Standardizing-Marine-Biological-Data

Standardizing Marine Biological Data Working Group

2021-11-04

Contents

Preface

Biological data structures, definitions, measurements, and linkages are neccessarily as diverse as the systems they represent. This presents a real challenge when integrating data across biological research domains such as ecology, oceanography, fisheries, and climate sciences.

Chapter 1

Introduction

The world of standardizing marine biological data can seem complex for the naive oceanographer, biologist, scientist, or programmer. Transforming and integrating data is about combining the right standards for your desired interoperability with other data types. For example, interoperating fish biology measurements with climate level variables. There are a few concepts necessary to make this possible such as standard data structures, controlled vocabularies and knowledge representations, along with metadata standards to facilitate data discovery. This will permit the inclusion of more data and broader access to better ecosystem based models. Many scientific domains data handling practices are currently being reshaped in light of recent advances in computing power, technology, and data science.

1.1 Data Structures

The OBIS-ENV Darwin Core Archive Data Structure.

OBIS manual

1.2 Ontologies

An ontology is a classification system for establishing a hierarchically related set of concepts. Concepts are often terms from controlled vocabularies.

From Marine Metadata: # TODO: add link

“Ontologies can include all of the following, but are not required to include them, depending on which perspective from above you adhere to:

Classes (general things, types of things) Instances (individual things) Relationships among things Properties of things Functions, processes, constraints, and rules relating to things”

TODO: Research Unified Modelling Language?

Environment Ontology (EnvO) EnvO is a community ontology for the concise, controlled description of environments.

1.3 Controlled Vocabularies

There are a number of controlled vocabularies that are used to describe parameters commonly used in specific research domains. This allows for greater interoperability of data sets within the domain, and ideally between domains. Here, we strive to document a number of relevant examples.

- Climate and Format (CF) Standard Names The purpose of the standard_name attribute is to provide a succinct and distinguishing description of a variable, in a way that encourages interoperability. These terms are typically for physical observations, however, there have been advancements in aligning biological taxa into the CF standard names (see here).
- NERC Vocabulary Server The NVS gives access to standardised and hierarchically-organized vocabularies.
  - Device categories using the SeaDataNet device categories
  - Device make/model using the SeaVoX Device Catalogue
  - Platform categories using SeaVoX Platform Categories
  - Platform instances using the ICES Platform Codes
  - Unit of measure
- GCMD Keywords (NASA)
- Geographic Domain/Features of Interest
- GeoLink base ontology was part of the EarthCube GeoLink Project

Note: To describe a measurement or fact of a biological specimen that conforms to Darwin Core standards, it’s necessary to use the ‘Biological entity described elsewhere’ method rather than taxon specific.

1.3.1 Taxonomy

- The World Registry of Marine Species (WoRMS) The aim of a World Register of Marine Species (WoRMS) is to provide an authoritative and comprehensive list of names of marine organisms, including information on synonymy. While the highest priority goes to valid names, other names in use are included so that this register can serve as a guide to interpret taxonomic literature.

1.3.2 Resources

1.3.3 Oceanography

Biological and Chemcial Oceanography Data Management Office

Marine metadata interoperability vocab resources

1.3.4 Biology

BioPortal Ecosystem Ontology

1.3.5 NERC Search Interfaces

- SeaDatanet Common Vocab Search Interface:
- SeaDataNet Common Vocabularies:
- SeaDataNet Vocab Library
- Measurement Types in OBIS

1.3.6 Geosciences

UDUNITSare more common unit measurements in geosciences

1.3.7 Eco/EnvO

Environment Ontology including genomics.

1.3.8 Wild Cards

Question: Not sure use case for this.

P01 Biological Entity Parameter Code Builder

1.4 Technologies

1.4.1 ERDDAP

ERDDAP can be thought of as a data server. It provides 'easier access to scientific data' by providing a consistent interface that aggregates many disparate data sources. It does this by providing translation services between many common file types for gridded arrarys ('net CDF' files) and tabular data (spreadsheets). Data access is also made easier because it unifies different types of data servers and access protocols. Here is a basic erddap installation that walks you through how to load a data set.

1.5 Notes on Integrating OBIS, Darwin Core as it relates to OOS's

1.6 Metadata

OBIS uses the GBIF EML profile (version 1.1). In case data providers use ISO19115/ISO19139, there is a mapping available here: http://rs.gbif.org/schema/eml-gbif-profile/1.1/eml2iso19139.xsl This will be important for integrating OBIS datasets to other CIOOS and IOOS metadata profiles.

1.7 Data QC

There are a number of tools available to check the quality of data or check your data format against the expected standard.

OBIS Datatools shows some great R packages for this.

1.7.1 Compliance Checking

LifeWatch Belgium provides a number of tools to check your data against. Specifically you can test OBIS data format and see a map of your sample locations to check if they are on land. See http://www.lifewatch.be/data-services/

There's also the GBIF data validator which allows anyone with a GBIF-relevant dataset to receive a report on the syntactical correctness and the validity of the content contained within the dataset.

1.7.2 Semantic Web and Darwin Core

Lessons learned from adapting the Darwin Core vocabulary standard for use in RDF

1.7.3 Resource Description Framework

Darwin Core Resource Description Framework Guide

Chapter 2

Applications

Some *significant* applications are demonstrated in this chapter.

2.1 Salmon Ocean Ecology Data

2.1.1 Intro

One of the goals of the Hakai Institute and the Canadian Integrated Ocean Observing System (CIOOS) is to facilitate Open Science and FAIR (findable, accessible, interoperable, reusable) ecological and oceanographic data. In a concerted effort to adopt or establish how best to do that, several Hakai and CIOOS staff attended an International Ocean Observing System (IOOS) Code Sprint in Ann Arbour, Michigan between October 7–11, 2019, to discuss how to implement FAIR data principles for biological data collected in the marine environment.

The Darwin Core is a highly structured data format that standardizes data table relations, vocabularies, and defines field names. The Darwin Core defines three table types: `event`, `occurrence`, and `measurementOrFact`. This intuitively captures the way most ecologists conduct their research. Typically, a survey (event) is conducted and measurements, counts, or observations (collectively measurementOrFacts) are made regarding a specific habitat or species (occurrence).

In the following script I demonstrate how I go about converting a subset of the data collected from the Hakai Institute Juvenile Salmon Program and discuss challenges, solutions, pros and cons, and when and what's worthwhile to convert to Darwin Core.

