Data Package Design for Special Cases

Members of the working group developing these documents: S. Beaulieu, R. Brown, J. Downing, S. Elmendorf, H. Garritt, G. Gastil-Buhl, C. Gries, J. Hollingsworth, H.-Y. Hsieh, L. Kui, M. Martin, G. Maurer, A. Nguyen, J. Porter, A. Sapp, M. Servilla, T. Whiteaker

In these documents we consider special cases for archiving research data based on their data type, format, or acquisition method, and recommend practices that ensure optimal re-usability of the data. Most recommendations here are aimed at improving documentation of data acquisition and processing to avoid misinterpretation. This includes the recommendation to publish raw data and/or processing code along with the data products. Others are aimed at usability in terms of data size/volume or connecting related data. Some recommendations involve including a metadata document formatted according to a new and emerging standard (e.g., codeMeta) or a data inventory table. Data inventory tables can cross the line between metadata and data and are intended to improve discoverability and navigation of archived data.

The intended audience for these best practice recommendations is the ecological research information manager (IM) community, and they are applicable to anyone operating in the context of an ecological research program. We assume that the target data repository is designed to handle ecological data, and that a given archive package will include metadata encoded in a community standard. This document references elements of the EML metadata standard, but many aspects would similarly apply to other metadata standards and these documents should be considered in the larger context of applicable metadata standard best practices. We refer to the Environmental Data Initiative (EDI) as an example data repository, though the same practices could be applied to other similar repositories.

Throughout the chapters we use the term *data package* to refer to a published unit of data and metadata together, which is the convention at the EDI repository. At other data repositories, equivalent terms for a data package, such as *dataset*, may be used. A data package may contain one or more *entities*, such as csv tables, spatial data, processing or modeling code, and other documents (pdf, jpg, zip). A basic discussion of data package design can be found as EDI's first phase of data publishing documentation and in the LTER Best Practices for Dataset Metadata in Ecological Metadata Language (EML).

Generally, we recommend archiving entities using standard file formats that are likely to be machine readable in the future. Exceptions to this may exist where the community standard for processing particular data types relies on specialized file formats (binary, closed specification, etc.) or proprietary software. In these cases, it may be appropriate to archive specialized file types and/or a copy that has been parsed into a format (e.g. ascii) that does not require proprietary software.

Main Topics:

1 Code

Contributors: An T. Nguyen, Tim Whiteaker

1.1 Introduction

This document describes best practices for archiving software, code, or scripts, such as a simulation model, data visualization package, or data manipulation scripts. The intention of these recommendations is to make research based on modeling or software more transparent rather than achieve exact reproducibility, i.e., provide sufficient documentation so that a knowledgeable person can understand algorithms, programming decisions, and their ramifications for the results, rather than run the model and obtain the same results.

Examples of candidate archives for code include CoMSES Net, which focuses on sharing models related to social and ecological sciences, and Zenodo, a popular DOI-minting all-purpose repository, that can conveniently archive a specific version of code in a GitHub repository. Alternatively, code may be archived in the EDI repository, either by itself or as part of a data package. The best practices in this document cover both archiving code in EDI and referencing code archived elsewhere.

While metadata for software may be described in detail using the EML <**software**> tree, there exists a project called CodeMeta which is specifically designed for software metadata. Therefore, one of the key recommendations in this document is to include a CodeMeta file when archiving software or code in EDI.

1.2 Recommendations for data packages

1.2.1 Considerations for archiving software or code

- If it is a model and/or a model-based dataset, please see the best practices for archiving model-based datasets (Chapter 2).

- How likely is it that the code will be well maintained into the future? For example, code packages submitted to established code repositories may stay there only while they comply with all testing requirements and may be removed if not well maintained (e.g., the R package repository CRAN). If that commitment to code maintenance is unlikely, such a package should be archived in a repository without maintenance requirements.
- Should the code be archived as a separate package or with the data?
    - If the code is used to generate several independent datasets it should be archived as a separate package.
    - The software authors wishing to place it under a different license from that of the associated data, or to obtain a DOI for only the code, may be reasons to separate code and data packages.
    - If deciding to package code separately, it may be archived on EDI or another repository. If archiving code outside of EDI, see section 2.2.4 for instructions on how to reference that code from related data packages in EDI.
    - In most other cases, it is recommended to archive code and data together for context.
- Large community software packages are usually maintained and available elsewhere. However, they may undergo significant updates and it may make sense to archive the code of a certain version with the data for transparency reasons. Consider whether prior versions of a software package are available wherever that software is distributed.
- When choosing a repository for the code, consider the ease of the archiving process and how well the code can be described. For example, Zenodo offers an easy pathway to archive code that is currently in GitHub, though metadata requirements are very light. Following the best practices described herein, you would create a CodeMeta file if you were going to archive with EDI. This is more rigorous than Zenodo, but then your code is better described, and in a machine-readable way.

1.2.2 Documenting software/code

When describing the code with EML, include the code as an otherEntity in a data package. Although a well-documented human readable text format of the code is preferred, in case of multiple scripts, and/or where directory structure is important, a zip archive may be used. For the formatName and entityType elements in EML, we recommend using format names from the DataONE format list when possible. Some format names are included in examples below. Always check the list for the most up-to-date version of these names.

Example 1.1: EML otherEntity snippet for a script file.

```
<otherEntity>
   <entityName>R script to process CTD data</entityName>
```

```
   <entityDescription>Annotated RMarkdown script to process, calibrate, and flag
raw CTD data.</entityDescription>
   <physical>
      <objectName>BLE_LTER_CTD_QAQC.Rmd</objectName>
      <size unit="byte">9674</size>
      <authentication
method="MD5">8547b7a63fcf6c1f0913a5bd7549d9d1</authentication>
      <dataFormat>
         <externallyDefinedFormat>
            <formatName>R Markdown file</formatName>
         </externallyDefinedFormat>
      </dataFormat>
   </physical>
   <entityType>script</entityType>
</otherEntity>
```

1.2.2.1 Software License

It is important to include a use license to make it clear how others can use your work. We recommend the Creative Commons "no copyright reserved" (CC0) license, which places the software in the public domain and makes it easiest for end users to adapt and use your work. If a more restrictive license is required, we recommend the Apache License, Version 2.0 license, a permissive license that allows others to reuse, modify, and redistribute your software.

If a mix of data and code needs to be archived, and they each fall under different licenses, then separating them into different packages is advisable to eliminate ambiguity on which license applies to which portion of a data package. When a license other than a public domain dedication is used, then in addition to specifying the license in the metadata (see the "intellectualRights" element in EML), consider including a copy of the license at the beginning of the code files themselves so that the license is readily apparent to end users who peruse the code.

1.2.2.2 CodeMeta

Include a CodeMeta JSON file for all code that is archived in EDI. The CodeMeta file should be named "codemeta.json" and listed as an EML otherEntity. The formatName should be "JavaScript Object Notation (JSON) file", the entityType should be "metadata", and the entityDescription should indicate that this is a CodeMeta file for a given software or script in the data package.

For unnamed projects, e.g., one-off scripts for data processing, analysis, and/or visualisation, a CodeMeta file might appear to be overkill; however, CodeMeta files are simple to generate, and we recommend the below bare minimum. If there are multiple scripts each in their own otherEntity tag, we recommend aggregating information about them into one codemeta.json.

Example 1.2 Minimum recommended codemeta.json example for unnamed projects.{

```
  "@context": ["https://doi.org/10.5063/schema/codemeta-2.0",
     "http://schema.org"
  ],
  "@type": "SoftwareSourceCode",
  "description": "RMarkdown script to calibrate and flag raw CTD data.",
  "author": {
     "@type": "Person",
     "givenName": "Christina",
     "familyName": "Bonsell",
     "email": "cbonsell@utexas.edu",
     "@id": "https://orcid.org/0000-0002-8564-0618"
  },
  "keywords": ["calibration", "CTD", "RMarkdown"],
  "license": "https://unlicense.org/",
  "dateCreated": "2013-10-19",
  "programmingLanguage": {
     "@type": "ComputerLanguage",
     "name": "R",
     "version": "3.6.2",
     "url": "https://r-project.org"
  }
}
```

Example 1.3 otherEntity metadata for example 2's codemeta.json.

```
<otherEntity>
   <entityName>CodeMeta file for BLE_LTER_CTD_QAQC.Rmd</entityName>
   <entityDescription>CodeMeta file for annotated RMarkdown script to process,
calibrate, and flag raw CTD data.</entityDescription>
   <physical>
      <objectName>codemeta.json</objectName>
      <size unit="byte">702</size>
      <authentication
method="MD5">8547b7a63abc6c1f0913a5bd7549d9d1</authentication>
      <dataFormat>
         <externallyDefinedFormat>
            <formatName>application/json</formatName>
         </externallyDefinedFormat>
      </dataFormat>
   </physical>
   <entityType>CodeMeta</entityType>
</otherEntity>
```

For named projects, also include the software name, and the version if applicable. The example below shows some additional metadata you can include. See also the more complete CodeMeta example and the available CodeMeta terms.

