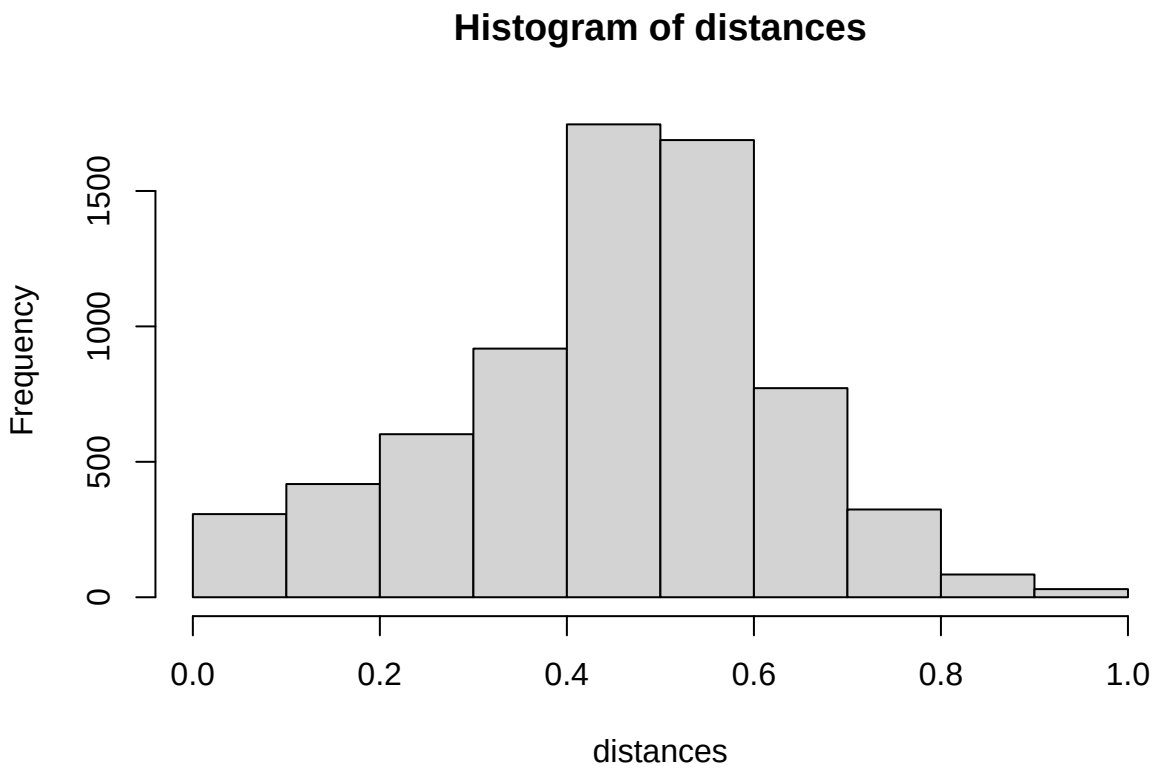
```
## calculate indices ------------------------------------------------------

raotrait <- lapply(resampled, function (x)
  lapply(x, function (y) qdecomp(y, distances, species)) |>
    bind_rows(.id = "step") |>
    mutate(step = as.numeric(step))
) |>
  bind_rows(.id = "site")

#scale steps in units of area

raotrait <- raotrait |>
  mutate(step = step / 100)

#add to list

indices[[2]] <- raotrait
```

```r
names(indices)[2] <- "aboveground"


## create curve set ------------------------------------------------------

curvesets <- vector(mode = "list", length = sum(map_lgl(raotrait, is.numeric))-1)

for (i in seq_along(curvesets)){
  column <- colnames(raotrait)[i+2]
  curves <- raotrait |>
    select(site, step, {{column}}) |>
    pivot_wider(names_from = site, values_from = 3) |>
    filter(if_all(where(is.numeric), ~ !is.na(.)))
