Human influence in SDMs: Literature Review (Part II)

Veronica F. Frans (email: verofrans@gmail.com)

February 1, 2024

Contents

1 Summary

This is the second R script of the literature review and synthesis for the article entitled, "Gaps and opportunities in modeling human influence on species distributions in the Anthropocene," by Veronica F. Frans and Jianguo Liu.

From June 2018 to April 2019, December 2020 to March 2021, and October 2022 to June 2023, we read 5,163 of 5,177 articles (14 were unavailable) and summarized their modeling procedures into a table.

We first assessed whether an article was relevant (eligible). Relevant articles are those that use correlative presence-absence SDMs and incorporated human predictors in model training and projection. We then recorded components of the modeling procedures onto a Microsoft Excel spreadsheet.This table of information was then exported from Excel as a CSV, and inputted here in this script for data cleaning.

The details of the literature review data fields that were extracted from the accepted articles are found in Table S1 of the corresponding article's supplementary materials. Note that some of those data fields were completed in the full article screening/reading step, while others are filled or finalized during the data cleanup conducted here and in Part II-V R scripts.

Here, in Part II of the synthesis, the following is accomplished:

(1) Upload of full article screening data field description table
(2) PRISMA framework summary
(3) Overview of themes and word associations
(4) Full article screening data cleanup for taxa, domain, and study focus and summary
(5) Summary counts of species modeled
(6) Summary of human vs. environmental predictor use
(7) Summary of qualitative evaluations of human predictor performance

The next script (Part III) uses the full article screening dataset to simplify and synthesize human predictor names across articles for generating figures of human predictor use across various categories and data types.

2 R Setup

We are using R version 4.3.0 (R Core Team 2023).

2.1 Libraries

Load libraries

```
# load libraries
  library("bibliometrix")  # for biblio-analytics
  library("dplyr")         # for table manipulations
  library("scales")        # for scales and formatting
  library("kableExtra")    # for table viewing in Rmarkdown
  library("tidyr")         # for table manipulations
  library("plyr")          # for table manipulations
  library("tidyverse")     # for graphics/table management
  library("ggplot2")       # for graphics
  library("RColorBrewer")  # for graphics
  library("alluvial")      # for graphics
  library("ggforce")       # for graphics (speeds up ggplot)
  library("ggalluvial")    # for graphics
```

```
  library("ggbreak")        # for graphics
  library("patchwork")      # for graphics
  library("tm")             # text analysis for word clouds
  library("SnowballC")      # text analysis for word clouds
  library("wordcloud")      # text analysis for word clouds
  library("ggwordcloud")    # text analysis for word clouds
  library("plotfunctions") # for data visualization
  library("svglite")        # for saving graphics in svg format
  library("PRISMA2020")     # for visualizing the PRISMA framework
```

2.2 Directories

The primary directory is the folder where the `hum_sdm_litrv_r.Rproj` is stored.

```
# create image folder and its directory
  dir.create(paste0("images"))
  image.dir <- paste0("images\\")

# create data folder and its directory
  dir.create(paste0("data"))
  data.dir <- paste0("data\\")
```

2.3 Load data

Upload data derived from Part I and from the full article screening and review.

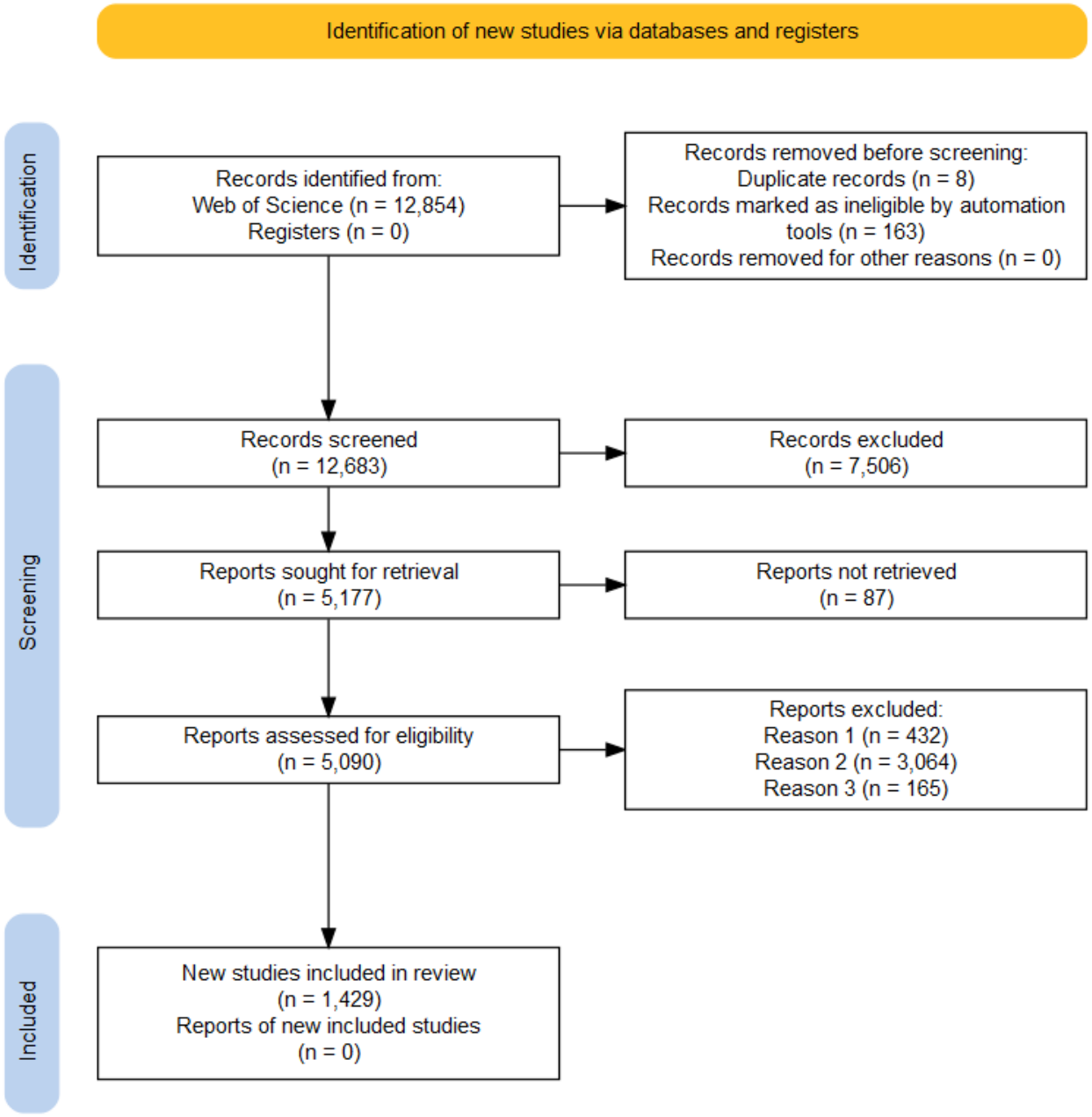
```
# bibtex data frame (with duplicates)
  bibs.df <- read.csv(paste0(data.dir,"bibtex_dataframe_RAW.csv"),
                              header=T, sep=",")

# bibtex data frame (duplicates removed)
  bibs2.df <- read.csv(paste0(data.dir,"bibtex_dataframe_duplicates_removed.csv"),
                              header=T, sep=",")

# abstract screening data set
  abs.df <- read.csv(paste0(data.dir,"abstracts_dataframe_duplicates_removed.csv"),
                              header=T, sep=",")

# final abstract screening data sets
  screened_final <- read.csv(paste0(data.dir,"screened_final.csv"),
                              header=T, sep=",")
  screened_no <- read.csv(paste0(data.dir,"screened_no.csv"),
                              header=T, sep=",")
  screened_yes <- read.csv(paste0(data.dir,"screened_yes.csv"),
                              header=T, sep=",")

# full article screening and review table
  rev.df <- read.csv(paste0(data.dir,"hum_sdm_lit_review_RAW.csv"),
                              header=T, sep=",")
```

3 Summary of article screening results (PRISMA framework)

Using the `PRISMA2020` package, we will visualize the PRISMA process thus far, using the values from the datasets generated in Script I. The outputs are formatted into a table following a CSV template from https://estech.shinyapps.io/prisma_flowdiagram/.

```
# load PRISMA template
  prisma.df <- read.csv(paste0(data.dir,"PRISMA_TEMPLATE.csv"),
                        header=T, sep=",")

# remove duplicates from the full article review dataframe
  revuniq.df <- rev.df[!duplicated(rev.df$uid),]

# insert data for template, based on results above
  # IDENTIFICATION
    # articles found in WoS
      condition <- prisma.df$data=='database_results'                    # template location
      prisma.df$n[condition] <- nrow(bibs.df)                            # update row value
      prisma.df$boxtext[condition] <- 'Web of Science'                    # change box text

    # articles excluded in Part I R code due to year >2021 or blank fields
      condition <- prisma.df$data=='excluded_automatic'                  # template location
      prisma.df$n[condition] <- (nrow(bibs.df)-nrow(abs.df))-(nrow(bibs.df)-nrow(bibs2.df))

    # duplicates removed
      condition <- prisma.df$data=='duplicates'                          # template location
      prisma.df$n[condition] <- nrow(bibs.df)-nrow(bibs2.df)             # update row value

  # SCREENING
    # total abstracts screened
      condition <- prisma.df$data=='records_screened'                    # template location
      prisma.df$n[condition] <- nrow(screened_final)                     # update row value

    # total abstracts rejected
      condition <- prisma.df$data=='records_excluded'                    # template location
      prisma.df$n[condition] <- nrow(screened_no)                        # update row value

    # total abstracts accepted (and sought for retrieval of full articles)
      condition <- prisma.df$data=='dbr_sought_reports'                  # template location
      prisma.df$n[condition] <- nrow(screened_yes)                       # update row value

    # total full articles not retrieved (inaccessible)
      condition <- prisma.df$data=='dbr_notretrieved_reports'            # template location
      prisma.df$n[condition] <- nrow(revuniq.df[revuniq.df$relevant=='UNK'|
                                          revuniq.df$year<2000,]) # update row value

    # total full articles accessed for screening
      condition <- prisma.df$data=='dbr_assessed'
      prisma.df$n[condition] <- length(
                                unique(rev.df$uid))-(
                                  nrow(revuniq.df[revuniq.df$relevant=='UNK'|
                                          revuniq.df$year<2000,])) # update row value

    # total full articles rejected (based on specific reasons)
```

```
      condition <- prisma.df$data=='dbr_excluded'                          # template location
      prisma.df$n[condition] <- paste0(                                    # update row value
        'Reason 1,',length(na.omit(rev.df$reject_code[rev.df$reject_code==1 &
                                                      rev.df$year>=2000])),
        '; Reason 2,',length(na.omit(rev.df$reject_code[rev.df$reject_code==2 &
                                                        rev.df$year>=2000])),
        '; Reason 3,',length(na.omit(rev.df$reject_code[rev.df$reject_code==3 &
                                                        rev.df$year>=2000])))

    # total full articles accepted for synthesis
      condition <- prisma.df$data=='new_studies'                           # template location
      prisma.df$n[condition] <- nrow(revuniq.df[
                                  revuniq.df$relevant=='yes'& revuniq.df$year>=2000,])

# convert to PRISMA package dataset
  pdata <- PRISMA_data(prisma.df)

# visualize PRISMA flow diagram
  PRISMA_plt <- PRISMA_flowdiagram(pdata,
                       # remove fields on previous and other data because not used here
                         previous = FALSE,
                         other = FALSE,
                       # adjust font
                         font = "Arial",
                         fontsize=12
                       )

# save PRISMA flow diagram
  PRISMA_save(PRISMA_plt,filename = paste0(image.dir, 'PRISMA_diagram_R.svg'),
              filetype = 'SVG', overwrite = TRUE)
  PRISMA_save(PRISMA_plt,filename = paste0(image.dir, 'PRISMA_diagram_R.png'),
              filetype = 'PNG', overwrite = TRUE)

# show plot here
  PRISMA_plt
```



