FACETS
a multidisciplinary open access science journal

NOTE

Using a linked table-based structure to encode self-describing multiparameter spatiotemporal data

Dewey W. Dunnington[a]* and Ian S. Spooner[b]

[a]Centre for Water Resources Studies, Department of Civil & Resource Engineering, Dalhousie University, 1360 Barrington Street, Halifax, NS B3H 4R2, Canada; [b]Department of Earth & Environmental Science, Acadia University, 12 University Avenue, Wolfville, NS B4P 2R6, Canada

*dewey.dunnington@dal.ca

Abstract

Multiparameter data with both spatial and temporal components are critical to advancing the state of environmental science. These data and data collected in the future are most useful when compared with each other and analyzed together, which is often inhibited by inconsistent data formats and a lack of structured documentation provided by researchers and (or) data repositories. In this paper we describe a linked table-based structure that encodes multiparameter spatiotemporal data and their documentation that is both flexible (able to store a wide variety of data sets) and usable (can easily be viewed, edited, and converted to plottable formats). The format is a collection of five tables (Data, Locations, Params, Data Sets, and Columns), on which restrictions are placed to ensure data are represented consistently from multiple sources. These tables can be stored in a variety of ways including spreadsheet files, comma-separated value (CSV) files, JavaScript object notation (JSON) files, databases, or objects in a software environment such as R or Python. A toolkit for users of R statistical software was also developed to facilitate converting data to and from the data format. We have used this format to combine data from multiple sources with minimal metadata loss and to effectively archive and communicate the results of spatiotemporal studies. We believe that this format and associated discussion of data and data storage will facilitate increased synergies between past, present, and future data sets in the environmental science community.

Key words: spatial data, temporal data, data formats, environmental data

Citation: Dunnington DW and Spooner IS. 2018. Using a linked table-based structure to encode self-describing multiparameter spatiotemporal data. FACETS 3: 326–337. doi:10.1139/facets-2017-0026

Handling Editor: Katrin Becker

Received: March 13, 2017

Accepted: December 18, 2017

Published: March 26, 2018

Published by: Canadian Science Publishing

Introduction

It has long been recognized that synergies between data sets are invaluable to researchers in the environmental sciences (Carpenter et al. 2009; Peters 2010; Marx 2013; Specht et al. 2015; McKay and Emile-Geay 2016). Many have noted the opportunities for synthesis among independently collected data sets (Carpenter et al. 2009; Peters 2010; Marx 2013; Specht et al. 2015), and some have called for software tools to ease the challenges of sharing data sets (Peters 2010). Spatiotemporal data, or data combining spatial variability with temporal variability, is an important subset of environmental data (Peters 2010) and includes data from long-term environmental monitoring, paleolimnological studies, dendrochronological studies, ice cores, and ocean cores. These data are often collected at high temporal resolution at great effort, the interpretation of which is facilitated by comparison to other

spatially or temporally proximal data sets. Although online repositories for environmental data have grown in number (e.g., DataONE, EcoTrends, NCEI Paleo), end-user software tools specifically dedicated to data exchange among researchers remain lacking.

Spatiotemporal data are by definition multidimensional and sometimes contain many parameters at high spatial and (or) temporal resolution. There is a high degree of variability in the storage format for these data sets, which hinders spatial and temporal comparisons of the data (Peters 2010; Reichman et al. 2011; Emile-Geay and Eshleman 2013; Marx 2013; McKay and Emile-Geay 2016). Furthermore, when data are not contained within an online repository, a comprehensive description of locations, parameters, and associated collection and measurement methods are not generally provided or stored with the data itself. These metadata are essential for both the comparison of a data set with other data sets (Fegraus et al. 2005; Reichman et al. 2011; Emile-Geay and Eshleman 2013) and the understanding of a data set by another researcher (Strasser et al. 2012).

All data storage is a balance between flexibility (what information can it store), efficiency (how much disk space is needed to store the data), and usability (how much manipulation is needed before something can be done with it; Anderson et al. 2009). An ideal format for environmental data must be flexible enough to store essential metadata, uncertainty information, and the data set itself in a single distributable container; efficient enough to store increasingly large data sets; and usable enough to encourage widespread use in the environmental community. Linked tables (Codd 1971, 1990) have long been used to store complex data in a single distributable container, but when they are used to distribute multiparameter spatiotemporal data, their use and format is often inconsistent. This paper describes an implementation of linked tables that provides a consistent, flexible, efficient, and usable data format to facilitate data exchange among a rapidly increasing number of multiparameter spatiotemporal data sets.