Example 1.4 A more complete CodeMeta example for named projects. Example taken from the CodeMeta project Github with edits for brevity.

```
{
   "@context": ["https://doi.org/10.5063/schema/codemeta-2.0",
      "http://schema.org"
   ],
   "@type": "SoftwareSourceCode",
   "name": "codemetar: Generate 'CodeMeta' Metadata for R Packages",
   "description": "A JSON-LD format for software metadata",
   "author": [{
         "@type": "Person",
         "givenName": "Carl",
         "familyName": "Boettiger",
         "email": "cboettig@gmail.com",
         "@id": "https://orcid.org/0000-0002-1642-628X"
      },
      {
         "@type": "Person",
         "givenName": "Maëlle",
         "familyName": "Salmon",
         "@id": "https://orcid.org/0000-0002-2815-0399"
      }
   ],
   "codeRepository": "https://github.com/ropensci/codemetar",
   "dateCreated": "2013-10-19",
   "license": "https://spdx.org/licenses/GPL-3.0",
   "version": "0.1.8",
   "programmingLanguage": {
      "@type": "ComputerLanguage",
      "name": "R",
      "version": "3.5.3",
      "url": "https://r-project.org"
   },
   "softwareRequirements": [{
         "@type": "SoftwareApplication",
         "identifier": "R",
         "name": "R",
         "version": ">= 3.0.0"
      },
      {
         "@type": "SoftwareApplication",
         "identifier": "git2r",
         "name": "git2r",
         "provider": {
            "@id": "https://cran.r-project.org",
            "@type": "Organization",
            "name": "Comprehensive R Archive Network (CRAN)",
            "url": "https://cran.r-project.org"
         }
      }
   ],
   "keywords": ["metadata", "codemeta", "ropensci"]
}
```

1.2.2.3 Metadata to enable reproducibility

When archiving software, we strongly recommend including a user guide with installation and usage instructions if such would not already be apparent to the typical user. Take into account that the user might not have access to certain inputs that the software/scripts require. Include when feasible at least some example data and configure the script so that it is ready to run with the example data.

Aside from the software/code itself and its dependencies, other pieces of information may be important should a user wish to reproduce results, such as the operating system and version and the system locale. Include this information in the data package's methods/methodStep/description. For certain tools, there are ways to easily generate this information, e.g., a call to `sessionInfo()` in the R console. If the system outputs this information in a standardly formatted plain text file, that might be included as an otherEntity.

1.2.3 Linking code and data

There are a few solutions for providing explicit machine-readable linkages between different entities/packages (the distinction between code/data doesn't matter too much here). For most cases we recommend the simplest approach, which is to use the methods/methodStep/description element of EML. More advanced users may wish to utilize the other solutions described herein.

1.2.3.1 Descriptive approach

In the dataset methods/methodStep/description element, include verbal descriptions such as "results.csv was derived from raw_data.csv using script.R" and repeat for all entities. If code and data reside in different packages, be sure to specify that.

1.2.3.2 The EML dataSource element

Nested under methods/methodStep, dataSource elements describe other data packages that serve as source for the current package. dataSource looks like a mini-EML tree describing the source data. Example: ecocomDP packages list the original packages under dataSource. dataSource does not describe relationships between entities in the same package, and as far as we know there is no explicit way in EML to do so.

1.2.3.3 ProvONE

ProvONE is a model developed by DataONE affiliates for provenance or denoting relationships between data entities. Each package on DataONE is described by a science metadata document (e.g., EML, ISO, FGDC) and a resource map document following ProvONE. The resource map powers a nice display of data relationships (see this package on the Arctic Data Center). This handles both relationships between entities in the same package and entities residing in different packages. However, note that EDI currently does not utilize this model.

1.2.4 External software

Large community-backed tools or proprietary software such as ArcGIS Pro or Microsoft Excel do not need to be archived. However, if they have had any impact on the final data (e.g., ArcGIS Pro was used to modify spatial rasters), the EML methods section should describe the routines performed. Within the data package, indicate linkage to external software as follows.

- Briefly describe the software/code and its relationship to the data in EML's methods/methodStep/description element.
- Names of all software used. Include both the common acronym and the full spelling.
- The URL(s) to all models/software used. Stable, persistent URLs pointing to exact version(s) are preferable, rather than generic links such as a project homepage. If the archived model has a DOI, then include a full citation to the model in the methods/methodStep/description text. The exception to this is when referencing tools such as Excel that have achieved global household name status.
- Broadly, the system setup used, if relevant.
- Information on exact versions for all code used (including dependencies). This is important, e.g., ArcGIS Pro 2.4.1 is very different from ArcGIS for Desktop 10.7.1. Different systems have methods to easily generate this information, e.g. a call to `sessionInfo()` in the R console.
- Consider, if applicable, to archive the "runfile" as its own data entity within the data package, i.e., the script(s) that sets parameters and/or calls on functions imported from external software.

Example 1.5 EML method description referring to external software.

```
<methods>
   <methodStep>
      <description>
         <para>
            The seagrass coverage raster was created in ArcGIS Pro (version
            2.4.3, by Esri) using the IDW geoprocessing tool on
            sampling_points.csv with a power of 2 and the nearest
            12 points.
         </para>
         <para>
            The raster was then refined using the seagrass-refiner package
            with the auto-refine option checked (Smith, 2017).
         </para>
         <para>
            Smith, J. (2017). seagrass-refiner: a package that does the cool
            seagrass stuff, Version 1.2, Zenodo.  https://doi.org/this-is/a-fake-
doi,
            2017.
         </para>
      </description>
```

```
    </methodStep>
  </methods>
```

1.3 Resources

CodeMeta website

CodeMeta crosswalks for a number of popular software

CodeMeta terms you can use for describing software

CoMSES

A description of some software licenses

ProvONE documentation

W3C PROV-O documentation

Licensing software as part of an EDI data package

Zenodo

2 Model-Based Datasets

Contributors: An T. Nguyen, Tim Whiteaker, Corinna Gries

2.1 Introduction

This document includes recommendations for archiving data packages composed of model-based datasets. These datasets may include the model code itself, input data, model parameter settings, and output data.

The range of cases for model-based datasets includes small one-off model code specific to one research question, through various code packages which are maintained in community repositories as long as they meet requirements (e.g., CRAN for R packages), to large community models maintained by groups of programmers and users.

The intention of these recommendations is to make research based on modeling more transparent rather than achieve exact reproducibility, i.e., provide sufficient documentation so that a knowledgeable person can understand algorithms, programming decisions, and their ramifications for the results, rather than run the model and obtain the same results.

It is not always easy to determine who among project personnel (IMs, scientists, programmers) is responsible for the different components of a model-based dataset. This is best decided on a case-by-case basis. A common division is that the code authors annotate the code, and the IM handles the archiving and linkage to data product(s); partially except in cases of large community models.

2.2 Recommendations for data packages

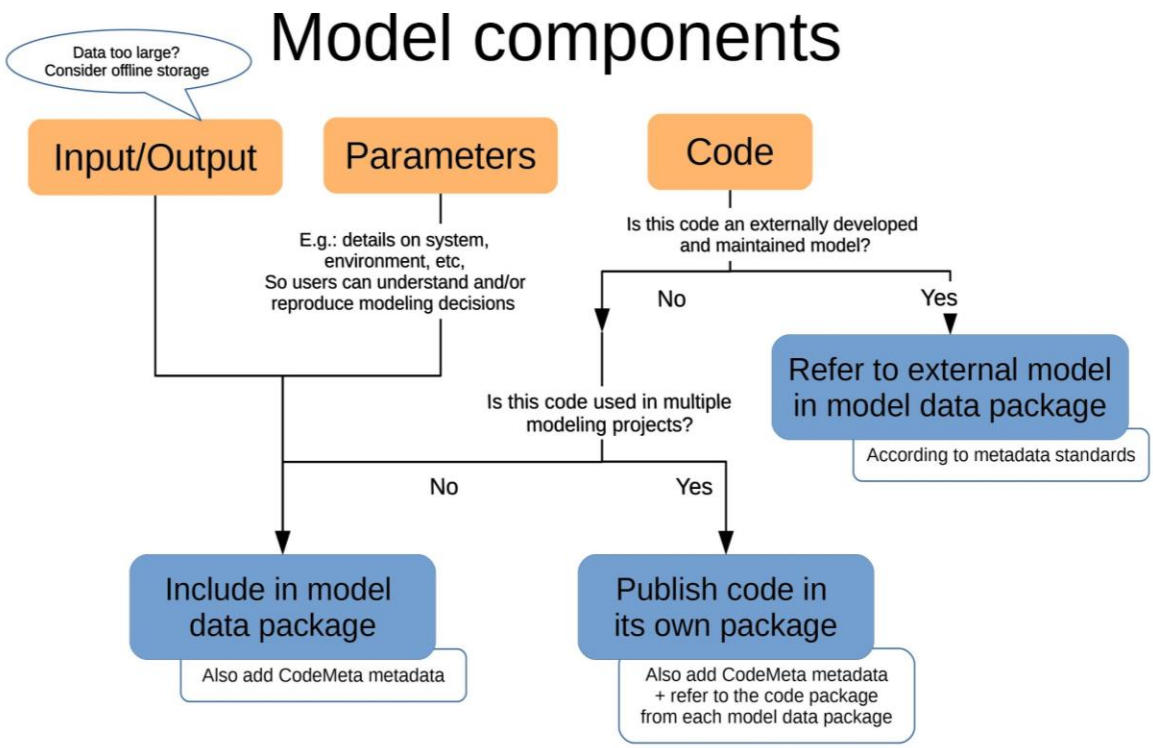

Figure 1 Flowchart for considering archival paths for various model components.referencing models in EML

For data packages related to a model, whether the model is archived within the same data package or not, indicate linkage to the model in EML following the best practices for archiving code (Chapter 1, see also the section on linking code and data below).

Example 2.1 EML snippet relating data to models via the method description

```
<methodStep>
   <description>
      <para>This methodStep contains data provenance information as specified in
the LTER EML Best Practices. Each dataSource element here lists entity-specific
information and links to source data used in the creation of this derivative data
package.</para>
   </description>
   <dataSource>
      <title>Source dataset title</title>
      <creator>
         <individualName>
            <givenName>first name</givenName>
            <surName>last name</surName>
         </individualName>
         <organizationName>organization name</organizationName>
         <electronicMailAddress>email@some.edu</electronicMailAddress>
      </creator>
```

```
      <distribution>
         <online>
            <onlineDescription>This is a link to an external online data resource
(describe resource and repository location).</onlineDescription>
            <url
function="information">https://pasta.lternet.edu/package/metadata/eml/knb-lter-
ntl/80/2</url>
         </online>
      </distribution>
      <contact>
         <positionName>Information Manager</positionName>
         <organizationName>organization name</organizationName>
         <electronicMailAddress>infomgr@some.edu</electronicMailAddress>
      </contact>
   </dataSource>
</methodStep>
```

2.2.1 Model code

The model used to produce certain data needs to be well documented and linked from the resulting data product(s). However, it is not always easy to decide where and how to archive the code, and whether or not in conjunction with the data product(s). We outline in sections below three common code archiving options.