3.1 Screened articles that use human predictors outside SDMs

We noted articles that considered human predictors, but only used them outside of the SDM training and projection step as an additional analysis (e.g., masking out urban areas, or running a regression against SDM results).

```
# extract semi-relevant papers published after 2000
  semi.df <- rev.df[(rev.df$relevant=="semi"),]
  semi.df <- semi.df[semi.df$year>=2000,]

# get summary
  paste("number of papers that use human predictors outside SDM:",
```

```
        length(unique(semi.df$uid)))
```

```
## [1] "number of papers that use human predictors outside SDM: 267"
```

4 Summary of literature review findings

Inspect column names of the completed literature review data table.

```
# show column names for data wrangling
  options(width=85) # ensure width
  names(rev.df)
```

```
##  [1] "uid"                    "year"                   "title"
##  [4] "author"                 "journal"                "doi"
##  [7] "relevant"               "study_focus"            "study_area_scale"
## [10] "study_area_country"     "time"                   "hum_time"
## [13] "time_start"             "time_end"               "future_time_start"
## [16] "future_time_end"        "taxa"                   "ttl_species"
## [19] "domain"                 "SDM_algorithm"          "SDM_algorithm_ensembles"
## [22] "past_hum_preds"         "present_hum_preds"      "future_hum_preds"
## [25] "num_past_preds"         "num_present_preds"      "num_future_preds"
## [28] "num_future_hum_preds"   "num_env_preds"          "num_hum_preds"
## [31] "worldclim"              "qual_eval"              "reject_code"
```

4.1 Published articles across years

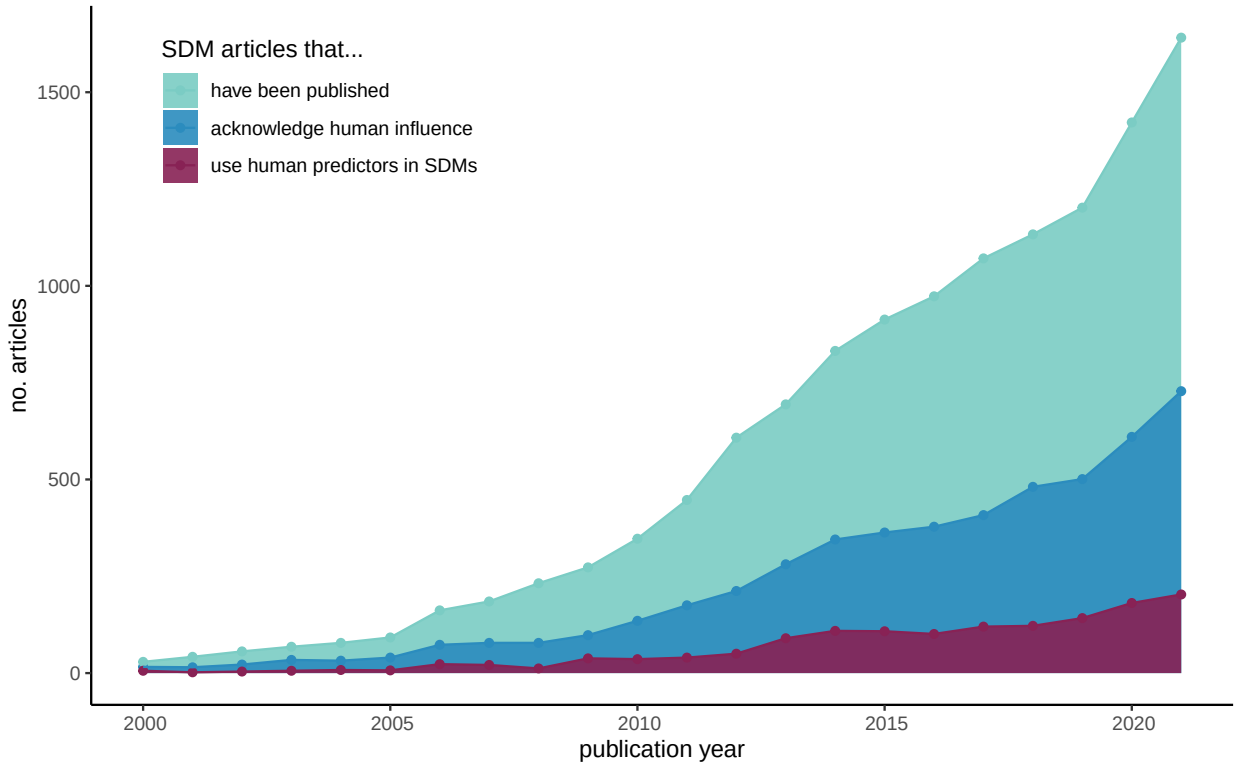
We will next make a figure showing articles across years. It will have two parts. In the first (main) part, we plot papers over time, categorized by three degrees of relevance: (1) SDM papers that have been published overall; (2) SDM papers that acknowledge human influence (at least at the abstract level); and (3) SDM papers that use human predictors in SDMs (determined in the full article screening and synthesis stage of the review). In the second (subfigure inset) part, we plot the percent interest in human influence across SDM articles over time and whether interest in modeling human influence in SDMs has also changed over time.

Note that while our original Web of Science search included publications for all years, we decided to restrict the analysis to the years 2000 and onwards. The truncation of this dataset is done below.

First, the main figure is plotted.

```
# data setup
  # make a copy of WoS search and subsets of papers across review process
    all.df <- screened_final
    scr.df <- screened_final[(screened_final$screened_abstracts=="selected"),]
    yes.df <- rev.df[(rev.df$relevant=="yes"),]

  # remove duplicated fields
    yes.df <- yes.df[!duplicated(yes.df$uid),]

  # remove articles lacking publication year or title
    all.df <- all.df[!(is.na(all.df$doi) & is.na(all.df$title)),]
```

```
  # Add a column to each data frame and relevance (DF for data frame subset) in each row
    all.df$DF <- "have been published"
    scr.df$DF <- "acknowledge human influence"
    yes.df$DF <- "use human predictors in SDMs"

  # Create new tables with only publication year (PY) and relevance (DF)
    all.yrs <- subset(all.df, select=c("year","DF"))
    scr.yrs <- subset(scr.df, select=c("year","DF"))
    yes.yrs <- subset(yes.df, select=c("year","DF"))

  # combine tables
    paper.yrs <- rbind(all.yrs,scr.yrs,yes.yrs)

  # remove papers prior to the year 2000
    paper.yrs$year <- as.integer(paper.yrs$"year")
    paper.yrs <- paper.yrs[paper.yrs$year>=2000,]

  # Get a count of records per year
    paper.yrs <- ddply(paper.yrs, .(year,DF), summarize, count=length(year))

  # Change to factors
    paper.yrs$year <- as.integer(paper.yrs$"year")
    paper.yrs$DF <- as.factor(paper.yrs$"DF")
    paper.yrs$DF <- factor(paper.yrs$DF, levels=c("have been published",
                                                  "acknowledge human influence",
                                                  "use human predictors in SDMs"))

# create figure
  # colors
    yrs.col <- c("#7bccc4","#2b8cbe",'#882255')

  # plot and save
    yrs.fig <- ggplot(paper.yrs, aes(x=year, y=count)) +
                        geom_area(position="identity",
                                  aes(y =count, fill = DF, group = DF),
                                  alpha=0.9) +
                        geom_point(aes(y =count, color = DF, group = DF))+
                         geom_line(aes(y =count, color = DF, group = DF))+
                         theme_classic() +
                         theme(axis.text.x=element_text(angle=0, hjust=.45)) +
                         scale_color_manual(name="SDM articles that...", values=yrs.col) +
                         scale_fill_manual(name="SDM articles that...", values=yrs.col) +
                         scale_x_continuous(breaks = seq(2000, 2020, by = 5)) +
                         xlim(2000,2021) +
                         xlab("publication year") + ylab("no. articles") +
                         theme(legend.position = c(0.2, 0.85))

  # save image
    ggsave(filename=paste0(image.dir,"Papers over time - relevant.png"),
           plot=yrs.fig, height = 5, width = 8)

  # display image
    yrs.fig
```



Get totals.

```
# show sums
  ddply(paper.yrs, .(DF), summarize, count=sum(count))
```