Example data

Throughout this paper we will use daily historical climate data from Environment Canada (2017) for Kentville, Nova Scotia, and Greenwood, Nova Scotia, from July and August of 1999 (**Fig. 1**). Climate data collected from climate stations are a common example of multiparameter data collected at discrete locations with a spatial component where the number of points in time is large. These data were obtained using the "rclimateca" package and R statistical software (R Core Team 2013; Dunnington 2017b).

Data formats

At least initially, much multiparameter time-series data are stored in a flat, wide format, with a single column for each measured parameter and a row for each time or depth measured. This format resembles the "tidy data" format described by Wickham (2014) and draws from the idea of location- and time-wide representations of spatiotemporal data described by Pebesma (2012). If multiple locations were measured in a study, the location names can be stored in a column or data from separate locations can be stored in separate tables. Value uncertainty information can be stored in additional columns (often one next to each parameter column), which requires a strict format if the data need to be machine readable. Other metadata that may apply to specific values (e.g., notes, additional uncertainty information) or parameters (e.g., units, measurement method) are generally lost in this format. Missing cell values may either indicate that a value was not measured or that it was below detection limit, and there is generally no way to distinguish the two outside the context of spreadsheet software. This data format is popular because of its simplicity and its usability. Spreadsheet software is amenable to visualizing and manipulating data in this format, which has likely aided its popularity. An example of this format is provided in **Table 1**.

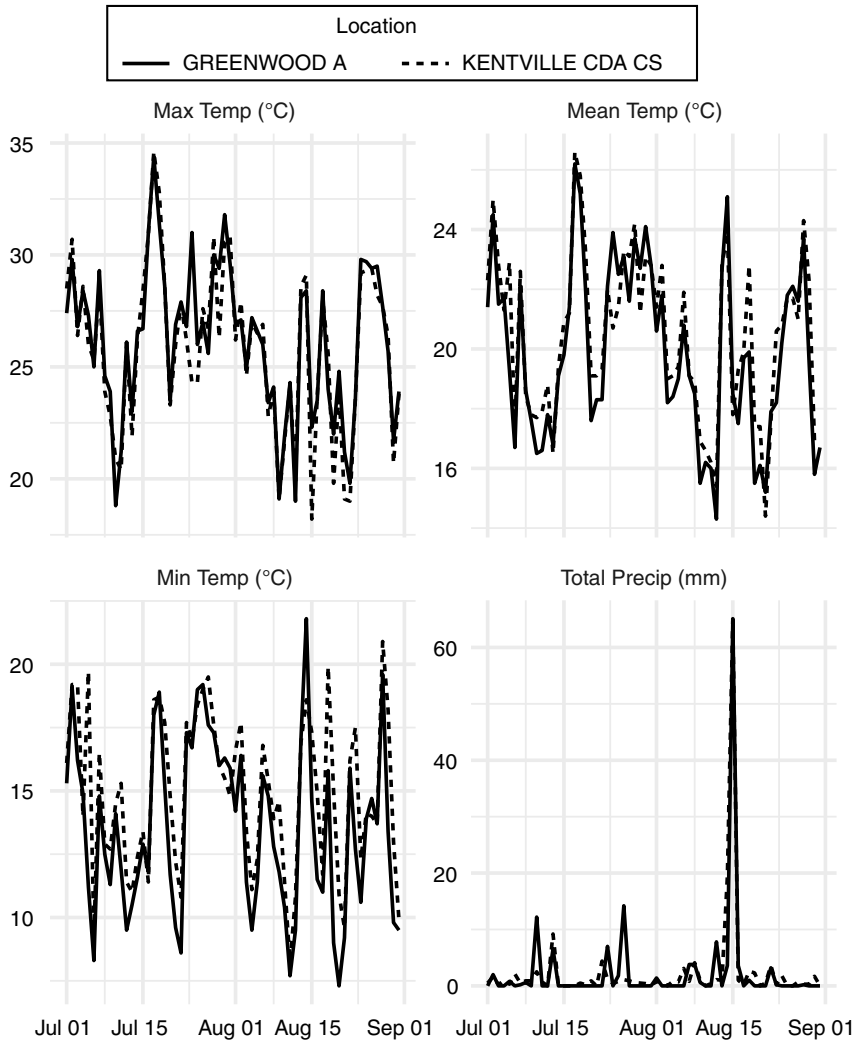

Fig. 1. Daily historical climate data from Environment Canada (2017). This data set features daily measurements of four parameters (Max Temp, Mean Temp, Min Temp, and Total Precip) for two locations (Kentville, Nova Scotia, and Greenwood, Nova Scotia) during July and August 1999.

