

Estimating species richness from inventory plots

Eric Marcon

March 14, 2022

1 Data

Simulate a spatialized community. Parameters:

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

Build the community:

```
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] 198863
```

Plot parameters

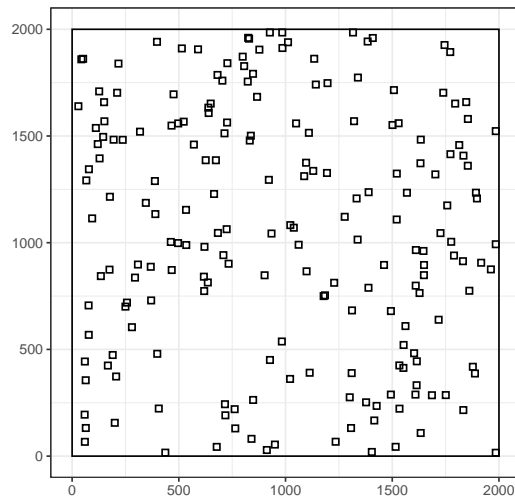
```
# Plot side length
side_length <- 30
# Number of plots
plots_n <- 200
```

Build the plots

```
# 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)
```

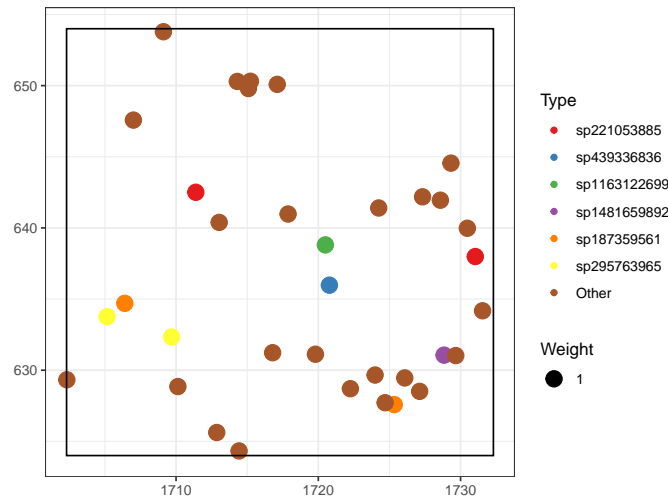
Draw the plots

```
# 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)
```



Draw the first plot

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



2 Inventory

Inventory of a plot

```
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)
}
```

Sample the plots

```
# 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
plots_inventory %>%
  group_by(Plot, Tree) %>%
  summarise(Abundance = n()) %>%
  pivot_wider(names_from = Tree, values_from = Abundance,
    values_fill = 0) -> plots_abundances
```

3 Agregate

Group plots in a grid

```
# 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)
}
```

Estimate total richness according to the size

```
richness_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 %>%
    group_by(Plot) %>%
    summarise_all(sum) %>%
    inner_join(plots_gridded) %>%
    select(-Plot) %>%
    group_by(x_grid, y_grid) %>%
    summarise_all(sum) -> plots_aggregated
  occurrences <- apply(plots_aggregated[, -(1:2)],
    2, function(x) sum(x > 0))
  n <- nrow(plots_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)
}
```

Estimate the number of species with a 4-cell grid:

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

```
## [1] 489.3365
```

Sensibility to the grid size

```
sapply(1:5, function(n) richness_gridded(plots_abundances,
  plots_coords, window_size/2^n))
```

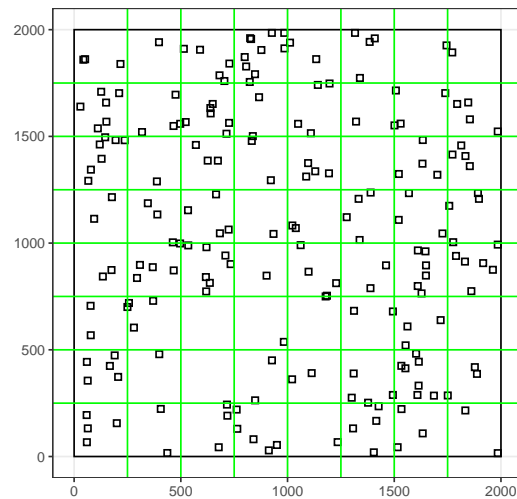
```
## [1] 481 481 481 481 481
```

Plot the smallest grid size

```

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")

```



Compare with the estimation of richness from abundance data:

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

```
## Jackknife 1
##          501
```