The conversion of a dataset to Darwin Core is much easier if your data are already tidy (normalized) in which you represent your data in separate tables that reflect the hierarchical and related nature of your observations. If your data are not already in a consistent and structured format, the conversion would likely be very arduous and not intuitive.

2.1.2 event

The first step is to consider what you will define as an event in your data set. I defined the capture of fish using a purse seine net as the `event`. Therefore, each row in the `event` table is one deployment of a seine net and is assigned a unique `eventID`.

My process for conversion was to make a new table called `event` and map the standard Darwin Core column names to pre-existing columns that serve the same purpose in my original `seine_data` table and populate the other required fields.

```
#TODO: Include abiotic measurements (YSI temp and salinity from 0 and 1 m) to hang off

event <- tibble(datasetName = "Hakai Institute Juvenile Salmon Program",
                eventID = survey_seines$seine_id,
                eventDate = date(survey_seines$survey_date),
                eventTime = paste0(survey_seines$set_time, "-0700"),
                eventRemarks = paste3(survey_seines$survey_comments, survey_seines$sei
                decimalLatitude = survey_seines$lat,
                decimalLongitude = survey_seines$long,
                locationID = survey_seines$site_id,
                coordinatePrecision = 0.00001,
                coordinateUncertaintyInMeters = 10,
                country = "Canada",
                countryCode = "CA",
                stateProvince = "British Columbia",
                habitat = "Nearshore marine",
                geodeticDatum = "EPSG:4326 WGS84",
                minimumDepthInMeters = 0,
                maximumDepthInMeters = 9, # seine depth is 9 m
                samplingProtocol = "http://dx.doi.org/10.21966/1.566666", # This is th
                language = "en",
                license = "http://creativecommons.org/licenses/by/4.0/legalcode",
                bibliographicCitation = "Johnson, B.T., J.C.L. Gan, S.C. Godwin, M. Kr
                references = "https://github.com/HakaiInstitute/jsp-data",
                institutionID = "https://www.gbif.org/publisher/55897143-3f69-42f1-810
                institutionCode = "Hakai"
               )
```

2.1.3 occurrence

Next you'll want to determine what constitutes an occurrence for your data set. Because each event captures fish, I consider each fish to be an occurrence. Therefore, the unit of observation (each row) in the occurrence table is a fish. To link each occurrence to an event you need to include the `eventID` column for every occurrence so that you know what seine (event) each fish (occurrence) came from. You must also provide a globally unique identifier for each occurrence. I already have a locally unique identifier for each fish in the original `fish_data` table called `ufn`. To make it globally unique I pre-pend the organization and research program metadata to the `ufn` column.

Not every fish is actually collected and given a Universal Fish Number (UFN) in our fish data tables, so in our field data sheets we record the total number of fish captured and the total number retained. So to get an occurrence row for every fish captured I create a row for every fish caught (minus the number taken) and create a generic numeric id (ie hakai-jsp-1) in one table and then join that to the fish table that includes a row for every fish retained that already has a UFN.

```
## make table long first
seines_total_long <- survey_seines %>%
  select(seine_id, so_total, pi_total, cu_total, co_total, he_total, ck_total) %>%
  pivot_longer(-seine_id, names_to = "scientificName", values_to = "n")

seines_total_long$scientificName <- recode(seines_total_long$scientificName, so_total = "Oncorhyn

seines_taken_long <- survey_seines %>%
  select(seine_id, so_taken, pi_taken, cu_taken, co_taken, he_taken, ck_taken) %>%
  pivot_longer(-seine_id, names_to = "scientificName", values_to = "n_taken")

seines_taken_long$scientificName <- recode(seines_taken_long$scientificName, so_taken = "Oncorhyn

## remove records that have already been assigned an ID because they were actually retained
seines_long <-  full_join(seines_total_long, seines_taken_long, by = c("seine_id", "scientificNam
  drop_na() %>%
  mutate(n_not_taken = n - n_taken) %>% #so_total includes the number taken so I subtract n_taken
  select(-n_taken, -n) %>%
  filter(n_not_taken > 0)

all_fish_not_retained <-
  seines_long[rep(seq.int(1, nrow(seines_long)), seines_long$n_not_taken), 1:3] %>%
  select(-n_not_taken) %>%
  mutate(prefix = "hakai-jsp-",
         suffix = 1:nrow(.),
         occurrenceID = paste0(prefix, suffix)
  ) %>%
  select(-prefix, -suffix)

#

# Change species names to full Scientific names
latin <- fct_recode(fish_data$species, "Oncorhynchus nerka" = "SO", "Oncorhynchus gorbu
  as.character()

fish_retained_data <- fish_data %>%
  mutate(scientificName = latin) %>%
  select(-species) %>%
  mutate(prefix = "hakai-jsp-",
         occurrenceID = paste0(prefix, ufn)) %>%
  select(seine_id, scientificName, occurrenceID)

occurrence <- bind_rows(all_fish_not_retained, fish_retained_data) %>%
  rename(eventID = seine_id) %>%  # rename = dplyr::rename; vs plyr::rename
  mutate(`Life stage` = "juvenile")

unique_taxa <- unique(occurrence$scientificName)
worms_names <- wm_records_names(unique_taxa) # library(worrms)
df_worms_names <- bind_rows(worms_names) %>%
  select(scientificName = scientificname,
         scientificNameAuthorship = authority,
         taxonRank = rank,
         scientificNameID = lsid
         )

#include bycatch species

unique_bycatch <- unique(bycatch$scientificName) %>%  glimpse()

by_worms_names <- wm_records_names(unique_bycatch) %>%
  bind_rows() %>%
  select(scientificName = scientificname,
         scientificNameAuthorship = authority,
         taxonRank = rank,
         scientificNameID = lsid
         )

bycatch_occurrence <- bycatch %>%
  select(eventID = seine_id, occurrenceID, scientificName, `Life stage` = bm_ageclass)
  filter(scientificName != "unknown")
```

```
bycatch_occurrence$`Life stage`[bycatch_occurrence$`Life stage` == "J"] <- "juvenile"
bycatch_occurrence$`Life stage`[bycatch_occurrence$`Life stage` == "A"] <- "adult"
bycatch_occurrence$`Life stage`[bycatch_occurrence$`Life stage` == "Y"] <- "Young of year"

combined_worms_names <- bind_rows(by_worms_names, df_worms_names) %>%
  distinct(scientificName, .keep_all = TRUE)

occurrence <- bind_rows(bycatch_occurrence, occurrence)

occurrence <- left_join(occurrence, combined_worms_names) %>%
    mutate(basisOfRecord = "HumanObservation",
        occurrenceStatus = "present")

write_csv(occurrence,"../datasets/hakai_salmon_data/raw_data/occurrence.csv") # here::here("..",

# This removes events that didn't result in any occurrences
event <- dplyr::semi_join(event, occurrence, by = 'eventID') %>%
  mutate(coordinateUncertaintyInMeters = ifelse(is.na(decimalLatitude), 1852, coordinateUncertain

simple_sites <- sites %>%
  select(site_id, ocgy_std_lat, ocgy_std_lon)

event <- dplyr::left_join(event, simple_sites, by = c("locationID" = "site_id")) %>%
  mutate(decimalLatitude = coalesce(decimalLatitude, ocgy_std_lat),
         decimalLongitude = coalesce(decimalLongitude, ocgy_std_lon)) %>%
  select(-c(ocgy_std_lat, ocgy_std_lon))

write_csv(event,"../datasets/hakai_salmon_data/raw_data/event.csv") # here::here("..", "datasets"
```