Table 1. Example data in a flat, wide format.

Date	maxtemp	meantemp	mintemp	totalprecip
1999-07-01	28.5	22.3	16.1	0.5
1999-07-02	30.7	25.0	19.2	1.6
1999-07-03	26.4	22.8	19.2	0.2
1999-07-04	28.6	21.3	14.0	1.0
1999-07-05	26.0	22.9	19.7	0.3
1999-07-06	25.3	17.3	9.3	2.1

An alternative to a flat, wide format is a flat, long format with a column that specifies which parameter was measured, a column that specifies the value of that parameter, and a row for each time–parameter combination (Pebesma 2012; Wickham 2014). As a result, the table may contain many more rows than the wide format and contains much repeated information (each time–parameter combination is included for every measured value, whereas in a wide format each time value is only included once). Uncertainty information and other metadata that apply to specific values can be included as an additional column to this table (e.g., notes, additional uncertainty information); however, metadata for each parameter–location is still difficult to incorporate. This format can be edited or manipulated in spreadsheet software, is able to natively retain far more of the original information than a wide format (e.g., value uncertainty), is able to accommodate varying observation times across multiple parameters, and allows rapid plot creation via plotting libraries that include a faceting mechanism (e.g., the "ggplot2" R package; Wickham et al. 2016). Missing values are explicit and can have a defined meaning in a long format because if a location–parameter–time combination was not measured, the row will not exist in the table (Wickham 2014). Conversion to the wide format described above is available in spreadsheet software (using pivot tables) as well as Python (via the "pandas" package; McKinney 2016) and R (via the "tidyr" package; Wickham and RStudio 2017). An example of the sample data in a flat, long format is provided in **Table 2**.

Many spatiotemporal data are distributed in geographic information system (GIS) data formats, which generally consist of location information linked to a flat, wide data table (i.e., the attribute table). When the number of locations is large compared with the number of points in time, multiparameter spatial data are effectively stored in these formats. When the number of points in time becomes large, handling these data in GIS data formats with flat, wide attribute tables becomes unwieldy, and other structures are needed to avoid attribute tables with a large number of columns (i.e., a column for each parameter–point in time combination). Pebesma (2012) described a number of R data structures, provided in the package "spacetime" (Pebesma et al. 2016), which combine GIS structures with time-series structures, together forming a powerful interface for the visualization and analysis of single-parameter spatiotemporal data.

Table 2. The first 12 rows of example data in a flat, long format.

Date	Parameter	Value
1999-07-01	maxtemp	28.5
1999-07-02	maxtemp	30.7
1999-07-03	maxtemp	26.4
1999-07-04	maxtemp	28.6
1999-07-05	maxtemp	26.0
1999-07-06	maxtemp	25.3
1999-07-01	meantemp	22.3
1999-07-02	meantemp	25.0
1999-07-03	meantemp	22.8
1999-07-04	meantemp	21.3
1999-07-05	meantemp	22.9
1999-07-06	meantemp	17.3

There are many examples of multiparameter spatiotemporal data distribution in flat, long format. Statistics Canada (statcan.gc.ca/) distributes location information (census tracts) in a spatial data format, which is linked to actual census data that are distributed in a flat, long format. The US National Water Quality Monitoring Council (waterqualitydata.us/) distributes historic water quality data from a number of water quality monitoring organizations in a flat, long format along with location data in a separate table. The Environment Canada NAtChem data set (ec.gc.ca/natchem/) occasionally distributes data in a file that contains several linked tables that include location metadata, parameter metadata, and column metadata alongside data values, although data values are in a flat, wide format.

The recently proposed linked paleo data (LiPD) format is a nested format for multiparameter spatiotemporal data implemented in JavaScript object notation (JSON) that is optimized for paleoclimate data (Emile-Geay and Eshleman 2013; McKay and Emile-Geay 2016). This format is highly structured and suggests required metadata that should be distributed alongside paleoclimate data. The nested nature and JSON implementation of this format inhibits editing data directly in spreadsheet software; however, the underlying data are still stored in tables with one table for each measured parameter, a column for time, a column for the value of the parameter, and a row for each time measured. This structure is similar to the flat, long format described above, except data for each parameter are split into separate tables. The LiPD format also proposes a formal encoding for the interpretation of paleoclimate parameters (Emile-Geay and Eshleman 2013; McKay and Emile-Geay 2016), similar to other formal encodings for ecological metadata (Fegraus et al. 2005).

