Describing landscapes: calculating forest amount and isolation for Atlantic Forest sampling points

Bernardo Niebuhr, Mauricio Vancine

2022-02-21

Introduction

Here we explore the amount of forest change, in absolute area and proportion, within several extents around each sampling point where interaction networks were sampled within the Atlantic Forest in Brazil. We consider both frugivory/seed dispersal networks (Bello et al. 2015) and pollination networks (REF, in prep?).

Loading data

Here we load the packages and the landscape and interaction data. To assess changes in forest amount, we used the land use maps from MapBiomas, from the years 1985 and 2020, and consider the class forest (detail the procedure here).

```
# library(devtools)
# devtools::install_github("bniebuhr/landscapemetrics", ref = "multibuffer")
# devtools::install_github("bniebuhr/landscapetools", ref = "multibuffer")

# packages
library(dplyr)
library(readxl)
library(ggplot2)

library(raster)
library(sf)
library(landscapetools)
library(landscapemetrics)
library(tmap)

# rasters
forest_1985 <- raster::raster("data/mapbiomas_v06_1985_forest.tif")
forest_2020 <- raster::raster("data/mapbiomas_v06_2020_forest.tif")
# plot(forest_1985)

# CRS
crs_atlantic <- raster::crs(forest_1985)
# resolution of the map in meters
resol <- 30

# datasets
pol <- readxl::read_xlsx("data/Atlantic_forest_floral_visitor_data_points.xlsx") %>%
```

```
  dplyr::rename(x = longitude, y = latitude) %>%
  sf::st_as_sf(coords = c("x", "y"), crs = crs_atlantic)
fru <- readxl::read_xlsx("data/Atlantic_forest_frugivory_data_points.xlsx") %>%
  dplyr::rename(x = longitude, y = latitude) %>%
  sf::st_as_sf(coords = c("x", "y"), crs = crs_atlantic)
```

Calculating forest change

To calculate the change in forest amount around each sampling point, we followed these steps: - Create a series of buffers, with extent from 250 m to 10 km - Calculate the change in absolute area of forest (in hectares) within each buffer, for each point - Calculate the change in proportion of forest within each buffer, for each point - Classify points where the changes were the highest (> 3% of change).

Pollination networks

Calculate changes

```
# pollination

# extract amount of forest
buffers <- c(250, 500, 750, 1000, 1500, 2000, 3000, 4000, 5000, 7500, 10000)
tictoc::tic()
pol_1985 <- landscapetools::util_extract_multibuffer(forest_1985, pol,
  buffer_width = buffers,
  rel_freq = TRUE, point_id_text = FALSE
) %>%
  dplyr::rename(freq1985 = freq, rel_freq1985 = rel_freq)
tictoc::toc() # 17 min, 11 buffer sizes, 200 points

# calculate isolation
# tictoc::tic()
# pol_1985_enn <- landscapemetrics::scale_sample(forest_1985, pol[1:2,], shape = "circle",
#                                                 size = buffers,
#                                                 what = c("lsm_c_enn_mn"))
# tictoc::toc()

# show_shareplot(multibuffer_df = pol_1985 %>% dplyr::filter(id %in% 1:5))
pol_2020 <- landscapetools::util_extract_multibuffer(forest_2020, pol,
  buffer_width = buffers,
  rel_freq = TRUE, point_id_text = FALSE
) %>%
  dplyr::rename(freq2020 = freq, rel_freq2020 = rel_freq)

# calculate isolation
# tictoc::tic()
# pol_2020_enn <- landscapemetrics::scale_sample(forest_2020, pol[1:2,], shape = "circle",
#                                                 size = buffers,
#                                                 what = c("lsm_c_enn_mn"))
# tictoc::toc()
```

```r
pol_forest_change <- pol_1985 %>%
  tibble::as_tibble() %>%
  dplyr::bind_cols(pol_2020[3:4]) %>%
  dplyr::mutate(
    change_area_ha = (freq2020 - freq1985) * resol**2 / 10e4,
    change_prop = 100 * (rel_freq2020 - rel_freq1985)
  )

# export table of changes
pol_forest_change %>%
  readr::write_csv(file = "output/polinization_forest_change.csv")
```

