Hawai‘i Climate Smart Agriculture Database Data Pipeline

MK Lau

Context

The City and County of Honolulu's Office of Climate Resilience has contracted OACA to produce a Climate Smart Agriculture database intended to inform food system professionals and policy makers on the potential ecosystem service impacts of CSA activities. The consultant (MK Lau) has been sub-contracted by OACA to complete the deliverables listed below in partial fulfillment of the larger contracted deliverable to HC&C.

Database Construction

- The goal is to construct a database of resources to support climate smart agricultural practices, using the USDA NRCS climate smart mitigation strategies as a framework.
- Via the data pipeline described in this document, data are ingested into the database starting with hand-extracted data gathered by Lucas McKinnon and Jackson Hart and then using a webcrawling to gather resources from existing websites, including the NRCS, NIFA, AMS, and ATTRA.
- A structured relational database in produced and saved to the main directory.

Ingest

Webcrawl

There are currently two webcrawl directories for NRCS. The first, "nrcs.usda.gov-292243", contains a very deep crawl with max depth = 4. The second, "nrcs.usda.gov-301855", is a shallower crawl with a max depth of 1, which is currenlty being used to compile data for the database.

```
nrcs.url <- paste0("https://www.nrcs.usda.gov/conservation-basics/",
                   "natural-resource-concerns/climate/climate-smart-mitigation-activities")

if ("nrcs.rds" %in% dir("data")){

    nrcs <- readRDS(file = "./data/nrcs.rds")

}else{

    Rcrawler(Website = nrcs.url,
             no_cores = 4, no_conn = 4,
             NetworkData = TRUE,
             NetwExtLinks = TRUE,
             ExtractXpathPat = "//*/a/@href",
             RequestsDelay = 0,01,
             ManyPerPattern = TRUE, MaxDepth = 1,
             saveOnDisk = FALSE
             )
    nrcs <- list()
```

```
    nrcs[[1]] <- INDEX
    nrcs[[2]] <- DATA
    nrcs[[3]] <- list(NetwIndex, NetwEdges)
    names(nrcs)[1] <- "INDEX"
    names(nrcs)[2] <- "DATA"
    names(nrcs)[3] <- "network"
    names(nrcs[[3]])[1] <- "NetwIndex"
    names(nrcs[[3]])[2] <- "NetwEdges"
    saveRDS(nrcs, file = "data/nrcs.rds", compress = TRUE)

}
```

