Supplementary material for *Using citizen science to parse climatic and landcover influences on bird occupancy within a tropical biodiversity hotspot*

Vijay Ramesh Pratik R. Gupte Morgan W. Tingley VV Robin Ruth DeFries

2020-12-23

Contents

1 Introduction

This is supplementary material for a project in preparation that models occupancy for birds in the southern Western Ghats, India. The main project can be found here: https://github.com/pratikunterwegs/eBirdOccupancy.

1.1 Attribution

Please contact the following in case of interest in the project.

- Vijay Ramesh (lead author)
  - PhD student, Columbia University
- Pratik Gupte (repo maintainer)
  - PhD student, University of Groningen

2 Predicting Species-specific Occupancy

This supplement plots species-specific probabilities of occupancy as a function of significant environmental predictors.

2.1 Prepare libraries

```r
# to load data
library(readxl)

# to handle data
library(dplyr)
library(readr)
library(forcats)
library(tidyr)
library(purrr)
library(stringr)

# plotting
library(ggplot2)
library(patchwork)
```

2.2 Read data

```r
# read data
data <- read_csv(”data/results/data_occupancy_predictors.csv”)
```

```r
# drop na
data <- select(
  data,
  -ci
) %>%
  drop_na() %>%
  nest(data = c(predictor, m_group, seq_x, mean, scale))
```

Figure code is hidden in versions rendered as HTML and PDF. Example output is shown below.

Figure here

3 Selecting species of interest

This script shows the proportion of checklists that report a particular species across every 25km by 25km grid across the Nilgiris and the Anamalais. Using this analysis, we arrived at a final list of species for occupancy modeling.

We derived this list from inclusion criteria adapted from the State of India's Birds 2020 (Viswanathan et al., 2020). Initially, we considered all 561 species in eBird that occurred within the outlines of our study area. We then considered only those species that had a minimum of 1000 detections each between 2013 and 2019 (reducing to 303 species). Next, the study area was divided into 25 x 25 km cells following (Viswanathan et al., 2020). We then kept only those species that occurred in at least 5% of all checklists across 50% of the 25 x 25 km cells from where they have been reported (reducing to 93 species). We used the above criteria to ensure as much uniform sampling of a species as possible across our study area and to reduce any erroneous associations between environmental drivers and species occupancy. Across our final list of 93 species, we analyzed a total of ~3.2 million detections (presences) between 2013 and 2019.

3.1 Prepare libraries

```r
# load libraries
library(data.table)
library(readxl)
library(magrittr)
library(stringr)
library(dplyr)
library(tidyr)
library(readr)

library(ggplot2)
library(ggthemes)
library(scico)

# round any function
round_any <- function(x, accuracy = 25000) {
  round(x / accuracy) * accuracy
}
```

3.2 Read species of interest

```r
# add species of interest
specieslist <- read.csv(”data/species_list.csv”)
speciesOfInterest <- specieslist$scientific_name
```

3.3 Load raw data for locations

```r
# read in shapefile of the study area to subset by bounding box
library(sf)
wg <- st_read(”data/spatial/hillsShapefile/Nil_Ana_Pal.shp”)
box <- st_bbox(wg)

# read in data and subset
ebd <- fread(”data/01_ebird-filtered-EBD-westernGhats.txt”)
ebd <- ebd[between(LONGITUDE, box[”xmin”], box[”xmax”]) &
  between(LATITUDE, box[”ymin”], box[”ymax”]), ]
ebd <- ebd[year(`OBSERVATION DATE`) >= 2013, ]

# make new column names
newNames <- str_replace_all(colnames(ebd), ” ”, ”_”) %>%
  str_to_lower()
setnames(ebd, newNames)

# keep useful columns
columnsOfInterest <- c(
  ”scientific_name”, ”observation_count”, ”locality”,
  ”locality_id”, ”locality_type”, ”latitude”,
  ”longitude”, ”observation_date”, ”sampling_event_identifier”
)

ebd <- ebd[, ..columnsOfInterest]
```

Add a spatial filter and assign grids of 25km x 25km.

```r
# strict spatial filter and assign grid
locs <- ebd[, .(longitude, latitude)]

# transform to UTM and get 20km boxes
coords <- setDF(locs) %>%
  st_as_sf(coords = c(”longitude”, ”latitude”)) %>%
  `st_crs<-`(4326) %>%
  bind_cols(as.data.table(st_coordinates(.))) %>%
  st_transform(32643) %>%
  mutate(id = 1:nrow(.))

# convert wg to UTM for filter
wg <- st_transform(wg, 32643)
coords <- coords %>%
  filter(id %in% unlist(st_contains(wg, coords))) %>%
  rename(longitude = X, latitude = Y) %>%
  bind_cols(as.data.table(st_coordinates(.))) %>%
  st_drop_geometry() %>%
  as.data.table()

# remove unneeded objects
```

```r
rm(locs)
gc()

coords <- coords[, .N, by = .(longitude, latitude, X, Y)]

ebd <- merge(ebd, coords, all = FALSE, by = c("longitude", "latitude"))

ebd <- ebd[(longitude %in% coords$longitude) &
  (latitude %in% coords$latitude), ]
```

