NPSdataverse: a suite of R packages for data processing, authoring Ecological Metadata Language metadata, checking data-metadata congruence, and accessing data

29 January 2025

Summary

The NPSdataverse is a suite of R packages developed to create, document, publish, and access data and metadata in open and machine-readable formats.NPSdataverse is modeled off of the tidyverse concept of several packages built with a common goal (Wickham et al. 2019).The NPSdataverse supports Ecological Metadata Language (EML) metadata and .csv data files. Some of the constituent R packages (EML and EMLassemblyline) are general-use and aimed at authoring EML documents. Other R packages (QCkit, EMLeditor, DPchecker and NPSutils) are designed and maintained by the National Park Service (NPS).Although many functions within the NPSdataverse packages are NPS-specific (particularly some API calls), whenever possible the functions are written so that they can also be used by the general public. Scientists conducting permitted research in NPS units can utilize the NPSdataverse to efficiently and consistently meet the data delivery requirements of their permits. Additionally, the packages will be useful for data management plans in a wide variety of grant proposals and for anyone that needs to create open data and machine-readable metadata. The ability to swiftly and easily author, edit, and check Ecological Metadata Language (EML) metadata in a reproducible fashion will be useful for data publication at any number of repositories or data journals. Finally, a scripted interface for downloading NPS data and leveraging metadata while loading it into R or other platforms for subsequent analyses and visualizations will be useful to researchers in the government, academia, and industry as well as the public.

Statement of Need

Following a movement for transparency in scientific research and data accessibility, the U.S. implemented the federal OPEN Government Data Act ("H. R. 4174" 2018). The Open Data Act mandates that federal agencies provide data in open formats with metadata. Subsequently, many funding agencies such as the National Science Foundation have required grant awardees make their data public, often including metadata (*The National Science Foundation Open Government Plan 3.5* 2015). Multiple publishers have followed suit (Wiley 2022; Springer 2023) and require data availability statements upon publication.

One goal of open science, and requirement of the recent "Nelson Memo" from the U.S. Office of Science and Technology Policy (Nelson et al. 2022) is to make data FAIR: findable, inter-operable, accessible, and reuseable (Wilkinson et al. 2016). These goals are often achieved by including structured, machine-readable metadata that conforms to a defined schema along with the data. Ecological Metadata Language Metadata (EML) is one metadata standard that is particularly amenable to studies with rich taxonomy (Jones et al. 2006, 2019). It has been adopted by multiple research organizations including the Ecological Data Initiative (EDI), National Ecological Observatory Network (NEON), Global Biodiversity Information Facility (GBIF), Swedish Biodiversity Data Infrastructure (SBDI), French Biodiversity Hub ("Pole National de Donnees de Biodiversite"), U.S. National Park Service, and others.

Nevertheless, actual availability of data and metadata varies (Federer 2018; Tedersoo et al. 2021), perhaps because there is a need for more infrastructure and tools to meet the goals of open data and open science (Huston, Edge, and Bernier 2019). Multiple solutions have been presented, including ezEML, a tool for authoring metadata in Ecological Metadata Language and publishing data and metadata to a repository (Vanderbilt et al. 2022). ezEML has an intuitive graphical user interface with a relatively low learning curve; however, it does have some drawbacks. For instance, ezEML is not scriptable, which makes repeated deployments of the same or similar workflows challenging and can limit reproducibility. ezEML also requires that the user upload their data to an external site for processing, which may not be suitable for sensitive data. Here we introduce the NPSdataverse, a series of R packages for authoring, editing, and checking EML metadata locally in a robust, repeatable, and scriptable fashion. R Packages within the NPSdataverse leverage earlier work using R to create and manipulate XML based EML files (Boettiger 2019a). Building upon that framework, we add user-friendly EML creation workflows; integration with taxonomic databases; fast, easy editing of existing metadata; congruence checks to test correspondence between data and metadata; and integration with public repositories such as the National Park Service's DataStore. R packages within the NPSdataverse also include functions that expedite data quality control, facilitate data interoperability, provide the ability to download data directly from DataStore, and leverage the rich EML associated with the data regardless of repository of origin.

NPSdataverse R package

The NPSdataverse package is a meta-package that loads packages within the NPSdataverse into R (Baker, Patterson, and DeVivo 2025). The NPSdataverse provides a convenient way to download, install, and load many of the R packages needed to create and access data packages, which consist of rich Ecological Metadata Language metadata and .csv data files:

```
pak::pkg_install("nationalparkservice/NPSdataverse")
library(NPSdataverse)
```

`NPSdataverse` will automatically check that the latest version of each R package is being loaded: either from the main development branch on GitHub.com or the latest version on CRAN. If updates are indicated, the user will be alerted and given instructions on how to update the relevant packages. To prevent API limits at GitHub (and to facilitate scripted workflows such as those at High Performance Computing facilities), `NPSdataverse` only checks for updates from an interactive R session and will skip checks when the system is not on-line or GitHub.com is not responding.