The following code scrapes key information from the NRCS Climate Mitigation Strategies webpage. It is then exported to the `data/hi-csa-es.db`.

```
## From the
## 1. Get a list of mitigation categories
## 2. Get a list of practices within each mitigation
## 3. Get URL links to resources
## From the remaining search,
## 1. Get a list of other resources



nrcs.csm <- read_html(nrcs.url)

h2 <- nrcs.csm %>% html_elements("h2")
h3 <- nrcs.csm %>% html_elements("h3")
h4 <- nrcs.csm %>% html_elements("h4")
p <- nrcs.csm %>% html_elements("p")
a <- nrcs.csm %>% html_elements("a")
div <- nrcs.csm %>% html_elements("div")

nrcs.csm %>% html_elements("h3")
nrcs.csm %>% html_elements("h4")
nrcs.csm %>% html_elements("body")
nrcs.csm %>% html_elements(".title")

headers <- nrcs.csm %>% html_elements("h3, h4, p")

# Load the web page
webpage <- read_html(nrcs.url)

# Extract all headers (h3, h4) and paragraphs (p)
elements <- webpage %>% html_elements("h3, h4, p")

# Initialize lists to store the associations
result <- list()
current_h3 <- NULL
current_h4 <- NULL

# Loop through each element and determine its tag type
for (element in elements) {
  # Get the text, tag name, and anchor tags within the current element
```

```
  element_text = element %>% html_text(trim = TRUE)
  tag_name = element %>% html_name()

  # Check if it's an h3 header
  if (tag_name == "h3") {
    # If it's an h3, update the current context
    current_h3 = element_text
    result[[current_h3]] = list("h4" = list(), "p" = list())
  } else if (tag_name == "h4" && !is.null(current_h3)) {
    # If it's an h4, update the current context
    current_h4 = element_text
    result[[current_h3]]$h4[[current_h4]] = list("p" = list())
  } else if (tag_name == "p") {
    # If it's a paragraph, add it to the corresponding context
    paragraph_data = list("text" = element_text, "links" = list())

    # Check for any anchor tags (links) within the paragraph
    anchors = element %>% html_elements("a")

    if (length(anchors) > 0) {
      paragraph_data$links = lapply(anchors, function(anchor) {
        list("text" = anchor %>% html_text(trim = TRUE), "href" = anchor %>% html_attr("href"))
      })
    }

    if (!is.null(current_h4) && !is.null(current_h3)) {
      result[[current_h3]]$h4[[current_h4]]$p <- append(result[[current_h3]]$h4[[current_h4]]$p, list(pa
    } else if (!is.null(current_h3)) {
      result[[current_h3]]$p <- append(result[[current_h3]]$p, list(paragraph_data))
    }
  }
}


# Define a function to convert the list into a data frame
list_to_dataframe <- function(result) {
  # Initialize an empty data frame
  data <- data.frame(H3 = character(), H4 = character(), p = character(), a = character(), stringsAsFac

  # Iterate through the result list to build the data frame
  for (h3_name in names(result)) {
    h3_entry <- result[[h3_name]]

    # Extract paragraphs for h3-level
    if ("p" %in% names(h3_entry)) {
      for (p_item in h3_entry$p) {
        # Check if the paragraph contains links
        if ("links" %in% names(p_item)) {
          for (link in p_item$links) {
            new_row <- data.frame(
              H3 = h3_name,
              H4 = NA,
              p = p_item$text,
```

```
              a = link$href,
              stringsAsFactors = FALSE
            )
            data <- bind_rows(data, new_row)
          }
        }
      }
    }

    # Extract h4-level entries
    if ("h4" %in% names(h3_entry)) {
      for (h4_name in names(h3_entry$h4)) {
        h4_entry <- h3_entry$h4[[h4_name]]

        # Extract paragraphs and links at h4-level
        if ("p" %in% names(h4_entry)) {
          for (p_item in h4_entry$p) {
            if ("links" %in% names(p_item)) {
              for (link in p_item$links) {
                new_row <- data.frame(
                  H3 = h3_name,
                  H4 = h4_name,
                  p = p_item$text,
                  a = link$href,
                  stringsAsFactors = FALSE
                )
                data <- bind_rows(data, new_row)
              }
            }
          }
        }
      }
    }
  }

  return(data)
}

## Use the function to convert the list into a data frame
nrcs.db <- list_to_dataframe(result)
## Add the full path for the nrcs urls
for (i in seq_along(nrcs.db[, "a"])){
    if (!grepl("http", nrcs.db[i, "a"])){
        nrcs.db[i, "a"] <- paste0("https://www.nrcs.usda.gov", nrcs.db[i, "a"])
    }else{}
}

## Prep for export
nrcs.db[is.na(nrcs.db[, "H4"]), "H4"] <- "Other"
colnames(nrcs.db) <- c("Mitigation", "Practice", "Description", "Resource")
```

```
attra.url <- "https://attra.ncat.org/climate-solutions/"

if ("attra.rds" %in% dir("data")){
```

```
    attra <- readRDS(file = "./data/attra.rds")

}else{

    Rcrawler(Website = attra.url,
             no_cores = 4, no_conn = 4,
             NetworkData = TRUE,
             NetwExtLinks = TRUE,
             ExtractXpathPat = "//*/a/@href",
             RequestsDelay = 0,01,
             ManyPerPattern = TRUE, MaxDepth = 1,
             saveOnDisk = FALSE
             )
    attra <- list()
    attra[[1]] <- INDEX
    attra[[2]] <- DATA
    attra[[3]] <- list(NetwIndex, NetwEdges)
    names(attra)[1] <- "INDEX"
    names(attra)[2] <- "DATA"
    names(attra)[3] <- "network"
    names(attra[[3]])[1] <- "NetwIndex"
    names(attra[[3]])[2] <- "NetwEdges"
    saveRDS(attra, file = "./data/attra.rds", compress = FALSE)
}
```

```
nifa.url <- "https://www.nifa.usda.gov/grants"

if ("nifa.rds" %in% dir("data")){

    rds <- readRDS(file = "./data/nifa.rds")

}else{

    Rcrawler(Website = nifa.url,
             no_cores = 4, no_conn = 4,
             NetworkData = TRUE,
             NetwExtLinks = TRUE,
             ExtractXpathPat = "//*/a/@href",
             RequestsDelay = 0,01,
             ManyPerPattern = TRUE, MaxDepth = 1,
             saveOnDisk = FALSE
             )
    nifa <- list()
    nifa[[1]] <- INDEX
    nifa[[2]] <- DATA
    nifa[[3]] <- list(NetwIndex, NetwEdges)
    names(nifa)[1] <- "INDEX"
    names(nifa)[2] <- "DATA"
    names(nifa)[3] <- "network"
    names(nifa[[3]])[1] <- "NetwIndex"
    names(nifa[[3]])[2] <- "NetwEdges"
    saveRDS(nifa, file = "./data/nifa.rds", compress = FALSE)
}
```

```
ams.url <- "https://www.ams.usda.gov/services/grants"

if ("ams.rds" %in% dir("data")){

    ams <- readRDS(file = "./data/ams.rds")

}else{

    Rcrawler(Website = ams.url,
             no_cores = 4, no_conn = 4,
             NetworkData = TRUE,
             NetwExtLinks = TRUE,
             ExtractXpathPat = "//*/a/@href",
             RequestsDelay = 0,01,
             ManyPerPattern = TRUE, MaxDepth = 1,
             saveOnDisk = FALSE
             )
    ams <- list()
    ams[[1]] <- INDEX
    ams[[2]] <- DATA
    ams[[3]] <- list(NetwIndex, NetwEdges)
    names(ams)[1] <- "INDEX"
    names(ams)[2] <- "DATA"
    names(ams)[3] <- "network"
    names(ams[[3]])[1] <- "NetwIndex"
    names(ams[[3]])[2] <- "NetwEdges"
    saveRDS(ams, file = "./data/ams.rds", compress = FALSE)

}
```