2.1.4 measurementOrFact

To convert all your measurements or facts from your normal format to Darwin Core you essentially need to put all your measurements into one column called measurementType and a corresponding column called MeasurementValue. This standardizes the column names are in the `measurementOrFact` table. There are a number of predefined `measurementType`s listed on the NERC database that should be used where possible. I found it difficult to navigate this page to find the correct `measurementType`.

Here I convert length, and weight measurements that relate to an event and an occurrence and call those `measurementTypes` as `length` and `weight`.

```
mof_types <- read_csv("https://raw.githubusercontent.com/HakaiInstitute/jsp-data/master/OBIS_data
fish_data$weight <- coalesce(fish_data$weight, fish_data$weight_field)
fish_data$fork_length <- coalesce(fish_data$fork_length, fish_data$fork_length_field)
fish_data$`Life stage` <- "juvenile"



measurementOrFact <- fish_data %>%
  mutate(occurrenceID = paste0("hakai-jsp-", ufn)) %>%
  select(occurrenceID, eventID = seine_id, "Length (fork length)" = fork_length,
         "Standard length" = standard_length, "Weight" = weight, `Life stage`) %>%
  pivot_longer(`Length (fork length)`:`Life stage`,
               names_to = "measurementType",
               values_to = "measurementValue",
               values_transform = list(measurementValue = as.character)) %>%
  filter(measurementValue != "NA") %>%
  left_join(mof_types,by = c("measurementType")) %>%
  mutate(measurementValueID = case_when(measurementValue == "juvenile" ~ "http://vocab
         measurementID = paste(eventID, measurementType, occurrenceID, sep = "-"))


write_csv(measurementOrFact,"../datasets/hakai_salmon_data/raw_data/extendedMeasuremen
```

```
#Check that every eventID in Occurrence occurs in event table
no_keys <- dm(event, occurrence, measurementOrFact)
only_pk <- no_keys %>%
  dm_add_pk(event, eventID) %>%
  dm_add_pk(occurrence, occurrenceID) %>%
  dm_add_pk(measurementOrFact, measurementID)
dm_examine_constraints(only_pk)

model <- only_pk %>%
  dm_add_fk(occurrence, eventID, event) %>%
  dm_add_fk(measurementOrFact, occurrenceID, occurrence)
dm_examine_constraints(model)

#TODO: Fix bookdown issues so that dm_draw shows data model html output. Perhaps add
# dm_draw(model, view_type = "all")
```

2.2 Hakai Seagrass

By: ZL Monteith, Hakai Institute

2.2.1 Setup

This section clears the workspace, checks the working directory, and installs packages (if required) and loads packages, and loads necessary datasets

```
# The following command will remove all objects** for a fresh start. Make
#   sure any objects you want to keep are saved before running!
rm(list = ls())

# Check working directory; set if necessary so document will compile
#   properly
getwd()

# Install packages; uncomment and run if packages not already installed
# install.packages(c("tidyverse", "uuid"))

# Load packages
lapply(c("tidyverse", "lubridate", "magrittr", "worrms"),
       library, character.only = TRUE)
```

2.2.1.1 Load Data

First load the seagrass density survey data, set variable classes, and have a quick look

```
# Load density data
seagrassDensity <-
  read.csv("https://raw.githubusercontent.com/ioos/bio_data_guide/main/datasets/hakai_seagrass_da
           colClass = "character") %>%
  mutate(date              = ymd(date),
         depth             = as.numeric(depth),
         transect_dist     = factor(transect_dist),
         collected_start   = ymd_hms(collected_start),
         collected_end     = ymd_hms(collected_end),
         density           = as.numeric(density),
         density_msq       = as.numeric(density_msq),
         canopy_height_cm  = as.numeric(canopy_height_cm),
         flowering_shoots  = as.numeric(flowering_shoots)) %T>%
  glimpse()
```

Next, load the habitat survey data, and same as above, set variable classes as necessary, and have a quick look.

```
# load habitat data, set variable classes, have a quick look
seagrassHabitat <-
  read.csv("https://raw.githubusercontent.com/ioos/bio_data_guide/main/datasets/hakai_s
           colClasses = "character") %>%
  mutate(date            = ymd(date),
         depth           = as.numeric(depth),
         hakai_id        = str_pad(hakai_id, 5, pad = "0"),
         transect_dist   = factor(transect_dist),
         collected_start = ymd_hms(collected_start),
         collected_end   = ymd_hms(collected_end)) %T>%
  glimpse()
```

Finally, load coordinate data for surveys, and subset necessary variables

```
coordinates <-
  read.csv("https://raw.githubusercontent.com/ioos/bio_data_guide/main/datasets/hakai_s
           colClass = c("Point.Name" = "character")) %>%
  select(Point.Name, Decimal.Lat, Decimal.Long) %T>%
  glimpse()
```

2.2.1.2 Merge Datasets

Now all the datasets have been loaded, and briefly formatted, we'll join together the habitat and density surveys, and the coordinates for these.