QCkit R package

QCkit ("Quality Control kit") is primarily a data processing package designed to prepare data for metadata creation and publication (Baker et al. 2025). This package serves two main functions: 1) Providing a suite of data quality control functions to be used across datasets regardless of the project, and 2) a suite of functions to apply data standards that promotes interoperability among datasets. For instance, `QCkit` includes functions that can help manage date-time formatting, can check data files for threatened or endangered species, and can help increase inter-operability by suggesting appropriate Darwin Core standards for naming data. `QCkit` also facilitates documenting data processing with functions that can generate a DataStore reference based on GitHub.com releases. The DataStore reference can hold processing scripts, code, or packages and have Digital Object Identifiers (DOIs) attached to them that are registered with DataCite once the DataStore reference is activated. `QCkit` is designed as an expandable framework that can adapt to new quality control tests or as new data standards are adopted.

EML R package

The EML ("Ecological Metadata Language") package is a fundamental package that allows for importing .xml files, creating and validating validating EML within R, and writing R objects back out to .xml files (Boettiger and Jones 2024). `EML` allows for creating fully fledged Ecological Metadata Language Metadata files using nested S3 lists within R while relying on the R/emld package (Boettiger 2019b).

EMLassembyline R package

The EMLassemblyline package builds upon `EML` and adds substantial functionality (Smith 2022). For instance, `EMLassemblyline` allows the user to supply .csv files, which are used to generate template .txt files. Users can adjust the template files as needed and use the `EMLassemblyline::make_eml()` function to generate an R-object that can be exported via `EML` as an EML-fomatted .xml file. `EMLassemblyline` includes the ability to generate entire taxonomic backbones from lists of scientific names via API calls to ITIS, GBIF, or Worms. `EMLassemblyline` will validate the R object against the EML schema and provide helpful hints on what might have gone wrong during the `EMLassemblyline::make_eml()` process. `EMLassemblyline` provides an efficient bridge between .csv data and EML metadata for users who are familiar with R but may not be experts on the EML schema or the detailed nested lists needed to create EML within R via the `EML` package. Products from the `EMLassemblyline` pipeline are suitable for publication at multiple repositories including the Environmental Data Initiative.

EMLeditor R package

The EMLeditor package allows users to quickly and easily view components of metadata in R and make on-the-fly edits to metadata (Baker and Patterson 2025). Edits made to EML objects using `EMLeditor` do not require re-running the `EMLassemblyline` functions to make EML. This is a significant improvement because running `EMLassemblyline` functions can be time consuming, especially if there are many taxa that need to be resolved. `EMLeditor` includes the ability to pick specific licenses (CC0, CC-BY, etc), add ORCIDs, include organizations as authors, and much more. `EMLeditor` also adds specific content necessary to be compliant with NPS's DataStore. With the proper permissions, `EMLeditor` can be used to generate draft references and reserve DOIs on DataStore as well as upload data and metadata files to DataStore. Finally, `EMLeditor` contains a .rmd template file that, after loading the package, is accessible in Rstudio under `Files > New File > R markdown`. The template provides an editable script that walks the user through using `EMLassemblyline`, `EMLeditor`, and `DPchecker` to create and validate EML metadata in R.

`EMLeditor` "set" class functions (which includes all functions that begin with "set_" such as "`EMLeditor::set_abstract()`") will add several NPS-specific items to the metadata using their default settings. For instance, these functions will set NPS as the publisher, Fort Collins as the publication location, and will add a "for or by NPS = TRUE" statement to the metadata. To invoke these functions without adding the NPS-specific metadata elements, set the parameter `NPS = FALSE` when calling each "set_" class function. Non-NPS publisher information can be added using the `EMLeditor::set_publisher()` function with the parameters `for_or_by_NPS` and `NPS` set to `FALSE`:

```
#set the abstract without NPS-specific information:

new_metadata1 <- set_abstract(eml_object = old_metadata,
                              abstract = "This is example abstract text",
                              NPS = FALSE)

#add custom publisher information:
```

```
new_metadata2 <- set_publisher(eml_object = new_metadata1,
                               org_name = "My Institution",
                               street_address = "1234 Sesame St.",
                               city = "Anytown",
                               State = "Delaware",
                               zip_code = "12345",
                               country = "USA",
                               URL = "https://www.myinstitution.us",
                               email = "publisher@myinstitution.us",
                               ror_id = "",
                               for_or_by_NPS = FALSE,
                               NPS = FALSE)
```