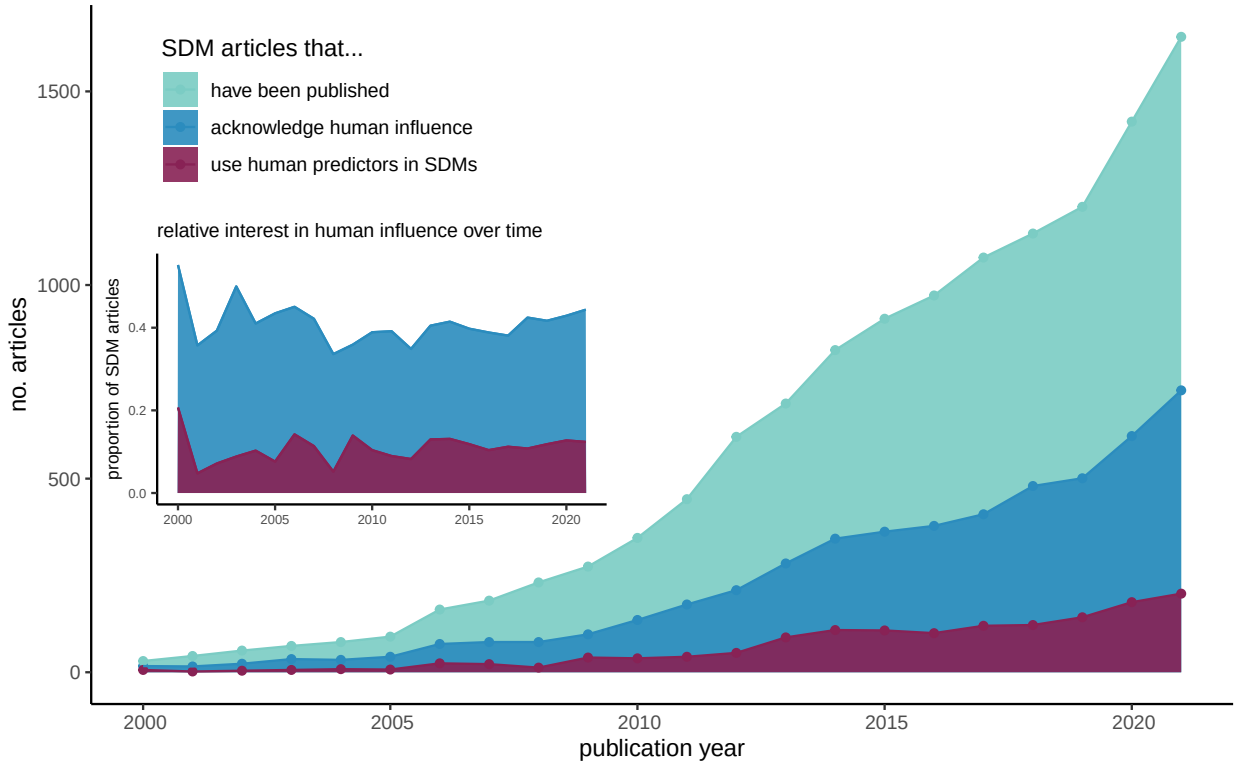
```
##                              DF count
## 1          have been published 12500
## 2  acknowledge human influence  5103
## 3 use human predictors in SDMs  1429
```

Next, calculate percent interest in modeling human influence, compared to all SDM articles and plot as an inset to the main figure.

```
# subset dataframe
  SDM_all <- paper.yrs[paper.yrs$DF=='have been published',]
  hum_all <- paper.yrs[paper.yrs$DF=='acknowledge human influence',]
  hum_SDM <- paper.yrs[paper.yrs$DF=='use human predictors in SDMs',]

# join for interest calculations
  interest_1 <- left_join(SDM_all,hum_all, by='year') # compare all SDMs to relevant abstracts
  interest_2 <- left_join(SDM_all,hum_SDM, by='year')   # compare all SDMs to relevant articles

# convert NA's to zero
  interest_1$count.y[is.na(interest_1$count.y)] <- 0
  interest_2$count.y[is.na(interest_2$count.y)] <- 0

# calc percent interest of mentioning humans in SDM paper abstracts
  interest_1$pc_SDM_all <- interest_1$count.y/interest_1$count.x
```

```
  interest_2$pc_SDM_hum <- interest_2$count.y/interest_2$count.x

# make new dataframe
  interest_1 <- subset(interest_1,select=c('year','pc_SDM_all'))
  interest_2 <- subset(interest_2,select=c('year','pc_SDM_hum'))
  interest   <- left_join(interest_1,interest_2,by='year')

# convert to long format dataframe for plotting
  interest <- gather(interest,              # data
                     level,                 # factor column name
                     percent,               # value column name
                     pc_SDM_all:pc_SDM_hum, # data columns to gather
                     factor_key=TRUE)

# re-organize factors and years
  interest$level <- factor(interest$level,
                           levels=c("pc_SDM_all","pc_SDM_hum"))
  interest$PY <- as.integer(as.character(interest$year))

# plot
  yrs.col <- c('#2b8cbe','#882255')
  yrs.fig2 <- ggplot(interest, aes(x=year, y=percent)) +
                      geom_area(position='identity',
                                aes(y=percent, fill=level, group=level, color=level),
                                alpha=0.9) +
                      theme_classic() +
                      # geom_point(aes(y=percent, fill=level, group=level, color=level,
                      #                size=0.5)) +
                      geom_line(aes(y=percent, fill=level, group=level, color=level)) +
                      scale_color_manual(name='SDM articles that...', values=yrs.col) +
                      scale_fill_manual(name='SDM articles that...', values=yrs.col) +
                      theme(axis.text.x=element_text(angle=0, hjust=0.5, size = 6),
                            axis.text.y=element_text(angle=0, hjust=0, size = 6),
                            axis.title.y=element_text(size = 8)) +
                      ylab('proportion of SDM articles') + xlab('publication year') +
                      scale_x_continuous(breaks = seq(2000, 2020, by = 5)) +
                      ggtitle('relative interest in human influence over time') +
                      theme(legend.position='none',
                            axis.title.x=element_blank(),
                            plot.title = element_text(size = 9)) # hjust=0.5

# save inset image (not shown here)
  ggsave(filename=paste0(image.dir,'Papers over time - proportion.png'),
         plot=yrs.fig2, height = 5, width = 8)

# make plot with inset
  library(cowplot)
  yrs.fig.final <- ggdraw(yrs.fig) +
                         draw_plot(yrs.fig2, x = 0.078, y = .3,
                                   .42,.42)

# save image
  ggsave(filename=paste0(image.dir,'Papers over time.png'),
```

```
        plot=yrs.fig.final, height = 5, width = 8)
  ggsave(filename=paste0(image.dir,'Papers over time.svg'),
        plot=yrs.fig.final, height = 5, width = 8)

# view image
  yrs.fig.final
```



4.2 Word cloud of common terms

From the papers that model human influence in SDMs, make a separate dataframe of keywords and create a wordcloud from the titles, keywords, and abstracts.

```
# Create new tables with only:
# title (TI), author keywords (DE), and abstract (AB)
# (keywords plus [ID] excluded since not from the authors)
  key.df <- merge(all.df,yes.df['uid'],by='uid')
  key.df <- key.df[key.df$year>=2000,]
  key.df  <- subset(key.df, select=c("title","keywords","abstract"))

# Save as a text file, with separators as just a space (" ")
  write.table(key.df, paste0(data.dir,"accepted_paper_keywords.txt"), sep=" ")
```

Load the data frame into the text mining package

```
# read the dataframe
  yes.words <- readLines(paste0(data.dir,"accepted_paper_keywords.txt"))
```

```
# convert it into a corpus
  yes.corp <- Corpus(VectorSource(yes.words))

# inspect the document(this is very long; activate as necessary)
  #inspect(yes.corp)
```

Next, transform the text to remove special characters and clean it up.

```
# transform special characters into a space
  toSpace <- content_transformer(function (x , pattern) gsub(pattern, " ", x))
  yes.corp <- tm_map(yes.corp, toSpace, "/")
  yes.corp  <- tm_map(yes.corp, toSpace, "@")
  yes.corp  <- tm_map(yes.corp, toSpace, "\\|")
  yes.corp  <- tm_map(yes.corp, toSpace, "\t")
  yes.corp  <- tm_map(yes.corp, toSpace, ",")
  yes.corp  <- tm_map(yes.corp, toSpace, ";")
  yes.corp  <- tm_map(yes.corp, toSpace, "_")
  yes.corp  <- tm_map(yes.corp, toSpace, "-")

# convert all text to lowercase
  yes.corp <- tm_map(yes.corp, content_transformer(tolower))

# Remove English common stopwords
  yes.corp <- tm_map(yes.corp, removeWords, stopwords("english"))
  yes.corp <- tm_map(yes.corp, removeWords,
                     c('however','per','also','may',
                       'will','well', 'can','non'))

# Remove numbers
  yes.corp <- tm_map(yes.corp, removeNumbers)

# Remove punctuations
  yes.corp <- tm_map(yes.corp, removePunctuation,preserve_intra_word_dashes = TRUE)

# ensure land use is one term by hyphenating all "land use" to "land-use"
  toHyphen <- content_transformer(function (x , pattern) gsub(pattern, "land-use", x))
  yes.corp <- tm_map(yes.corp, toHyphen, "land use")

# Remove additional spaces
  yes.corp <- tm_map(yes.corp, stripWhitespace)
```

Build a term document matrix, which is a table with word frequencies.