Note that these scenarios (model code archived with data, or standalone in EDI, or elsewhere) are not mutually exclusive. Any project that involves code might make use of both established and custom software hosted on many different platforms and might use some or all archiving options.

To decide between archiving options, consider the questions listed in best practices for publishing code (Chapter 1).

2.2.1.1 Model code and data in the same package

The goal of this practice is to ensure transparency of the data, and it applies to one-off models developed for the associated data, or occasionally to larger code bases for the reasons outlined in best practices for archiving code (Chapter 1). Include the code as a dataset/otherEntity. Additionally, it is recommended to include a CodeMeta file, which can also be handled and documented in EML as dataset/otherEntity. CodeMeta is a metadata standard for software and code compatible with schema.org. Refer to best practices for archiving code for how to document the code and create CodeMeta.

2.2.1.2 Model code as standalone package

If the model has been used to generate several datasets, i.e., is more widely applicable, it can be archived as its own package in EDI and assigned a DOI. Include the code as a dataset/otherEntity. Additionally, it is recommended to include a CodeMeta JSON-LD file, which can also be handled and documented in EML as dataset/otherEntity. CodeMeta is a metadata

standard for software and code compatible with schema.org. Refer to best practices for archiving code for how to document the code and create CodeMeta (Chapter 1).

2.2.1.3 Model code archived/maintained elsewhere

This might include complex community models/software maintained by many people, published and actively maintained R/Python packages, etc., or simply code archived in another repository such as CoMSES.Net. It may sometimes be advisable to archive a copy of the model code with the data, even if it appears to be maintained elsewhere. See recommendations above for referencing models in EML.

2.2.2 Model input and output data

These are considered data entities, which should be handled according to EML best practices for corresponding data types. However, if the resulting datasets are very large, one may consider if input/output from all individual model runs need to be archived. Are there specific model run results that are more useful for non-modelers? For example: results from model runs leading to a journal publication.

Very large model inputs/outputs may need to be archived offline. Refer to best practices for offline data (Chapter 7).

If the model requires a specific folder structure, you can zip model input files within the package to preserve that folder structure. A disadvantage of this approach is that you cannot elegantly describe each file with EML.

The EarthCube Research Coordination Network, "What About Model Data?" group is working on a rubric to help you determine how much model output data to save, based on assorted criteria on reproducibility/value of the data. Learn more about that group and their rubric on their Model Data RCN website.

Researchers at the Department of Energy's Environmental Systems Science are also working on assessing model archiving needs. In this preprint, Simmonds et al. 2020 discuss feedback from communications with modellers and propose preliminary solutions. With regards to input/output data, their feedback indicates two opposite opinions: some feel the whole gamut of raw to aggregated outputs needs to be archived, while others advocate for only high-level outputs corresponding to publication figures. They also found that spin-up simulations were not considered a high priority for archiving. See section 2.3 What is worth archiving and for how long does it remain useful?

2.2.3 Model parameters

Include model parameters whenever applicable. If code/input/output from multiple model runs are archived, make sure to archive all corresponding sets of parameters, and be explicit in linking the different components together.

Consider archiving model parameter files as their own data object(s) in both their native format and as a text (non-binary) version. If the “runfile” will be archived, consider including the parameters within that file with appropriate annotations.

2.3 Example data packages in EDI

Dataset Title	Description	EDI Package ID
North Temperate Lakes LTER General Lake Model Parameter Set for Lake Mendota, Summer 2016 Calibration	Parameters for specific GLM runs. GLM is a large community model, not managed and archived in EDI	knb-lter-ntl.348.2
SBC LTER: Regional Oceanic Modeling System (ROMS) Setup Files, Code, and Lagrangian Model Setup Files	All the necessary code, grid, forcing, initial, and boundary condition files for running the UCLA version of the Regional Oceanic Modeling System (ROMS) for the Santa Barbara Channel	knb-lter-sbc.126.1
Lake thermal structure drives inter-annual variability in summer anoxia dynamics in a eutrophic lake over 37 years	Dataset to run a 37-year simulation (1979-2015) of the Lake Mendota lake ecosystem using the vertical 1D GLM-AED2 model.	knb-lter-ntl.396.1

2.4 Resources

Janssen, Marco A., Lilian Na'ia Alessa, Michael Barton, Sean Bergin and Allen Lee (2008). ‘Towards a Community Framework for Agent-Based Modelling’. Journal of Artificial Societies and Social Simulation 11(2)6 http://jasss.soc.surrey.ac.uk/11/2/6.html.

Simmonds, Maegen, William J. Riley, Shreyas Cholia, and Charuleka Varadharajan (2020). 'Addressing Model Data Archiving Needs for the Department of Energy’s Environmental Systems Science Community'. EarthArXiv (preprint). https://doi.org/10.31223/osf.io/acdk4.

See sections 2.3 What is worth archiving and for how long does it remain useful, discussed above, plus 2.4 Model data archiving protocol, where the authors argue for better standardized reporting format for model data, e.g., top-level metadata and directory structure at a minimum. Section 4.1 Developing Model Data Archiving Guidelines proposes an organization scheme for model data.

3 Images and Documents as Data

Contributors: Renée F. Brown (lead), Stace Beaulieu, Sarah Elmendorf, Gastil Gastil-Buhl, Corinna Gries, Li Kui, Mary Martin, Greg Maurer, John Porter, Tim Whiteaker

3.1 Introduction

This chapter describes best practices for archiving images and other documents as data. The Environment Ontology (ENVO) defines a document as "*a collection of information content entities intended to be understood together as a whole*." Common examples include still images, audio and/or video multimedia files, field notebooks, written interview notes or transcribed oral accounts, historical document collections, and "paper" maps (non-digitized maps). For images that are already handled by specialized repositories (e.g., phenocam images, specimen images) refer to Chapter 7, for additional information on how to handle images from uncrewed (underwater or aerial) vehicles refer to Chapter 6, and for geospatial imagery refer to Chapter 5.

3.2 Recommendations for data packages

3.2.1 Reasons to archive documents as data

- **Enhance the credibility of associated datasets.** Many document types (field notes, still images, etc.) often provide additional metadata that cannot easily be encapsulated in the associated dataset(s) or were not considered important at the time of transcription. As such, these documents may provide opportunities to rectify transcription errors, retrospectively provide explanations of unusual data, and/or include additional observational or measured data, such as opportunistic measurements or calibration parameters.
- **Provide opportunities for new analyses.** New analytical methods may be employed on archived documents (especially still images) or documents that were never archived previously because the cost-to-benefit ratio was considered too high (e.g., pilot projects).
- **Improve ease of access.** In distributed projects, access to original and/or "hard-copy" documents may be limited to a particular institution or subset of people. By digitally archiving these documents in a data repository, the data become more findable, accessible, interoperable, and reusable (FAIR).

3.2.2 Considerations for data package structure

- **Balance file size and number of files.** A data package may contain document files individually or bundled as a compressed archive (e.g., zip). The decision of how to bundle documents into compressed archives and then into data packages should be guided by the overall goal of making data usable for the intended purpose of the documents. In most cases, this would involve finding specific documents by, for example, the date or location of the acquisition, or some other aspect of interest. In

addition, the effort of documenting documents (each individually vs. in groups) has to be taken into account. Also see Chapter 7.

- **Document grouping.** Data packages, or compressed archives within data packages, may be grouped spatially (e.g., by location) and/or temporally (e.g., by date, season, or year). For example, data outputs from a stationary camera may be archived in annual data packages, each containing monthly compressed archives if the number of images is large. While moving camera outputs may also be archived annually, these data packages may instead include compressed archives containing all still images for a single location.
- **Document naming.** To maximize searchability, document names should be unique and meaningful for a data reuser. It is recommended that individual documents be named according to their content, and compressed archives include date, location, and other relevant information in the filename.
- **Data inventory table.** An inventory table providing the structure and organization of the included document entities or groups of documents (see Table 4.1) is recommended, especially for larger collections of documents within a data package. The inventory table serves as an additional source of metadata and may also be used to link specific documents to additional information.
- **Archival frequency.** One should strive for archiving a fully processed group of documents when no more updates are expected (e.g., after a field season or annually) due to the large volume of documents to be handled repeatedly for each update.
- **Linking to related data packages.** In the case where the documents are useful to understanding another data package and vice versa (e.g., met station visitation logs and met station time series data), it is recommended to link the complementary data package in the methods section of both datasets. Alternatively, include the document(s) or compressed archive(s) in the existing dataset as otherEntity, as described in the next section.

3.2.3 Documenting data packages

3.2.3.1 Ecological Metadata Language

All data packages require good discovery-level metadata in Ecological Metadata Language (EML), which should be assembled using standard documented best practices. Documents (including compressed archives) should be included as otherEntity in the data package (e.g., see Example 4.1). Refer to the most recent version of EML Best Practices (currently v3) for guidance regarding the formatName and entityType EML elements. If a format for your document type is not covered, it is recommended to use the appropriate MIME type, if available.

Example 3.1 EML otherEntity snippet for a pdf file

```
<otherEntity>
   <entityName>site date</entityName>
   <entityDescription>Field notes at site and date.</entityDescription>
   <physical>
      <objectName>site_date.pdf</objectName>
      <size unit="byte">9674</size>
      <authentication
method="MD5">8547b7a63fcf6c1f0913a5bd7549d9d1</authentication>
      <dataFormat>
         <externallyDefinedFormat>
            <formatName>Portable Document Format</formatName>
         </externallyDefinedFormat>
      </dataFormat>
   </physical>
   <entityType>application/pdf</entityType>
</otherEntity>
```

The EML metadata should also include appropriate keywords describing the general purpose of the document or compressed archive (e.g., ice phenology, community composition, stream hydrology, etc.). For example, for still images, it is recommended to include keyword: image with the semantic annotation from the Information Artifact Ontology (IAO) :

> **Term IRI:** http://purl.obolibrary.org/obo/IAO_0000101
>
> **Definition:** An image is an affine projection to a two-dimensional surface, of measurements of some quality of an entity or entities repeated at regular intervals across a spatial range, where the measurements are represented as color and luminosity on the projected surface.

