Package: movepub (via r-universe)

August 21, 2024

Title Prepare Movebank Data for Publication

Version 0.3.0

Date 2024-04-24

Description Prepare animal tracking data from 'Movebank' (<https://movebank.org>) for publication in a research repository or the Global Biodiversity Information Facility ('GBIF' <https://www.gbif.org>).

License MIT + file LICENSE

URL https://github.com/inbo/movepub

BugReports https://github.com/inbo/movepub/issues

Depends R (>= 2.10)

Imports cli, dplyr, EML, frictionless, jsonlite, purrr, readr, rlang, tidyr, uuid, worrms

Suggests knitr, rmarkdown, stringr, testthat (>= 3.0.0), xml2

VignetteBuilder knitr

Remotes https://github.com/frictionlessdata/frictionless-r

Config/testthat/edition 3

Encoding UTF-8

LazyData true

Roxygen list(markdown = TRUE)

RoxygenNote 7.3.2

Repository https://ocean-tracking-network.r-universe.dev

RemoteUrl https://github.com/inbo/movepub

RemoteRef HEAD

RemoteSha 09b892cd7c20bc171c9c9e1bc9beaa0942a4d045

Contents

`add_resource` *Add Movebank data to a Frictionless Data Package*

Description

Adds Movebank data (`reference-data`, `gps`, `acceleration`, `accessory-measurements`) as a Data Resource to a Frictionless Data Package. The function extends `frictionless::add_resource()`. The title, definition, format and URI of each field are looked up in the latest version of the Movebank Attribute Dictionary and included in the Table Schema of the resource.

Usage

```
add_resource(package, resource_name, files, keys = TRUE)
```

Arguments

`package`	Data Package object, created with `read_package()` or `create_package()`.
`resource_name`	Name of the Data Resource.
`files`	One or more paths to CSV file(s) that contain the data for this resource, as a character (vector).
`keys`	Should `primaryKey` and `foreignKey` properties be added to the Table Schema?

Details

See Get started for examples.

Value

Provided package with one additional resource.

`datacite_to_eml` *Get DataCite metadata as EML*

Description

Get metadata from DataCite and transform to EML.

Usage

```
datacite_to_eml(doi)
```

Arguments

`doi`	DOI of a dataset.

Value

EML list that can be extended and/or written to file with `EML::write_eml()`.

See Also

Other support functions: `get_aphia_id()`, `get_mvb_term()`

`get_aphia_id` *Get WoRMS AphiaID from a taxonomic name*

Description

This function wraps `worrms::wm_name2id_()` so that it returns a data frame rather than a list. It also silences "not found" warnings, returning `NA` instead.

Usage

```
get_aphia_id(x)
```

Arguments

`x`	A (vector with) taxonomic name(s).

Value

Data frame with `name`, `aphia_id`, `aphia_lsid` and `aphia_url`.

See Also

Other support functions: `datacite_to_eml()`, `get_mvb_term()`

Examples

```
get_aphia_id("Mola mola")
get_aphia_id(c("Mola mola", "not_a_name"))
```

get_mvb_term *Get term from the Movebank Attribute Dictionary*

Description

Search a term by its label in the Movebank Attribute Dictionary (MVB). Returns in order: term with matching `prefLabel`, matching `altLabel` or error when no matching term is found.

Usage

```
get_mvb_term(label)
```

Arguments

`label`	Label of the term to look for. Case will be ignored and `-`, `_`, `.` and `:` interpreted as space.

Value

List with term information.

See Also

Other support functions: `datacite_to_eml()`, `get_aphia_id()`

Examples

```
get_mvb_term("animal_id")

get_mvb_term("Deploy.On.Date")
```

`o_assen` *Sample Movebank dataset with GPS tracking data*

Description

A sample Movebank dataset with GPS tracking data, formatted as a Frictionless Data Package and read by `read_package()`.

Usage

```
o_assen
```

Format

An object of class `datapackage` (inherits from `list`) of length 7.

Details

This sample is derived from the Zenodo-deposited dataset Dijkstra et al. (2022), but excludes the acceleration data.

Source

https://doi.org/10.5281/zenodo.10053903

Examples

```
## Not run:
# The data in o_assen was created with the code below
o_assen <-
  read_package("https://zenodo.org/records/10053903/files/datapackage.json") %>%
  remove_resource("acceleration")
o_assen$title <- "O_ASSEN - Eurasian oystercatchers (Haematopus ostralegus, Haematopodidae) breeding in Assen (the
o_assen$licenses[[1]]$name <- "CC0-1.0"
o_assen$contributors[[1]]$title <- "Vogelwerkgroep Assen"
o_assen$contributors[[1]]$role <- "rightsholder"
usethis::use_data(o_assen, overwrite = TRUE)

## End(Not run)
```

write_dwc — *Transform Movebank data to a Darwin Core Archive*

Description

Transforms a Movebank dataset (formatted as a Frictionless Data Package) to a Darwin Core Archive.