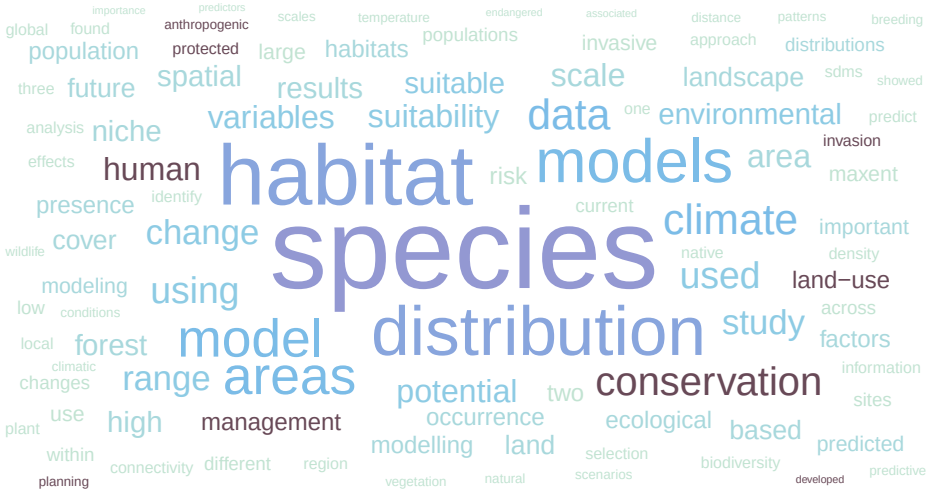
```
# Create term matrix
  term.mtrx <- TermDocumentMatrix(yes.corp)
  mtrx <- as.matrix(term.mtrx)
  rows.mtrx <- sort(rowSums(mtrx),decreasing=TRUE)
  terms.df <- data.frame(word = names(rows.mtrx),freq=rows.mtrx)

# Show frequency of the top 20 terms
  options(width=85) # ensure width
  head(terms.df, 20)
```

```
##                         word freq
## species              species 6909
## habitat              habitat 4610
## distribution    distribution 3953
## models               models 2860
## model                 model 2288
## areas                 areas 2150
## conservation   conservation 1838
## climate             climate 1660
## data                   data 1476
## suitability     suitability 1318
## variables         variables 1310
## change               change 1302
## using                 using 1275
## used                   used 1252
## potential         potential 1245
## environmental environmental 1243
## range                 range 1180
## study                 study 1101
## human                 human 1082
## suitable           suitable 1067
```

Generate a word cloud of the top 100 words across the accepted papers. We will also highlight any human-related words in the image.

```
# subset of top 100 most frequent words
  terms.df2 <- terms.df[1:100,]

# set up logarithmic bin breaks from 1 to 10 (need 6 bins for color assignments)
  bin_breaks <- as.integer(10^seq(log10(1), log10(101), length.out = 7))

# assign colors to top 100, using bin breaks
  terms.df2$row <- 1:100
  terms.df2$colors <- cut(terms.df2$row, breaks = bin_breaks,
                          labels = c("#9398D2", "#88A5DD", "#7BBCE7", "#8DCBE4",
                                     "#A8D8DC", "#C2E3D2" ), right = FALSE)

# assign a dark red color to highlight anthropogenic terms
  #change to character
    terms.df2$colors <- as.character(terms.df2$colors)

  # select terms from top 100 list and edit color
    terms.df2$colors[grepl(
    "anthropogenic|human|land-use|conservation|management|protected|developed|invasion|planning",
            terms.df2$word)] <- "#684957"

# set seed for image
  set.seed(1234)

# generate word cloud
  cloud_fig <- ggplot(terms.df2,
                      aes(label = word, size = freq, color= colors)) +
                      geom_text_wordcloud_area(shape = "square") +
                      scale_size_area(max_size = 15) +
```

```
                    theme_minimal()+
                    scale_color_identity()

# save image
  ggsave(filename=paste0(image.dir,"word_cloud_labeled.svg"),
         plot=cloud_fig,
         height = 4, width = 5)
  ggsave(filename=paste0(image.dir,"word_cloud_labeled.png"),
         plot=cloud_fig,
         height = 4, width = 5, dpi = 600)

# show here
  cloud_fig
```



4.3 Anthropogenic word associations

Find correlative word associations with the terms 'anthropogenic', 'human', 'people', 'loss', 'impact', 'use', 'urban', and 'risk', using a correlation limit of 0.2 (if very few) or 0.3 (if many).

```
options(width=85) # ensure width
findAssocs(term.mtrx, terms = "anthropogenic", corlimit = 0.2)

options(width=85) # ensure width
findAssocs(term.mtrx, terms = "human", corlimit = 0.2)

options(width=85) # ensure width
findAssocs(term.mtrx, terms = "people", corlimit = 0.3)
```

```r
options(width=85) # ensure width
findAssocs(term.mtrx, terms = "loss", corlimit = 0.2)

options(width=85) # ensure width
findAssocs(term.mtrx, terms = "impact", corlimit = 0.2)

options(width=85) # ensure width
findAssocs(term.mtrx, terms = "urban", corlimit = 0.3)

options(width=85) # ensure width
findAssocs(term.mtrx, terms = "risk", corlimit = 0.3)
```

```
## $anthropogenic
##          eulemur            lemur        cascading        coincides            corax
##             0.33             0.31             0.27             0.27             0.27
##     disentangled       enterprise          greener multiplicatively            raven
##             0.27             0.27             0.27             0.27             0.27
##           ravens       subsidized       tridentata        subsidies        expansive
##             0.27             0.27             0.27             0.26             0.24
##        featuring     presentation        semiurban        residence
##             0.23             0.22             0.22             0.21
##
## $human
##     footprint   disturbance    activities     unfilling     influence    expansions
##          0.34          0.27          0.27          0.26          0.25          0.24
##    tolerances      compound dissemination          nbms       novelty          shop
##          0.24          0.23          0.23          0.23          0.23          0.23
##        slider           tse        humans     dominated  conservatism     trachemys
##          0.23          0.23          0.22          0.22          0.22          0.21
##      conflict
##          0.20
##
## $people
##      achieveminimum          agronomist           authority               bite
##                0.48                0.48                0.48                0.48
##        condensation             despair           emergency         fivecommon
##                0.48                0.48                0.48                0.48
##                 gpu             healers            hospital           hospitals
##                0.48                0.48                0.48                0.48
##       improvisation           inflicted          inflicting                ols
##                0.48                0.48                0.48                0.48
##           panchayat               phcs         sensitizing             sikkim
##                0.48                0.48                0.48                0.48
##      snakeshospital            therisk             torment       unintentional
##                0.48                0.48                0.48                0.48
##             victims           snakebite            venomous              author
##                0.48                0.47                0.45                0.42
##              belief              death        envenomation          facilities
##                0.42                0.42                0.42                0.41
##            medicine             medical              health               cases
##                0.41                0.39                0.37                0.37
##     epidemiological             snakes  alligatoralligator     contemporaneous
##                0.36                0.36                0.36                0.36
```

```
## crocodilecrocodylus         crocodilian        crocodilians              digest
##                0.36                0.36                0.36                0.36
##          ectotherms          endotherms  mississippiensisin         niloticusin
##                0.36                0.36                0.36                0.36
##            alliance             amenity          blossoming              bumble
##                0.36                0.36                0.36                0.36
##         deprivation            edinburgh          engagement           gardening
##                0.36                0.36                0.36                0.36
##          greenspace               homes           municipal              pledge
##                0.36                0.36                0.36                0.36
##                 win             attacks              attack           compiling
##                0.36                0.35                0.34                0.33
##                gram       preparedness
##                0.33                0.33
##
## $loss
##        brazils           para          paras            ref           toll
##           0.30           0.30           0.30           0.30           0.30
##            lost     deforested         change      arunachal        bellied
##            0.27           0.24           0.23           0.22           0.22
##           heron       insignis          lohit       namdapha       sticking
##            0.22           0.22           0.22           0.22           0.22
##           verge            wbh        classic    intervening       rubecula
##            0.22           0.22           0.22           0.22           0.22
##   scelorchilus      sharpened  fragmentation      wildfires        flowing
##            0.22           0.22           0.21           0.21           0.20
##     equivalent evolutionarily    cornerstone
##            0.20           0.20           0.20
##
## $impact
##                andaman             bridgehead bullfroghoplobatrachus
##                   0.29                   0.29                   0.29
##           dicroglossid                   forh    genushoplobatrachus
##                   0.29                   0.29                   0.29
##         hoplobatrachus              mascarene            necessitate
##                   0.29                   0.29                   0.29
##                nicobar                    ofh             recipients
##                   0.29                   0.29                   0.29
##                  snout              tigerinus           tigerinusand
##                   0.29                   0.29                   0.29
##            tigerinusas                   vent                  bases
##                   0.29                   0.29                   0.29
##                    eta              skylarks                 skylark
##                   0.29                   0.29                   0.28
##        infrastructural              appraise               heighten
##                   0.26                   0.25                   0.25
##                  citri             diaphorina                 lagged
##                   0.25                   0.25                   0.25
##                psyllid             cumulative                    cti
##                   0.25                   0.24                   0.24
##                profile                   wind                climate
##                   0.23                   0.23                   0.21
##                  farms              leveraged            concomitant
##                   0.21                   0.21                   0.20
```

```
##               averages                 divoire
##                   0.20                    0.20
##
## $urban
##  spaces   ninox strenua    apex
##    0.33    0.33    0.31    0.30
##
## $risk
##    invasion depredation
##        0.33        0.32
```

5 Study focus

5.1 Number of studies among accepted papers

Here, we synthesize and edit words that summarize the focus of each study. **Note that the total number of studies will differ from the total number of accepted papers in the review, as studies counted for each domain (terrestrial, freshwater, and/or marine) modeled in the paper.** In other words, an accepted article that models human influence on species distributions in a terrestrial domain is just one study; meanwhile, an article that models human influence on species distributions in both terrestrial and freshwater environments (as separate models, typically, for e.g., multi-species papers) counts as two studies.

```
# subset of accepted papers
# (we redo `yes.df` since from here onward, we need UID duplicates with domain, etc.)
  yes.df <- rev.df[(rev.df$relevant=="yes"),]
  yes.df <- yes.df[(yes.df$year>=2000),]

# get count of studies
  paste("total number of accepted papers:",length(unique(yes.df$uid)))
```

```
## [1] "total number of accepted papers: 1429"
```

```
  paste("total number of studies:",nrow(yes.df))
```

```
## [1] "total number of studies: 1441"
```

5.2 Synthesize terms for study focus

```
# extract subset of table of relevant papers
  focus.df <- data.frame(subset(yes.df,
                          select = c("uid","relevant",
                                     "domain","taxa","study_focus")))

# convert to factors
  focus.df[1:5] <- lapply(focus.df[1:5],factor)

# get summary
  summary(focus.df$study_focus)
```

```
##             coexistence              collisions            conservation
##                       7                      13                     341
##          conservationn              distubance             disturbance
##                       1                       1                     136
##               economics             exploratory         habiitat_change
##                      15                     339                       1
##          habitat_change            habitat_loss human-wildlife_conflict
##                      84                       1                      33
##       human_disturbance         human_economics     human_food_security
##                       4                      18                      23
##             human_gain            human_health  human_illegal_activity
##                      21                      77                       1
##  human_land_abandonment         illegal_activity               invasians
##                       1                       4                       1
##              invasions              protection             recreation
##                     212                       4                       1
##          reintroduction             restoration          urban_planning
##                      70                      31                       1
```