Note, IAO includes at least one subcategory for image (e.g., photograph). It is recommended the most specific applicable concept be used.

3.2.3.2 Data Inventory Table

We recommend that an additional level of metadata be provided through a data inventory table that effectively serves as a document catalog (see Table 4.1). The detail provided in this table should be guided by the same principles as stated above -- to enable optimal usability of the documents. For example, still images from a stationary camera require latitude and longitude only in the EML file, not for each individual image. However, images from a moving camera may need that information for every image, or at least for every location (e.g., site, quadrat, transect). Additionally, Exif metadata from photographic images may be programmatically extracted to supplement the inventory table (refer to the *Tips and Tricks* section of Chapter 6).

The data inventory table should be structured such that each column represents a particular attribute, described in EML as a dataTable entity, and each row represents an individual document or a compressed archive of a group of documents. At minimum, the table should

include an attribute for the document/archive filename, as well as any other essential attributes that vary per each document/archive. Additional attributes may include information on the date and/or time, but for this information to be useful, be consistent and use a controlled vocabulary for these fields so that a user can effectively search on them.

Table 3.1 Data inventory table structure for documents and images

Column	Attribute Description
Filename	Filename of each document or compressed archive, including file extension (e.g., “site_date.jpg”). For compressed archives, include the relative path of the document, with respect to the uncompressed directory structure (e.g., “2018/SITE3/quadrat4.jpg”).
Link/URL/URI	Link to download a document if it is available on a different system (also see Chapter 4). Persistent identifiers are recommended, if available.
Creator(s)	Name(s) of the creator(s) of the original document (e.g., photographer, field technician, interviewer). Multiple creators should be entered into a single cell using the pipe delimiter.
Datetime	Date (and time) associated with the document, in ISO 8601 format (e.g., 2007-04-05T12:30-02:00).
Project specific datetime attributes	One or more appropriately labeled columns containing project specific date and time information for easier search and retrieval of documents (e.g., year, season, campaign).
Location	One or more location columns as appropriate, such as latitude and longitude in decimal degrees, site name, transect name, altitude, depth, habitat, etc.
Document specific attributes	One or more columns as appropriate to the document type, such as weather conditions, organism name, instrument type, etc.

3.3 Example data packages in EDI

Each of the Environmental Data Initiative (EDI) data packages listed below include images or other documents as data. Some of these packages contain data inventory tables (as dataTable entities) described in the EML metadata.

Dataset Title	Description	EDI Package ID
Annual ground-based photographs taken at 15 net primary production (NPP)	Compressed archives of images grouped by year. Includes data inventory file.	knb-lter-jrn.210011005.105

study sites at Jornada Basin LTER, 1996-ongoing		
McMurdo Dry Valleys LTER: Landscape Albedo in Taylor Valley, Antarctica from 2015 to 2019	Compressed archives of aerial images, grouped by flight date, and associated reflectance data.	knb-lter-mcm.2016.2
MCR LTER: Coral Reef: Computer Vision: Multi-annotator Comparison of Coral Photo Quadrat Analysis	5090 coral reef survey images, and 251,988 random-point annotations by coral ecology experts.	knb-lter-mcr.5013.3
Abundance and biovolume of taxonomically-resolved phytoplankton and microzooplankton imaged continuously underway with an Imaging FlowCytobot along the NES-LTER Transect in winter 2018	144,281 images from a plankton imaging system with annotations and extracted size data.	knb-lter-nes.9.1
Calling activity of Birds in the White Mountain National Forest: Audio Recordings (2016 and 2018)	Compressed archive containing 410 audio files in wav format. Includes data inventory table.	knb-lter-hbr.268.1

3.4 Resources

3.4.1 Considerations for digitizing documents

Following are some general considerations and recommendations for digitizing paper or other "hard-copy" documents for archival. This is not meant to be an exhaustive list. For further and more detailed information, please refer to the U.S. National Archives and Records Administration (NARA)'s Technical Guidelines for Digitizing Archival Materials for Electronic Access.

- **Effort.** The decision to digitize documents, as well as the digitization method, involves trade-offs in the accessibility and ease of using particular hardware and/or software technologies, the quality of the digitization, and the overall effort spent. Digitization efforts may be significant, for example, when dealing with a large number of documents requiring meaningful file names, text recognition, and/or high resolution for improved accessibility.
- **Equipment.** Instruments for digitizing hard-copy documents range from high resolution scanners (less accessible, less user-friendly, more expensive, better quality) to smartphone cameras (ubiquitous, easy-to-use, lower quality). For example, taking a

smartphone image in the field may be utilized for quick and easy digitization of field notes.

- **Document resolution and file size.** This is an important consideration that should be guided by the content and purpose of the document. Detailed paper maps should probably be scanned at high resolution and large file size, while field sheets may not need as much detail.
- **Optical Character Recognition (OCR):** When digitizing documents that include text, we recommend using scanning or other software with OCR capabilities (e.g., Adobe, ABBYY, Tesseract) to convert the text into machine readable characters so that the documents are searchable and thus, more usable. OCR does not work well for handwritten text, older fonts, or documents with busy backgrounds (speckled, dirty, faded, etc.).
- **Sensitive Information and Human Subjects:** Regardless of the digitization method, one should be mindful of sensitive information that shouldn't be archived or otherwise redacted (e.g., photographs of human subjects, field notebooks containing personal messages, gate combinations, and/or telephone numbers). In all cases in which human subjects are involved, Institutional Review Board (IRB) restrictions must be heeded. A signed IRB consent form for the associated research project represents a contract between researcher and human subject. It is important to note that IRB restrictions can differ among research studies within the same project. For further information, see the EDI Data Initiative Data Policy.

While transcription is a digitization method that can be performed on certain types of documents (e.g., audio/video recordings, field notebooks) and can enhance search capabilities, transcript generation requires substantially more effort than other digitization methods, and is prone to error. Moreover, in the case where the original documents contain drawings, transcripts may be incomplete or otherwise inaccurate. *Thus, we recommend digitizing documents by other means, using the considerations described above.*

4 Data in Other Repositories

Contributors: Greg Maurer (lead), Stace Beaulieu, Renée Brown, Sarah Elmendorf, Hap Garritt, Gastil Gastil-Buhl, Corinna Gries, Li Kui, An Nguyen, John Porter, Margaret O'Brien, Tim Whiteaker

4.1 Introduction

A wide variety of data repositories are available for publishing biological, environmental, and Earth observation data, and the choice of where to publish a particular dataset is determined by many competing factors. For example, a funding agency or journal may require a certain repository (e.g., NSF BCO-DMO, NSF ADC, USDA ADC, DOE ESS-DIVE); the research subject or data type may be best served by a specialized repository (e.g., AmeriFlux, GenBank);

or datasets may be submitted to a general-purpose repository with minimal metadata requirements to simplify and speed data publishing (e.g., DRYAD, Figshare, Zenodo). For these and other reasons related datasets are sometimes published in disparate data repositories, the same data needs to be discoverable in more than one repository, or multiple datasets from one or more repositories may be used to create a new, derived dataset. In such cases, it can be advantageous to establish links between datasets in different repositories such that provenance, supplementation, duplication or other relationships are explicit. Clearly, this subject goes well beyond the single repository and better standards and approaches for linking resources and documenting data provenance are being developed elsewhere (e.g. DataONE, ProvONE, WholeTale). Here we concentrate on specific cases in the context of large and multidisciplinary projects, such as LTER sites, that wish to enhance data discovery and preserve data relationships across multiple repositories.

4.2 Recommendations for data packages

4.2.1 Considerations for creating linked data

In practice, links to data in other repositories can be achieved using metadata only, by including a data inventory file, or, although not recommended, by duplicating the data in the new repository record. **Generally, duplicating data in multiple repositories is not recommended because it creates two problems.** First, it is a burden to maintain multiple copies of a dataset and avoid divergence between them. Second, it can create confusion for data re-users who may download or cite the same data multiple times. Care must be taken to clearly identify such duplications for data users when they are created. Whenever linked datasets are created, it is strongly recommended that both repositories are aligned with FAIR data principles, outlined here, so that users have unfettered access to all data and metadata.

In addition to these considerations, there are a number of reasons to create a new repository record that is linked to data in other repositories. Each of these reasons, which are outlined below, has pros and cons that will need to be weighed from the different perspectives of the data user, data provider and research project management requirements.

- **Requirements dictate multiple repositories:** Large research projects or sites are frequently funded by different agencies and programs. Data collection may be supported by several such funding streams and, hence, fall in the purview of more than one requirement to archive data in a particular repository. In some cases, data repositories already accommodate such requirements by linking or replicating data appropriately. Examples of this are LTER data in EDI, NSF BCO-DMO and NSF ADC.
- **Adding important metadata**: If data were originally submitted to a general-purpose repository with minimal metadata requirements (e.g., DRYAD, figshare) additional metadata (e.g., EML) may be needed for discoverability, reusability, and integration. By creating a new repository record that identifies and is linked to the original published dataset, richer and more useful metadata can be added to the new record and utilized.

