

Estimating species richness from an aggregation of inventory plots

Eric Marcon

March 16, 2022

1 Rationale

The Chao2 estimator (Chao, 1987) can be used to estimate the richness of a community from occurrence data in sampling plots.

It has been applied by Cazzolla Gatti et al. (2022) to abundance data of small sample plots aggregated into occurrence data on a grid. The question addressed here is the relationship between the aggregation level and the resulting estimation.

The estimation actually does not depend on the aggregation level.

2 Data

A large, spatialized community is simulated by package *SpatDiv*. The community is log-normal. Spatially, tree species are clustered according to a Thomas process.

Parameters are:

- the size of the square window of the simulation, including the name of units,
- the number of trees per unit area,
- the number of species,
- the parameters of aggregation: the standard deviation of the displacements of points around the cluster centers (`thomas_scale`) and the average number of trees per cluster (`thomas_mu`).

The simulation may take time and memory over 100,000 trees.

```
library("SpatDiv")
window_size <- 2000
unit_name <- c("meter", "meters")
trees_n_per_area <- 500/10000
species_n <- 500
thomas_scale <- window_size/10
thomas_mu <- 100
```

The community is simulated:

```
library("spatstat")
rSpCommunity(n = 1, size = window_size^2 * trees_n_per_area,
  S = species_n, Spatial = "Thomas", scale = thomas_scale,
  mu = thomas_mu, win = square(r = window_size, unitname = unit_name)) ->
  spCommunity
# Number of trees
spCommunity$n
```

```
## [1] 195650
```

Square, random inventory plots are simulated in the community.
Their parameters are: - their size, - their number.

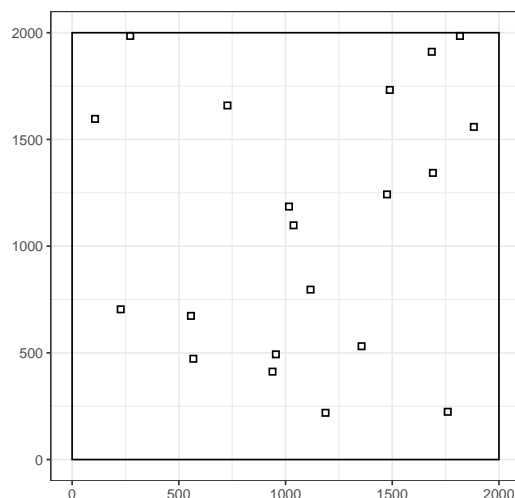
```
# Plot side length
side_length <- 30
# Number of plots
plots_n <- 20
```

Random coordinates are drawn:

```
# Draw random plots
X_0 <- runif(plots_n, max = window_size)
Y_0 <- runif(plots_n, max = window_size)
# List of windows
plots_windows <- solapply(seq_along(X_0), function(i) owin(xrange = c(X_0[i],
  X_0[i] + side_length), yrange = c(Y_0[i], Y_0[i] +
  side_length), unitname = unit_name))
# Coordinates of the plots in a dataframe
plots_coords <- data.frame(Plot = seq_len(plots_n),
  X = X_0, Y = Y_0)
```