Calculate changes

Below we plot the absolute change in forest area (in hectares) and the change in the proportion of forests. Please note that the y axes follow different scales. In both plots the points were sorted according to the value of absolute change in forest area (in the x axis).

```r
# plot - total area
pol_forest_change %>%
  dplyr::filter(layer == 1) %>%
  dplyr::arrange(change_area_ha) %>%
  tibble::rowid_to_column(var = "rown") %>%
  ggplot(aes(x = rown, y = change_area_ha)) +
  geom_point() +
  facet_wrap(~buffer, scales = "free_y") +
  labs(y = "Change in area 1985-2020 (ha)", x = "")
```
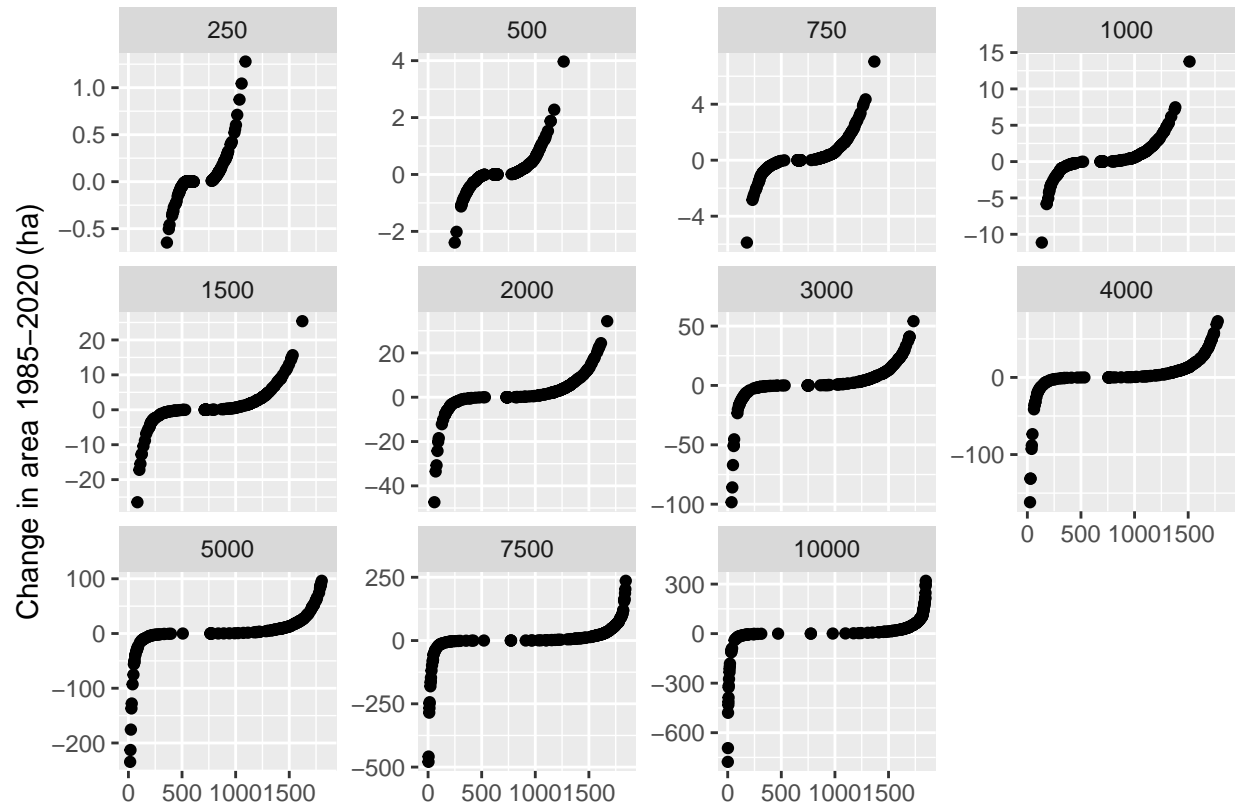
```
# plot - proportion
pol_forest_change %>%
  dplyr::filter(layer == 1) %>%
  dplyr::arrange(change_area_ha) %>%
  tibble::rowid_to_column(var = "rown") %>%
  ggplot(aes(x = rown, y = change_prop)) +
  geom_point() +
  facet_wrap(~buffer, scales = "free_y") +
  labs(y = "Change in proportion of forest within the buffer, 1985-2020", x = "")
```
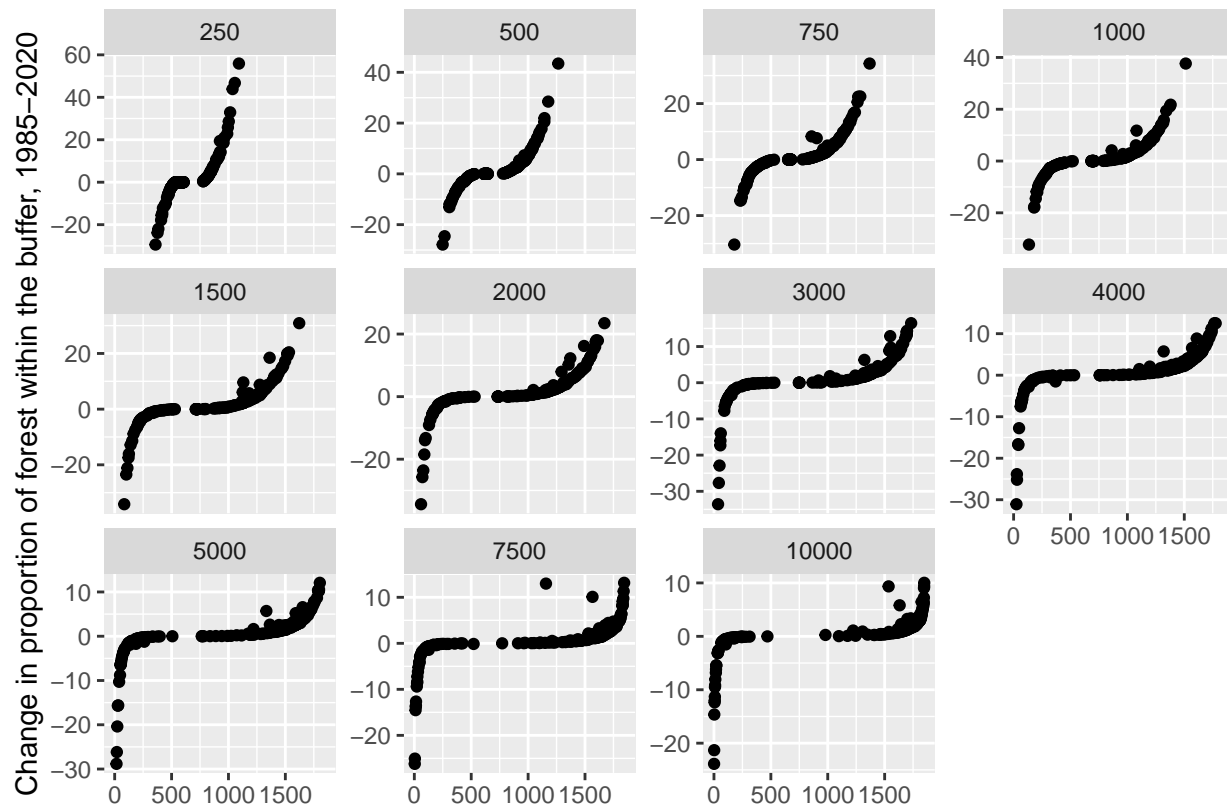
Join with spatial data and save output