- **Use of specialist repositories for related data:** There are sometimes advantages to publishing particular data types in specialized repositories. Specialized data repositories (e.g., GenBank, AmeriFlux) usually enforce strict data formatting, provide quality standards, enhanced search, discovery and reuse of particular types of data across projects in a way that is not possible using a generalized metadata format (EML) and repository (EDI). However, these data may not be discoverable with other, related project data taken at the same location and time. Creating links between related datasets held in specialist and generalist repositories helps preserve this context.
- **A derived data product is archived in a different repository than the source (raw) data:** A wide range of cases fall into this category, from a direct one to one relationship of, e.g., a gene sequence and its OTU identification, a metagenome analysis and its community diversity metrics, to several datasets being combined in synthesis or meta-analysis studies. In these cases, links between source data and derived data products that are published in separate repositories need to be created and clearly documented.
- **Linking to site- or project-relevant data from other research groups or agencies:** Although it may help with some aspects of data discovery it is generally not recommended to create records in EDI for data collected and managed by entirely different research groups or agencies. ***In these cases, however, it is recommended to place a pointer to such repositories on a project website or develop other means for data users to discover relevant resources.***

4.2.2 General metadata for linked data packages in EDI

In EDI, the linked data package can be assembled using standard practices and EML metadata elements, but the included metadata and data entities must clearly lead the data user to files held in outside repositories. In addition, the package metadata should communicate the essential elements needed for data discovery (subject matter, authors, location, time-frame, etc.) and a brief description of how the data may be accessed, re-used, and cited via the outside repository as needed. General guidance on the content and structure of key metadata elements in an EDI data package linked to data in other repositories are described below.

- **Abstract:** Describe the key features of the data package. If the data package contains only links to data held in other repositories, or data duplicated from another repository, clearly state that the original data are located in a different repository and direct the user to the correct data citation. Describe the target data in sufficient detail that users can determine whether these data are fit for their use, and instruct them on how to find and re-use the data.
- **Methods:** Collection/generation methods for any data entities included or linked to. If the methods are well-described in the metadata at another repository, this element can simply refer users there. If the new data package includes ancillary data or derived data, describe how those data were collected or derived.
- **Geographic description** and **coordinates:** At a minimum these elements should define a bounding box that will make the data package discoverable through EDI, DataOne, or

other geographic search interfaces. Additional, more detailed coordinates may be given in the inventory file entity as described below.

- **Keywords:** Since some linked data packages include an inventory of data held at a different repository, include the keyword "data inventory" and thematic keywords that describe the data entities in the other repository.

4.2.3 Common use cases and their structure in EDI

There are several common use cases for creating a new linked data package in EDI. The new package may establish either a one-to-one link from EDI to a dataset in another repository, or a one-to-many relationship that is more complex. Three possible cases are described below in terms of what entities to publish, where to publish them, metadata elements to be created in EML, and the contents of included data entities. There are likely to be other use cases for linking EDI data packages to other repositories.

Case 1: One dataset needs to be discoverable in more than one data repository. The data remain the same, but the metadata in the new data package at EDI may be upgraded beyond what exists in the other repository.

- The metadata in EDI must clearly state, preferably in the abstract or another obvious location, that this data package is already published in another repository. Include the original unique identifier and instruct users to cite that original data, if appropriate.
- Include instructions on how to access and cite the original data if the original repository is lacking in such guidance.
- If data are duplicated (which is not recommended), metadata should include information on how versions in different repositories are kept synchronized. If such synchronization is not feasible, users should be warned to inspect both sources for the latest data..
- In EML the <additionalIdentifier> field may be used to store the persistent identifier (DOI), or a link (URL) that refers to the data held in another repository to make the link machine readable. Where an external repository supplies both a URL and DOI, use the DOI as URLs may not be maintained through time.

Case 2: A list of data records held in a specialized repository needs to be linked to ancillary or supporting data that are being published in EDI (for derived data see Case 3).

- This case applies when a collection of datasets, or similar scientific resources, is held in a specialized repository and closely related ancillary or supporting data and metadata needs to be archived in a more generalist data repository like EDI. For example, ancillary environmental data or laboratory analyses held in EDI could be linked to collections of sequence reads held in NCBI GenBank or museum voucher specimens archived with Darwin Core metadata. See complete examples in Table 5.3.

- The new EDI data package should include a “data inventory” (or manifest of holdings) file as a data entity. This is most likely a simple tabular data file, such as a CSV, that lists and describes the repository records held in the specialized data repository and has its column attributes described in EML as a dataTable entity.
- The inventory table must have a row for each outside repository record (or some meaningful grouping of records, e.g., project in NCBI) being linked to, with columns that include persistent unique identifiers of the data in the other repository, and relevant descriptors of the data. The complete content of the inventory will be dictated by the structure of the other repository and the data entities and metadata held there. Suggested columns are presented in Table 5.1.
- The inventory table may also provide additional contextual information for each individual data resource in another repository. Table 5.2 presents examples of these contextual columns. They are, however, subject dependent and may vary for different projects. For more examples, see the discussion on sequencing and genomic data later in this document.

Table 4.1 Suggested columns for identifying the external data in the data inventory table.

Column	Description
External unique ID	Unique identifier for the data resource in the other repository. E.g. Accession number
External access URL	A unique, persistent link to the data resource in the other repository.
Title/description	Title and/or brief description of the data resource
Filename(s)	Dataset or file name at the other repository
Format	File format of above
Repository URL	URL of the repository being linked to

Table 4.2 Examples of additional contextual columns in the data inventory table for external data.

Column	Description
Latitude/Longitude	Latitude and longitude in standard format for each data resource in the other repository.
Location name	Locally used name of collection site
Treatment level	Experimental treatment applied to the outside dataset
Start/End datetime	Starting/ending datetime of the data resource (NA for End if data collection is ongoing)

Reference publication	DOI of publication providing in-depth context for data

Case 3: One or more datasets in other repositories are used to create derived data products that need to be archived in EDI.

- In this case the new dataset is directly or indirectly derived from the “source” dataset(s) in other repositories. Such derived data may serve a wide range of research purposes, including use in cross-site synthesis, re-analysis, or meta-analysis studies.
- Provenance metadata should be used to describe the relationship between the source and derived datasets, which ensures reproducibility and preserves data lineage. In a new EDI data package that archives derived data, the provenance metadata should be inserted in the EML file utilizing <dataSource> elements. The <dataSource> elements should be nested within a <methodStep> element and will establish the links to any source datasets located in another repository. An example snippet of provenance EML is shown in Figure 1.
- Other cross-repository standards for provenance metadata are still being developed and are not widely adopted, e.g., ProvONE.
- The EDI portal interface provides automatic generation of provenance metadata EML snippets for datasets in EDI. The EMLassemblyline and MetaEgress (in connection with LTER-core-metabase) R packages for EML creation will also generate provenance metadata.

Example 4.1 EML snippet with a data provenance methodStep

```
<methodStep>
   <description>
      <para>This methodStep contains data provenance information as specified in
the LTER EML Best Practices. Each dataSource element here lists entity-specific
information and links to source data used in the creation of this derivative data
package.</para>
   </description>
   <dataSource>
      <title>Source dataset title</title>
      <creator>
         <individualName>
            <givenName>first name</givenName>
            <surName>last name</surName>
         </individualName>
         <organizationName>organization name</organizationName>
         <electronicMailAddress>email@some.edu</electronicMailAddress>
      </creator>
      <distribution>
         <online>
            <onlineDescription>This is a link to an external online data resource
(describe resource and repository location).</onlineDescription>
```

```
            <url
function="information">https://pasta.lternet.edu/package/metadata/eml/knb-lter-
ntl/80/2</url>
         </online>
      </distribution>
      <contact>
         <positionName>Information Manager</positionName>
         <organizationName>organization name</organizationName>
         <electronicMailAddress>infomgr@some.edu</electronicMailAddress>
      </contact>
   </dataSource>
</methodStep>
```

4.3 Nucleotide sequence and genomic data

Nucleotide sequence data consists of the order and arrangement of DNA or RNA bases extracted from individual organisms or environmental samples. Similarly, genomic data refers to the complete genetic information (either DNA or RNA) of an organism, while metagenomic data refers to the study of genomes recovered from environmental samples. Sequencing, genomic and metagenomic datasets can be very large and complex, and researchers in these fields benefit from particular methods of data access, analysis, and collaboration. Therefore, these data have specialized requirements for data archiving.

Archiving nucleotide sequence and genomic (or other 'omics') data are a common use case for creating linked datasets. Data that originate from nucleotide sequencing techniques are most often stored in specialized repositories such as National Center for Biotechnology Information (NCBI) GenBank and the European Nucleotide Archive. However, while sequences or assembled genomes constitute important raw data, ancillary and derived data products related to these raw data are frequently published in repositories specializing in ecological data. For example, data derived from sequence data, such as operational taxonomic units (OTUs) or functional assignments, and ancillary data that describe the environmental, biochemical, or experimental context of the sequencing data, are often included in scientific publications, and do not always fit within the scope of a specialized sequence or genome data repository.

4.3.1 Recommendations for sequencing or genomic datasets

Linking to genomics data is an example of Case 2 described above. Summaries or inventories of data records held in a repository like NCBI GenBank are linked to their derived products or additional measurements published in a more generalist repository such as EDI.

In addition to the metadata typically included with any data package published by the site or research group, include metadata that is descriptive specifically of sequencing and genomics datasets. It is recommended to refer to the MixS templates for standard terminology, especially in the keyword section:

Keywords that can help users discover the sequencing or genomic dataset include:

1. General data type descriptions (‘nucleotide sequence’, ‘genomics’, ‘metagenomics’)
2. Names of target genes or subfragments (‘16S rRNA’, ‘18S rRNA’, ‘nif’, ‘amoA’, ‘rpo’, ‘ITS’)
3. Names of the sequencing technique (‘Sanger’, ‘pyrosequencing’, ‘ABI-solid’)
4. Names of the linked repository (‘SRA’, ‘EMBL’, ‘Ensembl’)
5. Descriptors of included ancillary data (‘nitrogen’, ‘soil’, ‘drought’)
6. Descriptors of derived data products (‘OTU’, ‘functional annotation’, ‘population’)

Inventory tables are of central importance to datasets that index data resources in a sequencing or genomics repository. It is recommended that this inventory should have the columns described in Table 5.1. Note that the unique identifiers included will depend on the granularity of the links to the outside repository. For example, in NCBI, there are accession numbers and URLs for a project, samples within the project, and sequence datasets from a given sample.

