DwCA_gen

Eduardo Klein and Enrique Montes

10/7/2020

Contents

This code transforms long format data tables of the Marine Biodiversity Observation Network Pole to Pole of the Americas (MBON Pole to Pole) into Darwin Core Archive files for data sharing through the Ocean Biodiversity Information System (OBIS). It also generates anintegrated file ready for analysis. Just copy the code chunks below and paste into your R console.

Basic Setup

You need few packages to run this code. Also you need to create two new folders ("Data", IPT" and "Analysis") under your working directory. The "Data" folder is where you copy your P2P long format data table.

```
library(lutz)
library(countrycode)
library(readxl)
library(reshape2)
library(lubridate)
library(dplyr)
library(ggplot2)
library(kableExtra)
options(dplyr.summarise.inform = FALSE)
```

```
# Create a "Data", IPT" and "Analysis" folder under your selected directory
baseDataDir = "Data"
baseIPT = "IPT"
baseAnalysis = "Analysis"
```

Load your data table

Set `fileName`to the name of your P2P longformat data table

```
# Select your data table
fileName = "DataSheet_longformat_TEST.xlsx"
```

Read file sheets

```
## Extract information about your sampling site from the SiteInfo tab of your data table file

DF.sites = read_xlsx(file.path(baseDataDir, fileName),sheet = "SiteInfo")
DF.sites = DF.sites[!is.na(DF.sites$COUNTRY),]
```

convert time to UTC

```
## get number of seconds from midnight
secsSTART = (1 -abs(as.numeric(julian(DF.sites$TIME_START)) - (as.integer(julian(DF.sites$TIME_START)))
secsEND = (1 - abs(as.numeric(julian(DF.sites$TIME_END)) - (as.integer(julian(DF.sites$TIME_END))))) *
dateChar = paste(DF.sites$YEAR, DF.sites$MONTH, DF.sites$DAY, sep="-")

## get timezone and timezone offset
timeZone = tz_lookup_coords(mean(DF.sites$LATITUDE, na.rm=T), mean(DF.sites$LONGITUDE, na.rm=T), method=
dateOffset = tz_offset(dateChar, timeZone)$utc_offset_h

## create data and time UTC
DF.sites$eventDate = as.POSIXct(dateChar, tz="UTC")
DF.sites$TIME_START = DF.sites$eventDate + seconds(secsSTART) + hours(dateOffset)
DF.sites$TIME_END = DF.sites$eventDate + seconds(secsEND) + hours(dateOffset)
DF.sites$eventTime = paste(format(DF.sites$TIME_START, "%H:%M:%SZ"), format(DF.sites$TIME_END, "%H:%M:%S

print(timeZone)
```

```
## [1] "America/Argentina/Catamarca"
```

```
kable(DF.sites[1:5, c(3:5, 13:14)]) %>% kable_styling("striped")
```

COUNTRY	LOCALITY	SITE	TIME_START	TIME_END
ARGENTINA	PUERTO MADRYN	PUNTA ESTE	2018-10-25 07:45:00	2018-10-25 10:45:00
ARGENTINA	PUERTO MADRYN	PUNTA ESTE	2018-10-25 07:45:00	2018-10-25 10:45:00
ARGENTINA	PUERTO MADRYN	PUNTA ESTE	2018-10-25 07:45:00	2018-10-25 10:45:00
ARGENTINA	PUERTO MADRYN	PUNTA CUEVAS	2018-11-05 07:45:00	2018-11-05 10:45:00
ARGENTINA	PUERTO MADRYN	PUNTA CUEVAS	2018-11-05 07:45:00	2018-11-05 10:45:00

Extract other fields

```
# Country code
DF.sites$datasetName = paste0("MBON-P2P-biodiversity-",unique(DF.sites$countryCode))
DF.sites$countryCodeISO = countrycode(DF.sites$COUNTRY, "country.name","iso3c")

# Sampling protocol
DF.sites$samplingProtocol = "MBON-P2P_bestpractices-rockyshores"

# Sampling size value
DF.sites$samplingSizeValue = 0.25

# Sampling unit
DF.sites$samplingSizeUnit = "square meter"

print(DF.sites$countryCodeISO[1])
```

```
## [1] "ARG"
```