3.4 Get proportional obs counts in 25km cells

```r
# round to 25km cell in UTM coords
ebd[, `:=`(X = round_any(X), Y = round_any(Y))]

# count checklists in cell
ebd_summary <- ebd[, nchk := length(unique(sampling_event_identifier)),
  by = .(X, Y)
]

# count checklists reporting each species in cell and get proportion
ebd_summary <- ebd_summary[, .(nrep = length(unique(
  sampling_event_identifier
))),
by = .(X, Y, nchk, scientific_name)
]

ebd_summary[, p_rep := nrep / nchk]

# filter for soi
ebd_summary <- ebd_summary[scientific_name %in% speciesOfInterest, ]

# complete the dataframe for no reports
# keep no reports as NA --- allows filtering based on proportion reporting
ebd_summary <- setDF(ebd_summary) %>%
  complete(
    nesting(X, Y), scientific_name # ,
    # fill = list(p_rep = 0)
  ) %>%
  filter(!is.na(p_rep))
```

3.5 Which species are reported sufficiently in checklists?

```r
# A total of 42 unique grids (of 25km by 25km) across the study area
# total number of checklists across unique grids

tot_n_chklist <- ebd_summary %>%
  distinct(X, Y, nchk)

# species-specific number of grids
spp_grids <- ebd_summary %>%
  group_by(scientific_name) %>%
```

```
  distinct(X, Y) %>%
  count(scientific_name,
    name = ”n_grids”
  )

# Write the above two results
write_csv(tot_n_chklist, ”data/nchk_per_grid.csv”)
write_csv(spp_grids, ”data/ngrids_per_spp.csv”)

# left-join the datasets
ebd_summary <- left_join(ebd_summary, spp_grids, by = ”scientific_name”)

# check the proportion of grids across which this cut-off is met for each species
# Is it > 90% or 70%?
# For example, with a 3% cut-off, ~100 species are occurring in >50%
# of the grids they have been reported in

p_cutoff <- 0.05 # Proportion of checklists a species has been reported in
grid_proportions <- ebd_summary %>%
  group_by(scientific_name) %>%
  tally(p_rep >= p_cutoff) %>%
  mutate(prop_grids_cut = n / (spp_grids$n_grids)) %>%
  arrange(desc(prop_grids_cut))

grid_prop_cut <- filter(
  grid_proportions,
  prop_grids_cut > p_cutoff
)

# Write the results
write_csv(grid_prop_cut, ”data/chk_5_percent.csv”)

# Identifying the number of species that occur in potentially <5% of all lists
total_number_lists <- sum(tot_n_chklist$nchk)

spp_sum_chk <- ebd_summary %>%
  distinct(X, Y, scientific_name, nrep) %>%
  group_by(scientific_name) %>%
  mutate(sum_chk = sum(nrep)) %>%
  distinct(scientific_name, sum_chk)

# Approximately 90 to 100 species occur in >5% of all checklists
prop_all_lists <- spp_sum_chk %>%
  mutate(prop_lists = sum_chk / total_number_lists) %>%
  arrange(desc(prop_lists))
```

3.6 Figure: Checklist distribution

```
# add land
library(rnaturalearth)
land <- ne_countries(
  scale = 50, type = ”countries”, continent = ”asia”,
  country = ”india”,
```

```r
  returnclass = c("sf")
)

# crop land
land <- st_transform(land, 32643)
```

3.7 Prepare the species list

```r
# write the new list of species that occur in at least 5% of checklists across a minimum of 50% of the grids they h

new_sp_list <- semi_join(specieslist, grid_prop_cut, by = "scientific_name")

write_csv(new_sp_list, "data/03_list-of-species-cutoff.csv")
```

4 Climate in Relation to Landcover

This script showcases how climate data varies as a function of land cover types across our study area.