These data will need to be edited for typos, etc. To do this, first we make a dataframe of the edits to make, and then run a for-loop across the dataframe to incorporate these edits.

```
# create a dataframe of edits
  # dataframe with all names repeated in both columns
    edit.df <- data.frame(original=levels(as.factor(focus.df$study_focus)),
                          edit=levels(as.factor(focus.df$study_focus)))

  # confirm start and end of character strings in original column
    edit.df$original <- paste("^", edit.df$original, "$", sep = "")

  # list of original terms to edit
    edit.df$edit[edit.df$edit=='coexistence'] <- 'conservation'
    edit.df$edit[edit.df$edit=='conservationn'] <- 'conservation'
    edit.df$edit[edit.df$edit=='protection'] <- 'conservation'
    edit.df$edit[edit.df$edit=='urban_planning'] <- 'conservation'
    edit.df$edit[edit.df$edit=='collisions'] <- 'conflict/collisions'
    edit.df$edit[edit.df$edit=='human_illegal_activity'] <- 'conflict/collisions'
    edit.df$edit[edit.df$edit=='illegal_activity'] <- 'conflict/collisions'
    edit.df$edit[edit.df$edit=='human-wildlife_conflict'] <- 'conflict/collisions'
    edit.df$edit[edit.df$edit=='distubance'] <- 'disturbance/habitat change'
    edit.df$edit[edit.df$edit=='disturbance'] <- 'disturbance/habitat change'
    edit.df$edit[edit.df$edit=='habitat_change'] <- 'disturbance/habitat change'
    edit.df$edit[edit.df$edit=='habiitat_change'] <- 'disturbance/habitat change'
    edit.df$edit[edit.df$edit=='human_disturbance'] <- 'disturbance/habitat change'
    edit.df$edit[edit.df$edit=='human_land_abandonment'] <- 'disturbance/habitat change'
    edit.df$edit[edit.df$edit=='habitat_loss'] <- 'disturbance/habitat change'
    edit.df$edit[edit.df$edit=='economics'] <- 'food/economics'
    edit.df$edit[edit.df$edit=='human_economics'] <- 'food/economics'
    edit.df$edit[edit.df$edit=='human_food_security'] <- 'food/economics'
    edit.df$edit[edit.df$edit=='human_gain'] <- 'food/economics'
    edit.df$edit[edit.df$edit=='recreation'] <- 'food/economics'
    edit.df$edit[edit.df$edit=='reintroduction'] <- 'reintroduction/restoration'
    edit.df$edit[edit.df$edit=='restoration'] <- 'reintroduction/restoration'
    edit.df$edit[edit.df$edit=='invasians'] <- 'invasions'
```

```
    edit.df$edit[edit.df$edit=='human_health'] <- 'human health/safety'

# Edit data fields
  # for-loop to edit values in dataframe
    for (i in 1:nrow(edit.df)) {
                                # get the current row from edit.df
                                  edit_row <- edit.df[i, ]

                                # extract the values for replacement
                                  search_value <- edit_row$original
                                  replace_value <- edit_row$edit

                                # update the corresponding values in the dataframe
                                  focus.df <- data.frame(lapply(focus.df, function(x) {
                                                gsub(search_value, replace_value, x)
                                }))
    }

# Convert to factor
  focus.df$study_focus <- as.factor(focus.df$study_focus)
  focus.df$domain <- as.factor(focus.df$domain)

# get summary
  options(width=85) # ensure width
  summary(focus.df$study_focus)
```

```
##        conflict/collisions              conservation disturbance/habitat change
##                         51                       354                         228
##                exploratory            food/economics        human health/safety
##                        339                        78                          77
##                  invasions reintroduction/restoration
##                        213                       101
```

6 Study taxa

6.1 Synthesize taxa names

First, separate data frames for each domain will be used, as this will help with creating new rows per paper.

```
# Split multiple taxa listed in a row into other new rows
  library("tidyr")
  taxa.df <- separate_rows(focus.df, taxa, sep="; ", convert = TRUE)
  taxa.df$taxa <- as.factor(taxa.df$taxa)
  taxa.df$relevant <- as.factor(taxa.df$relevant)

# get summary
  options(width=85) # ensure width
  summary(taxa.df$taxa)
```

```
##           algae      amphibians      arthropods     assemblages            bird
##               1              62               3               1               1
```

```
##           birds      bryophytes     butterflies          corals     crustaceans
##             344               2               1               1              15
##         diatoms            fish           fungi          fungus          insect
##               1              76               2               2               1
##         insects   invertebrates          mamma         mammals      marsupials
##             148              43               1             504              10
## micro-organisms  microorganisms        molluscs         plants        reptiles
##               1              27              28             175              73
##           shrub          shrubs          trees          worms
##               2              19              51               2
```

Next, we edit the names, following the same editing procedure as above, using an edit table and for-looping through the dataset.

```
# create a dataframe of edits
  # dataframe with all names repeated in both columns
    edit.df <- data.frame(original=levels(as.factor(taxa.df$taxa)),
                          edit=levels(as.factor(taxa.df$taxa)))

  # confirm start and end of character strings in original column
    edit.df$original <- paste("^", edit.df$original, "$", sep = "")

 # list of original terms to edit
    edit.df$edit[edit.df$edit=='bird'] <- 'birds'
    edit.df$edit[edit.df$edit=='bryophytes'] <- 'herbaceous plants'
    edit.df$edit[edit.df$edit=='plants'] <- 'herbaceous plants'
    edit.df$edit[edit.df$edit=='shrub'] <- 'trees/shrubs'
    edit.df$edit[edit.df$edit=='shrubs'] <- 'trees/shrubs'
    edit.df$edit[edit.df$edit=='trees'] <- 'trees/shrubs'
    edit.df$edit[edit.df$edit=='mamma'] <- 'mammals'
    edit.df$edit[edit.df$edit=='marsupials'] <- 'mammals'
    edit.df$edit[edit.df$edit=='algae'] <- 'microorganisms'
    edit.df$edit[edit.df$edit=='diatoms'] <- 'microorganisms'
    edit.df$edit[edit.df$edit=='assemblages'] <- 'microorganisms'
    edit.df$edit[edit.df$edit=='fungi'] <- 'microorganisms'
    edit.df$edit[edit.df$edit=='fungus'] <- 'microorganisms'
    edit.df$edit[edit.df$edit=='micro-organisms'] <- 'microorganisms'
    edit.df$edit[edit.df$edit=='arthropods'] <- 'invertebrates'
    edit.df$edit[edit.df$edit=='insect'] <- 'invertebrates'
    edit.df$edit[edit.df$edit=='insects'] <- 'invertebrates'
    edit.df$edit[edit.df$edit=='corals'] <- 'invertebrates'
    edit.df$edit[edit.df$edit=='butterflies'] <- 'invertebrates'
    edit.df$edit[edit.df$edit=='crustaceans'] <- 'invertebrates'
    edit.df$edit[edit.df$edit=='molluscs'] <- 'invertebrates'
    edit.df$edit[edit.df$edit=='worms'] <- 'invertebrates'

# Edit data fields
  # for-loop to edit values in dataframe
    for (i in 1:nrow(edit.df)) {
                                  # get the current row from edit.df
                                    edit_row <- edit.df[i, ]

                                  # extract the values for replacement
                                    search_value <- edit_row$original
```

```
                                  replace_value <- edit_row$edit

                                # update the corresponding values in the dataframe
                                  taxa.df <- data.frame(lapply(taxa.df, function(x) {
                                                gsub(search_value, replace_value, x)
                                }))
    }

# Convert to factor
  taxa.df$taxa <- as.factor(taxa.df$taxa)
  taxa.df$domain <- as.factor(taxa.df$domain)

# get summary
  options(width=85) # ensure width
  summary(taxa.df$taxa)
```

```
##        amphibians             birds              fish herbaceous plants
##                62               345                 76               177
##     invertebrates           mammals    microorganisms          reptiles
##               242               515                 35                73
##      trees/shrubs
##                72
```

Next, make a summary and table and save as a CSV.

```
# Make a summary table with count of papers across taxa, separated by domain and study focus
  library("plyr")
  taxa.sum <- ddply(taxa.df, .(domain,taxa,study_focus),
                    summarize,
                  # count number of studies using each predictor
                    count=length(taxa),
                  # list of paper UIDs that used the predictors
                    papers=paste(unique(uid),collapse="; "),
                  # count number of papers using each predictor
                    count_papers=paste(length(unlist(strsplit(papers, ";"))))
                    )

# save as a CSV
  write.csv(taxa.sum,paste0(data.dir,"domain_taxa_focus_count_papers.csv"),
            row.names = FALSE)
```

7 Domain, taxa and study focus summaries

7.1 Proportion of studies across domains

```
# change to integer
  taxa.sum$count <- as.integer(taxa.sum$count)

# get sum
  totals <- sum(taxa.sum$count)
```

domain	count	percent
terrestrial	1375	86.10
freshwater	184	11.52
marine	38	2.38

taxa	count	percent
mammals	515	32.25
birds	345	21.60
invertebrates	242	15.15
herbaceous plants	177	11.08
fish	76	4.76
reptiles	73	4.57
trees/shrubs	72	4.51
amphibians	62	3.88
microorganisms	35	2.19

```
# get summary
  domain_ttls <- ddply(taxa.sum, .(domain), summarize,
                       # total count of entries
                         count=sum(count),
                         percent=round((sum(count)/totals)*100,digits = 2)) %>%
                # sort
                  arrange(desc(percent))

# view table
  kableExtra::kbl(domain_ttls,booktabs=T) %>%
      kable_styling(latex_options = c("striped"))
```

7.2 Proportion of studies across taxa

```
# get sum
  totals <- sum(taxa.sum$count)

# get summary
  taxa_ttls <- ddply(taxa.sum, .(taxa), summarize,
                     # total count of entries
                       count=sum(count),
                       percent=round((sum(count)/totals)*100,digits = 2)) %>%
              # sort
                arrange(desc(percent))

# view table
  kableExtra::kbl(taxa_ttls,booktabs=T) %>%
      kable_styling(latex_options = c("striped"))
```

study_focus	count	percent
conservation	378	23.67
exploratory	370	23.17
invasions	287	17.97
disturbance/habitat change	238	14.90
reintroduction/restoration	108	6.76
food/economics	82	5.13
human health/safety	80	5.01
conflict/collisions	54	3.38

7.3 Proportion of studies across study focus

```
# get sum
  totals <- sum(taxa.sum$count)

# get summary
  foc_ttls <- ddply(taxa.sum, .(study_focus), summarize,
                    # total count of entries
                      count=sum(count),
                      percent=round((sum(count)/totals)*100,digits = 2)) %>%
              # sort
                arrange(desc(percent))

# view table
  kableExtra::kbl(foc_ttls,booktabs=T) %>%
      kable_styling(latex_options = c("striped"))
```