Extrat data, taxa list and codes

```
## data
DF.data = read_xlsx(file.path(baseDataDir, fileName),sheet = "DATA")
DF.data = DF.data[!is.na(DF.data$LOCALITY),]

## spp list
DF.spp = read_xlsx(file.path(baseDataDir, fileName),sheet = "sppList")

## codes
DF.countryCodes = read_xlsx(file.path(baseDataDir, fileName),sheet = "Countries")
DF.localityCodes = read_xlsx(file.path(baseDataDir, fileName),sheet = "Locality")
DF.siteCodes = read_xlsx(file.path(baseDataDir, fileName),sheet = "Sites")
DF.habitatCodes = read_xlsx(file.path(baseDataDir, fileName),sheet = "Habitat")

kable(DF.data[1:5, 1:7]) %>% kable_styling("striped")
```

LOCALITY	SITE	STRATA	SAMPLE	scientificName	Variable	Value
PUERTO MADRYN	PUNTA CUEVAS	HIGH	S02	substrate_BAREROCK	COVER	22
PUERTO MADRYN	PUNTA CUEVAS	MID	S01	substrate_BAREROCK	COVER	2
PUERTO MADRYN	PUNTA CUEVAS	MID	S02	substrate_BAREROCK	COVER	1
PUERTO MADRYN	PUNTA CUEVAS	MID	S03	substrate_BAREROCK	COVER	5
PUERTO MADRYN	PUNTA CUEVAS	MID	S04	substrate_BAREROCK	COVER	3

```
kable(DF.spp[1:5,]) %>% kable_styling("striped")
```

scientificName	AphiaID	Authority	Rank
Aaptos	132064	Gray, 1867	Genus
Abietinaria	117225	Kirchenpauer, 1884	Genus
Abietinaria abietina	117870	(Linnaeus, 1758)	Species
Abra alba	141433	(W. Wood, 1802)	Species
Acanthais brevidentata	395268	(W. Wood, 1828)	Species

Generate IDs

```
## add codes: SITES
DF.sites = left_join(DF.sites, DF.countryCodes, by = "COUNTRY")
DF.sites = left_join(DF.sites, DF.localityCodes, by = "LOCALITY")
DF.sites = left_join(DF.sites, DF.siteCodes, by = "SITE")
DF.sites = left_join(DF.sites, DF.habitatCodes, by = "HABITAT")

DF.sites$PARENT_UNIT_ID = paste(DF.sites$countryCode, DF.sites$localityCode, DF.sites$siteCode, DF.sites
                                paste0(DF.sites$YEAR, DF.sites$MONTH, DF.sites$DAY), sep="_")
DF.sites$UNIT_ID = paste(DF.sites$PARENT_UNIT_ID, DF.sites$STRATA, sep="_")

print(DF.sites$UNIT_ID[1:6])
```

```
## [1] "ARG_PMADRY_PESTE_RS_20181025_HIGH" "ARG_PMADRY_PESTE_RS_20181025_MID"
## [3] "ARG_PMADRY_PESTE_RS_20181025_LOW"  "ARG_PMADRY_PCUEVA_RS_2018115_HIGH"
## [5] "ARG_PMADRY_PCUEVA_RS_2018115_MID"  "ARG_PMADRY_PCUEVA_RS_2018115_LOW"
```