Now we join this information with the sampling point positions and export it as a csv, as well as a shapefile. We classify all points where the proportion of forest change was < 3% (at the highest extent, 10 km buffer) as stable, and the others as "forest loss" and "forest gain".

```
# join tables
pol <- pol %>%
  dplyr::left_join(
    pol_forest_change %>%
      dplyr::filter(layer == 1 & buffer == 10000) %>%
      dplyr::select(-layer) %>%
      dplyr::mutate(id = as.numeric(id)),
    by = c("points" = "id")
  ) %>%
  dplyr::mutate(stability = case_when(
    change_prop < -3 ~ "forest loss",
    change_prop < 3 ~ "stable",
    TRUE ~ "forest gain"
  ))


# How many forest loss points do we have?
pol %>%
  dplyr::filter(stability == "forest loss") %>%
```

```
  nrow()
```

[1] 16

Here we see we have only 16 points with forest loss.

```
# save shapefile
sf::st_write(pol,
  dsn = "output/Atlantic_forest_floral_visitor_data_points_forest_change.gpkg",
  delete_dsn = TRUE
)
```


Plot spatial data

We finally plot the spatial data.

```
# change
tm_shape(pol) +
  tmap::tm_dots(col = "change_prop", size = 0.1)
```
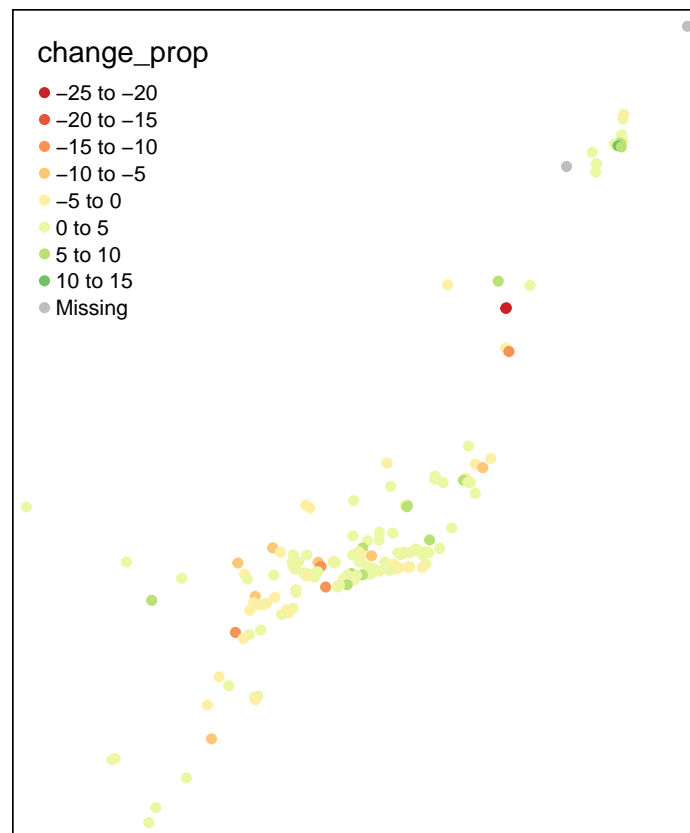


```
# stability
tm_shape(pol) +
  tmap::tm_dots(col = "stability", size = 0.1, palette = c("blue", "red", "yellow"))
```
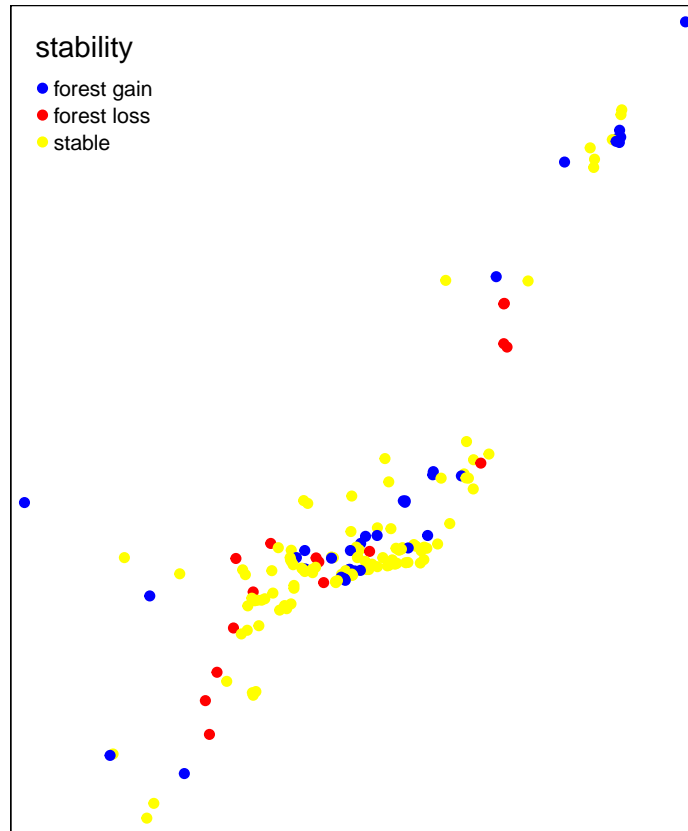
Furgivory networks

Now we do the same for frugivory networks.