Other data to integrate

- https://gofarmhawaii.org/farmer-resources-2/
- https://www.fsa.usda.gov/programs-and-services/farm-loan-programs/
- https://www.fns.usda.gov/fm/grant-opportunities
- https://www.rd.usda.gov/

```
nrc.wc[nrc.wc[, "Level"] == 2, "Url"]
nrc.url <-
nrc.url <- nrc.url[grepl("natural-resource-concerns/", nrc.url)]
nrc.cat <- strsplit(nrc.url, "natural-resource-concerns/")
nrc.url <- nrc.url[lapply(nrc.cat, length) == 2]
nrc.cat <- nrc.cat[lapply(nrc.cat, length) == 2]
nrc.cat <- lapply(nrc.cat, function(x) x[2])
nrc.cat <- lapply(nrc.cat, strsplit, split = "\\/")
nrc.cat <- lapply(nrc.cat, unlist)

for (i in seq_len(length(nrc.cat))){
    if (length(nrc.cat[[i]]) < max(unlist(lapply(nrc.cat, length)))){
        nrc.cat[[i]] <- c(nrc.cat[[i]], rep("", times = max(unlist(lapply(nrc.cat, length))) - length(ni
    }else{}
}

nrc.tab <- cbind(do.call(rbind, nrc.cat), nrc.url)
```

```r
colnames(nrc.tab) <- c("Category",
                       paste0("Sub-Category ", seq(1, ncol(nrc.tab)-2)),
                       "Resource")
```