4.1 Prepare libraries

```r
# load libs
library(raster)
library(glue)
library(purrr)
library(dplyr)
library(tidyr)

# plotting options
library(ggplot2)
library(ggthemes)
library(scico)

# get ci func
ci <- function(x) {
  qnorm(0.975) * sd(x, na.rm = T) / sqrt(length(x))
}
```

4.2 Prepare environmental data

```r
# read landscape prepare for plotting
landscape <- stack("data/spatial/landscape_resamp01km.tif")

# get proper names
elev_names <- c("elev", "slope", "aspect")
chelsa_names <- c("bio_01", "bio_12")

names(landscape) <- as.character(glue('{c(elev_names, chelsa_names, "landcover")}'))
```

```r
# make duplicate stack
land_data <- landscape[[c("landcover", chelsa_names)]]

# convert to list
```

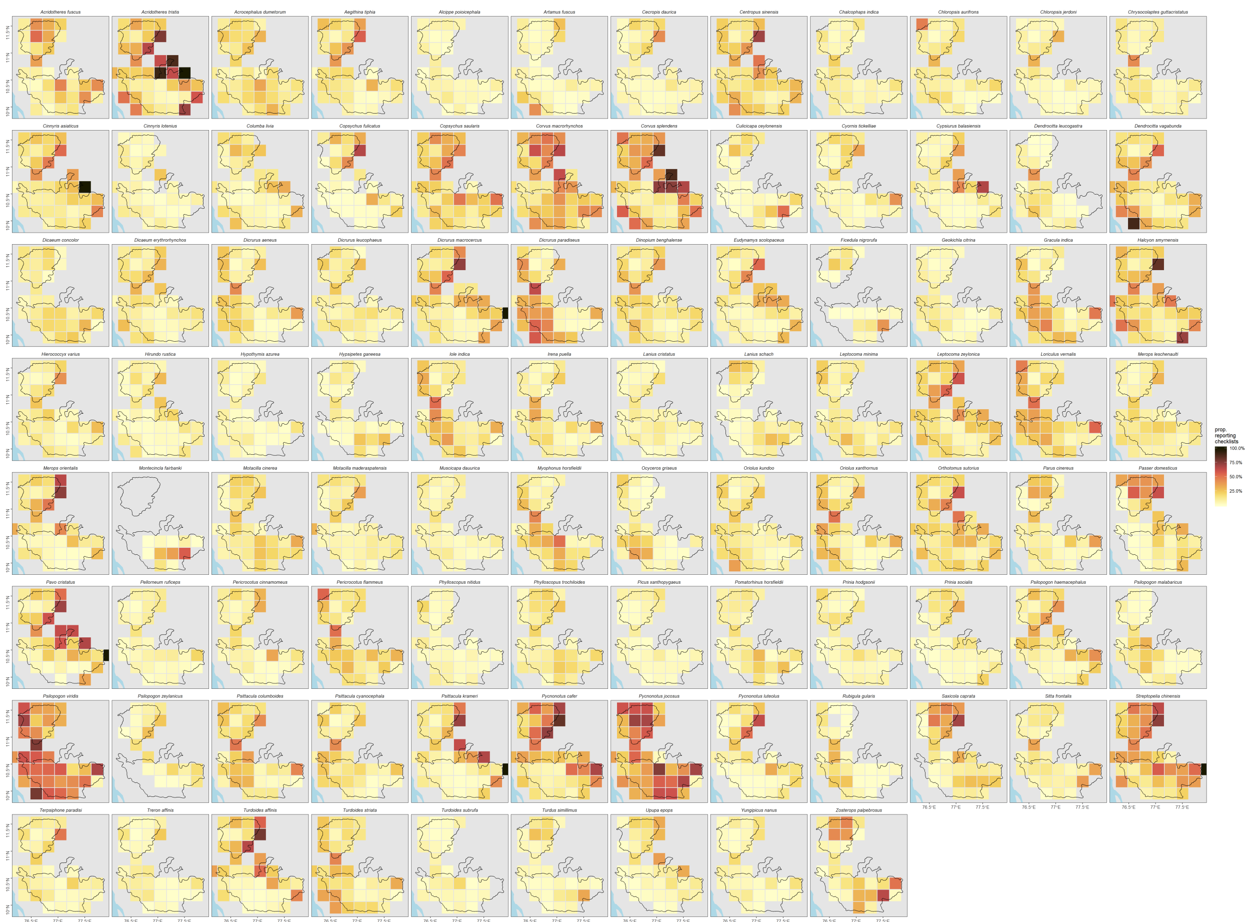

Figure 1: Proportion of checklists reporting a species in each grid cell (25km side) between 2013 and 2019. Checklists were filtered to be within the boundaries of the Nilgiris and the Anamalai hills (black outline), but rounding to 25km cells may place cells outside the boundary. Deeper shades of red indicate a higher proportion of checklists reporting a species.

```
land_data <- as.list(land_data)

# map get values over the stack
land_data <- purrr::map(land_data, raster::getValues)
names(land_data) <- c("landcover", chelsa_names)

# conver to dataframe and round to 100m
land_data <- bind_cols(land_data)
land_data <- drop_na(land_data) %>%
  filter(landcover != 0) %>%
  pivot_longer(
    cols = contains("bio"),
    names_to = "clim_var"
  ) # %>%
# group_by(landcover, clim_var) %>%
# summarise_all(.funs = list(~mean(.), ~ci(.)))
```

4.3 Climatic variables over landcover

Figure code is hidden in versions rendered as HTML and PDF.