Calculate changes

```r
# frugivory

# extract amount of forest
buffers <- c(250, 500, 750, 1000, 1500, 2000, 3000, 4000, 5000, 7500, 10000)
fru_1985 <- landscapetools::util_extract_multibuffer(forest_1985, fru,
  buffer_width = buffers,
  rel_freq = TRUE, point_id_text = FALSE
) %>%
  dplyr::rename(freq1985 = freq, rel_freq1985 = rel_freq)

# calculate isolation
# tictoc::tic()
# fru_1985_enn <- landscapemetrics::scale_sample(forest_1985, fru[1:2,], shape = "circle",
#                                                 size = buffers, what = c("lsm_c_enn_mn"))
# tictoc::toc()

# show_shareplot(multibuffer_df = fru_1985 %>% dplyr::filter(id %in% 1:5))
fru_2020 <- landscapetools::util_extract_multibuffer(forest_2020, fru,
```

```
    buffer_width = buffers,
    rel_freq = TRUE, point_id_text = FALSE
) %>%
  dplyr::rename(freq2020 = freq, rel_freq2020 = rel_freq)

# calculate isolation
# tictoc::tic()
# fru_2020_enn <- landscapemetrics::scale_sample(forest_2020, fru[1:2,], shape = "circle",
#                                                size = buffers, what = c("lsm_c_enn_mn"))
# tictoc::toc()

fru_forest_change <- fru_1985 %>%
  tibble::as_tibble() %>%
  dplyr::bind_cols(fru_2020[3:4]) %>%
  dplyr::mutate(
    change_area_ha = (freq2020 - freq1985) * resol**2 / 10e4,
    change_prop = 100 * (rel_freq2020 - rel_freq1985)
  )

# export table of changes
fru_forest_change %>%
  readr::write_csv(file = "output/frugivory_forest_change.csv")
```