- **External unique ID and URL:** For NCBI GenBank this would be the accession number for a collection. For most sequence and genomic datasets an access URL would include an accession number (e.g., https://www.ncbi.nlm.nih.gov/nuccore/AY741555). Referring to a range of accession numbers, may involve providing a search URL that will return the desired list, e.g. (https://www.ncbi.nlm.nih.gov/popset/?term=AY741555). The recommendation is to link to the widest level of sequence or genomic granularity that is useful to interpret data being archived in the new dataset.

The following are suggestions for additional contextual columns in the inventory table. This information is generally associated with the data in the genomics repository and should only be duplicated if deemed useful for reuse, or if missing in the original data.

- **Sequencing method**: the name of the sequencing method used; e.g., Sanger, pyrosequencing, ABI-solid. This attribute is used in MIxS templates, where it is called seq_meth.
- **Environment (biome, feature, or material) descriptors**: These are descriptors of the environmental context and are standardized by the genomics community in the MixS templates and EnvO.
- **Taxon description**: If applicable, e.g., Binomial name, or taxonomic group

Data packages of metadata and inventory tables will aid in discovering genomic data within an ecological data repository (EDI) and will aid in clarifying the context in which they were collected. Most use cases, however, employ this inventory table to link specific genetic data to derived data. Such products frequently are community or population metrics where species, OTUs or traits have been determined from the sequence data.

4.4 Example data packages in EDI

Each of the EDI data packages below are linked to data in outside repositories. Some contain data inventory tables (as dataTable entities) that link to the datasets held in outside repositories and are described in the EML metadata. The EML abstract and methods elements in each give detailed access and citation instructions.

Title	Description	EDI packageID
Mass and energy fluxes from the US-Jo2 AmeriFlux eddy covariance tower in Tromble Weir experimental watershed at the Jornada Basin LTER site, 2010-ongoing	This data package links to eddy covariance data from a Jornada Basin LTER tower. The data are held at the AmeriFlux data repository (https://ameriflux.lbl.gov)	knb-lter-jrn.210338005
Catalog of GenBank sequence read archive (SRA) entries of 16S and 18S rRNA genes from bacterial and protistan planktonic communities along the Eastern Beaufort Sea coast, North Slope, Alaska, 2011-2013	Data inventory of runs, samples, and experiments held at GenBank.	knb-lter-ble.10
Correlation of native and exotic species richness: a global meta-analysis finds no invasion paradox across scales	This data package re-publishes data held in a package in Dryad. The metadata has been substantially enriched relative to the original dataset.	edi.548.1
Vascular Flora of the Harvard Farm at Harvard Forest since 2014	This data package includes an inventory table with information on voucher specimens held in the Harvard Herbarium.	knb-lter-hfr.236.3
Biological responses to landscape change in the McMurdo Dry Valleys, Antarctica	This data package links to genomic data in NCBI, and includes additional data from biogeochemical analyses performed on each sample.	knb-lter-mcm.262.1

4.5 Resources

This section aggregates information helpful at the time this document was written, particularly regarding nucleotide sequence and genomic data repositories in widespread use at this time. Given the rapid rate of change in the field, this info may fall out of date quickly.

4.5.1 Sequence and genomic repository information

It is generally preferable that sequencing and genomic data are archived in community repositories that are specialized for their data type, rather than in a generalist repository such as the Environmental Data Initiative (EDI). There are many such specialized repositories; a fairly comprehensive listing is provided by the journal Nucleic Acids Research (summarized on this page). Metadata standards and collaborative structures among these repositories are governed by the International Nucleotide Sequence Database Collaboration (INSDC, more guidance here).Often these repositories provide or are accessible to specialized tools for searching, accessing, and analyzing the data (e.g., BLAST, MG-RAST). Furthermore, some products derived from sequence or genomic data are best archived in another specialized repository (e.g., metagenome-assembled genomes, or MAGs). As a general rule, these specialized repositories assign unique identifiers to projects, samples, and/or single sequences (often referred to as accession numbers) that can be used to locate sequences or genomic data. Note that each repository may have its own mechanism for reverse linking to related data held in another repository (such as EDI), and these mechanisms are beyond the scope of this document.

- NCBI Databases - list of various databases with search capabilities. See also How to submit data to GenBank.
- NCBI Accession Number prefixes - Explanation of accession number prefix codes.
- DNA DataBank of Japan (DDBJ) - list of various databases with search capabilities. See also Submissions.
- European Nucleotide Archive (ENA) - list of various databases with search capabilities. See also Submit and update.
- Integrated Microbial Genomes & Microbiomes (IMG/M) system from the Joint Genome Institute
- MG-RAST (technically an analysis pipeline not a primary repository but replicates to primary repositories) Replicates to the European Bioinformatics Institute (EMBL-EBI), which in turn replicates to the NCBI Sequence Read Archive (such that data submitted on MG-RAST will automatically appear on all three).
- Barcode of Life DataSystems (BOLD) DNA barcoding is a taxonomic method that uses one or more standardized short genetic markers in an organism's DNA to identify it as belonging to a particular species. Through this method unknown DNA samples are identified to registered species based on comparison to a reference library. The Centre for Biodiversity Genomics in Canada maintains the BOLD public data portal, a cloud-based data storage and analysis platform.

4.5.2 Tips for locating metadata in sequence and genomic data repositories

Where information for populating metadata in EML has not been supplied directly to the IM from the research group, metadata that investigators provided when submitting data may be found in the genomics repository.

- For data in NCBI, go to the NCBI website and search using the accession number. Or search by accession number in a specific NCBI Database, for example Genes PopSet (the PopSet database is a collection of related DNA sequences derived from population, phylogenetic, mutation and ecosystem studies that have been submitted to NCBI).
- For sequences submitted to the NCBI Sequence Read Archive, there are some easily accessible online tools for generating tables of linked sequence data and their metadata. For an example, go to the example dataset at https://www.ncbi.nlm.nih.gbov/bioproject/305753, and click the number next to **SRA Experiments** to see a list of all experiments. Then click **Send results to Run selector** to see a table summarizing geolocations and associated metadata which could be archived at EDI or used to extract metadata for EML preparation.
- A full Data Carpentry tutorial on accessing data on the NCBI SRA database can be found here: Examining Data on the NCBI SRA Database
- BCO-DMO examples for contributing sequence accession numbers.

4.5.3 Darwin Core standard for sequence data

For sequence data to conform with the Darwin Core standard, a column header "associatedSequences" (https://dwc.tdwg.org/terms/#dwc:associatedSequences) may be used in the inventory table populated with a unique identifier (or list of identifiers) for the sequence data (e.g., SNLBE002-17, a sequence in Barcode Of Life Data system, aka BOLD) or full URL (e.g., http://www.boldsystems.org/index.php/Public_RecordView?processid=SNLBE002-17).

5 Spatial Data

Contributors: Tim Whiteaker, John Porter, Mary Martin

5.1 Introduction

This document/chapter contains recommendations on data package structure and metadata for spatial datasets. Over the timeline of Long Term Ecological Research (LTER) Network's use of the Ecological Metadata Language (EML), both spatial data formats and data curation options have evolved. In this document, focus on best practices that can be widely adopted with the goal of enhancing data discoverability and usability, and the understanding that there are multiple solutions to creating these data packages.

5.2 Recommendations for data packages

5.2.1 Considerations for archiving spatial data

5.2.1.1 Data formats

To maximize reuse, avoid proprietary formats. The formats listed below can be read or imported by most mainstream GIS programs or with code using libraries such as GDAL.

Strongly recommended formats:

- **GeoTIFF** - An open format for storing spatial raster data and metadata in a TIFF file.
- **GeoPackage** - A standard format from the Open Geospatial Consortium (OGC) for storing vector and raster data in a SQLite database file.

Some other formats to consider are listed below.

- **KML/KMZ** - Keyhole Markup Language (KML) file and its zipped version for storing vector data. This format was popularized by Google Earth and is now an OGC standard. KML is best visualized in Google software and may not render as well in other GIS software.
- **GeoJSON** - A format for storing vector data as text in JavaScript Object Notation (JSON). GeoJSON data are limited to the WGS84 coordinate system.
- **netCDF/HDF5** - binary formats originally designed for storing multidimensional arrays of spatial data typically organized onto a grid, but which now can accommodate vector data following the NetCDF Climate and Forecast Conventions (version 1.8 or higher).

A couple of Esri formats are worth mentioning and are listed below.

- **File geodatabase** - One of Esri's formats for storing vector and raster information. Several feature classes and rasters can be stored in this folder and file based structure. GDAL's OpenFileGDB driver enables non-Esri software to view at least the data layers in a file geodatabase, but usage of more advanced file geodatabase components such as topology rules or geometric networks may not be available outside of Esri software. Field types may not be imported correctly either. Export to GeoPackage instead, unless geodatabase is the only format that supports the advanced representation of your GIS data. Just know that you limit potential reuse of your data if you use this format.
- **Shapefile** - A legacy format for vector data which is widely supported. Be aware of shapefile limitations when considering this format. A shapefile consists of several individual files; include them as a single zip file in the data package. If the package has more than one shapefile, create a separate zip file for each shapefile.

Although other open formats exist, their implementation in popular GIS software may be less common. If a proprietary format must be used to capture the full meaning of the data, consider also including a version of the data in an open format such as a simple data table along with

metadata explaining its limitations in that format, or instructions on how to utilize the proprietary format. For example, an Esri layer package could be used when including recommended symbols for drawing vector features in a GIS is desired, in which case one could note that the vector data can be extracted by treating the layer package as a zip file.

Formats that are composed of more than one file, such as shapefiles, should be zipped. Include one dataset per zip file. For example, if you have 10 shapefiles, you would create 10 zip files.