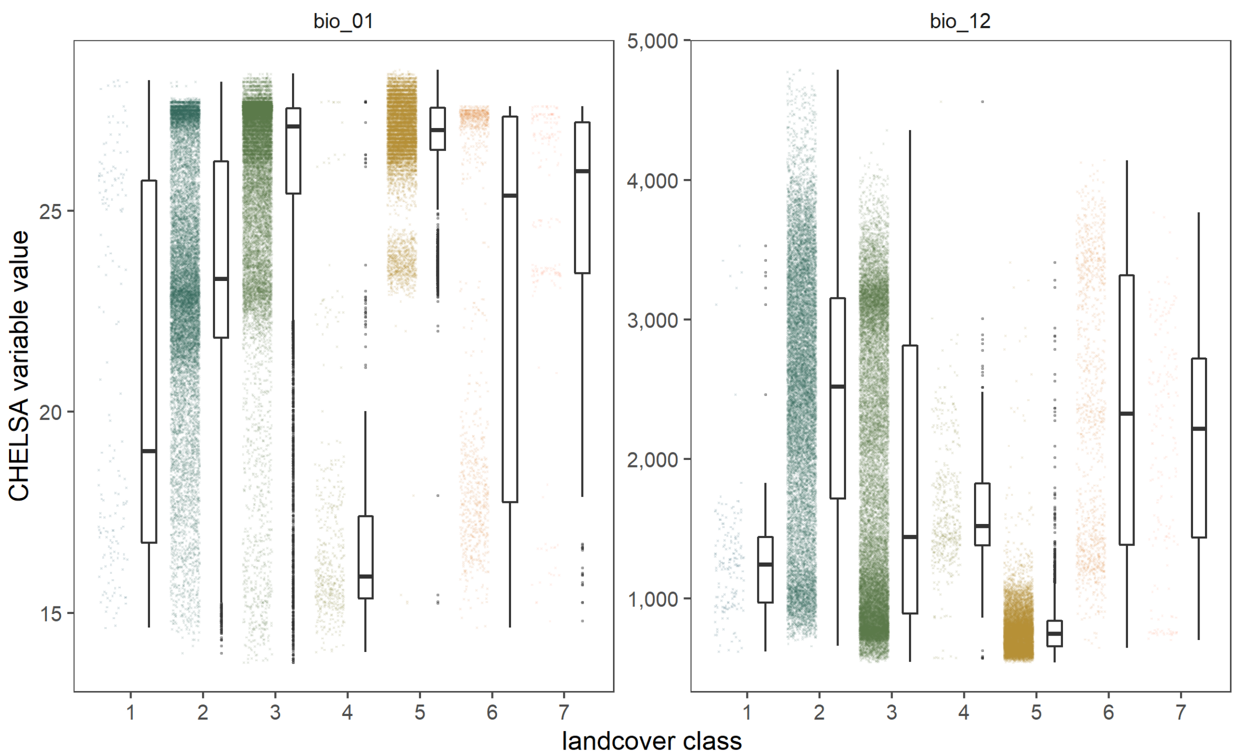

Figure 2: CHELSA climatic variables as a function of landcover class. Grey points in the background represent raw data.

5 Distribution of Observer Expertise

This script plots observer expertise over time (2013-2019) as well as across land cover types. ## Prepare libraries

```r
# load libs
library(raster)
library(glue)
library(purrr)
library(dplyr)
library(tidyr)
library(readr)
library(scales)

# plotting libs
library(ggplot2)
library(ggthemes)
library(scico)

# get ci func
ci <- function(x) {
  qnorm(0.975) * sd(x, na.rm = T) / sqrt(length(x))
}
```

5.1 Load observer expertise scores and checklist covariates

```r
# read in scores and checklist data and link
scores <- read_csv(”data/03_data-obsExpertise-score.csv”)
data <- read_csv(”data/03_data-covars-perChklist.csv”)

data <- left_join(data, scores, by = c(”observer” = ”observer”))
data <- dplyr::select(data, score, nSp, nSoi, landcover, year) %>%
  filter(!is.na(score))
```