Calculate changes

Below we plot the absolute change in forest area (in hectares) and the change in the proportion of forests. Please note that the y axes follow different scales. In both plots the points were sorted according to the value of absolute change in forest area (in the x axis).

```
# plot - total area
fru_forest_change %>%
  dplyr::filter(layer == 1) %>%
  dplyr::arrange(change_area_ha) %>%
  tibble::rowid_to_column(var = "rown") %>%
  ggplot(aes(x = rown, y = change_area_ha)) +
  geom_point() +
  facet_wrap(~buffer, scales = "free_y") +
  labs(y = "Change in area 1985-2020 (ha)", x = "")
```
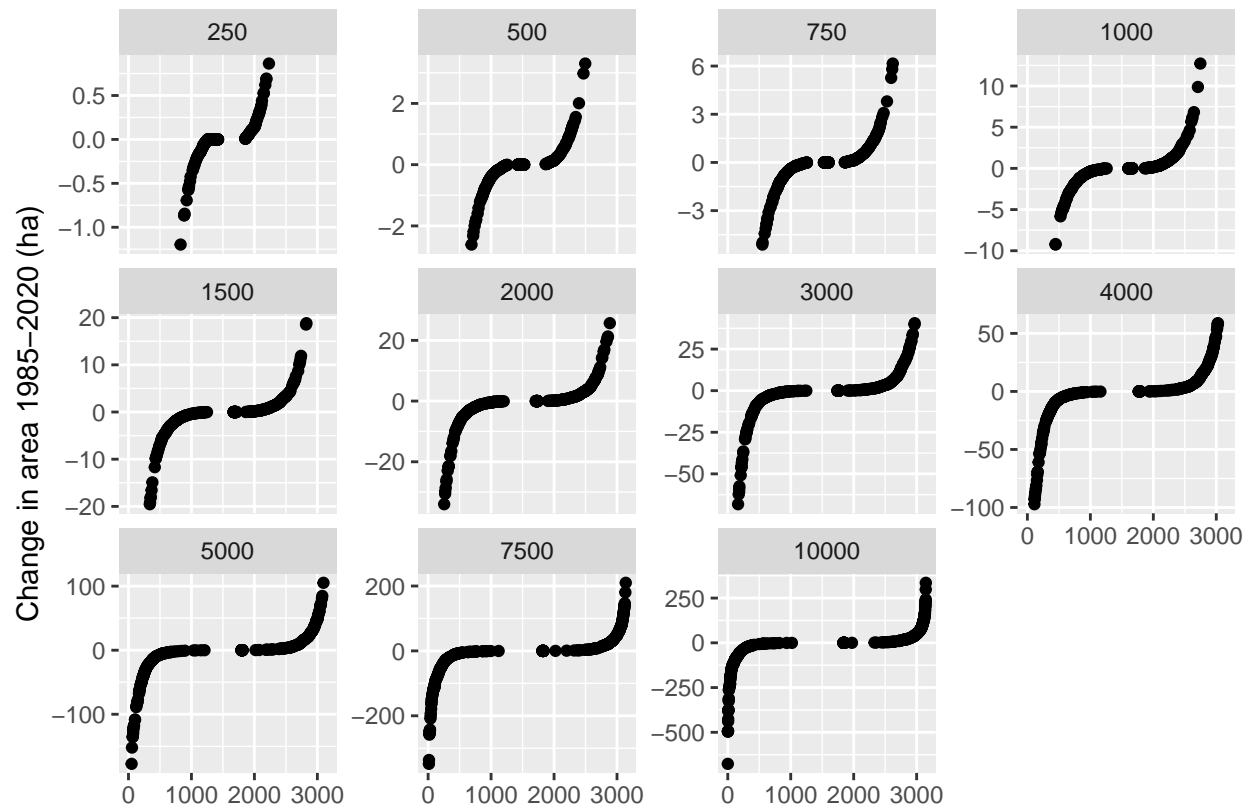
```r
# plot - proportion
fru_forest_change %>%
  dplyr::filter(layer == 1) %>%
  dplyr::arrange(change_area_ha) %>%
  tibble::rowid_to_column(var = "rown") %>%
  ggplot(aes(x = rown, y = change_prop)) +
  geom_point() +
  facet_wrap(~buffer, scales = "free_y") +
  labs(y = "Change in proportion of forest within the buffer, 1985-2020", x = "")
```
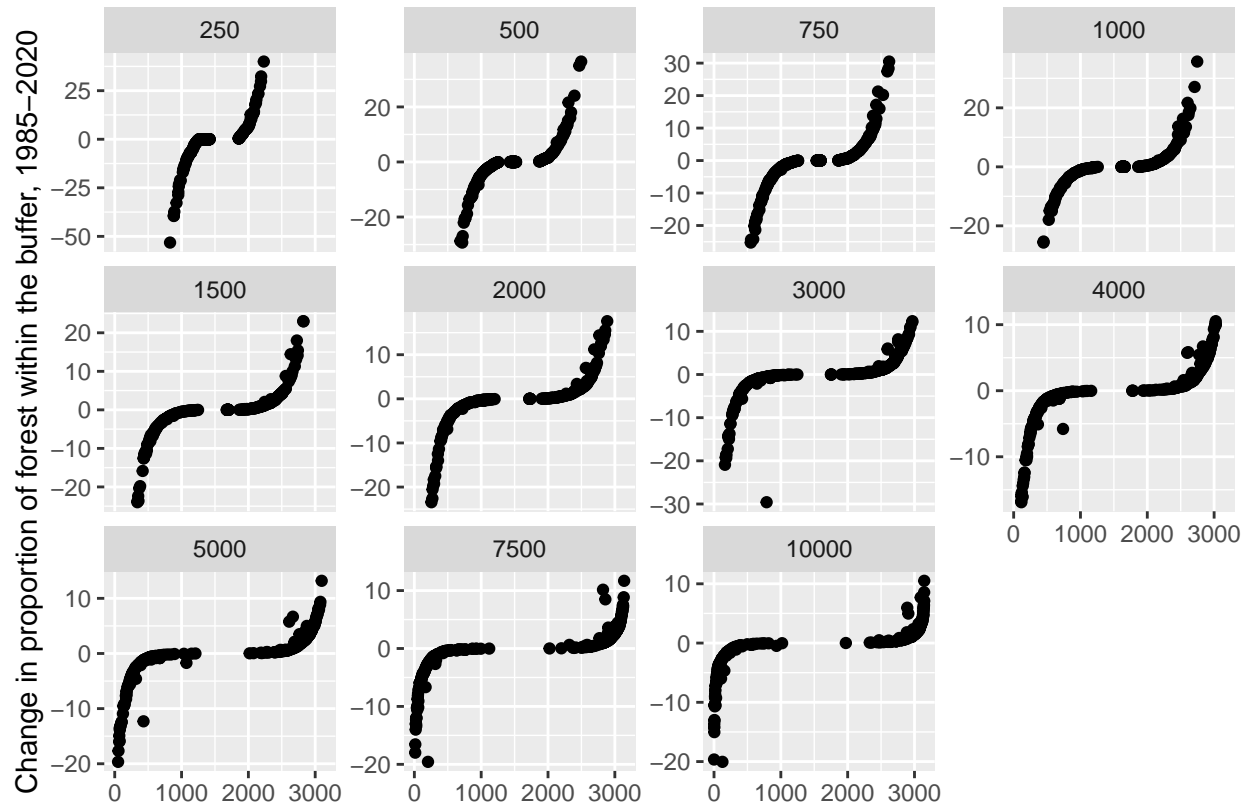
Join with spatial data and save output