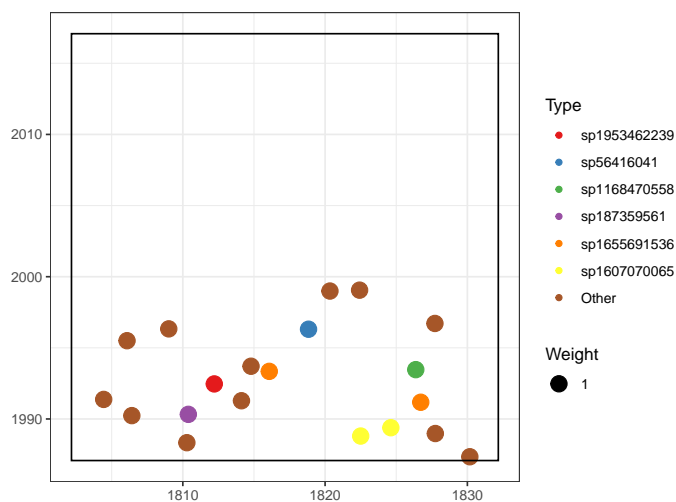
The plots are mapped:

```
# spatstat
# plot(intersect.owin(spCommunity>window,
# union.owin(plots_windows)))
# ggplot
window_all <- data.frame(xmin = c(spCommunity>window$xrange[1],
  pmin(X_0, window_size - side_length)), xmax = c(spCommunity>window$xrange[2],
  pmin(X_0 + side_length, window_size)), ymin = c(spCommunity>window$yrange[1],
  pmin(Y_0, window_size - side_length)), ymax = c(spCommunity>window$yrange[2],
  pmin(Y_0 + side_length, window_size)))
library("ggplot2")
gg_plots <- ggplot() + geom_rect(data = window_all,
  mapping = aes(xmin = xmin, xmax = xmax, ymin = ymin,
    ymax = ymax), color = "black", fill = "transparent") +
  coord_fixed()
print(gg_plots)
```



The sampling intensity is 0.45% of the total area.
Any plot can be mapped with its content. The first one is:

```
plot_to_draw <- plots_windows[[1]]
library("dbmss")
autoplot(spCommunity[plot_to_draw])
```



3 Inventory

The inventory of a plot is made by the `plot_trees()` function which return a boolean vector for all the trees of the community. Its value is `TRUE` when a tree is in the plot.

If the plot is partly outside the window, it is moved inside.

```

plot_trees <- function(spCommunity, side_length, x_0,
  y_0) {
  x_max <- spCommunity$window$ xrange[2]
  y_max <- spCommunity$window$ xrange[2]
  # The plot must be in the window
  if ((x_0 + side_length) > x_max)
    x_0 <- x_max - side_length
  if ((y_0 + side_length) > y_max)
    y_0 <- y_max - side_length
  # Trees in the plot
  is_in <- (spCommunity$x >= x_0) & (spCommunity$x <=
    x_0 + side_length) & (spCommunity$y >= y_0) &
    (spCommunity$y <= y_0 + side_length)
  return(is_in)
}

```

The inventory is simulated: the trees inside each plot are listed and a abundance by plot dataframe is produced.

```

# Initialize a dataframe with factors
plots_inventory <- data.frame()
# Inventory
for (i in seq_len(plots_n)) {
  plot_inventory <- data.frame(Plot = i, Tree = spCommunity$marks$PointType[plot_trees(spCommunity,
    side_length, X_0[i], Y_0[i])])
  plots_inventory <- rbind(plots_inventory, plot_inventory)
}
# Transform the inventory into an abundance table
library("tidyverse")
plots_inventory %>%
  group_by(Plot, Tree) %>%
  summarise(Abundance = n()) %>%
  pivot_wider(names_from = Tree, values_from = Abundance,
    values_fill = 0) -> plots_abundances

```

4 Aggregation

plots are grouped in the cells of a grid by the function `group_plots()`. It returns a three-column dataframe with the plot names and the x and y coordinates of the left lower corner of the cell of the grid each plot belongs to.

```

# Group plots in a grid
group_plots <- function(plots, grid_x, grid_y) {
  grid_xy <- matrix(0, nrow = nrow(plots), ncol = 2)
  for (row in seq_len(nrow(plots))) {
    grid_xy[row, 1] <- grid_x[max(which(plots$X[row] >
      grid_x))]
    grid_xy[row, 2] <- grid_y[max(which(plots$Y[row] >
      grid_y))]
  }
  colnames(grid_xy) <- c("x_grid", "y_grid")
  return(grid_xy)
}

```

```

abundance_gridded <- function(abundances, plots, grid_size,
  x_min = 0, x_max = 1, y_min = 0, y_max = 1) {
  grid_x <- seq(x_min, x_max, by = grid_size)
  grid_y <- seq(y_min, y_max, by = grid_size)
  plots_gridded <- cbind(plots["Plot"], group_plots(plots,
    grid_x, grid_y))
  abundances %>%