5.2 Species observed in relation to observer expertise

```r
# summarise data by rounded score and year
data_summary01 <- data %>%
  mutate(score = plyr::round_any(score, 0.2)) %>%
  dplyr::select(score, year, nSp, nSoi) %>%
  pivot_longer(
    cols = c(”nSp”, ”nSoi”),
    names_to = ”variable”, values_to = ”value”
  ) %>%
  group_by(score, year, variable) %>%
  summarise_at(vars(value), list(~ mean(.), ~ ci(.)))

# make plot and export
fig_nsp_score <-
  ggplot(data_summary01) +
  geom_jitter(
    data = data, aes(x = score, y = nSp),
    col = ”grey”, alpha = 0.2, size = 0.1
  ) +
  geom_pointrange(aes(
    x = score, y = mean,
    ymin = mean - ci, ymax = mean + ci,
```

```
    col = as.factor(variable)
  ),
  position = position_dodge(width = 0.05)
  ) +
  facet_wrap(~year) +
  scale_y_log10() +
  #  coord_cartesian(ylim=c(0,50))+
  scale_colour_scico_d(palette = ”cork”, begin = 0.2, end = 0.8) +
  labs(x = ”CCI”, y = ”Number of Species Reported”) +
  theme_few() +
  theme(legend.position = ”none”)

# export figure
ggsave(filename = ”figs/fig_nsp_score.png”, width = 12, height = 7, device = png(), dpi = 300)
dev.off()
```

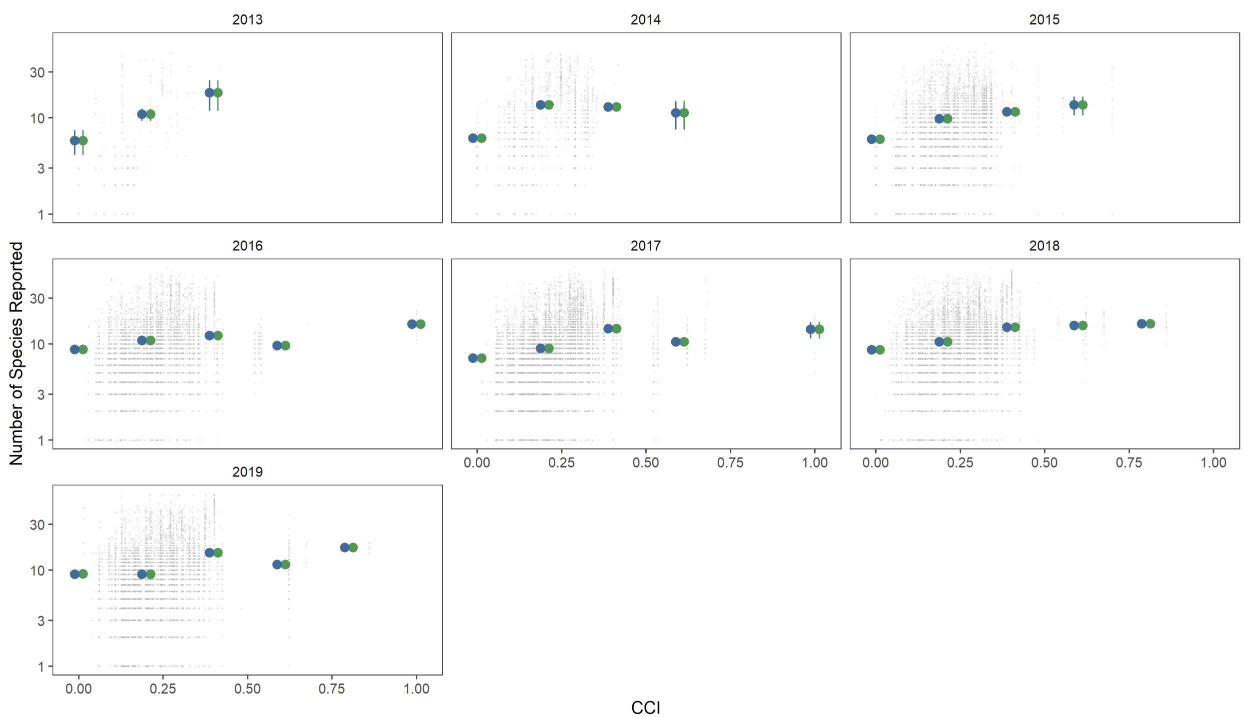

Figure 3: Total number of species (blue) and species of interest to this study (green) reported in checklists from the study area over the years 2013 – 2018, as a function of the expertise score of the reporting observer. Points represent means, with bars showing the 95% confidence intervals; data shown are for expertise scores rounded to multiples of 0.2, and the y-axis is on a log scale. Raw data are shown in the background (grey points).

5.3 Observer expertise in relation to landcover

Figure code is hidden in versions rendered as HTML or PDF.