The seagrass density surveys collect data at discrete points (ie. 5 metres) along the transects, while the habitat surveys collect data over sections (ie. 0 - 5 metres) along the transects. In order to fit these two surveys together, we'll narrow the habitat surveys from a range to a point so the locations will match. Based on how the habitat data is collected, the point the habitat survey is applied to will be the distance at the end of the swath (ie. 10-15m will become 15m). To account for no preceeding distance, the 0m distance will use the 0-5m section of the survey.

First, well make the necessary transformations to the habitat dataset.

```
# Reformat seagrassHabitat to merge with seagrassDensity
## replicate 0 - 5m transect dist to match with 0m in density survey;
## rest of habitat bins can map one to one with density (ie. 5 - 10m -> 10m)
seagrass0tmp <-
  seagrassHabitat %>%
  filter(transect_dist %in% c("0 - 5", "0 - 2.5")) %>%
  mutate(transect_dist = factor(0))

## collapse various levels to match with seagrassDensity transect_dist
```

```
seagrassHabitat$transect_dist <-
  fct_collapse(seagrassHabitat$transect_dist,
               "5" = c("0 - 5", "2.5 - 7.5"),
               "10" = c("5 - 10", "7.5 - 12.5"),
               "15" = c("10 - 15", "12.5 - 17.5"),
               "20" = c("15 - 20", "17.5 - 22.5"),
               "25" = c("20 - 25", "22.5 - 27.5"),
               "30" = c("25 - 30", "27.5 - 30"))

## merge seagrass0tmp into seagrassHabitat to account for 0m samples,
## set class for date, datetime variables
seagrassHabitatFull <-
  rbind(seagrass0tmp, seagrassHabitat) %>%
  filter(transect_dist != "0 - 2.5")  %>% # already captured in seagrass0tmp
  droplevels(.)  # remove now unused factor levels
```

With the distances of habitat and density surveys now corresponding, we can now merge these two datasets plus there coordinates together, combine redundant fields, and remove unnecessary fields.

```
# Merge seagrassHabitatFull with seagrassDensity, then coordinates
seagrass <-
  full_join(seagrassHabitatFull, seagrassDensity,
            by = c("organization",
                   "work_area",
                   "project",
                   "survey",
                   "site_id",
                   "date",
                   "transect_dist")) %>%
  # merge hakai_id.x and hakai_id.y into single variable field;
  # use combination of date, site_id, transect_dist, and field uid (hakai_id
  # when present)
  mutate(field_uid = ifelse(sample_collected == TRUE, hakai_id.x, "NA"),
         hakai_id = paste(date, "HAKAI:CALVERT", site_id, transect_dist, sep = ":"),
         # below, aggregate metadata that didn't merge naturally (ie. due to minor
         # differences in watch time or depth gauges)
         dive_supervisor = dive_supervisor.x,
         collected_start = ymd_hms(ifelse(is.na(collected_start.x),
                                          collected_start.y,
                                          collected_start.x)),
         collected_end   = ymd_hms(ifelse(is.na(collected_start.x),
                                          collected_start.y,
                                          collected_start.x)),
         depth_m         = ifelse(is.na(depth.x), depth.y, depth.x),
         sampling_bout   = sampling_bout.x) %>%
  left_join(., coordinates,  # add coordinates
            by = c("site_id" = "Point.Name")) %>%
  select( - c(X.x, X.y, hakai_id.x, hakai_id.y,  # remove unnecessary variables
              dive_supervisor.x, dive_supervisor.y,
              collected_start.x, collected_start.y,
              collected_end.x, collected_end.y,
              depth.x, depth.y,
              sampling_bout.x, sampling_bout.y)) %>%
  mutate(density_msq = as.character(density_msq),
         canopy_height_cm = as.character(canopy_height_cm),
         flowering_shoots = as.character(flowering_shoots),
         depth_m = as.character(depth_m)) %T>%
  glimpse()
```

2.2.2 Convert Data to Darwin Core - Extended Measurement or Fact format

The Darwin Core ExtendedMeasurementOrFact (eMoF) extension bases records around a core event (rather than occurrence as in standard Darwin Core), allowing for additional measurement variables to be associated with occurrence data.

2.2.2.1 Add Event ID and Occurrence ID variables to dataset

As this dataset will be annually updated, rather than using natural keys (ie. using package::uuid to autogenerate) for event and occurence IDs, here we will use surrogate keys made up of a concatenation of date survey, transect location, observation distance, and sample ID (for occurrenceID, when a sample is present).

```
# create and populate eventID variable
## currently only event is used, but additional surveys and abiotic data
## are associated with parent events that may be included at a later date
seagrass$eventID <- seagrass$hakai_id

# create and populate occurrenceID; combine eventID with transect_dist
# and field_uid
## in the event of <NA> field_uid, no sample was collected, but
## measurements and occurrence are still taken; no further subsamples
## are associated with <NA> field_uids
seagrass$occurrenceID <-
```

```
  with(seagrass,
       paste(eventID, transect_dist, field_uid, sep = ":"))
```

2.2.2.2 Create Event, Occurrence, and eMoF tables

Now that we've created eventIDs and occurrenceIDs to connect all the variables together, we can begin to create the Event, Occurrence, and extended Measurement or Fact table necessary for DarwinCore compliant datasets

```
# subset seagrass to create event table
seagrassEvent <-
  seagrass %>%
  distinct %>%  # some duplicates in data stemming from database conflicts
  select(date,
         Decimal.Lat, Decimal.Long, transect_dist,
         depth_m, eventID) %>%
  rename(eventDate                     = date,
         decimalLatitude               = Decimal.Lat,
         decimalLongitude              = Decimal.Long,
         coordinateUncertaintyInMeters = transect_dist,
         minimumDepthInMeters          = depth_m,
         maximumDepthInMeters          = depth_m) %>%
  mutate(geodeticDatum  = "WGS84",
         samplingEffort = "30 metre transect") %T>% glimpse

# save event table to csv
write.csv(seagrassEvent, "../datasets/hakai_seagrass_data/processed_data/hakaiSeagrassDwcEvent.cs
```