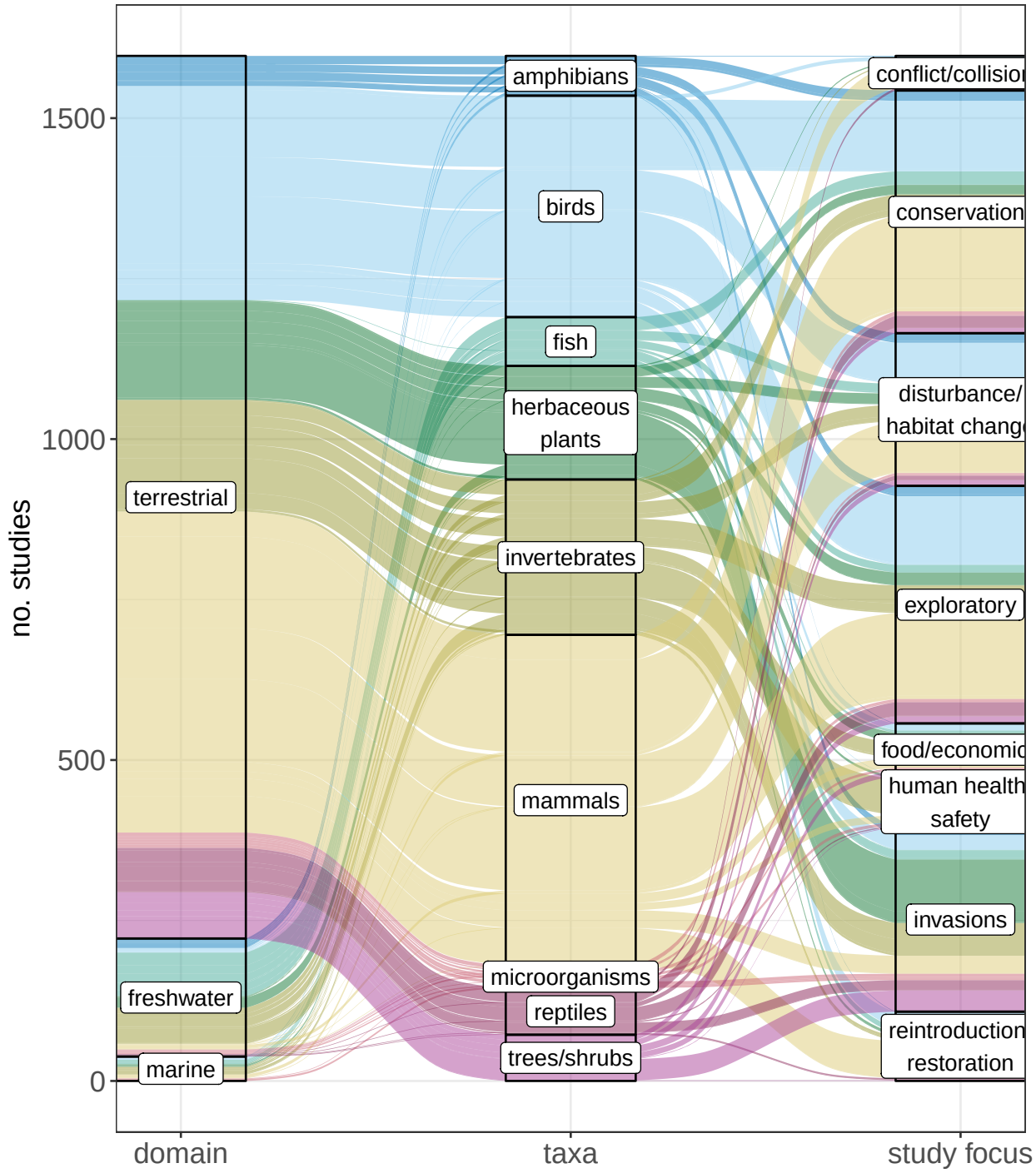
7.4 Alluvial plot across domain, taxa and study focus

An alluvial plot takes a population of dataset and shows relationships across its categorical features. Here, we will make an alluvial plot of domain, taxa and study focus.

```
# edit dataset
  # make new lines for long study focus names
    taxa.sum <- data.frame(lapply(taxa.sum, function(x) {
                                  gsub("^disturbance/habitat change$",
                                       "disturbance/\nhabitat change",x)
                                  }))
    taxa.sum <- data.frame(lapply(taxa.sum, function(x) {
                                  gsub("^reintroduction/restoration$",
                                       "reintroduction/\nrestoration",x)
                                  }))
    taxa.sum <- data.frame(lapply(taxa.sum, function(x) {
                                  gsub("^human health/safety$","human health/\nsafety",x)
                                  }))
    taxa.sum <- data.frame(lapply(taxa.sum, function(x) {
                                  gsub("^climate_change$","climate change",x)
                                  }))
    taxa.sum <- data.frame(lapply(taxa.sum, function(x) {
```

```
                                gsub("^herbaceous plants$","herbaceous\nplants",x)
                                }))

  # reorder columns
    taxa.sorted <- taxa.sum[order(taxa.sum$domain,
                                  taxa.sum$taxa,
                                  taxa.sum$study_focus),]
    taxa.sorted$n <- seq.int(nrow(taxa.sorted))
    #str(taxa.sorted)
    taxa.sorted$count <- as.integer(as.character(taxa.sorted$count))
    taxa.sorted$domain <- as.factor(as.character(taxa.sorted$domain))
    taxa.sorted$taxa <- as.factor(as.character(taxa.sorted$taxa))
    taxa.sorted$study_focus <- as.factor(as.character(taxa.sorted$study_focus))

  # reorder domain factors
    taxa.sorted$domain <- factor(taxa.sorted$domain,
                                 levels = c("terrestrial", "freshwater", "marine"))

# colors
  taxa.col <- c('#0077BB','#88CCEE','#44AA99','#117733','#999933',
                '#DDCC77','#CC6677','#882255','#AA4499','#BBBBBB')  #colorblind-friendly

# alluvial plot
  allu.plot <- ggplot(as.data.frame(taxa.sorted),
                      aes(y=count, axis1=domain,
                          axis2=taxa, axis3=study_focus))+
                      geom_alluvium(aes(fill=taxa), width = 1/3,
                                    knot.pos = 0.3)+
                      geom_stratum(width=1/3,
                                   fill=NA, color="black")+
                      geom_label(stat = "stratum",
                                 aes(label = after_stat(stratum),label.size=NA,
                                     label.border="white")) +
                      scale_x_discrete(limits = c("domain", "taxa", "study focus"),
                                       expand = c(.05, .05)) +
                      ylab('no. studies') +
                      scale_fill_manual(name="taxa", values=taxa.col)+
                      theme_bw()+
                      theme(legend.position="none", axis.text = element_text(size=14),
                            axis.title.y = element_text(size=14))

  ggsave(filename=paste0(image.dir,"taxa_focus_alluvial.svg"),
         plot=allu.plot,
         height = 11, width = 10)
  ggsave(filename=paste0(image.dir,"taxa_focus_alluvial_600dpi.png"),
         plot=allu.plot,
         height = 11, width = 10, dpi = 600)
```

```
# display here
  allu.plot
```



8 Total number of species modeled

Most studies reported the numbers of species modeled. We can take the sum of those to get an estimate of the species modeled across all studies. As this number is not representative of unique species across all studies (e.g., many studies on the same species of mosquito have been done), we still believe we can safely assume that the number below represents the minimal number of total species modeled across all papers.

This is because many papers did not report species numbers. Such papers are those modeling at higher taxonomic levels, or worked with large multi-species datasets.

```
# make data field numeric
  yes.df$ttl_species <- as.integer(yes.df$ttl_species)

# get sum
  paste('total reported sum of species across studies:',
        sum(yes.df$ttl_species, na.rm = TRUE))

# get min and max
  paste('minimum reported number of species per study:',
        min(yes.df$ttl_species, na.rm = TRUE))
  paste('maximum reported number of species per study:',
        max(yes.df$ttl_species, na.rm = TRUE))
```

```
## [1] "total reported sum of species across studies: 58161"
## [1] "minimum reported number of species per study: 1"
## [1] "maximum reported number of species per study: 7427"
```

How many papers modeled only one species?

```
# how many papers modeled only one species
  paste('number of single-species studies:',
        length(yes.df$ttl_species[yes.df$ttl_species==1]))
```

```
## [1] "number of single-species studies: 933"
```

```
# proportion of studies modeling only one species
  paste('proportion of single-species studies:',
        round(length(
          yes.df$ttl_species[yes.df$ttl_species==1])/length(unique(yes.df$uid)),
              digits = 4))
```

```
## [1] "proportion of single-species studies: 0.6529"
```

Also get count of papers that did not report the number of species modeled.

```
# how many papers did not report number of species
  paste('number of papers not reporting species numbers:',
        length(yes.df$ttl_species[is.na(yes.df$ttl_species)]))
```

```
## [1] "number of papers not reporting species numbers: 74"
```

```
# proportion of studies not reporting number of species
  paste('proportion of papers not reporting species numbers:',
        round(length(
          yes.df$ttl_species[is.na(yes.df$ttl_species)])/length(unique(yes.df$uid)),
              digits = 4))
```

```
## [1] "proportion of papers not reporting species numbers: 0.0518"
```

Which papers did not report the number of species?

```
# view table
  options(width=85) # ensure width
  yes.df$uid[is.na(yes.df$ttl_species)]
```

```
##  [1]   172   327   541   985  1184  1244  1484  1722  1726  1965  2068  2129  3019
## [14]  3134  3366  3452  3585  3585  3617  3638  3642  4724  4846  4905  5094  5470
## [27]  5936  5976  6191  6191  6191  6482  6484  6523  6544  6622  6978  6978  7252
## [40]  7432  7564  7575  7605  7605  7605  7845  8112  8193  8261  8261  8261  8261
## [53]  8451  9082  9360  9392  9545  9577  9590  9785  9879  9879 10039 10699 10747
## [66] 10883 10899 10935 11259 11449 11832 11861 11871 12386
```

9 Comparing the proportionate use of human and environmental predictors in SDMs

We next compare the amount of human predictors used in a model compared to environmental predictors.