An ideal data format combines the strengths of all of the above data formats: the ability to view and edit data in spreadsheet software, the ability to store uncertainty information, the ability to easily visualize the data without complicated manipulation, and the ability to encode metadata that are critical for interpreting and contextualizing environmental data. In particular, we would argue that making data formats readable in spreadsheet software is critical to its widespread adoption, as most researchers have been trained in this and are comfortable with its use.

Methods

A mostly universal data structure

It is not possible to encode location data, data set metadata, parameter metadata, and the data itself in a single table without significantly undermining the usability and efficiency of the data structure. Therefore, we propose several linked tables to describe multiparameter spatiotemporal data: Data, Locations, Params, Data Sets, and Columns (**Tables 3–7**). The idea of using linked tables to describe data is not new (Codd 1971, 1990), and it is likely used in the background of many existing databases that contain data of this type. Its use to distribute data to the user, however, appears to be rare.

The Data table includes data set, location, param, *x*, value, and any additional columns that are pertinent to the measured value (e.g., uncertainty information). Here *x* is a placeholder for the axes along which the data vary, which may be any combination of date, datetime, depth in core, depth in water, or some other measure that in combination with data set, location, and param, uniquely identify a row in the data table. The Locations table includes data set, location, and any other columns that are pertinent to describe the location (e.g., latitude, longitude). The Params table includes data set, param, and any other columns that are necessary to describe the parameter (e.g., units, measurement method). The Data Sets table includes data set and any other columns that are necessary to describe the data set. Finally, the Columns table includes data set, table, and column, and any other columns necessary to describe the meaning of columns that may be included in any of the Data, Locations, Params, or Data Sets tables (e.g., data type, units). This allows column descriptions

Table 3. The first six rows of an example Data table.

Data set	Location	Param	Date	Value	Flags
ecclimate	KENTVILLE CDA CS	maxtemp	1999-07-01	28.5	NA
ecclimate	KENTVILLE CDA CS	maxtemp	1999-07-02	30.7	NA
ecclimate	KENTVILLE CDA CS	maxtemp	1999-07-03	26.4	NA
ecclimate	KENTVILLE CDA CS	maxtemp	1999-07-04	28.6	NA
ecclimate	KENTVILLE CDA CS	maxtemp	1999-07-05	26.0	NA
ecclimate	KENTVILLE CDA CS	maxtemp	1999-07-06	25.3	NA

Note: The data set column identifies the source of the datum, which is further described in the Data Sets table; the location column identifies the location at which the datum was collected, which is further described in the Locations table; the param column identifies the parameter whose value is given in the value column, and the date column (the *x* column) identifies the point in time represented by the datum. The flags column is an example of arbitrary extra columns that can be included in the Data table to fully describe the datum. Other examples of columns that could be included in the Data table are measures of uncertainty such as standard deviation or flags to indicate values that were below the detection limit.

Table 4. An example Locations table.

Data set	Location	Station ID	Latitude	Longitude
ecclimate	GREENWOOD A	6354	44.98	−64.92
ecclimate	KENTVILLE CDA CS	27141	45.07	−64.48

Note: The data set column identifies the source of the location, which is further described in the Data Sets table; the location column provides the identifier that is used in the Data table; and other columns are included to provide metadata about each location.

Table 5. An example Params table.

Data set	Param	Label	Units
ecclimate	maxtemp	Max Temp (°C)	°C
ecclimate	mintemp	Min Temp (°C)	°C
ecclimate	meantemp	Mean Temp (°C)	°C
ecclimate	totalprecip	Total Precip (mm)	mm

Note: The data set column identifies the source of the parameter, which is further described in the Data Sets table; the param column provides the identifier that is used in the Data table; and other columns are included to provide metadata about each location.

and data types to be distributed alongside the data. Each of the Locations, Params, Data Sets, and Columns tables are intended to be tables in a tidy format (Wickham 2014), with rows corresponding to the values found in the Data table and columns as variables describing their usage. We will refer to this format as the mostly universal data (MUData) format. An example of data encoded this form is provided in Tables 3–7.