Manual

Data from NRCS were extracted manually using tabula.

```r
csa.mit.tab <- read.csv("data/tabula-NRCS-CSAF-Mitigation-Activities.csv", header = FALSE)
csa.mit.head <- csa.mit.tab[grepl("Mitigation Categories", csa.mit.tab[, 1]), ]
csa.mit.head <- gsub("\\[.*?\\]", "", csa.mit.head)
csa.mit.head <- gsub("  ", " ", csa.mit.head)
colnames(csa.mit.tab) <- csa.mit.head
csa.mit.tab <- apply(csa.mit.tab, 2, gsub, pattern = "\\[.*?\\]", replace = "")
csa.mit.tab <- apply(csa.mit.tab, 2, gsub, pattern = "  ", replace = " ")

## Removing narrative crosswalk
csa.cwk <- csa.mit.tab[seq(grep("Waste Storage Structure", csa.mit.tab[, 2]), nrow(csa.mit.tab)), ]
colnames(csa.cwk)[4] <- "Narrative"
csa.mit.tab <-csa.mit.tab[-seq(grep("Waste Storage Structure", csa.mit.tab[, 2]), nrow(csa.mit.tab)), ]

## Generate urls for codes
get.codes <- function(x){
    x <- paste0(x, collapse = " ")
    x <- unlist(strsplit(x, split = " "))
    x <- x[grep("E[0-9][0-9][0-9][a-z,A-Z]", x)]
    return(x)
}

csa.mit.codes <- unlist(lapply(apply(csa.mit.tab, 1, get.codes), function(x) x[1]))
csa.mit.tab[, "Code"] <- csa.mit.codes
csa.mit.tab[!(grepl("E", csa.mit.tab[, "Code"])), "Code"] <- ""
csa.mit.url <- paste0("https://www.nrcs.usda.gov/sites/default/files/2022-11/",
                      csa.mit.tab[, "Code"],
                      "_July_2022.pdf")
csa.mit.url <- gsub(" ", "-", csa.mit.url)
csa.mit.url[csa.mit.tab[, "Code"] == ""] <- ""
csa.mit.tab <- data.frame(csa.mit.tab, "URL" = csa.mit.url)
```

```r
db.og <- as.data.frame(read_sheet("https://docs.google.com/spreadsheets/d/1AMlsLPDnwt01eEsBLdRe1hvhNSa3
db.jh <- as.data.frame(read_sheet("https://docs.google.com/spreadsheets/d/1AMlsLPDnwt01eEsBLdRe1hvhNSa3

db.mg <- db_merge(db.og, db.jh) %>%
    distinct

colnames(db.og)[colnames(db.og) == "Resources (Links)"] <- "Resource"
colnames(nrc.tab)[colnames(nrc.tab) == "Sub-Category 1"] <- "Sub-Category"
```

Merge Data Streams

```r
hicsa.db <- db_merge(db.og, nrc.tab)
```

Saving Database

```
hicsa.db <- nrcs.db
saveRDS(hicsa.db, file = "./data/hi-csa-db.rds")
```

Preview

Mitigation	Practice	Description	Resource
Additional Resources: Climate-Smart Agriculture and Forestry	Irrigation Water Management	Producers and landowners interested in climate-smart agriculture and forestry are encouraged to contact the NRCS office at their local USDA Service Center for additional information, including one-on-one support specific to their operation. Visit farmers.gov/service-locator to find your local office.	https://www.farmers.gov/working-with-us/service-center-locator
Additional Resources: Climate-Smart Agriculture and Forestry	Irrigation Water Management	Visit farmers.gov/climate-smart for additional information on climate solutions for your working land, including USDA programs and digital tools. You may access state-specific application ranking dates for NRCS conservation programs here.	https://www.farmers.gov/conservation/climate-smart
Additional Resources: Climate-Smart Agriculture and Forestry	Irrigation Water Management	Visit farmers.gov/climate-smart for additional information on climate solutions for your working land, including USDA programs and digital tools. You may access state-specific application ranking dates for NRCS conservation programs here.	https://www.nrcs.usda.gov/ranking-dates
Additional Resources: Climate-Smart Agriculture and Forestry	Irrigation Water Management	Visit USDA's Climate Solutions webpage for Department-wide resources, tools and announcements to support agricultural producers and rural communities in making informed, science-based decisions to support climate change mitigation and build climate resilience.	https://www.usda.gov/topics/climate-solutions
Agroforestry; Forestry and Wildlife Habitat	Critical Area Planting	Critical area planting is establishing permanent vegetation on sites that have, or are expected to have, high erosion rates, and on sites that have physical, chemical, or biological conditions that prevent the establishment of vegetation with normal practices. Producers who practice critical area planting may increase soil carbon sequestration in perennial biomass and soils while delivering the co-benefits of reducing soil erosion, building soil health, providing wildlife habitat, and increasing plant productivity and health. Watch the video	https://youtu.be/aRNOLSrr6Ok

Mitigation	Practice	Description	Resource
Agroforestry, Forestry and Wildlife Habitat	Windbreak and Shelterbelt Establishment and Renovation	This practice establishes, enhances or renovates windbreaks, which are single or multiple rows of trees or shrubs planted in linear or curvilinear configurations. Producers who establish windbreaks may increase carbon sequestration in perennial biomass and soils while delivering the co-benefits of reducing erosion, protecting crops, livestock and buildings from wind-related damage, enhancing moisture management and improving ambient air quality. Watch the video	https://youtu.be/EUoMCc3J4dA