9.1 Table clean-up and setup

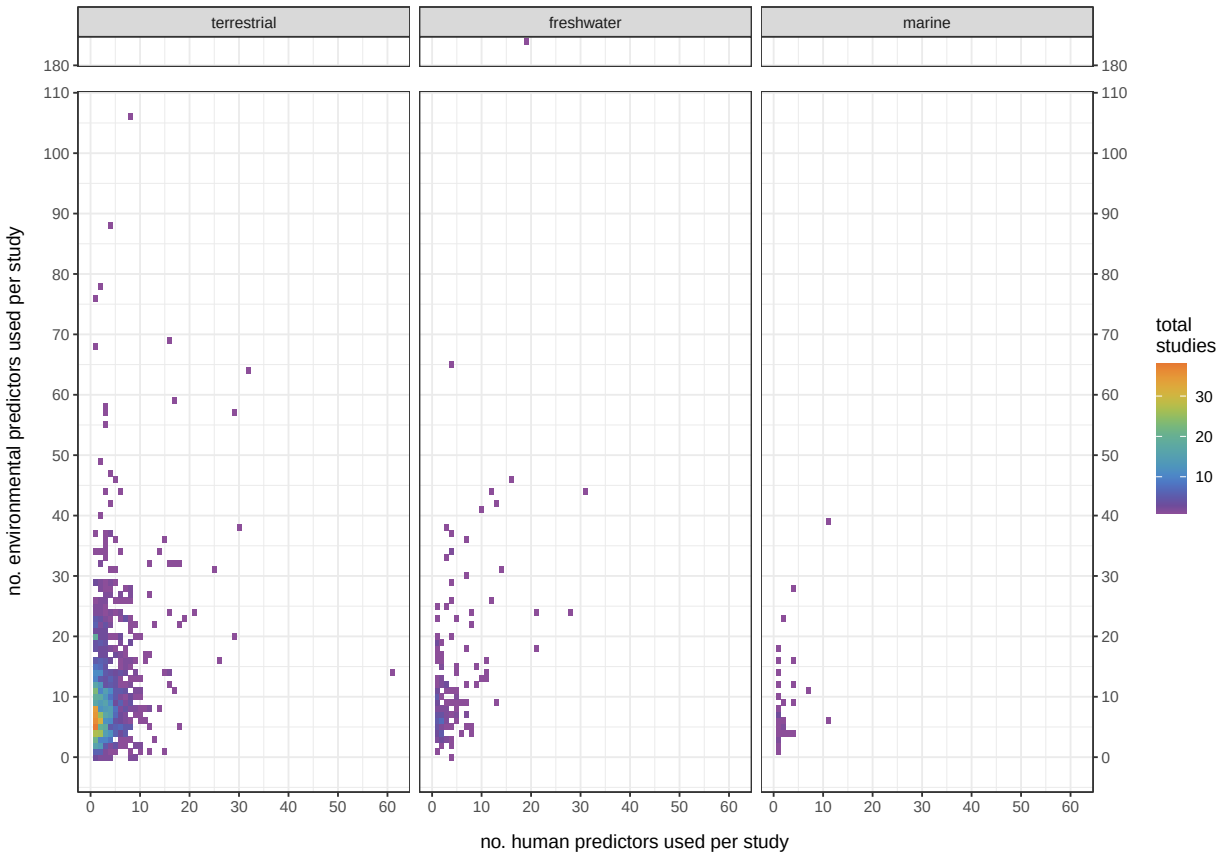
Make a subset of relevant papers only and summarize

```
# Pull up original table and subset
  preds.df <- data.frame(subset(yes.df,
                                select = c("uid","domain",
                                "num_present_preds","num_env_preds","num_hum_preds")))
# set up factors
  preds.df$domain <- as.factor(preds.df$domain)
  preds.df$domain <- factor(preds.df$domain,
                            levels = c("terrestrial", "freshwater", "marine"))

# get summary
  options(width=85) # ensure width
  summary(preds.df)
```

```
##       uid                domain     num_present_preds num_env_preds
##  Min.   :    2   terrestrial:1259   Min.   :  1.00    Min.   :  0.00
##  1st Qu.: 2885   freshwater : 151   1st Qu.:  8.00    1st Qu.:  5.00
##  Median : 5968   marine     :  31   Median : 12.00    Median :  8.00
##  Mean   : 6050                      Mean   : 14.73    Mean   : 11.27
##  3rd Qu.: 9082                      3rd Qu.: 17.00    3rd Qu.: 14.00
##  Max.   :12484                      Max.   :203.00    Max.   :184.00
##                                     NA's   :5
##  num_hum_preds
##  Min.   : 1.000
##  1st Qu.: 1.000
##  Median : 2.000
##  Mean   : 3.494
##  3rd Qu.: 4.000
##  Max.   :61.000
##
```

9.2 Density plot of predictor use by domain

```
# convert the number of predictors to factor
  preds.df$num_hum_preds <- factor(as.integer(preds.df$num_hum_preds))
  preds.df$num_env_preds <- factor(as.integer(preds.df$num_env_preds))

# get summary
  preds.dens <- ddply(preds.df, .(domain, num_hum_preds, num_env_preds), summarize,
                      # total count of entries
                      frequency=length(domain))

# Convert the number of predictors back to numeric
  preds.dens$num_hum_preds <- as.integer(as.character(preds.dens$num_hum_preds))
  preds.dens$num_env_preds <- as.integer(as.character(preds.dens$num_env_preds))

# Set up the color scale
  color_scale <- scale_fill_gradientn("total\nstudies",       # colorblind-friendly colors
                                      colors = c("#8C4E99", "#6F4C9B", "#6059A9", "#5568B8",
                                                 "#4E79C5", "#4D8AC6", "#4E96BC", "#549EB3",
                                                 "#59A5A9", "#60AB9E", "#69B190", "#77B77D",
                                                 "#8CBC68", "#A6BE54", "#BEBC48", "#D1B541",
                                                 "#DDAA3C", "#E49C39", "#E78C35", "#E67932"))

# Create the density grid plot with facets
  density_plot <- ggplot(preds.dens,
                         aes(x = num_hum_preds, y = num_env_preds, fill = frequency)) +
                         geom_tile() +
                         color_scale +
                         xlab("no. human predictors used per study") +
                         ylab("no. environmental predictors used per study") +
                         theme_bw() +
                         facet_wrap(~ domain, nrow = 1) +
                         theme(legend.position = "right") +
                         scale_y_continuous(breaks = seq(0, 190, by = 10)) +
                         scale_x_continuous(breaks = seq(0, 60, by = 10))

# Make y-axis cuts using `ggbreak` package options
  density_plot <- density_plot +
                  scale_y_break(c(105,180), ticklabels = c(180), space = .4)

# Save the plot as an image
  ggsave(filename = paste0(image.dir, "predictor_density_grid_facets_600dpi.png"),
         plot = density_plot, dpi = 600,
         height = 7, width = 10)
  ggsave(filename = paste0(image.dir, "predictor_density_grid_facets.svg"),
         plot = density_plot,
         height = 7, width = 10)

# show here
  density_plot
```

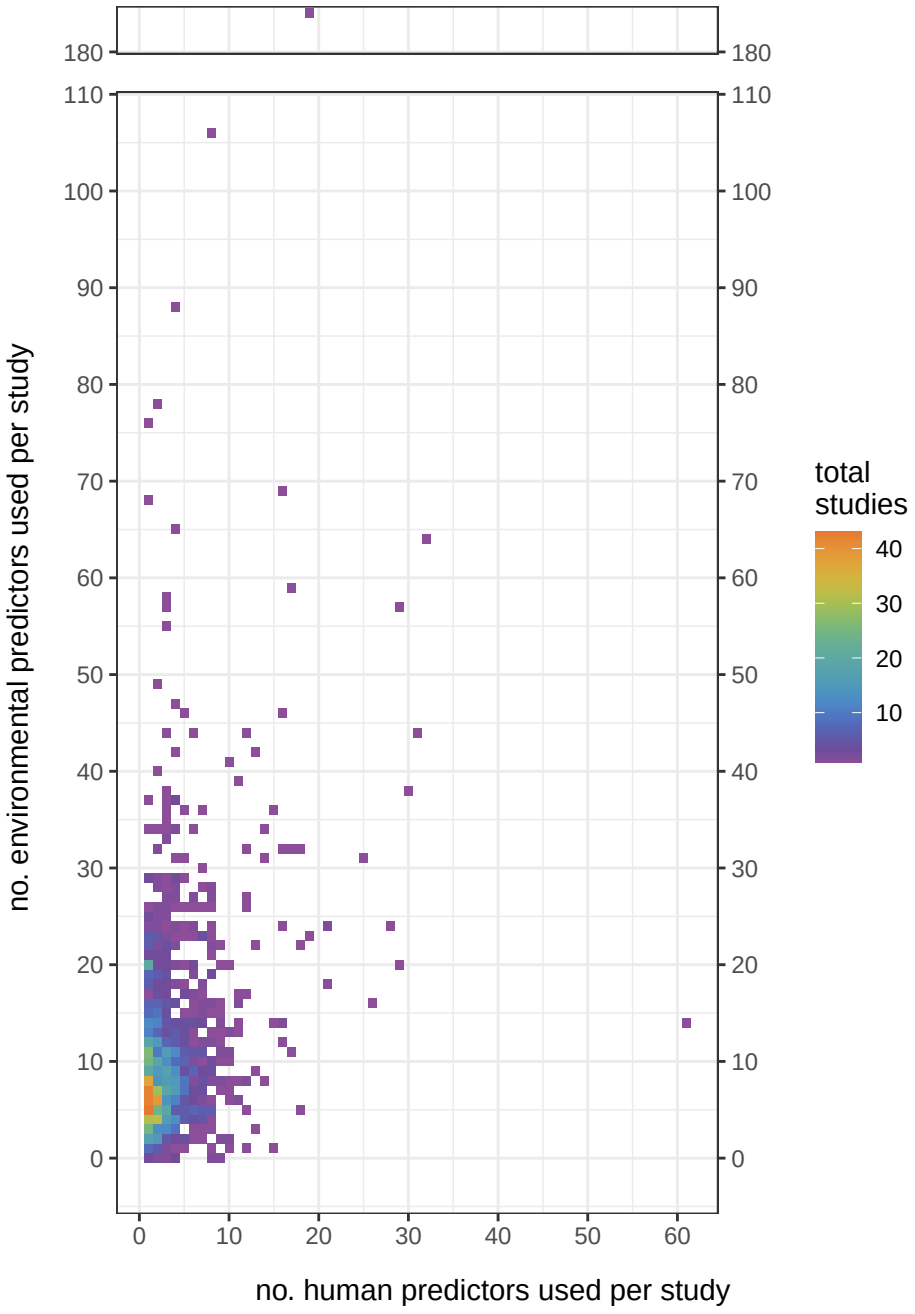


The numbers on the y-axis on the right-hand side will be manually deleted from the SVG using an imaging software, for use in the final manuscript.