  curvesets[[i]] <- curve_set(obs = as.data.frame(curves[-1]), r = curves[[1]])
  names(curvesets)[i] <- column
}

indexcurves[[2]] <- curvesets
names(indexcurves)[2] <- "aboveground"

# clonal ----------------------------------------------------------------

## calculate distances ---------------------------------------------------

distances <- vegdist(data[8:11], method = "mahalanobis", diag = T, upper = T) |>
  as.matrix()

#scaling to 0-1

distances <- distances / max(distances)

#double checking the squared euclidean property

ade4::is.euclid(as.dist(sqrt(distances)))
```

```
## Warning in ade4::is.euclid(as.dist(sqrt(distances))): Zero distance(s)
```

```
## [1] TRUE
```

```r
#check distance distribution

hist(distances)
```




```
#pairs with distance 0

as.data.frame(as.table(distances)) |>
  filter(Freq == 0 & Var1 != Var2)
```

```
##    Var1 Var2 Freq
## 1    10    1    0
## 2    29    1    0
## 3    44    1    0
## 4    49    1    0
## 5    67    1    0
## 6     1   10    0
## 7    29   10    0
## 8    44   10    0
## 9    49   10    0
## 10   67   10    0
## 11   19   15    0
## 12   37   15    0
## 13   34   17    0
## 14   75   17    0
## 15   15   19    0
## 16   37   19    0
## 17   48   28    0
## 18   66   28    0
## 19    1   29    0
```

```
## 20   10   29    0
## 21   44   29    0
## 22   49   29    0
## 23   67   29    0
## 24   35   31    0
## 25   36   31    0
## 26   40   31    0
## 27   77   31    0
## 28   17   34    0
## 29   75   34    0
## 30   31   35    0
## 31   36   35    0
## 32   40   35    0
## 33   77   35    0
## 34   31   36    0
## 35   35   36    0
## 36   40   36    0
## 37   77   36    0
## 38   15   37    0
## 39   19   37    0
## 40   61   39    0
## 41   31   40    0
## 42   35   40    0
## 43   36   40    0
## 44   77   40    0
## 45    1   44    0
## 46   10   44    0
## 47   29   44    0
## 48   49   44    0
## 49   67   44    0
## 50   28   48    0
## 51   66   48    0
## 52    1   49    0
## 53   10   49    0
## 54   29   49    0
## 55   44   49    0
## 56   67   49    0
## 57   39   61    0
## 58   28   66    0
## 59   48   66    0
## 60    1   67    0
## 61   10   67    0
## 62   29   67    0
## 63   44   67    0
## 64   49   67    0
## 65   17   75    0
## 66   34   75    0
## 67   31   77    0
## 68   35   77    0
## 69   36   77    0
## 70   40   77    0
## 71   82   80    0
## 72   80   82    0
```

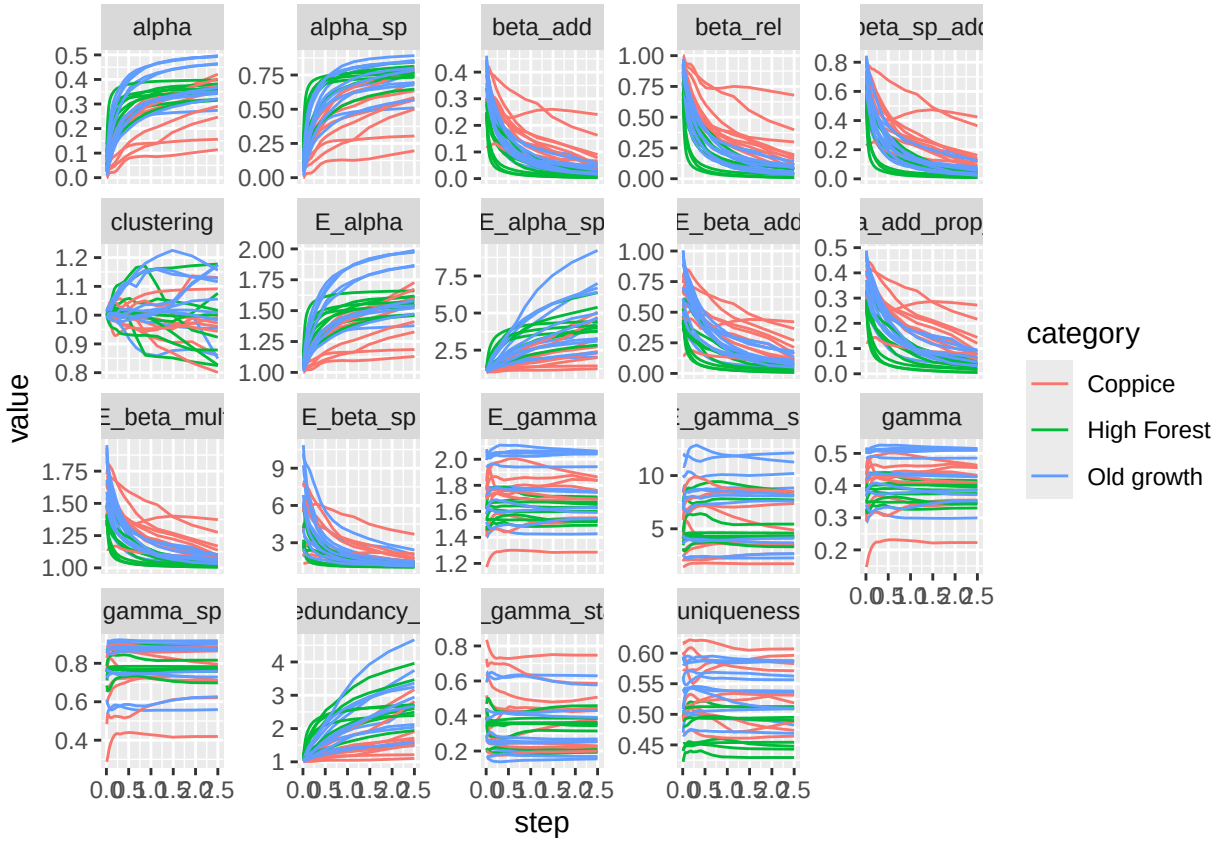
```r
## calculate indices ------------------------------------------------------

raotrait <- lapply(resampled, function (x)
  lapply(x, function (y) qdecomp(y, distances, species)) |>
    bind_rows(.id = "step") |>
    mutate(step = as.numeric(step))
) |>
  bind_rows(.id = "site")

#scale steps in units of area

raotrait <- raotrait |>
  mutate(step = step / 100)

#add to list

indices[[3]] <- raotrait
names(indices)[3] <- "clonal"


## create curve set ------------------------------------------------------

curvesets <- vector(mode = "list", length = sum(map_lgl(raotrait, is.numeric))-1)

for (i in seq_along(curvesets)){
  column <- colnames(raotrait)[i+2]
  curves <- raotrait |>
    select(site, step, {{column}}) |>
    pivot_wider(names_from = site, values_from = 3) |>
    filter(if_all(where(is.numeric), ~ !is.na(.)))