By default, `EMLeditor` functions provide verbose user feedback and may require user input to confirm some operations. These checks are intended to help guide users, reduce mistakes, and limit unnecessary API calls. However, requiring user input can hamper highly scripted approaches and limits reproducibility. Therefore, all `EMLeditor` functions can be set to circumvent these requirements using the parameter `force = TRUE`.

```
#example setting the abstract while suppressing user feedback and input:

new_metadata <- set_abstract(eml_object = old_metadata,
                             abstract = "This is example abstract text",
                             force = TRUE)
```

DPchecker R Package

The DPchecker ("Data Package checker") package provides feedback on data-metadata congruence (Baker and Wright 2025). Here, a "data package" consists of the EML metadata file with a filename that ends in *_metadata.xml and one or more data files in .csv format, all of which are in a single directory (and the directory contains no extraneous .csv or .xml files). `DPchecker` is useful for both data package authors and reviewers. `DPchecker` goes beyond validating EML objects in R against the EML schema. Using the `DPchecker::run_congruence_checks` function, `DPchecker` will conduct a series of 46 tests. These are divided into several categories to check whether:

1. Metadata are well formatted (file names are not duplicated, files specify the field delimiter, data files have URLs, the proper delimiter and header row numbers are present, etc.).
2. Metadata elements necessary for DataStore automated extraction are present (creators have valid surnames, publication date is present and in the correct ISO-8601 format, keywords are present, abstract and methods are present and well formatted, etc).
3. Recommended EML elements are present including ORCiDs and a notes section.
4. Metadata and data are in congruence including all files listed in metadata refer to data files, the columns in the metadata match the columns in the data files, missing fields in data files are properly documented in metadata, and dates in data files fall within the date ranges given in the metadata, etc.
5. Data and metadata are in compliance with (a subset of) federal regulations including tests for information that should not be released to the public such as non-.gov emails.

For each test, the data package may fail with an error, fail with a warning, or pass. When possible, warnings and error messages indicate the appropriate `EMLeditor` function to address the problem. `DPchecker` will often throw a warning even if an EML element exists and is properly formatted but could by improved to increase the FAIR characteristics of the metadata. For instance, `DPchecker` will throw a warning if an abstract is less than 20 words long as it is unlikely the creator is able to meaningfully describe the data collection and processing in less than 20 words.

NPSutils R Package

The NPSutils ("NPS utilities") package serves primarily as a way to access data (Baker, DeVivo, and Patterson 2025). `NPSutils` provides avenues for directly downloading data from DataStore using R. `NPSutils` can also import data downloaded from any repository (provided it is properly formatted as a data package) into R and take advantage of rich EML metadata to call column types. `NPSutils` provides some basic meta-analysis capability. `NPSutils` can also be used to import data and metadata into common data visualization tools.

Example of how to download and access an example data package titled, "Mojave Desert Network Springs Data Package 2016 - 2023" (Bailard and Lehman 2024):

```
# download a data package from datastore:
# the data package will be downloaded to ./data/2300498

NPSutils::get_data_package(2300498)

# load the data package into R, and use the metadata to call column types
# returns a list of tibbles; each tibble corresponds to a single data file

mojn <- NPSutils::load_data_package(2300498, assign_attributes = TRUE)
```

Acknowledgements

We acknowledge contributions from across the National Park Service, but in particular from the Inventory and Monitoring Division. Members of the NPS Long Term Data Management Governance Board provided critical guidance and insight (in addition to several of the authors, these include Kristen Bonebrake, Adam Kozlowski, Ryan Monello, Mark Isley, and Megan Swan). Justin Mills (currently at U.S. Fish and Wildlife Service) and Derrick Dardano helped with navigating API and Active Directory interfaces, Marsha Leavitt made and explained numerous updates to DataStore. Dan Gussett, Kate Miller, and Pete Budde facilitated software availability, and Meg White supported and endorsed the project. We are particularly indebted to our strong user base and their very helpful feedback including Alison Loar, Christina Appleby, Kirk Sherrill, Lisa Nelson and Tom Phillipi. Numerous Student Conservation Association interns made contributions to the code base including Sarah Kelso, James Brown, and Amy Sherman. Alissa Graff (currently at the Internal Revenue Service) provided important input on early versions of NPSutils.

References

Bailard, Jennifer, and Mark Lehman. 2024. "Mojave Desert Network Desert Springs Data Package 2016 - 2023." National Park Service.

Baker, Robert, Joe DeVivo, and Judd Patterson. 2025. *NPSutils: Collection of Functions to Read and Manipulate Information from the NPS DataStore*. https://github.com/nationalparkservice/NPSutils.

Baker, Robert, and Judd Patterson. 2025. *EMLeditor: View and Edit EML Metadata*. https://github.com/nationalparkservice/EMLeditor.