5.2.2 Documenting spatial data packages

5.2.2.1 Document as spatial[Raster, Vector] vs. otherEntity in EML

There is a noticeable divergence in EDI spatial data packages, specifically, in the use of otherEntity vs spatial[Raster,Vector]. Here we discuss pros and cons of why one might choose to document spatial data with one type of EML entity over another. Either method is acceptable, and we recommend using spatial[Raster,Vector] when feasible. The documentation that follows provides best practices that will maximize discoverability and useability of spatial data, regardless of the entity type used.

5.2.2.2 otherEntity

- Pros
    - EML preparation is simpler than with the spatial EML types
    - Allows aggregated data structures (e.g., file geodatabases)
- Cons
    - Spatial data stored as <otherEntity> might be harder to discover because it may be difficult to determine if data in an <otherEntity> is spatial data or some other type when searching or browsing
    - There is currently no controlled keywording to identify spatial data files that are included as otherEntity in EML
    - Tabular attributes of geometric entities may not be described in detail
    - Units (latitude/longitude vs meters vs feet) and projections may not be identified

5.2.2.3 spatial[Raster,Vector]

- Pros
    - EML more fully describes vector attributes
    - There is a well documented path from Esri metadata to EML
    - An EML metadata search (on EDI or elsewhere) clearly identifies these as spatial datasets through the use of spatialRaster or spatialVector entities
    - LTER has built applications based on spatial[Raster,Vector] entities

- Cons
    - Data may not originate in ArcGIS, requiring a custom workflow to generate spatial entity EML
    - Spatial[Raster,Vector] can't describe multi-layer aggregates of GIS data (e.g., geodatabases containing multiple feature classes)

5.2.2.4 Keywords

Clearly identifying a dataset as spatial in nature is important to discoverability. This can be achieved by the use of keywords in the EML keyword elements as well as in the title/abstract and methods where appropriate. Keywords frequently searched include: GIS, geographic information system, spatial data, plus the more specific format names like shapefile, geoTIFF etc. Consider including as appropriate.

Do include the keywords **spatial vector** and **spatial raster** as appropriate for your data. These keywords should be used especially if the data are archived as otherEntity.

You may also include keywords that describe broad spatial data layers, e.g., digital elevation model, elevation, boundary, land use, land cover, census, parcel, imagery, as well as keywords that describe the specifics associated with a broad spatial data layer, e.g., land cover types such as water and vegetation types, land use types such as urban and forest, and so on.

5.2.2.5 GIS software compatible metadata

GIS platforms will not ingest EML metadata. If your GIS software creates its own metadata file specific to that software, then it may be included as otherEntity. Be sure to populate this metadata, for example with descriptions and units for attributes in vector data or raster attribute tables. However, metadata in the standard ISO 19115 or CSDGM format to enable the metadata to be read by other GIS software is more useful.

5.2.2.6 Attribute and coordinate system detail for otherEntity

While the GIS software compatible metadata included in the package typically describes attributes and coordinate systems of the data, such descriptions should also be included in the EML metadata to help users determine fit for use prior to data download. The EML spatialVector and spatialRaster types include elements for this purpose. EML otherEntity can also include attribute descriptions; however, inclusion of attributes in this more generic element may not be as common, and the element does not formally support a description of coordinate systems.

When using otherEntity instead of spatialVector or spatialRaster, include coordinate system details in the otherEntity/entityDescription element. If not including a description of attributes in the otherEntity/attributeList element, at least include a summary description of attributes in otherEntity/entityDescription. If the spatial dataset and its associated metadata files are the only items in the data package, then you can include these descriptions in higher level EML elements such as the dataset abstract in addition to or in place of descriptions at the entity level.

5.2.2.7 Standardized content for formats and entity types

In EML physical/dataFormat/externallyDefinedFormat, include a **formatName** indicating the spatial data file format. We recommend using format names from the DataONE format list when possible. Some spatial items from that list are shown below. Always check the list for the most up-to-date version of these names.

- Esri Shapefile (zipped)
- Google Earth Keyhole Markup Language (KML)
- Google Earth Keyhole Markup Language (KML) Compressed archive
- Network Common Data Format, version 4
- Hierarchical Data Format version 5 (HDF5)
- GeoTIFF
- GeoPackage Encoding Standard (OGC) Format Family
- Esri File Geodatabase (zipped)
- GeoJSON, version RFC 7946

If your format is not included in the DataONE list, consider submitting an issue to that GitHub repository's issue tracker so that the format can be added.

EML **formatVersion**, a sibling of formatName can be used to indicate the format version as in the example EML snippet below.

Example 5.1 EML for Externally defined format

```
<externallyDefinedFormat>
  <formatName>Network Common Data Format, version 4</formatName>
  <formatVersion>netCDF-4 classic</formatVersion>
</externallyDefinedFormat>
```

For otherEntity, when populating the **entityType** element, use **spatial raster** or **spatial vector** as appropriate.

6 Data Gathered with Small Moving Platforms

Contributors: Sarah Elmendorf (lead), Tim Whiteaker, Lindsay Barbieri, Jane Wyngaard, Greg Maurer, Hap Garritt, Adam Sapp, Corinna Gries, Stace Beaulieu

6.1 Introduction

Modern advances in technology have increasingly allowed the collection of ecological data using small, often uncrewed, moving platforms. Systems variously known as small Uncrewed Aircraft Systems (sUAS), Uncrewed Surface Vehicles (USV), Autonomous, Uncrewed Underwater Vehicles (AUV or UUV) or “drones,” more generally, now frequently serve as sensor carrying platforms. Moving platforms may also include gliders or animals with sensors affixed. Depending on the sensor(s) installed on the moving platform, data collected may include environmental measurements (temperature, concentration of chemicals), imagery (digital photos, multi- or hyperspectral sensors), or other remote-sensing acquisitions (ranging data, ground-penetrating radar). Example research applications include studies of vegetation cover and phenology, snowpack cover and depth, ground surface temperature, terrain elevation, bathymetry, species distribution or abundance, and many others.

Raw drone data can be voluminous and challenging to archive, but after processing, derived drone datasets typically resemble the more conventional spatial datasets that are regularly used in ecological research. In this document we focus on best practices for archiving raw and derived drone data, with particular attention to metadata and processing code that is specific to drone datasets. Note that this chapter does not specifically address data collected by large moving platforms like airplanes and satellites, or by human and animal platforms.

6.2 Recommendations for data packages

General considerations for archiving data from moving sensor platforms

- **Repository**: We are currently unaware of many specialized repositories for these data, and therefore, EDI is used as the representative data repository for many cases presented here. Repositories other than EDI may have specific metadata formatting requirements, but the general recommendations with regard to content could presumably be applied. For LiDAR based UAV data, consider contributing to Open Topography https://opentopography.org/); for AUV data, the U.S. Marine Geoscience Data System (MGDS) (http://www.marine-geo.org/index.php) which serves the IEDA "MGDL" node in DataONE is a good option. Glider data may be contributed to the U.S. IOOS Glider DAC (https://gliders.ioos.us/data/), archived at the National Centers for Environmental Information (NCEI) thus fulfilling NSF OCE Data Policy. If a decision is made to archive an LTER drone dataset in an external (i.e. non-EDI) repository but links to EDI data packages are desired, recommendations in Chapter 4 may apply.
- **Size of data set**: The file size of raw data from drone imagery can be substantial. If large volumes of raw data (>100 GB in total) are to be archived on EDI, please coordinate with EDI and follow the best practices for Large Datasets (Chapter 7) Even if raw data are in a proprietary binary format and specific software is required for processing, publishing them may be important for reprocessing when software improves.
- **Designing a data package**: In many applications of moving sensor data, raw images/measurements must be processed to arrive at data products that can be analyzed to answer research questions. To enable a fully reproducible analysis pipeline,

we recommend archiving three components: the raw data, any key derived data products (e.g., orthomosaic images, DEMs, DSMs, NDVI, landcover, snow depth, or surface temperature maps), and the processing code. These three components may be archived in separate data packages or together, and each should follow accepted best practices for its data type. To archive raw image collections, for example, see the considerations on grouping images into compressed archives (.zip, .tar) and creating an inventory file, as described in Images and Documents as Data (Chapter 3). For derived geospatial files, such as DEMs, refer to Spatial Data (Chapter 5). Custom processing code should be archived with the data following recommendations in Code in EDI (Chapter 1). If a standalone program is used to process data, reference the program in the methods metadata with adequate details to ensure reproducibility (name, version, date, configuration, etc.).

6.2.1 Metadata for moving platform data packages at EDI

The data package should include metadata elements that, at a minimum, (a) identify it as being collected by a moving platform, (b) deliver basic information about the data collection platform, instrument payload (camera, sensors), and procedure (flight information or similar), and (c) deliver necessary information about post-processing of the raw camera or sensor data, if any. Accordingly, these recommendations vary based on whether the data package contains raw or derived data.

High level metadata pertaining to the entire data package are easily provided in the EML file (e.g., a geographic bounding box). Data packages from drones or other moving platforms commonly include numerous point measurements, images, or other granular data entities, either separately or inside a compressed archive file. Detailed metadata pertaining to these data entities may be included as additional files in the data package. Inventory tables, usually a simple CSV file, are one such additional metadata format. For example, an inventory table could be used to list individual data files in the data package (e.g., images from one drone flight) and provide metadata (e.g. point location) about each. In addition to inventory tables, files that enable or supplement common processing pipelines, such as flight or mission logs, may be included. A flight/mission log may be provided in a proprietary binary format, but because software for parsing these formats may become obsolete, we recommend archiving the log in the format most useful for contemporary analysis software and extracting and appending the information to the inventory file where appropriate. Exif (Exchangeable image file format) metadata in images may also be programmatically extracted to supplement the inventory file.

Clearly, there are many possible ways to combine raw data, derived data, and metadata files into a moving platform data package. No matter the combination, the critical metadata categories and the recommended contents below should be considered and included where possible. The decision on whether to provide the metadata in EML or at a more granular level, such as an inventory table, will depend on the given dataset.