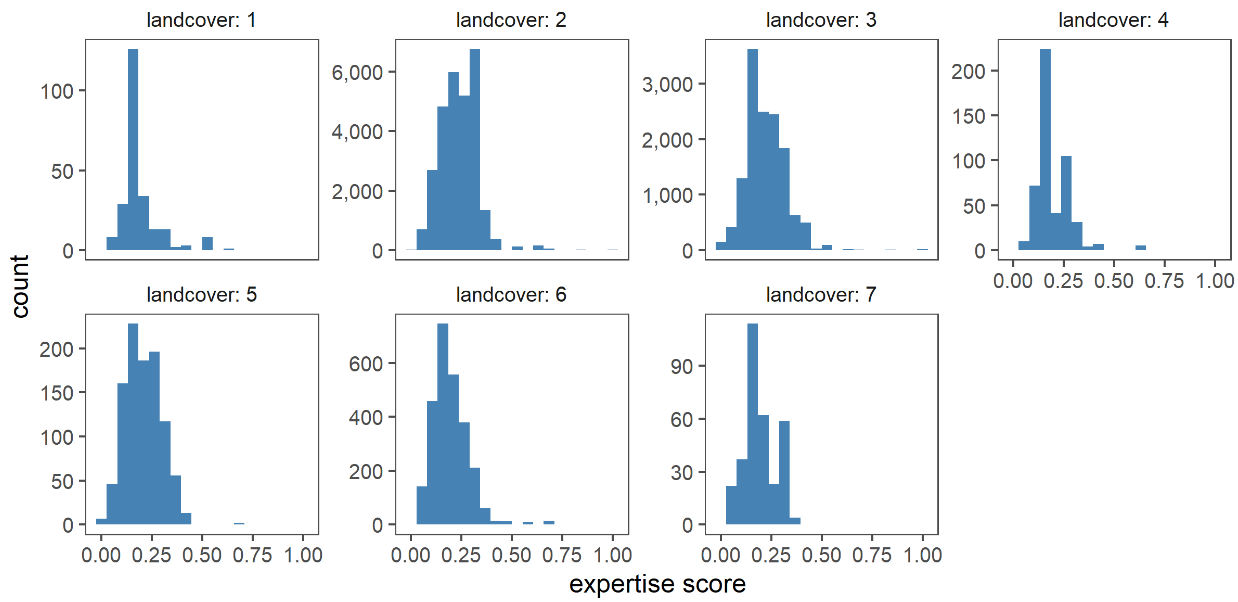

Figure 4: Distribution of expertise scores in the seven landcover classes present in the study site.

6 Spatial Autocorrelation of Climatic Predictors

6.1 Load libraries

```
# load libs
library(raster)
library(gstat)
library(stars)
library(purrr)
library(tibble)
library(dplyr)
library(tidyr)
library(glue)
library(scales)
library(gdalUtils)
library(sf)

# plot libs
library(ggplot2)
library(ggthemes)
library(scico)
library(gridExtra)
library(cowplot)
library(ggspatial)

#' make custom functiont to convert matrix to df
raster_to_df <- function(inp) {

  # assert is a raster obj
  assertthat::assert_that("RasterLayer" %in% class(inp),
    msg = "input is not a raster"
```

```
  )

  coords <- coordinates(inp)
  vals <- getValues(inp)

  data <- tibble(x = coords[, 1], y = coords[, 2], value = vals)

  return(data)
}
```

6.2 Prepare data

```
# list landscape covariate stacks
landscape_files <- ”data/spatial/landscape_resamp01_km.tif”
landscape_data <- stack(landscape_files)

# get proper names
elev_names <- c(”elev”, ”slope”, ”aspect”)
chelsa_names <- c(”bio_01”, ”bio_12”)
names(landscape_data) <- c(elev_names, chelsa_names, ”landcover”)


# get chelsa rasters
chelsa <- landscape_data[[chelsa_names]]
chelsa <- purrr::map(as.list(chelsa), raster_to_df)
```

6.3 Calculate variograms of environmental layers

```
# prep variograms
vgrams <- purrr::map(chelsa, function(z) {
  z <- drop_na(z)
  vgram <- gstat::variogram(value ~ 1, loc = ~ x + y, data = z)
  return(vgram)
})

# save temp
save(vgrams, file = ”data/chelsa/chelsaVariograms.rdata”)

# get variogram data
vgrams <- purrr::map(vgrams, function(df) {
  df %>% select(dist, gamma)
})
vgrams <- tibble(
  variable = chelsa_names,
  data = vgrams
)
```

```
wg <- st_read(”data/spatial/hillsShapefile/Nil_Ana_Pal.shp”) %>%
  st_transform(32643)
bbox <- st_bbox(wg)

# add lamd
library(rnaturalearth)
```

```r
land <- ne_countries(
  scale = 50, type = ”countries”, continent = ”asia”,
  country = ”india”,
  returnclass = c(”sf”)
)

# crop land
land <- st_transform(land, 32643)
```

