TRhizo-localAdaptation

Urbanization Metrics

David Murray-Stoker

Contents

Urbanization Metrics

Description goes here...

Distance from the Urban Center

Distance from the urban center was calculated for each site as the distance on an ellipsoid (i.e. the geodesic distance). Coordinates for the urban center were selected as the Toronto City Hall (43.651536, -79.383276).

```
distance.data <- data.frame("Distance" = numeric(length = 40))

for (x in 1:40) {
  site.from.UC <- distm(
    x = site.data[x, c(3, 2)],
    y = c(-79.383276, 43.651536),
    fun = distGeo
  )

  site.from.UC.km <- site.from.UC / 1000

  distance.data[x, ] <- site.from.UC.km
}

## Add a Population identifier to each row
distance.data$Population <- site.data$Population

## Set column structure
distance.data$Population <- as_factor(distance.data$Population)
```

Impervious Surface Cover

Impervious surface cover (ISC) was estimated using the 30-m resolution Global Man-Made Impervious Surface raster dataset (Brown de Colstoun et al. 2017), with ISC values averaged within a 250-m buffer surrounding each population.

> Brown de Colstoun, E. C., C. Huang, P. Wang, J. C. Tilton, B. Tan, J. Phillips, S. Niemczura, P.-Y. Ling, and R. E. Wolfe. 2017. Global man-made impervious surface (GMIS) dataset from landsat. Palisades, NY.

```r
## Set the path to the ISC raster
ISC_raster_path <- "~/Downloads/CAN_gmis_impervious_surface_percentage_geographic_30m.tif"

## Set the function to extract ISC values
calculate_ISC <- function(df, raster_path, buffer = 250) {
  # Load in raster for country
  raster <- raster::raster(raster_path)

  # Create spatial point dataframe from latitude and longitude
  spdf <- sp::SpatialPointsDataFrame(
    coords = df %>%
      dplyr::select(Longitude, Latitude),
    proj4string = raster@crs,
    data = df
  )

  # Calculate ISC data for population
  ISC_data <- raster::extract(x = raster, y = spdf, method = "simple", buffer = buffer)
  ISC_data <- ifelse(is.na(ISC_data), 255, ISC_data) # Missing data needs to be 255

  # Take mean ISC across all cells included within buffer
  ISC_data_mod <- ISC_data %>%
    # Convert ISC values of 200 to 0, as per documentation
    # Convert ISC values of 255 to missing
    map(., function(x) {
      case_when(
        x == 200 ~ 0,
        x == 255 ~ NA_real_,
        TRUE ~ x
      )
    }) %>%
    map(., mean, na.rm = TRUE) %>% # Ignore cells with missing ISC values when calculating mean
    unlist()

  # Add column with ISC values
  df_out <- df %>%
    mutate(
      Mean_ISC = ISC_data_mod,
      Mean_ISC = ifelse(Mean_ISC == 255, NA, Mean_ISC)
    )

  # Reorganize the data
  df_out <- df_out %>%
    dplyr::select(Population, Mean_ISC)
```

```
  return(df_out)
}

## Calculate ISC data in a dataframe
ISC.data <- calculate_ISC(df = site.data, raster_path = ISC_raster_path) %>%
  as_tibble()
```

Human Influence Index

The Human Influence Index (HII) is a globally-distributed dataset (1-km grid cells) containing data on human population density, land use, infrastructure, and human access in a composite variable of HII.

> Wildlife Conservation Society (WCS) and Center for International Earth Science Information Network (CIESIN). 2005. Last of the wild project, version 2, 2005 (LWP-2): Global human influence index (HII) dataset.

```r
## Set the path to the HII raster
HII_raster_path <- "~/Downloads/HII_raster/hii_v2geo/w001001.adf"

## Set the function to extract HII values
calculate_HII <- function(df, raster_path) {
  # Load in the raster
  raster <- raster::raster(raster_path)

  # Create spatial point dataframe from latitude and longitude
  spdf <- sp::SpatialPointsDataFrame(
    coords = df %>%
      dplyr::select(Longitude, Latitude),
    proj4string = raster@crs,
    data = df
  )

  # Calculate HII data for each population
  HII_data <- raster::extract(x = raster, y = spdf, method = "simple")

  # Add column with HII values
  df_out <- df %>%
    mutate(
      Human_Influence_Index = HII_data,
      Human_Influence_Index = ifelse(Human_Influence_Index == 255, NA, Human_Influence_Index)
    )

  # Reorganize the data
  df_out <- df_out %>%
    dplyr::select(Population, Human_Influence_Index)

  return(df_out)
}

## Calculate HII data in a dataframe
HII.data <- calculate_HII(df = site.data, raster_path = HII_raster_path) %>%
  as_tibble()
```

Data Management

Description goes here…

Old:

Response variables from the experiment (i.e., aboveground biomass, belowground biomass, nodule density, and percent fixing nodules) were merged with urbanization metrics. Additionally, an identifier vector, `Microbiome_Global`, was used to identify broader nonlocal treatment effects (i.e., all nonlocal treatments combined, not just $\text{Nonlocal}_{\text{Rural}}$ or $\text{Nonlocal}_{\text{Urban}}$). All data were aggregated to population means for later analyses.

```
## Full urbanization data
full.urbanization.data <- site.data %>%
  full_join(distance.data, by = "Population") %>%
  full_join(HII.data, by = "Population") %>%
  full_join(ISC.data, by = "Population")
```

```
## Export full urbanization data
write_rds(full.urbanization.data, file = "data/full_urbanization_data.rds")
```

R Session Information

Table 1: Packages required for data management and analysis.

Package	Loaded Version	Date
dplyr	1.0.10	2022-09-01
forcats	0.5.2	2022-08-19
geosphere	1.5-14	2021-10-13
ggplot2	3.3.6	2022-05-03
kableExtra	1.3.4	2021-02-20
knitr	1.40	2022-08-24
purrr	0.3.5	2022-10-06
raster	3.6-3	2022-09-18
readr	2.1.3	2022-10-01
sp	1.5-0	2022-06-05
stringr	1.4.1	2022-08-20
tibble	3.1.8	2022-07-22
tidyr	1.2.1	2022-09-08
tidyverse	1.3.2	2022-07-18

References