- **Methods:** unique identifier for a given flight or mission; summary information from a flight log; weather conditions; accuracy of sensor and geographic location information; data processing method; ground sample distance; image overlap; flight height; whether

UAS followed terrain elevation vs fixed-height flight; location of UAS launch (since some image metadata are derived from this); general description of software used and for what purpose; sensor calibration date and procedure; general description of payload type, such as multispectral camera and spectral bands.

- **Instrumentation:** make and model of platform, sensor, and camera, including manufacturer and specific model names and numbers. Include make and model of any interchangeable lenses in cameras. Specifics like spectral bands, temperature range, sensor accuracy, etc.
- **Software:** (see also Chapter 1) list of software used. Especially when code is proprietary or archived elsewhere, the name, version, and configuration of any software used are advisable, as are corrections applied (e.g., correction for sensor angle or heat/air flow). Ideally, a .pdf report generated by processing software can be archived as an otherEntity together with the imagery itself to convey much of the necessary information. Also include data used as a ground truth or calibration points for post processing (e.g. spectral calibration image/biomass sample/wind speed/etc) and their date of collection.
- **People with specified role:** drone operator, image processor
- **Geographic Information:** (see also Chapter 5) a general bounding box should be included in EML, while the individual location of images or point measurements should be handled in the inventory table, or directly in the included data files. Also include the coordinate reference system (e.g., WGS 84) used for images and (if different) ground control points, projection type if needed, altitude of image/measurement acquisition, spatio-temporal coordinates, pitch, roll, and yaw from flight log or image data points. It should be noted that there are special considerations for underwater vehicles, especially with regard to metadata to explain how geographic positions were obtained. With autonomous underwater vehicles (AUVs), there can be error sources in the topside GPS localization, the underwater acoustic positioning system (e.g., Long Baseline, Ultrashort Baseline), as well as any sensors used for dead reckoning (e.g., accelerometer, Doppler Velocity Log). At a minimum, it would be useful to know which sensors were used to produce the localization data and whether the navigation tracklines were post-processed with benchmarks.
- **Temporal Information:** may also be provided at the EML level or as timestamps for individual data/image points, either in inventory tables or in the data files themselves. Time of day critically affects useability for image-based datasets; so ensure that the time of day is clear from the metadata available prior to download, either in the EML temporal coverage or via the methods.
- **Keywords:** Use of appropriate keywords aids in data discovery. Keywords that identify datasets as drone-related are therefore recommended (e.g., drone, UAV, UAS). Keywords describing the type of data collected are also recommended (e.g., image collection, aerial imagery, thermal imagery, NDVI, digital elevation map). For drone mapping data products, keyword recommendations in Chapter 5 are largely applicable.

6.2.2 Examples and additional metadata guidance

Several EDI data packages for data from moving platforms are presented as examples in Table 1. Many more detailed, “drone-specific” metadata terms and values can be included in data packages for drones and other moving platforms. For completeness we have developed a comprehensive list of recommended and optional metadata terms based on the work of Wyngaard et al. (2019), Thorner et al. (2020), with mappings to select relevant ontologies, viewable here. For each metadata element, we assessed its utility in terms of data discovery, evaluating fitness for use, and actual data reuse. The minimum recommended subsets of metadata that are included in the section above were derived from this table.

Title	Description	EDI packageID
Orthophoto and elevation models from UAV overflights at the G-IBPE study site at Jornada Basin LTER in 2019	Approximately 599 RGB images and data derived from uncrewed aerial vehicle (UAV) overflights of the G-IBPE study site at the Jornada Basin LTER in southern New Mexico, USA.	knb-lter-jrn.210543001
Aerial imagery from unmanned aerial systems (UAS) flights and ground control points: Plum Island Estuary and Parker River NWR (PRNWR), February 27th, 2018.	USGS Aerial imagery UAS flights at the Parker River National Wildlife Refuge, Massachusetts, USA, includes ground control, multispectral and true color child items which each have data entities that include ground control or a file catalog of images	ScienceBase
Spatial variability in water chemistry of four Wisconsin aquatic ecosystems - High speed limnology Environmental Science and Technology datasets	water chemistry sensors embedded in a high-speed water intake system to document spatial variability.	knb-lter-ntl.337.4
Thermal infrared, multispectral, and photogrammetric data collected by drone for hydrogeologic analysis of the East River beaver-impacted corridor near Crested Butte, Colorado	infrared, multispectral, visual image data, and derivative products (orthomosaic and digital surface model) collected along a beaver-impacted section of the East River from August 12-17, 2017 and July 28-August 2, 2018.	ScienceBase

6.3 Resources

6.3.1 Tips and Tricks

For making an image catalog (.csv) from a directory of images, consider using the exif tool https://exiftool.org/. For example, the command “exiftool.exe -csv -r mydirectory > image_catalog.csv” will extract the entirety of the exif tags from all files stored under mydirectory into a comma-delimited table and write it to the the file image_catalog.csv

6.3.2 Semantic Annotation

Semantic annotation of drone imagery is a rapidly developing field. Ontologies that provide relevant terms include: dronetology; sensorML; FGDC content standard for digital geospatial metadata (not officially an ontology but a structured metadata format with defined terms); Semantic Sensor Network ontology (SSN, including the SOSA core); Semantic Web for Earth and Environment Technology ontology (SWEET); and Environment Ontology.

6.3.3 References

Thomer, Andrea K., Swanz, Sarah, Barbieri, Lindsay, Wyngaard, Jane. (2020). A minimum information framework the FAIR collection of earth and environmental science data with drones. DOI: 10.5281/zenodo.4017647

Wyngaard, J.; Barbieri, L.; Thomer, A.; Adams, J.; Sullivan, D.; Crosby, C.; Parr, C.; Klump, J.; Raj Shrestha, S.; Bell, T. Emergent Challenges for Science sUAS Data Management: Fairness through Community Engagement and Best Practices Development. *Remote Sens.* **2019**, *11*, 1797.

7 Large Data Sets

Contributors: Margaret O’Brien, Corinna Gries, Mark Servilla

7.1 Introduction

Data entities are kept offline when they are too large to be handled easily by the HTTP protocol, are expected to be rarely requested, and can be mailed on an external drive. If you suspect your data fall into this category, contact EDI for advice (support@environmentaldatainitiative.org). Below are recommendations for the EDI repository’s handling of data packages that have an offline component.

Standard practice is to handle data entities (both upload and download) via the HTTP protocol, using a URL. However, for very large datasets HTTP can fail due to physical limits. The limit for “too large” is somewhat subjective; EDI’s current limit for datasets that are “too large for HTTP” is 100 GB (all data and metadata).

7.2 Recommendations for data packages

7.2.1 Physical Storage

- The use of a Solid-state Drive (SSD) is strongly recommended for all offline data storage. The SSD should be formatted using one of the following file systems: 1) exFAT, 2) NTFS, or 3) ext4. Each of these file systems can accommodate individual file sizes greater than 1 TB.
- Add data to external drive in native (non-compressed, non-tarred, non-zipped) format, deliver to EDI (e.g., by physical mail).
- EDI will store three copies, one external hard drive each in New Mexico and in Wisconsin, one copy in general EDI backup cloud storage.
- Please purchase two external hard drives, copy your data and mail one copy each to:

Attn: Mark Servilla UNM Biology, Castetter Hall 1480 MSC03-2020, 219 Yale Blvd NE Albuquerque, NM 87131-0001

Attn: Corinna Gries University of Wisconsin Center for Limnology 680 North Park Street Madison WI 53706-1413

7.2.2 Data package

- The external hard drive should contain at least two entities: the data (which will be offline) and an inventory or manifest that describe the contents of the external hard drive.
- Content of the manifest (inventory table of holdings) would be dictated by the type of data entity. The **manifest will be available as an online entity** (through the EDI Data Portal) so that potential requesters can evaluate the offline resource before requesting it.
- Suggested columns are:
    - Filename(s)
    - Format (netCDF, tabular csv, etc.)
    - Start_datetime
    - End_datetime
    - Location_lat
    - Location_lon
    - (other params the PIs may feel are essential)
    - Checksum

7.2.3 Package Metadata

(in EDI metadata template and converted to EML - generally, as for any data package)

- Abstract: describe the collection generally. If individual files require specific software to read, provide the name of that software.
- Creators
- Contact (will be responsible for sending out copies as requested.) positionName: EDI Repository Manager Email: support@environmentaldatainitiative.org
- Methods - detailed collection/generation methods for the offline data entities. Detailed information for re-using the data. (May instead be included in the manifest table if different for different offline files.)
- Data Entities
    - Offline Entity:
        - Describe as you would for an online resource. Restate the software needed to read the individual files if this is important to a user. See Table 7.1 and Sample XML (Example 7.1 below).
    - Manifest (inventory of the offline holdings)
        - Column descriptions as for any data table

7.2.4 EML

In addition to basic resource-level metadata, at least two entities should be described:

- Manifest (inventory) should be a tableEntity: will be the online entity and described as all
- Offline entity:
    - Fill out high-level fields as for an online resource. Restate the software needed to read the individual files if this is important to a user.
    - Distribution node will be `offline` (See Table 1, code block)

Table 7.1 Three required fields in an inventory table for an offline distribution

physical/objectName	As for any entity, this is the name of the file or data object
dataFormat/ExternallyDefinedFormat/formatName	The name of the format the data object is in. If there is a special compression applied, list it here.
distribution/offline/mediumName	Instead of a data URL, you will have an offline distribution node. The name of

	almost all offline media is “external drive”, because that is how you will deliver the data to a requestor.

Example 7.1 Sample XML, offline entity

```
<physical>
  <objectName>mainl_2005acc.zip</objectName>
  <dataFormat>
    <externallyDefinedFormat>
      <formatName>netCDF file</formatName>
    </externallyDefinedFormat>
  </dataFormat>
  <distribution>
    <offline>
      <mediumName>External drive</mediumName>
    </offline>
  </distribution>
</physical>
```

7.3 Potential Issues

- SSD formatting (eventually, whatever we use, it will become unusable).
- Even with cloud storage, eventually a binary format will become unusable.