Usage

```
write_dwc(
  package,
  directory,
  dataset_id = package$id,
  dataset_name = package$title,
  license = NULL,
  rights_holder = NULL
)
```

Arguments

`package`	A Frictionless Data Package of Movebank data, as returned by `read_package()`. It is expected to contain a `reference-data` and `gps` resource.
`directory`	Path to local directory to write files to.
`dataset_id`	An identifier for the dataset.
`dataset_name`	Title of the dataset.
`license`	License of the dataset.
`rights_holder`	Acronym of the organization owning or managing the rights over the data.

Details

The resulting files can be uploaded to an IPT for publication to GBIF and/or OBIS. A corresponding `eml.xml` metadata file can be created with `write_eml()`. See `vignette("movepub")` for an example.

Value

CSV and `meta.xml` files written to disk. And invisibly, a list of data frames with the transformed data.

Transformation details

This function **follows recommendations** suggested by Peter Desmet, Sarah Davidson, John Wieczorek and others and transforms data to:

- An Occurrence core.

- An Extended Measurements Or Facts extension
- A `meta.xml` file.

Key features of the Darwin Core transformation:

- Deployments (animal+tag associations) are parent events, with tag attachment (a human observation) and GPS positions (machine observations) as child events. No information about the parent event is provided other than its ID, meaning that data can be expressed in an Occurrence core with one row per observation and `parentEventID` shared by all occurrences in a deployment.
- The tag attachment event often contains metadata about the animal (sex, life stage, comments) and deployment as a whole. The sex and life stage are additionally provided in an Extended Measurement Or Facts extension, where values are mapped to a controlled vocabulary recommended by OBIS.
- No event/occurrence is created for the deployment end, since the end date is often undefined, unreliable and/or does not represent an animal occurrence.
- Only `visible` (non-outlier) GPS records that fall within a deployment are included.
- GPS positions are downsampled to the **first GPS position per hour**, to reduce the size of high-frequency data. It is possible for a deployment to contain no GPS positions, e.g. if the tag malfunctioned right after deployment.
- Parameters or metadata are used to set the following record-level terms:
  - `dwc:datasetID`: `dataset_id`, defaulting to `package$id`.
  - `dwc:datasetName`: `dataset_name`, defaulting to `package$title`.
  - `dcterms:license`: `license`, defaulting to the first license name (e.g. `CC0-1.0`) in `package$licenses`.
  - `dcterms:rightsHolder`: `rights_holder`, defaulting to the first contributor in `package$contributors` with role `rightsHolder`.

See Also

Other dwc functions: `write_eml()`

Examples

```
write_dwc(o_assen, directory = "my_directory")

# Clean up (don't do this if you want to keep your files)
unlink("my_directory", recursive = TRUE)
```

`write_eml` *Transform Movebank metadata to EML*

Description

Transforms the metadata of a published Movebank dataset (with a DOI) to an Ecological Metadata Language (EML) file.

Usage

```
write_eml(
  doi,
  directory,
  contact = NULL,
  study_id = NULL,
  derived_paragraph = TRUE
)
```

Arguments

`doi`	DOI of the original dataset, used to get metadata.
`directory`	Path to local directory to write files to.
`contact`	Person to be set as resource contact and metadata provider. To be provided as a `person()`.
`study_id`	Identifier of the Movebank study from which the dataset was derived (e.g. `1605797471` for this study).
`derived_paragraph`	If `TRUE`, a paragraph will be added to the abstract, indicating that data have been transformed using `write_dwc()`.

Details

The resulting EML file can be uploaded to an IPT for publication to GBIF and/or OBIS. A corresponding Darwin Core Archive can be created with `write_dwc()`. See `vignette("movepub")` for an example.

Value

`eml.xml` file written to disk. And invisibly, an EML::eml object.

Transformation details

Metadata are derived from the original dataset by looking up its `doi` in DataCite (example) and transforming these to EML. The following properties are set:

- **title**: Original dataset title.
- **description**: Original dataset description. If `derived_paragraph = TRUE` a generated paragraph is added, e.g.:

  Data have been standardized to Darwin Core using the movepub R package and are downsampled to the first GPS position per hour. The original data are available in Dijkstra et al. (2023, `https://doi.org/10.5281/zenodo.10053903`), a deposit of Movebank study 1605797471.
- **license**: License of the original dataset.
- **creators**: Creators of the original dataset.
- **contact**: `contact` or first creator of the original dataset.
- **metadata provider**: `contact` or first creator of the original dataset.

- **keywords**: Keywords of the original dataset.
- **alternative identifier**: DOI of the original dataset. As a result, no new DOI will be created when publishing to GBIF.
- **external link** and **alternative identifier**: URL created from `study_id` or the first `derived from` related identifier in the original dataset.

The following properties are not set:

- **type**
- **subtype**
- **update frequency**
- **publishing organization**
- **geographic coverage**
- **taxonomic coverage**
- **temporal coverage**
- **associated parties**
- **project data**
- **sampling methods**
- **citations**
- **collection data**: not applicable.

See Also

Other dwc functions: `write_dwc()`

Examples

```
(write_eml(doi = "10.5281/zenodo.10053903", directory = "my_directory"))

# Clean up (don't do this if you want to keep your files)
unlink("my_directory", recursive = TRUE)
```

Index