Assign codes to DATA

```
## Add Aphia ID and taxa rank
DF.data = left_join(DF.data, DF.spp[,c("scientificName", "AphiaID", "Rank")])
DF.data = left_join(DF.data, DF.sites[,c("UNIT_ID", "LOCALITY", "SITE", "STRATA")])
DF.data = DF.data %>% group_by(LOCALITY, SITE, STRATA, SAMPLE) %>%
  mutate(sampleOrganismID = 1:n(), scientificName, AphiaID, Rank, Variable, Value)
DF.data$occurrenceID = paste(DF.data$UNIT_ID, DF.data$SAMPLE, sprintf("%03d", DF.data$sampleOrganismID)

print(DF.data$occurrenceID[1:20])
```

```
##  [1] "ARG_PMADRY_PCUEVA_RS_2018115_HIGH_S02_001"
##  [2] "ARG_PMADRY_PCUEVA_RS_2018115_MID_S01_001"
##  [3] "ARG_PMADRY_PCUEVA_RS_2018115_MID_S02_001"
##  [4] "ARG_PMADRY_PCUEVA_RS_2018115_MID_S03_001"
##  [5] "ARG_PMADRY_PCUEVA_RS_2018115_MID_S04_001"
##  [6] "ARG_PMADRY_PCUEVA_RS_2018115_MID_S05_001"
##  [7] "ARG_PMADRY_PCUEVA_RS_2018115_MID_S06_001"
##  [8] "ARG_PMADRY_PCUEVA_RS_2018115_MID_S07_001"
```

```
##  [9] "ARG_PMADRY_PCUEVA_RS_2018115_MID_S08_001"
## [10] "ARG_PMADRY_PCUEVA_RS_2018115_MID_S09_001"
## [11] "ARG_PMADRY_PCUEVA_RS_2018115_MID_S10_001"
## [12] "ARG_PMADRY_PESTE_RS_20181025_HIGH_S01_001"
## [13] "ARG_PMADRY_PESTE_RS_20181025_HIGH_S02_001"
## [14] "ARG_PMADRY_PESTE_RS_20181025_HIGH_S03_001"
## [15] "ARG_PMADRY_PESTE_RS_20181025_HIGH_S04_001"
## [16] "ARG_PMADRY_PESTE_RS_20181025_HIGH_S05_001"
## [17] "ARG_PMADRY_PESTE_RS_20181025_HIGH_S06_001"
## [18] "ARG_PMADRY_PESTE_RS_20181025_HIGH_S07_001"
## [19] "ARG_PMADRY_PESTE_RS_20181025_HIGH_S08_001"
## [20] "ARG_PMADRY_PESTE_RS_20181025_HIGH_S09_001"
```

convert abundance

```
## to count per square meter
DF.data$Value[DF.data$Variable=="ABUNDANCE"] = DF.data$Value[DF.data$Variable=="ABUNDANCE"] * 4

print(DF.data$Value[DF.data$Variable=="ABUNDANCE"][1:10])
```

```
##  [1] 12 16  4  4  4  4  4  4  8  4
```

Assign other IPT fields

```
DF.data$basisOfRecord = "HumanObservation"
DF.data$occurrenceStatus = "present"
DF.data$scientificNameID = paste0("lsid:marinespecies.org:taxname:", DF.data$AphiaID)

## fields for the eMoF
DF.data$measurementTypeID = ifelse(DF.data$Variable=="COVER",
                                   "http://vocab.nerc.ac.uk/collection/P01/current/SDBIOL10/",  ##Cover
                                   "http://vocab.nerc.ac.uk/collection/P06/current/UPMS/") ## number pe

DF.data$measurementUnit = ifelse(DF.data$Variable=="COVER", "percent", "count")
DF.data$measurementUnitID = ifelse(DF.data$Variable=="COVER",
                                   "http://vocab.nerc.ac.uk/collection/P06/current/UPCT/",    ## percen
                                   "http://vocab.nerc.ac.uk/collection/P06/current/UPMS/")   ## number

DF.data = DF.data %>% arrange(occurrenceID, scientificName)

kable(DF.data[1:5, ]) %>% kable_styling("striped")
```

LOCALITY	SITE	STRATA	SAMPLE	scientificName	Variable	Value	Remar
PUERTO MADRYN	PUNTA CUEVAS	HIGH	S01	Brachidontes rodriguezii	COVER	1	NA
PUERTO MADRYN	PUNTA CUEVAS	HIGH	S01	Ulva prolifera	COVER	13	NA
PUERTO MADRYN	PUNTA CUEVAS	HIGH	S02	substrate_BAREROCK	COVER	22	NA
PUERTO MADRYN	PUNTA CUEVAS	HIGH	S02	Balanus glandula	COVER	2	NA
PUERTO MADRYN	PUNTA CUEVAS	HIGH	S02	Brachidontes rodriguezii	COVER	2	NA