Now we join this information with the sampling point positions and export it as a csv, as well as a shapefile. We classify all points where the proportion of forest change was $< 3\%$ (at the highest extent, 10 km buffer) as stable, and the others as "forest loss" and "forest gain".

```r
# join tables
fru <- fru %>%
  dplyr::left_join(
    fru_forest_change %>%
      dplyr::filter(layer == 1 & buffer == 10000) %>%
      dplyr::select(-layer) %>%
      dplyr::mutate(id = as.numeric(id)),
    by = c("points" = "id")
  ) %>%
  dplyr::mutate(stability = case_when(
    change_prop < -3 ~ "forest loss",
    change_prop < 3 ~ "stable",
    TRUE ~ "forest gain"
  ))

# How many forest loss points do we have?
fru %>%
  dplyr::filter(stability == "forest loss") %>%
  nrow()
```

```
## [1] 61
```

Here we see we have 61 points with forest loss in the frugivory dataset.

```
# save shapefile
sf::st_write(fru,
    dsn = "output/Atlantic_forest_frugivory_data_points_forest_change.gpkg",
    delete_dsn = TRUE
)
```

Plot spatial data

We finally plot the spatial data. Here there seems to be some outliers that we must either check or remove.

```
# change
fru %>%
    dplyr::filter(!is.na(change_prop)) %>%
    tm_shape() +
    tmap::tm_dots(col = "change_prop", size = 0.1)
```
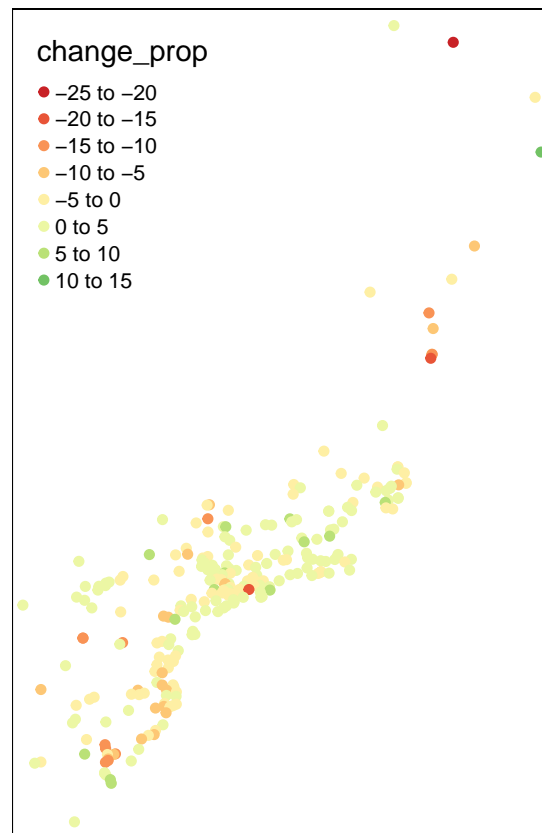


```
# stability
fru %>%
    dplyr::filter(!is.na(change_prop)) %>%
    tm_shape() +
    tmap::tm_dots(col = "stability", size = 0.1, palette = c("blue", "red", "yellow"))
```