2.2.2.2.1 Event Table

```
# subset seagrass to create occurrence table
seagrassOccurrence <-
  seagrass %>%
  distinct %>%  # some duplicates in data stemming from database conflicts
  select(eventID, occurrenceID) %>%
  mutate(basisOfRecord = "HumanObservation",
         scientificName   = "Zostera subg. Zostera marina",
         occurrenceStatus = "present")
```

```
# Taxonomic name matching
# in addition to the above metadata, DarwinCore format requires further
# taxonomic data that can be acquired through the WoRMS register.
## Load taxonomic info, downloaded via WoRMS tool
# zmWorms <-
#   read.delim("raw_data/zmworms_matched.txt",
#              header = TRUE,
#              nrows  = 1)

zmWorms <- wm_record(id = 145795)

# join WoRMS name with seagrassOccurrence create above
seagrassOccurrence <-
  full_join(seagrassOccurrence, zmWorms,
            by = c("scientificName" = "scientificname")) %>%
  select(eventID, occurrenceID, basisOfRecord, scientificName, occurrenceStatus, AphiaI
         url, authority, status, unacceptreason, taxonRankID, rank,
         valid_AphiaID, valid_name, valid_authority, parentNameUsageID,
         kingdom, phylum, class, order, family, genus, citation, lsid,
         isMarine, match_type, modified) %T>%
  glimpse

# save occurrence table to csv
write.csv(seagrassOccurrence, "../datasets/hakai_seagrass_data/processed_data/hakaiSeag
```

2.2.2.2.2 Occurrence Table

```
seagrassMof <-
  seagrass %>%
  # select variables for eMoF table
  select(date,
         eventID, survey, site_id, transect_dist,
         substrate, patchiness, adj_habitat_1, adj_habitat_2,
         vegetation_1, vegetation_2,
         density_msq, canopy_height_cm, flowering_shoots) %>%
  # split substrate into two variables (currently holds two substrate type in same var
  separate(substrate, sep = ",", into = c("substrate_1", "substrate_2")) %>%
  # change variables names to match NERC database (or to be more descriptive where non
  rename(measurementDeterminedDate   = date,
         SubstrateTypeA              = substrate_1,
         SubstrateTypeB              = substrate_2,
         BarePatchLengthWithinSeagrass = patchiness,
         PrimaryAdjacentHabitat      = adj_habitat_1,
         SecondaryAdjacentHabitat    = adj_habitat_2,
         PrimaryAlgaeSp              = vegetation_1,
         SecondaryAlgaeSp            = vegetation_2,
         BedAbund                    = density_msq,
         CanopyHeight                = canopy_height_cm,
         FloweringBedAbund           = flowering_shoots) %>%
  # reformat variables into DwC MeasurementOrFact format
  # (single values variable, with measurement type, unit, etc. variables)
  pivot_longer( - c(measurementDeterminedDate, eventID, survey, site_id, transect_dist),
                names_to = "measurementType",
                values_to = "measurementValue",
                values_ptypes = list(measurementValue = "character")) %>%
  # use measurement type to fill in remainder of variables relating to
  # NERC vocabulary and metadata fields
  mutate(
    measurementTypeID = case_when(
      measurementType == "BedAbund" ~ "http://vocab.nerc.ac.uk/collection/P01/current/SDBIOL02/",
      measurementType == "CanopyHeight" ~ "http://vocab.nerc.ac.uk/collection/P01/current/OBSMAXL
      # measurementType == "BarePatchWithinSeagrass" ~ "",
      measurementType == "FloweringBedAbund" ~ "http://vocab.nerc.ac.uk/collection/P01/current/SD
    measurementUnit = case_when(
      measurementType == "BedAbund" ~ "Number per square metre",
      measurementType == "CanopyHeight" ~ "Centimetres",
      measurementType == "BarePatchhLengthWithinSeagrass" ~ "Metres",
      measurementType == "FloweringBedAbund" ~ "Number per square metre"),
    measurementUnitID = case_when(
      measurementType == "BedAbund" ~ "http://vocab.nerc.ac.uk/collection/P06/current/UPMS/",
      measurementType == "CanopyHeight" ~ "http://vocab.nerc.ac.uk/collection/P06/current/ULCM/",
      measurementType == "BarePatchhLengthWithinSeagrass" ~ "http://vocab.nerc.ac.uk/collection/P
      measurementType == "FloweringBedAbund" ~ "http://vocab.nerc.ac.uk/collection/P06/current/UP
    measurementAccuracy = case_when(
      measurementType == "CanopyHeight" ~ 5),
    measurementMethod = case_when(
      measurementType == "BedAbund" ~ "25cmx25cm quadrat count",
      measurementType == "CanopyHeight" ~ "in situ with ruler",
      measurementType == "BarePatchhLengthWithinSeagrass" ~ "estimated along transect line",
      measurementType == "FloweringBedAbund" ~ "25cmx25cm quadrat count")) %>%
  select(eventID, measurementDeterminedDate, measurementType, measurementValue,
         measurementTypeID, measurementUnit, measurementUnitID, measurementAccuracy,
         measurementMethod) %T>%
#  select(!c(survey, site_id, transect_dist)) %T>%
  glimpse()

# save eMoF table to csv
```

```
write.csv(seagrassMof, "../datasets/hakai_seagrass_data/processed_data/hakaiSeagrassDwc
```

2.2.2.2.3 Extended MeasurementOrFact table

2.2.3 Session Info

Print session information below in case necessary for future reference

```
# Print Session Info for future reference
sessionInfo()
```

Chapter 3

Final Words

We have finished a nice book.

Chapter 4

Tools

Below are some of the tools and packages used in workflows. R and Python package “Type” is BIO for packages specifically for biological applications, and GEN for generic packages.

4.1 R

Package	Type	Description
bdveRse	BIO	A family of R packages for biodiversity data.
ecocomDP	BIO	Work with the Ecological Community Data Design Pattern. ‘ecocomDP’ is a flexible data model for harmonizing ecological community surveys, in a research question agnostic format, from source data published across repositories, and with methods that keep the derived data up-to-date as the underlying sources change.

Package	Type	Description
EDIorg/EMLasseblyline	BIO	For scientists and data managers to create high quality EML metadata for dataset publication.
finch	BIO	Parse Darwin Core Files
iobis/obistools	BIO	Tools for data enhancement and quality control.
robis	BIO	R client for the OBIS API
ropensci/EML	BIO	Provides support for the serializing and parsing of all low-level EML concepts
taxize	BIO	Interacts with a suite of web ‘APIs’ for taxonomic tasks, such as getting database specific taxonomic identifiers, verifying species names, getting taxonomic hierarchies, fetching downstream and upstream taxonomic names, getting taxonomic synonyms, converting scientific to common names and vice versa, and more.
worrms	BIO	Client for World Register of Marine Species. Includes functions for each of the API methods, including searching for names by name, date and common names, searching using external identifiers, fetching synonyms, as well as fetching taxonomic children and taxonomic classification.