6.4 Visualise variograms of environmental data

```r
# make ggplot of variograms
yaxis <- c(”semivariance”, ””)
xaxis <- c(””, ”distance (km)”)
fig_vgrams <- purrr::pmap(list(vgrams$data, yaxis, xaxis), function(df, ya, xa) {
  ggplot(df) +
    geom_line(aes(x = dist / 1000, y = gamma), size = 0.2, col = ”grey”) +
    geom_point(aes(x = dist / 1000, y = gamma), col = ”black”) +
    scale_x_continuous(labels = comma, breaks = c(seq(0, 100, 25))) +
    scale_y_log10(labels = comma) +
    labs(x = xa, y = ya) +
    theme_few() +
    theme(
      axis.text.y = element_text(angle = 90, hjust = 0.5, size = 8),
      strip.text = element_blank()
    )
})
# fig_vgrams <- purrr::map(fig_vgrams, ggplot2::ggplotGrob)

# make ggplot of chelsa data
chelsa <- as.list(landscape_data[[chelsa_names]]) %>%
  purrr::map(stars::st_as_stars)

# colour palettes
pal <- c(”bilbao”, ”davos”)
title <- c(
  ”a Annual Mean Temperature”,
  ”b Annual Precipitation”
)
direction <- c(1, 1)
lims <- list(
  range(values(landscape_data$bio_01), na.rm = T),
  range(values(landscape_data$bio_12), na.rm = T)
)
fig_list_chelsa <-
  purrr::pmap(
    list(chelsa, pal, title, direction, lims),
    function(df, pal, t, d, l) {
      ggplot() +
        stars::geom_stars(data = df) +
        geom_sf(data = land, fill = NA, colour = ”black”) +
        geom_sf(data = wg, fill = NA, colour = ”black”, size = 0.3) +
        scale_fill_scico(
```

```
          palette = pal, direction = d,
          label = comma, na.value = NA, limits = l
        ) +
        coord_sf(
          xlim = bbox[c(”xmin”, ”xmax”)],
          ylim = bbox[c(”ymin”, ”ymax”)]
        ) +
        ggspatial::annotation_scale(location = ”tr”, width_hint = 0.4, text_cex = 1) +
        theme_few() +
        theme(
          legend.position = ”top”,
          title = element_text(face = ”bold”, size = 8),
          legend.key.height = unit(0.2, ”cm”),
          legend.key.width = unit(1, ”cm”),
          legend.text = element_text(size = 8),
          axis.title = element_blank(),
          axis.text.y = element_text(angle = 90, hjust = 0.5),
          panel.background = element_rect(fill = ”lightblue”),
          legend.title = element_blank()
        ) +
        labs(x = NULL, y = NULL, title = t)
    }
  )
# fig_list_chelsa <- purrr::map(fig_list_chelsa, ggplotGrob)
```

```
# fig_list_chelsa <- append(fig_list_chelsa, fig_vgrams)
# lmatrix <- matrix(c(c(1, 2, 3, 4, 5), c(1, 2, 3, 4, 5), c(6, 7, 8, 9, 10)),
#   nrow = 3, byrow = T
# )
# plot_grid <- grid.arrange(grobs = fig_list_chelsa, layout_matrix = lmatrix)
#
# ggsave(
#   plot = plot_grid, filename = ”figs/fig_chelsa_variograms.png”,
#   dpi = 300, width = 12, height = 6
# )
# dev.off()

library(patchwork)
fig_variogram <- wrap_plots(append(fig_list_chelsa, fig_vgrams))
ggsave(fig_variogram,
  filename = ”figs/fig_chelsa_variograms.png”,
  dpi = 300,
  width = 6, height = 6
)
```

7 Climatic raster resampling

7.1 Prepare landcover

```
# read in landcover raster location
landcover <- ”data/landUseClassification/classifiedImage-UTM.tif”
# get extent
e <- bbox(raster(landcover))
```