Cleanup

```
## EventCore file
IPT.event = DF.sites %>%
  select(datasetName,
         parentEventID=PARENT_UNIT_ID,
         eventID = UNIT_ID,
         samplingProtocol,
         samplingSizeValue,
         samplingSizeUnit,
         eventDate,
         eventTime,
         year = YEAR,
         month = MONTH,
         day = DAY,
         habitat = HABITAT,
         eventRemarks = REMARKS,
         country = COUNTRY,
         countryCode = countryCodeISO,
         locality = LOCALITY,
         decimalLatitude = LATITUDE,
         decimalLongitude = LONGITUDE,
         coordinateUncertaintyInMeters = GPS_ERROR,
         geodeticDatum = DATUM,
         strata=STRATA)

## Remove substrate type records
DF.data.noSubstrate = DF.data %>%
  filter(! grepl("substrate", scientificName, fixed = T))

## OccurrenceCore file
IPT.occurrence = DF.data.noSubstrate %>% ungroup() %>%
  select(eventID = UNIT_ID,
         basisOfRecord,
         occurrenceID,
         scientificNameID,
         scientificName,
         taxonRank = Rank)

## Event Measurement or Fact (eMOF) file
IPT.mof = data.frame(eventID = DF.data.noSubstrate$UNIT_ID,
                     occurrenceID = DF.data.noSubstrate$occurrenceID,
                     measurementType = tolower(DF.data.noSubstrate$Variable),
                     measurmenetTypeID = DF.data.noSubstrate$measurementTypeID,
                     measurementValue = DF.data.noSubstrate$Value,
                     measurementUnit = DF.data.noSubstrate$measurementUnit,
                     measurementUnitID = DF.data.noSubstrate$measurementUnitID
                     )
print("EventCore")
```

```
## [1] "EventCore"
```

```
kable(IPT.event[1:3, ]) %>% kable_styling("striped")
```

datasetName	parentEventID	eventID
MBON-P2P-biodiversity-	ARG_PMADRY_PESTE_RS_20181025	ARG_PMADRY_PESTE_RS_20181025_HIGH
MBON-P2P-biodiversity-	ARG_PMADRY_PESTE_RS_20181025	ARG_PMADRY_PESTE_RS_20181025_MID
MBON-P2P-biodiversity-	ARG_PMADRY_PESTE_RS_20181025	ARG_PMADRY_PESTE_RS_20181025_LOW

```
print("OccurrenceCore")
```

```
## [1] "OccurrenceCore"
```

```
kable(IPT.occurrence[1:3, ]) %>% kable_styling("striped")
```

eventID	basisOfRecord	occurrenceID
ARG_PMADRY_PCUEVA_RS_2018115_HIGH	HumanObservation	ARG_PMADRY_PCUEVA_RS_2018115_HI
ARG_PMADRY_PCUEVA_RS_2018115_HIGH	HumanObservation	ARG_PMADRY_PCUEVA_RS_2018115_HI
ARG_PMADRY_PCUEVA_RS_2018115_HIGH	HumanObservation	ARG_PMADRY_PCUEVA_RS_2018115_HI

```
print("MoF")
```

```
## [1] "MoF"
```

```
kable(IPT.mof[1:3, ]) %>% kable_styling("striped")
```

eventID	occurrenceID	meas
ARG_PMADRY_PCUEVA_RS_2018115_HIGH	ARG_PMADRY_PCUEVA_RS_2018115_HIGH_S01_001	cover
ARG_PMADRY_PCUEVA_RS_2018115_HIGH	ARG_PMADRY_PCUEVA_RS_2018115_HIGH_S01_002	cover
ARG_PMADRY_PCUEVA_RS_2018115_HIGH	ARG_PMADRY_PCUEVA_RS_2018115_HIGH_S02_002	cover

Generate data anaylisis files

```
## reformat to wide
DF.dataWide = dcast(occurrenceID+LOCALITY+SITE+STRATA+SAMPLE+scientificName+AphiaID+Rank~Variable, valu
```

Save files

```
rootFileName = paste(unique(DF.sites$countryCodeISO), paste0(unique(DF.sites$localityCode, collapse="-"
                     unique(DF.sites$HABITAT), gsub("-","", min(DF.sites$eventDate)), sep="_")

## IPT files
readr::write_csv(IPT.event, path = file.path(baseIPT,paste0(rootFileName, "_IPT-event.csv")))
readr::write_csv(IPT.occurrence, path = file.path(baseIPT,paste0(rootFileName, "_IPT-occurrence.csv")))
readr::write_csv(IPT.mof, path = file.path(baseIPT,paste0(rootFileName, "_IPT-mof.csv")))
```

```r
## Analysis file
readr::write_csv(DF.dataWide, path = file.path(baseAnalysis,paste0(rootFileName, "_analysis.csv")))
readr::write_csv(DF.sites, path = file.path(baseAnalysis,paste0(rootFileName, "_site.csv")))
```