Package	Type	Description
Hmisc	GEN	Contains many functions useful for data analysis, high-level graphics, utility operations, functions for computing sample size and power, simulation, importing and annotating datasets, imputing missing values, advanced table making, variable clustering, character string manipulation, conversion of R objects to LaTeX and html code, and recoding variables. Particularly check out the describe() function.
lubridate	GEN	Functions to work with date-times and time-spans: fast and user friendly parsing of date-time data, extraction and updating of components of a date-time (years, months, days, hours, minutes, and seconds), algebraic manipulation on date-time and time-span objects.
stringr	GEN	Simple, Consistent Wrappers for Common String Operations

Package	Type	Description
tidyverse	GEN	The ‘tidyverse’ is a set of packages that work in harmony because they share common data representations and ‘API’ design. This package is designed to make it easy to install and load multiple ‘tidyverse’ packages in a single step.
uuid	GEN	Tools for generating and handling of UUIDs (Universally Unique Identifiers).

4.2 Python

Package	Type	Description
metapype	BIO	A lightweight Python 3 library for generating EML metadata
python-dwca-reader	BIO	A simple Python package to read and parse Darwin Core Archive (DwC-A) files, as produced by the GBIF website, the IPT and many other biodiversity informatics tools.
pyworms	BIO	Python client for the World Register of Marine Species (WoRMS) REST service.

Package	Type	Description
numpy	GEN	NumPy (Numerical Python) is an open source Python library that's used in almost every field of science and engineering. It's the universal standard for working with numerical data in Python, and it's at the core of the scientific Python and PyData ecosystems.
pandas	GEN	pandas is a fast, powerful, flexible and easy to use open source data analysis and manipulation tool, built on top of the Python programming language. Super helpful when manipulating tabular data!
uuid	GEN	This module provides immutable UUID objects (class UUID) and the functions uuid1(), uuid3(), uuid4(), uuid5() for generating version 1, 3, 4, and 5 UUIDs as specified in RFC 4122. Built in – part of the Python standard library.

4.3 Google Sheets

Package	Description
Google Sheet DarwinCore Archive Assistant add-on	Google Sheet add-on which assists the creation of Darwin Core Archives (DwCA) and publising to Zenodo. DwCA's are stored into user's Google Drive and can be downloaded for upload into IPT installations or other software which is able to read DwC-archives.

4.4 Validators

Name	Description
Darwin Core Archive Validator	This validator verifies the structural integrity of a Darwin Core Archive. It does not check the data values, such as coordinates, dates or scientific names.
GBIF DATA VALIDATOR	The GBIF data validator is a service that allows anyone with a GBIF-relevant dataset to receive a report on the syntactical correctness and the validity of the content contained within the dataset.
LifeWatch Belgium	Through this interactive section of the LifeWatch.be portal users can upload their own data using a standard data format, and choose from several web services, models and applications to process the data.

References

Allaire, JJ, Yihui Xie, Jonathan McPherson, Javier Luraschi, Kevin Ushey, Aron Atkins, Hadley Wickham, Joe Cheng, Winston Chang, and Richard Iannone. 2021. *Rmarkdown: Dynamic Documents for r*. https://CRAN.R-project.org/package=rmarkdown.

Barbier, Edward B. 2017. "Marine Ecosystem Services." *Current Biology*. Cell Press. https://doi.org/10.1016/j.cub.2017.03.020.

Benson, Abigail, Cassandra M. Brooks, Gabrielle Canonico, Emmett Duffy, Frank Muller-Karger, Heidi M. Sosik, Patricia Miloslavich, and Eduardo Klein. 2018. "Integrated Observations and Informatics Improve Understanding of Changing Marine Ecosystems." *Frontiers in Marine Science*. Frontiers Media S.A. https://doi.org/10.3389/fmars.2018.00428.

Benson, Abigail, Diana LaScala-Gruenewald, Robert McGuinn, Erin Satterthwaite, Stace Beaulieu, Mathew Biddle, Lynn deWitt, et al. 2021. "Biological Observation Data Standardization - a Primer for Data Managers." ESIP. https://doi.org/10.6084/m9.figshare.16806712.v1.

Benson, Abigail, Tylar Murray, Gabrielle Canonico, Enrique Montes, Frank Muller-Karger, Maria T. Kavanaugh, Joaquin Trinanes, and Lynn M. deWitt. 2021. "Data Management and Interactive Visualizations for the Evolving Marine Biodiversity Observation Network." *Oceanography*. https://doi.org/10.5670/oceanog.2021.220.

Borer, Elizabeth T., Eric W. Seabloom, Matthew B. Jones, and Mark Schildhauer. 2009. "Some Simple Guidelines for Effective Data Management." *Bulletin of the Ecological Society of America* 90 (April). https://doi.org/10.1890/0012-9623-90.2.205.

Canonico, Gabrielle, Pier Luigi Buttigieg, Enrique Montes, Frank E. Muller-Karger, Carol Stepien, Dawn Wright, Abigail Benson, et al. 2019. "Global Observational Needs and Resources for Marine Biodiversity." *Frontiers in Marine Science*. Frontiers Media S.A. https://doi.org/10.3389/fmars.2019.00367.

Crystal-Ornelas, Robert, Charuleka Varadharajan, Ben Bond-Lamberty, Kristin Boye, Madison Burrus, Shreyas Cholia, Michael Crow, et al. 2021. "A Guide to Using GitHub for Developing and Versioning Data Standards and Reporting Formats." *Earth and Space Science*, July, e2021EA001797. https://doi.org/10.1029/2021EA001797.

Davies, Neil, John Deck, Eric C Kansa, Sarah Whitcher Kansa, John Kunze, Christopher Meyer, Thomas Orrell, et al. 2021. "Internet of Samples (iSamples): Toward an Interdisciplinary Cyberinfrastructure for Material Samples." *GigaScience* 10: 1–5. https://doi.org/10.1093/gigascience/giab028.