9.3 Density plot of predictor use across all studies

```
# get summary (for all, so not by domain anymore)
  preds.dens2 <- ddply(preds.df, .(num_hum_preds, num_env_preds), summarize,
                       # total count of entries
                       frequency=length(domain))

# Convert the number of predictors back to numeric
  preds.dens2$num_hum_preds <- as.integer(as.character(preds.dens2$num_hum_preds))
  preds.dens2$num_env_preds <- as.integer(as.character(preds.dens2$num_env_preds))

# Set up the color scale
  color_scale <- scale_fill_gradientn("total\nstudies",       # colorblind-friendly colors
                                      colors = c("#8C4E99", "#6F4C9B", "#6059A9", "#5568B8",
                                                 "#4E79C5", "#4D8AC6", "#4E96BC", "#549EB3",
                                                 "#59A5A9", "#60AB9E", "#69B190", "#77B77D",
                                                 "#8CBC68", "#A6BE54", "#BEBC48", "#D1B541",
                                                 "#DDAA3C", "#E49C39", "#E78C35", "#E67932"))

# Create the density grid plot
  density_plot2 <- ggplot(preds.dens2,
```

```
                          aes(x = num_hum_preds, y = num_env_preds, fill = frequency)) +
                          geom_tile() +
                          color_scale +
                          xlim(0, 60) + #ylim(0, 100) +
                          xlab("no. human predictors used per study") +
                          ylab("no. environmental predictors used per study") +
                          theme_bw() +
                          theme(legend.position = "right") +
                          scale_y_continuous(breaks = seq(0, 190, by = 10)) +
                          scale_x_continuous(breaks = seq(0, 60, by = 10))

# Make y-axis cuts using `ggbreak` package options
  density_plot2 <- density_plot2 +
                   scale_y_break(c(105,180), ticklabels = c(180), space = .4)

# Save the plot as an image
  ggsave(filename = paste0(image.dir, "predictor_density_grid_all_600dpi.png"),
         plot = density_plot2, dpi= 600,
         height = 7, width = 5)
  ggsave(filename = paste0(image.dir, "predictor_density_grid_all.svg"),
         plot = density_plot2,
         height = 7, width = 5)

# show here
  density_plot2
```



The numbers on the y-axis on the right-hand side will be manually deleted from the SVG using an imaging software, for use in the final manuscript.

9.4 Summary table of predictor use

```
# calculate total predictors across studies (this will help include past-only papers)
  preds.df$total_preds <- as.integer(preds.df$num_env_preds) +
                          as.integer(preds.df$num_hum_preds)
```

```
# drop `num_present_preds`
  preds.df <- subset(preds.df,select=-c(num_present_preds))

# change from wide- to long-format dataframe
  library("reshape2")
  preds.long <- melt(preds.df, id.vars=c("uid","domain"))

# change to numeric format
  preds.long$value <- as.integer(as.character(preds.long$value))

# function to calculate mode
  calc_mode <- function(x) unique(x)[which.max(table(x))]

# calculate summary statistics
  preds.means <- ddply(preds.long, .(domain,variable),
                       summarize,
                       num_studies=length(value),
                       mean_preds=mean(value),
                       sd=sd(value),
                       se=sd/sqrt(num_studies),
                       min=min(value),max=max(value),
                       mode=calc_mode(value))

# view table
  kableExtra::kbl(preds.means,booktabs=T,longtable=T) %>%
      kable_styling(latex_options = c("striped"))
```

domain	variable	num_studies	mean_preds	sd	se	min	max	mode
terrestrial	num_env_preds	1259	10.983320	9.524074	0.2684169	0	106	2
terrestrial	num_hum_preds	1259	3.424146	3.683149	0.1038022	1	61	6
terrestrial	total_preds	1259	15.185068	9.761296	0.2751025	2	79	9
freshwater	num_env_preds	151	13.960265	17.604878	1.4326647	0	184	16
freshwater	num_hum_preds	151	4.264901	4.895851	0.3984187	1	31	2
freshwater	total_preds	151	18.205298	13.904109	1.1315003	3	79	37
marine	num_env_preds	31	9.806452	8.239819	1.4799153	1	39	3
marine	num_hum_preds	31	2.580645	2.668010	0.4791888	1	11	1
marine	total_preds	31	13.387097	9.755947	1.7522197	3	51	19

Get summary of values across all studies.

```
# calculate summary statistics
  preds.means.all <- ddply(preds.long, .(variable),
                           summarize,
                           num_studies=length(value),
                           mean=mean(value),
                           sd=sd(value),
                           se=sd/sqrt(num_studies),
                           min=min(value),max=max(value),
                           mode=calc_mode(value))
```

```
# view table
  kableExtra::kbl(preds.means.all,booktabs=T,longtable=T) %>%
      kable_styling(latex_options = c("striped"))
```

variable	num_studies	mean	sd	se	min	max	mode
num_env_preds	1441	11.269951	10.668619	0.2810452	0	184	2
num_hum_preds	1441	3.494101	3.818468	0.1005905	1	61	6
total_preds	1441	15.462873	10.310669	0.2716157	2	79	8

10 Comparing use of common environmental predictors (Worldclim)

Check the use of more popular environmental predictors, such as those for climate. Here, we examine how many papers use Worldclim.

```
# Make a subset table of studies
  wc.list.df <- subset(yes.df, select = c("uid", "worldclim"))

# Get a count of articles using Worldclim
  wc.list.df <- ddply(wc.list.df, .(worldclim), summarize,
                      count=length(worldclim),
                      perc=count/nrow(yes.df))

# show table
  kableExtra::kbl(wc.list.df,booktabs=T,longtable=T) %>%
      kable_styling(latex_options = c("striped"))
```

worldclim	count	perc
no	966	0.6703678
UNK	5	0.0034698
yes	470	0.3261624

11 Studies' qualitative evaluations on human predictor performance

Some papers have modeled SDMs with and without human predictors. In some of these cases, comparative statements were made by the authors about SDM performance. These evaluations of performance were based on quantitative results (accuracy, predictor importance, significance, overlap), based on more holistic evaluations (comparing prediction with prior knowledge), or a combination of both. These evaluations were either in the Results or Discussion sections of the articles. We summarized these mainly qualitative evaluations to see if there is any consensus or detectable trend about using human predictors in SDMs.

```
# convert to factor
  yes.df$qual_eval <- as.factor(yes.df$qual_eval)
```

```
# get total number of papers that made comparisons
  options(width=85) # ensure width
  paste(length(yes.df$qual_eval[!is.na(yes.df$qual_eval)]),
        "papers evaluated human predictor performance");
```

```
## [1] "130 papers evaluated human predictor performance"
```

```
  paste("in SDMs compared to SDMs without human predictors")
```

```
## [1] "in SDMs compared to SDMs without human predictors"
```

Next, make a summary table showing the different results of the qualitative evaluation

```
# Get count of summaries
  qual_eval <- ddply(yes.df, .(qual_eval),
                             summarize,
                             # list of the types of papers by study focus
                             focus=paste(unique(study_focus),collapse="; "),
                             # list of UIDs that used the variables
                             papers=paste(unique(uid),collapse="; "),
                             # count of papers
                             count=paste(length(unlist(strsplit(papers, ";"))))
                             )

# make NA field for NAs since not relevant
  qual_eval$papers[is.na(qual_eval$qual_eval)] <- NA
  qual_eval$focus[is.na(qual_eval$qual_eval)] <- NA

# save table
  write.csv(qual_eval,paste0(data.dir,"qualitative_predictor_eval_summary.csv"),
            row.names = FALSE)

# show table of only the counts (longer table is saved)
  kableExtra::kbl(qual_eval,booktabs=T, longtable=T) %>%
    kable_styling(latex_options = c("striped","repeat_header")) %>%
    column_spec(1, width="5em") %>%
    column_spec(2, width="10em") %>%
    column_spec(3, width="10em") %>%
    column_spec(4, width="5em")
```

qual_eval	focus	papers	count
better	reintroduction; invasions; conservation; disturbance; exploratory; human_food_security; protection; human_health; human-wildlife_conflict; habitat_change; human_economics	755; 1658; 2935; 3040; 3112; 4550; 4563; 4581; 4762; 4944; 5174; 5334; 5394; 5554; 5829; 5903; 5999; 6360; 7016; 7240; 7605; 7874; 7990; 8300; 8654; 8712; 8750; 8761; 9108; 9398; 9521; 9539; 9740; 9785; 10187; 10360; 10545; 10640; 10747; 10813; 11413; 11640; 11814	43
depends	exploratory; invasions; habitat_change; restoration; disturbance; conservation; human_health; reintroduction	408; 1048; 2367; 2484; 3854; 5192; 5294; 5411; 5863; 6422; 7201; 8404; 8482; 8516; 8663; 8840; 9326; 9678; 9782; 9957; 10431; 11427; 11474; 11618; 11958; 12246	26
no_differenc	invasions; exploratory; conservation; human_food_security; habitat_change	632; 3134; 4355; 4595; 4918; 5284; 5427; 5594; 5643; 6343; 6522; 6622; 6856; 7537; 8287; 9501; 10984; 11008	18
not_stated	conservation; invasions; reintroduction; human_food_security; human_disturbance; exploratory; protection; disturbance; human_economics; human_health	106; 188; 1576; 3130; 3238; 3601; 4159; 4396; 4818; 6211; 6300; 6347; 6834; 7468; 7618; 7944; 8548; 8736; 8858; 9182; 9235; 9673; 9938; 9939; 10047; 10829; 10959; 11230; 11404; 11425	30
worse	exploratory; habitat_change; human_gain; invasions; illegal_activity; conservation	498; 1573; 3302; 5627; 6787; 6978; 8486; 10334; 10989; 11625	10
NA	NA	NA	1302

12 Save

```
# save progress
  save.image("SDMs_human_lit_review_II.RData")
```

THIS IS THE END OF THE SCRIPT.

See "Human Influence in SDMs: Literature Review (Part III)" for next steps.