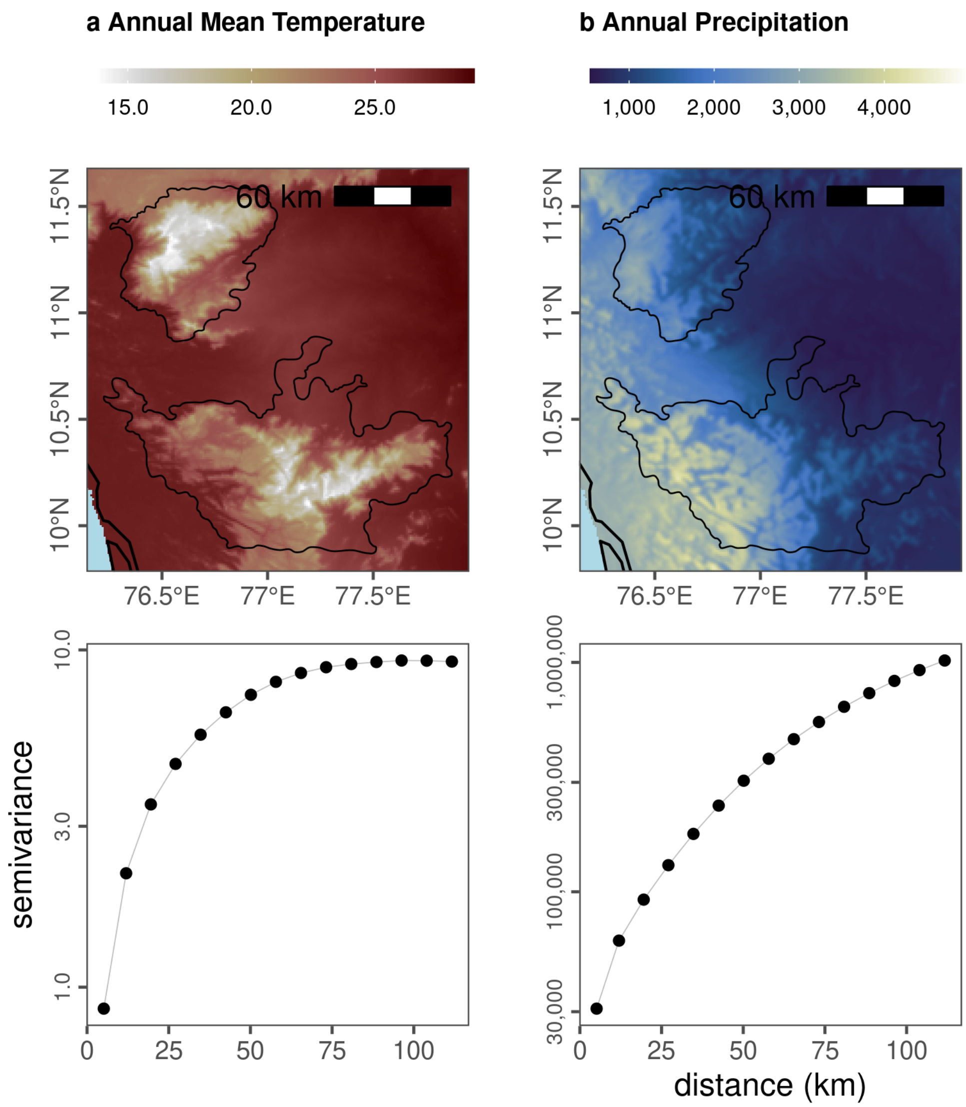

Figure 5: CHELSA rasters with study area outline, and associated semivariograms. Semivariograms are on a log-transformed y-axis.

```r

# init resolution
res_init <- res(raster(landcover))
# res to transform to 1000m
res_final <- map(c(100, 250, 500, 1e3, 2.5e3), function(x) {
  x * res_init
})

# use gdalutils gdalwarp for resampling transform
# to 1km from 10m
for (i in 1:length(res_final)) {
  this_res <- res_final[[i]]
  this_res_char <- stringr::str_pad(this_res[1], 5, pad = ”0”)
  gdalUtils::gdalwarp(
    srcfile = landcover,
    dstfile = as.character(glue(”data/landUseClassification/lc_{this_res_char}m.tif”)),
    tr = c(this_res), r = ”mode”, te = c(e)
  )
}
```

```r
# read in resampled landcover raster files as a list
lc_files <- list.files(”data/landUseClassification/”, pattern = ”lc”, full.names = TRUE)
lc_data <- map(lc_files, raster)
```

7.2 Prepare spatial extent

```r
# load hills
library(sf)
hills <- st_read(”data/spatial/hillsShapefile/Nil_Ana_Pal.shp”)
hills <- st_transform(hills, 32643)
buffer <- st_buffer(hills, 3e4) %>%
  st_transform(4326)
bbox <- st_bbox(hills)
```

7.3 Prepare CHELSA rasters

```r
# list chelsa files
chelsaFiles <- list.files(”data/chelsa/”, full.names = TRUE, pattern = ”*.tif”)

# gather chelsa rasters
chelsaData <- purrr::map(chelsaFiles, function(chr) {
  a <- raster(chr)
  crs(a) <- crs(buffer)
  a <- crop(a, as(buffer, ”Spatial”))
  return(a)
})

# stack chelsa data
chelsaData <- raster::stack(chelsaData)
names(chelsaData) <- c(”chelsa_bio10_01”, ”chelsa_bio10_12”)
```