Djurhuus, Anni, Collin J. Closek, Ryan P. Kelly, Kathleen J. Pitz, Reiko P. Michisaki, Hilary A. Starks, Kristine R. Walz, et al. 2020. "Environmental DNA Reveals Seasonal Shifts and Potential Interactions in a Marine Community." *Nature Communications* 11 (December): 1–9. https://doi.org/10.1038/s41467-019-14105-1.

Duffy, J. Emmett, Linda A. Amaral-Zettler, Daphne G. Fautin, Gustav Paulay, Tatiana A. Rynearson, Heidi M. Sosik, and John J. Stachowicz. 2013. "Envisioning a Marine Biodiversity Observation Network." *BioScience* 63 (May): 350–61. https://doi.org/10.1525/bio.2013.63.5.8.

Fornwall, M, R Gisiner, S E Simmons, H Moustahfid, G Canonico, P Halpin, P Goldstein, et al. 2012. "Expanding Biological Data Standards Development Processes for US IOOS: Visual Line Transect Observing Community for Mammal, Bird, and Turtle Data." IOOS. https://www.researchgate.net/publication/255681522.

Hardisty, Alex R., William K. Michener, Donat Agosti, Enrique Alonso García, Lucy Bastin, Lee Belbin, Anne Bowser, et al. 2019. "The Bari Manifesto: An Interoperability Framework for Essential Biodiversity Variables." *Ecological Informatics* 49 (January): 22–31. https://doi.org/10.1016/j.ecoinf.2018.11.003.

Heberling, J Mason, Joseph T Miller, Daniel Noesgaard, Scott B Weingart C, Dmitry Schigel, and Douglas E Soltis. 2021. "Data Integration Enables Global Biodiversity Synthesis." *Proceedings of the National Academy of Sciences of the United States of America*. https://doi.org/10.1073/pnas.2018093118/-/DCSupplemental.

Jonathan A. Hare, Mark W. Nelson, Wendy E. Morrison. n.d. "A Vulnerability Assessment of Fish and Invertebrates to Climate Change on the Northeast u.s. Continental Shelf." *PLoS ONE* 11 (2): e0146756. https://doi.org/10.1371/journal.pone.0146756.

Jones, Matthew B., Mark P. Schildhauer, O. J. Reichman, and Shawn Bowers. 2006. "The New Bioinformatics: Integrating Ecological Data from the Gene to the Biosphere." *Annual Review of Ecology, Evolution, and Systematics*. https://doi.org/10.1146/annurev.ecolsys.37.091305.110031.

Kavanaugh, Maria T., Matthew J. Oliver, Francisco P. Chavez, Ricardo M. Letelier, Frank E. Muller-Karger, and Scott C. Doney. 2016. "Seascapes as a New Vernacular for Pelagic Ocean Monitoring, Management and Conservation." *ICES Journal of Marine Science* 73 (July): 1839–50. https://doi.org/10.1093/icesjms/fsw086.

Kot, Connie Y., Ei Fujioka, Lucie J. Hazen, Benjamin D. Best, Andrew J. Read, and Patrick N. Halpin. 2010. "Spatio-Temporal Gap Analysis of OBIS-SEAMAP Project Data: Assessment and Way Forward." *PLoS ONE* 5 (September): 12990. https://doi.org/10.1371/journal.pone.0012990.

Lab, Malin Pinksy. n.d. "OceanAdapt." *GitHub*. https://oceanadapt.rutgers.

edu/.

Lamprecht, Anna-Lena, Leyla Garcia, Mateusz Kuzak, Carlos Martinez, Ricardo Arcila, Eva Martin Del Pico, Victoria Dominguez Del Angel, et al. 2019. "Towards FAIR Principles for Research Software." Edited by Paul Groth. *Data Science*, 1–23. https://doi.org/10.3233/DS-190026.

Maria Dornelas, Brian McGill, Nicholas J. Gotelli. 2014. "Assemblage Time Series Reveal Biodiversity Change but Not Systematic Loss." *Science* 344 (6181): 296–99. https://doi.org/10.1126/science.1248484.

McKenna, Megan F., Simone Baumann-Pickering, Annebelle C. M. Kok, William K. Oestreich, Jeffrey D. Adams, Jack Barkowski, Kurt M. Fristrup, et al. 2021. "Advancing the Interpretation of Shallow Water Marine Soundscapes." *Frontiers in Marine Science* 0 (September): 1426. https://doi.org/10.3389/FMARS.2021.719258.

Miloslavich, Patricia, Nicholas J. Bax, Samantha E. Simmons, Eduardo Klein, Ward Appeltans, Octavio Aburto-Oropeza, Melissa Andersen Garcia, et al. 2018. "Essential Ocean Variables for Global Sustained Observations of Biodiversity and Ecosystem Changes." *Global Change Biology* 24 (June): 2416–33. https://doi.org/10.1111/gcb.14108.

Montes, Enrique, Anni Djurhuus, Frank E. Muller-Karger, Daniel Otis, Christopher R. Kelble, and Maria T. Kavanaugh. 2020. "Dynamic Satellite Seascapes as a Biogeographic Framework for Understanding Phytoplankton Assemblages in the Florida Keys National Marine Sanctuary, United States." *Frontiers in Marine Science* 7 (July): 575. https://doi.org/10.3389/fmars.2020.00575.

Moustahfid, Hassan, and Philip Goldstein. 2014. "IOOS Biological Data Services Enrollment Procedures."

Moustahfid, Hassan, Jim Potemra, Philip Goldstein, Roy Mendelssohn, and Annette Desrochers. 2011. "Making United States Integrated Ocean Observing System (u.s. IOOS) Inclusive of Marine Biological Resources." https://www.researchgate.net/publication/254013004.

Muller-Karger, Frank E., Erin Hestir, Christiana Ade, Kevin Turpie, Dar A. Roberts, David Siegel, Robert J. Miller, et al. 2018. "Satellite Sensor Requirements for Monitoring Essential Biodiversity Variables of Coastal Ecosystems." *Ecological Applications* 28 (April): 749–60. https://doi.org/10.1002/eap.1682.

Muller-Karger, Frank E., Patricia Miloslavich, Nicholas J. Bax, Samantha Simmons, Mark J. Costello, Isabel Sousa Pinto, Gabrielle Canonico, et al. 2018. "Advancing Marine Biological Observations and Data Requirements of the Complementary Essential Ocean Variables (EOVs) and Essential Biodiversity Variables (EBVs) Frameworks." *Frontiers in Marine Science*. Frontiers Media S.A. https://doi.org/10.3389/fmars.2018.00211.