  curvesets[[i]] <- curve_set(obs = as.data.frame(curves[-1]), r = curves[[1]])
  names(curvesets)[i] <- column
}

indexcurves[[3]] <- curvesets
names(indexcurves)[3] <- "clonal"


# plotting indices ------------------------------------------------------
categories <- read_csv("data/classification.csv")
```

```
## Rows: 32 Columns: 2

## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (2): site, category
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```r
indices$overall |>
  left_join(categories) |>
  pivot_longer(3:21, names_to = "index", values_to = "value") |>
```

```
  ggplot(aes(x = step, y = value, group = site, colour = category))+
  geom_line()+
  facet_wrap(~index, scales = "free_y")
```

```
## Joining with `by = join_by(site)`
```




```
# save results ------------------------------------------------------------

saveRDS(indices, "indices.rds")
saveRDS(indexcurves, "curves.rds")

# session info ------------------------------------------------------------

sessionInfo()
```

```
## R version 4.5.2 (2025-10-31)
## Platform: x86_64-redhat-linux-gnu
## Running under: Nobara Linux 43 (KDE Plasma Desktop Edition)
##
## Matrix products: default
## BLAS/LAPACK: FlexiBLAS OPENBLAS-OPENMP;  LAPACK version 3.12.1
##
## locale:
##  [1] LC_CTYPE=it_IT.UTF-8       LC_NUMERIC=C
```

```
##  [3] LC_TIME=it_IT.utf8        LC_COLLATE=it_IT.UTF-8
##  [5] LC_MONETARY=it_IT.utf8    LC_MESSAGES=it_IT.UTF-8
##  [7] LC_PAPER=it_IT.utf8       LC_NAME=C
##  [9] LC_ADDRESS=C              LC_TELEPHONE=C
## [11] LC_MEASUREMENT=it_IT.utf8 LC_IDENTIFICATION=C
##
## time zone: Europe/Rome
## tzcode source: system (glibc)
##
## attached base packages:
## [1] stats     graphics  grDevices datasets  utils     methods   base
##
## other attached packages:
##  [1] vegan_2.7-2     permute_0.9-8   GET_1.0-7       lubridate_1.9.4
##  [5] forcats_1.0.1   stringr_1.6.0   dplyr_1.1.4     purrr_1.2.0
##  [9] readr_2.1.6     tidyr_1.3.2     tibble_3.3.0    ggplot2_4.0.1
## [13] tidyverse_2.0.0
##
## loaded via a namespace (and not attached):
##  [1] utf8_1.2.6         generics_0.1.4     stringi_1.8.7      lattice_0.22-7
##  [5] hms_1.1.4          digest_0.6.39      magrittr_2.0.4     evaluate_1.0.5
##  [9] grid_4.5.2         timechange_0.3.0   RColorBrewer_1.1-3 fastmap_1.2.0
## [13] Matrix_1.7-4       gridExtra_2.3      mgcv_1.9-4         viridisLite_0.4.2
## [17] scales_1.4.0       CoprManager_0.5.7  ade4_1.7-23        cli_3.6.5
## [21] crayon_1.5.3       rlang_1.1.6        bit64_4.6.0-1      splines_4.5.2
## [25] withr_3.0.2        yaml_2.3.12        tools_4.5.2        parallel_4.5.2
## [29] tzdb_0.5.0         vctrs_0.6.5        R6_2.6.1           lifecycle_1.0.4
## [33] bit_4.6.0          vroom_1.6.7        MASS_7.3-65        cluster_2.1.8.1
## [37] pkgconfig_2.0.3    pillar_1.11.1      gtable_0.3.6       Rcpp_1.1.0
## [41] glue_1.8.0         xfun_0.55          tidyselect_1.2.1   rstudioapi_0.17.1
## [45] knitr_1.50         farver_2.1.2       nlme_3.1-168       htmltools_0.5.9
## [49] labeling_0.4.3     rmarkdown_2.30     compiler_4.5.2     S7_0.2.1
```