```

```

    inner_join(plots_gridded) %>%
    select(-Plot) %>%
    group_by(x_grid, y_grid) %>%
    summarise_all(sum) -> abundances_aggregated
  return(abundances_aggregated)
}

```

The `richness_gridded()` function return the estimation of richness by the Chao2 estimator after aggregating the plots of each cell of the grid.

```

richness_gridded <- function(abundances, plots, grid_size,
  x_min = 0, x_max = 1, y_min = 0, y_max = 1) {
  abundances_aggregated <- abundance_gridded(abundances,
    plots, grid_size, x_min, x_max, y_min, y_max)
  occurrences <- apply(abundances_aggregated[, -(1:3)],
    2, function(x) sum(x > 0))
  n <- nrow(abundances_aggregated)
  s_obs <- sum(occurrences > 0)
  s_1 <- sum(occurrences == 1)
  s_2 <- sum(occurrences == 2)
  s_chao2 <- ifelse(s_2 > 0, s_obs + (n - 1) * s_1^2/2/n/s_2,
    s_obs + (n - 1) * s_1 * (s_1 - 1)/2/n)
  return(s_chao2)
}

```

The estimation of the number of species with a 4-cell grid is:

```

# Example
richness_gridded(plots_abundances, plots_coords, grid_size = window_size/2,
  x_max = window_size, y_max = window_size)

```

```
## [1] 389.4082
```

The sensitivity to the grid size is evaluated by dividing it by 2 several times (from 1/2 to 1/32 of the community size).

```

sapply(1:5, function(n) richness_gridded(plots_abundances,
  plots_coords, grid_size = window_size/2^n, x_max = window_size,
  y_max = window_size))

```

```
## [1] 389.4082 390.0658 389.9691 383.8129 380.2960
```

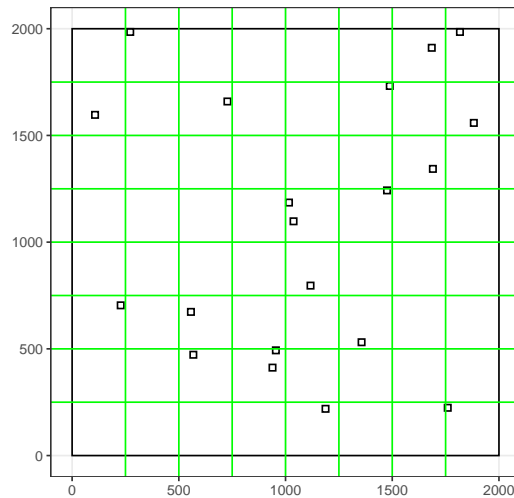
The estimation is very stable.

A grid size equal to 1/8th of the window (64 cells) is shown here:

```

grid_interval <- window_size/2^3
grid_seq <- seq(grid_interval, window_size * 0.9999,
  by = grid_interval)
gg_plots + geom_hline(yintercept = grid_seq, col = "green") +
  geom_vline(xintercept = grid_seq, col = "green")

```



The estimation of richness could be made directly from the abundance data:

```
Richness(colSums(plots_abundances[, -1]))
```

```
## Jackknife 1
##          402
```

5 Turing's relation

```
turing <- function(distribution) {
  n <- sum(distribution)
  s_1 <- sum(distribution == 1)
  cat("Actual number of singletons:", s_1, "\n")
  s_2 <- sum(distribution == 2)
  s_3 <- sum(distribution == 3)
  s_4 <- sum(distribution == 4)
  s_1_hat <- (n - 1)/n * 2 * s_2 * (5 * s_2/6/s_3 -
    s_3/4/s_4)
  cat("Expected number of singletons:", s_1_hat,
    "\n")
  return(s_1_hat)
}
# Abundances
turing(colSums(plots_abundances[, -1]))
```

```
## Actual number of singletons: 101
## Expected number of singletons: 186.0138
```

```
## [1] 186.0138
```

```
# Aggregated
abundances_aggregated <- abundance_gridded(plots_abundances,
  plots_coords, grid_size = window_size/2^3, x_max = window_size,
  y_max = window_size)
occurences <- apply(abundances_aggregated[, -(1:3)],
  2, function(x) sum(x > 0))
turing(occurences)
```

```
## Actual number of singletons: 122
## Expected number of singletons: 139.6517

## [1] 139.6517
```