Neeley, Aimee, Stace E. Beaulieu, Chris Proctor, Ivona Cetinić, Joe Futrelle, Inia Soto Ramos, Heidi M. Sosik, et al. 2021. "Standards and Practices for Reporting Plankton and Other Particle Observations from Images." https://doi.org/10.1575/1912/27377.

O'Brien, Margaret, Colin A. Smith, Eric R. Sokol, Corinna Gries, Nina Lany, Sydne Record, and Max C. N. Castorani. 2021. “ecocomDP: A Flexible Data Design Pattern for Ecological Community Survey Data.” *Ecological Informatics* 64 (September): 101374. https://doi.org/10.1016/J.ECOINF.2021.101374.

Pooter, Daphnis De, Ward Appeltans, Nicolas Bailly, Sky Bristol, Klaas Deneudt, Menashè Eliezer, Ei Fujioka, et al. 2017. “Toward a New Data Standard for Combined Marine Biological and Environmental Datasets - Expanding OBIS Beyond Species Occurrences.” *Biodiversity Data Journal* 5 (January): 10989. https://doi.org/10.3897/BDJ.5.e10989.

Provoost, Pieter. n.d. “Iobis/Ebsa.” *GitHub*. https://github.com/iobis/ebsa.

R Core Team. 2021. *R: A Language and Environment for Statistical Computing*. Vienna, Austria: R Foundation for Statistical Computing. https://www.R-project.org/.

Rücknagel, J., P. Vierkant, R. Ulrich, G. Kloska, E. Schnepf, D. Fichtmüller, E. Reuter, et al. 2015. “Metadata Schema for the Description of Research Data Repositories. Version 3.0.” https://doi.org/10.2312/re3.008.

Rule, Adam, Amanda Birmingham, Cristal Zuniga, Ilkay Altintas, Shih-Cheng Huang, Rob Knight, Niema Moshiri, et al. 2019. “Ten Simple Rules for Writing and Sharing Computational Analyses in Jupyter Notebooks.” *PLOS Computational Biology* 15 (July): e1007007. https://doi.org/10.1371/JOURNAL.PCBI.1007007.

Santora, Jarrod A., Elliott L. Hazen, Isaac D. Schroeder, Steven J. Bograd, Keith M. Sakuma, and John C. Field. 2017. “Impacts of Ocean Climate Variability on Biodiversity of Pelagic Forage Species in an Upwelling Ecosystem.” *Marine Ecology Progress Series* 580 (September): 205–20. https://doi.org/10.3354/meps12278.

Schmid, Moritz S, Dominic Daprano, Kyler M Jacobson, Christopher Sullivan, Christian Briseño-Avena, Jessica Y Luo, and Robert K Cowen. 2021. *A Convolutional Neural Network Based High-Throughput Image Classification Pipeline - Code and Documentation to Process Plankton Underwater Imagery Using Local HPC Infrastructure and NSF’s XSEDE*. Zenodo. https://doi.org/10.5281/ZENODO.4641158.

Taylor, Gordon T., Frank E. Muller-Karger, Robert C. Thunell, Mary I. Scranton, Yrene Astor, Ramon Varela, Luis Troccoli Ghinaglia, et al. 2012. “Ecosystem Responses in the Southern Caribbean Sea to Global Climate Change.” *Proceedings of the National Academy of Sciences of the United States of America* 109 (November): 19315–20. https://doi.org/10.1073/pnas.1207514109.

Tittensor, Derek P, Camilo Mora, Walter Jetz, Heike K Lotze, Daniel Ricard, Edward Vanden Berghe, and Boris Worm. 2010. “Global Patterns and Predictors of Marine Biodiversity Across Taxa.” *Nature* 466 (7310): 1098.

Warren, R., J. Price, E. Graham, N. Forstenhaeusler, and J. VanDerWal. 2018. “The Projected Effect on Insects, Vertebrates, and Plants of Limiting Global Warming to 1.5°c Rather Than 2°c.” *Science* 360 (May): 791–95. https://doi.org/10.1126/science.aar3646.

Wieczorek, John, David Bloom, Robert Guralnick, Stan Blum, Markus Döring, Renato Giovanni, Tim Robertson, and David Vieglais. 2012. "Darwin Core: An Evolving Community-Developed Biodiversity Data Standard." *PLoS ONE* 7 (January): 29715. https://doi.org/10.1371/journal.pone.0029715.

Wilkinson, Mark D., Michel Dumontier, IJsbrand Jan Aalbersberg, Gabrielle Appleton, Myles Axton, Arie Baak, Niklas Blomberg, et al. 2016. "The FAIR Guiding Principles for Scientific Data Management and Stewardship." *Scientific Data* 3 (March): 1–9. https://doi.org/10.1038/sdata.2016.18.

Xie, Yihui. 2014. "Knitr: A Comprehensive Tool for Reproducible Research in R." In *Implementing Reproducible Computational Research*, edited by Victoria Stodden, Friedrich Leisch, and Roger D. Peng. Chapman; Hall/CRC. http://www.crcpress.com/product/isbn/9781466561595.

———. 2015b. *Dynamic Documents with R and Knitr*. 2nd ed. Boca Raton, Florida: Chapman; Hall/CRC. http://yihui.name/knitr/.

———. 2015a. *Dynamic Documents with R and Knitr*. 2nd ed. Boca Raton, Florida: Chapman; Hall/CRC. https://yihui.org/knitr/.

———. 2016. *Bookdown: Authoring Books and Technical Documents with R Markdown*. Boca Raton, Florida: Chapman; Hall/CRC. https://bookdown.org/yihui/bookdown.

———. 2021a. *Bookdown: Authoring Books and Technical Documents with r Markdown*. https://CRAN.R-project.org/package=bookdown.

———. 2021b. *Knitr: A General-Purpose Package for Dynamic Report Generation in r*. https://yihui.org/knitr/.

Xie, Yihui, J. J. Allaire, and Garrett Grolemund. 2018. *R Markdown: The Definitive Guide*. Boca Raton, Florida: Chapman; Hall/CRC. https://bookdown.org/yihui/rmarkdown.

Xie, Yihui, Christophe Dervieux, and Emily Riederer. 2020. *R Markdown Cookbook*. Boca Raton, Florida: Chapman; Hall/CRC. https://bookdown.org/yihui/rmarkdown-cookbook.