Baker, Robert, Judd Patterson, and Joe DeVivo. 2025. *NPSdataverse: Tools and Packages for Data and Metadata Manipulation*. https://github.com/nationalparkservice/NPSdataverse.

Baker, Robert, Judd Patterson, Joe DeVivo, Issac Quevedo, and Sarah Wright. 2025. *QCkit: NPS Inventory and Monitoring Quality Control Toolkit*. https://github.com/nationalparkservice/QCkit/.

Baker, Robert, and Sarah E. Wright. 2025. *DPchecker: Checks Data Packages for Congruence*. https://nationalparkservice.github.io/DPchecker/.

Boettiger, Carl. 2019a. "Ecological Metadata as Linked Data." *Journal of Open Source Software* 4 (34): 1276.

———. 2019b. "Ecological Metadata as Linked Data. Journal of Open Source Software." *The Journal of Open Source Software* 4 (34): 1276. https://doi.org/10.21105/joss.01276.

Boettiger, Carl, and Matthew B. Jones. 2024. *EML: Read and Write Ecological Metadata Language Files*. https://docs.ropensci.org/EML/.

Federer, Christopher W. AND Joubert, Lisa M. AND Belter. 2018. "Data Sharing in PLOS ONE: An Analysis of Data Availability Statements." *PLOS ONE* 13 (5): 1–12. https://doi.org/10.1371/journal.pone.0194768.

"H. R. 4174." 2018. *Law*. H.R.4174 - 115th Congress. https://www.congress.gov/bill/115th-congress/house-bill/4174.

Huston, P, VL Edge, and E Bernier. 2019. "Open Science/Open Data: Reaping the Benefits of Open Data in Public Health." *Canada Communicable Disease Report* 45 (11): 252.

Jones, Matthew, Margaret O'Brien, Bryce Mecum, Carl Boettiger, Mark Schildhauer, Mitchell Maier, Timothy Whiteaker, Stevan Earl, and Steven Chong. 2019. "Ecological Metadata Language Version 2.2.0." https://doi.org/10.5063/f11834t2.

Jones, Matthew, Mark P. Schildhauer, O. J. Reichman, and Shawn Bowers. 2006. "The New Bioinformatics: Integrating Ecological Data from the Gene to the Biosphere." Journal Article. *Annual Review of Ecology, Evolution, and Systematics* 37 (Volume 37, 2006): 519–44. https://doi.org/https://doi.org/10.1146/annurev.ecolsys.37.091305.110031.

Nelson, Alondra et al. 2022. "Memorandum for the Heads of Executive Departments and Agencies: Ensuring Free, Immediate, and Equitable Access to Federally Funded Research."

Smith, Colin. 2022. *EMLassemblyline: A Tool Kit for Building EML Metadata Workflows*.

Springer. 2023. "Data Availability Statement." https://www.springer.com/gp/editorial-policies/data-availability-statement?srsltid=AfmBOoq9OGxFR-H9UXUfYx_Nl1fRgfnBfCIFl3nbUqkNcRey1oaTBNqn.

Tedersoo, Leho, Rainer Küngas, Ester Oras, Kajar Köster, Helen Eenmaa, Äli Leijen, Margus Pedaste, et al. 2021. "Data Sharing Practices and Data Availability Upon Request Differ Across Scientific Disciplines." *Scientific Data* 8 (1): 192.

The National Science Foundation Open Government Plan 3.5. 2015. Alexandria, VA, USA: National Science Foundation; NSF Document Number nsf15093. https://www.nsf.gov/publications/pub_summ.jsp?ods_key=nsf15094.

Vanderbilt, Kristin, Jon Ide, Corinna Gries, Susanne Grossman-Clarke, Paul Hanson, Margaret O'Brien, Mark Servilla, Colin Smith, Robert Waide, and Kyle Zollo-Venecek. 2022. "Publishing Ecological Data in a Repository: An Easy Workflow for Everyone." *The Bulletin of the Ecological Society of America* 103 (4): e2018.

Wickham, Hadley, Mara Averick, Jennifer Bryan, Winston Chang, Lucy D'Agostino McGowan, Romain François, Garrett Grolemund, et al. 2019. "Welcome to the Tidyverse." *Journal of Open Source Software* 4 (43): 1686. https://doi.org/10.21105/joss.01686.

Wiley. 2022. "Wiley's Data Sharing Policies." https://authorservices.wiley.com/author-resources/Journal-Authors/open-access/data-sharing-citation/data-sharing-policy.html.

Wilkinson, Mark D., Michel Dumontier, Ijsbrand Jan Allbersberg, Gabrielle Appleton, Myles Axton, Arie Baak, Niklas Blomberg, et al. 2016. "The FAIR Guiding Principles for Scientific Data Management and Stewardship." *Scientific Data* 3 (1): 160018. https://doi.org/10.1038/sdata.2016.18.