References

- Cazzolla Gatti, R., P. B. Reich, J. G. P. Gamarra, T. Crowther, C. Hui, A. Morera, J.-F. Bastin, S. de-Miguel, G.-J. Nabuurs, J.-C. Svenning, J. M. Serra-Diaz, C. Merow, B. Enquist, M. Kamenetsky, J. Lee, J. Zhu, J. Fang, D. F. Jacobs, B. Pijanowski, A. Banerjee, R. A. Giaquinto, G. Alberti, A. M. Almeyda Zambrano, E. Alvarez-Davila, A. Araujo-Murakami, V. Avitabile, G. A. Aymard, R. Balazy, C. Baraloto, J. G. Barroso, M. L. Bastian, P. Birnbaum, R. Bitariho, J. Bogaert, F. Bongers, O. Bouriaud, P. H. S. Brancalion, F. Q. Brearley, E. N. Broadbent, F. Bussotti, W. Castro da Silva, R. G. César, G. Češljár, V. Chama Moscoso, H. Y. H. Chen, E. Cienciala, C. J. Clark, D. A. Coomes, S. Dayanandan, M. Decuyper, L. E. Dee, J. Del Aguila Pasquel, G. Derroire, M. N. K. Djuikouo, T. Van Do, J. Dolezal, I. Đ. Đorđević, J. Engel, T. M. Fayle, T. R. Feldpausch, J. K. Fridman, D. J. Harris, A. Hemp, G. Hengeveld, B. Herault, M. Herold, T. Ibanez, A. M. Jagodzinski, B. Jaroszewicz, K. J. Jeffery, V. K. Johannsen, T. Jucker, A. Kangur, V. N. Karminov, K. Kartawinata, D. K. Kennard, S. Kepfer-Rojas, G. Keppel, M. L. Khan, P. K. Khare, T. J. Kileen, H. S. Kim, H. Korjus, A. Kumar, A. Kumar, D. Laarmann, N. Labrière, M. Lang, S. L. Lewis, N. Lukina, B. S. Maitner, Y. Malhi, A. R. Marshall, O. V. Martynenko, A. L. Monteagudo Mendoza, P. V. Ontikov, E. Ortiz-Malavasi, N. C. Pallqui Camacho, A. Paquette, M. Park, N. Parthasarathy, P. L. Peri, P. Petronelli, S. Pfautsch, O. L. Phillips, N. Picard, D. Piotto, L. Poorter, J. R. Poulsen, H. Pretzsch, H. Ramírez-Angulo, Z. Restrepo Correa, M. Rodeghiero, R. D. P. Rojas Gonzáles, S. G. Rolim, F. Rovero, E. Rutishauser, P. Saikia, C. Salas-Eljatib, D. Schepaschenko, M. Scherer-Lorenzen, V. Šebeň, M. Silveira, F. Slik, B. Sonké, A. F. Souza, K. J. Stereńczak, M. Svoboda, H. Taedoumg, N. Tchebakova, J. Terborgh, E. Tikhonova, A. Torres-Lezama, F. van der Plas, R. Vásquez, H. Viana, A. C. Vibrans, E. Vilanova, V. A. Vos, H.-F. Wang, B. Westerlund, L. J. T. White, S. K. Wiser, T. Zawila-Niedzwiecki, L. Zemagho, Z.-X. Zhu, I. C. Zo-Bi, and J. Liang (2022). The number of tree species on Earth. *Proceedings of the National Academy of Sciences* 119(6), e2115329119.
- Chao, A. (1987). Estimating the population size for capture-recapture data with unequal catchability. *Biometrics* 43(4), 783–791.