Table 6. An example Data Sets table.

Data set	url
ecclimate	climate.weather.gc.ca/

Note: The data set column provides the identifier that is used in the Data, Locations, Params, and Columns tables, and other columns provide metadata about each data set.

Table 7. The first six rows of an example Columns table.

Data set	Table	Column	Type	Description
ecclimate	Data	Data set	Character	NA
ecclimate	Data	Location	Character	NA
ecclimate	Data	Param	Character	NA
ecclimate	Data	Date	Date	The date in YYYY-MM-DD format
ecclimate	Data	Value	Double	The measured value (see params table)
ecclimate	Data	Flags	Character	Flags provided by Environment Canada

Note: The data set column identifies the data set for which the column metadata is valid, which is further described in the Data Sets table; the table column describes the table for which the column metadata applies; the column column describes the column name; the type column describes which one of type of integer, double, character, date, datetime, wkt, or json applies to the values in that column; and other columns add additional metadata that apply to each column.

As a theoretical construct, this format is almost infinitely flexible; the limitations that restrict the amount, type, and content of the data are dependent on the file format in which the tables are saved (e.g., database, comma-separated values (CSV), JSON) and not the theoretical data structure itself. However, without constraints on the amount, type, and content of the data, this structure remains without utility. Therefore we propose the following restrictions to ensure storing data in this format facilitates the end goal of increasing comparability among data sets:

- Values in the location, param, and data set columns should be in a human-readable format. In traditional linked database storage, values in linked columns are numerical identifiers. This implementation is certainly possible within the MUData structure and may save disk space or memory when storing large data sets; however, it limits the utility of the data outside the context of database storage. The values in these columns should be short and alpha-numeric if possible. Additional details (e.g., units, special labels for labeling plots with non-ASCII characters) should be delegated to columns in the Locations, Params, Data Sets, and Columns tables as applicable.
- There should be a row in the Locations table for every unique data set–location combination in the Data table.
- There should be a row in the Params table for every unique data set–param combination in the Data table.
- There should be a row in the Data set table for every unique value in the data set column of the Data table. The scope of a data set is intended to mean data from a given source, and columns in the Data set table could include url, doi, and citation, among others.
- There should be a row in the Columns table for every non-required column in any of the tables. The values in the type column in the Columns table should be one of integer, double, character, date, datetime, json, or wkt. Including the ability for JSON-encoded data (json) to be included as

a column type allows more complex data types to be included in MUData tables; including the ability for well-known text (wkt) to be included as a column type allows for non-point geometries (e.g., lines and polygons) to be included in MUData tables, which is particularly useful in the Locations table.

- The name of the *x* columns (e.g., date, datetime, or depth in core) in the Data table must remain constant within an implementation of the MUData structure. Various examples of multiparameter spatiotemporal data lend themselves to specific time–date formats (e.g., depth in core, depth in water, tree-ring number, ^{14}C years before present, years AD, years before present, calendar date–time), some of which apply only to some disciplines. The purpose of the *x* columns is to uniquely identify rows in the Data table in combination with data set, location, and param. For the purposes of efficient storage, manipulation, and plotting, it is necessary for this column to be of the same type (e.g., integer, floating point number, date–time) for all of the data sets included in the Data table. If possible, this value should be numeric or a calendar date–time, such that plotting routines can understand the value as a continuous variable. The position of the *x* columns should be between the param and value columns such that they can be readily identified.

- Every unique combination of data set, location, param, and *x* column(s) should identify a single value in the Data table. Because there can be multiple *x* columns, replicates can be included with an *x* column that numbers replicates 1 to *n*, where *n* is the number of replicates for each data set, location, param, and *x* column(s). That each row can be uniquely identified with a set of identifiers is useful, and because multiple *x* columns are allowed, this constraint is rarely an issue.

- The encoding of dates and datetimes in the *x* column(s) (if applicable) should be in ISO 8601 format (e.g., year–month–day for dates, or year–month–day hour–minute–second for datetimes). Datetimes should represent an unambiguous point in time and specify a timezone in the Columns table.

- The data type for the value column in the Data table should remain constant. We recognize that not all parameters may produce measurements of the same data type (e.g., some parameters values may be a presence–absence measurement or require text to fully describe the value); however, in practice each column in each table must be of the same data type or automated plotting–manipulation routines will not produce the expected results. If this constraint is important in an implementation of the MUData structure, an additional column (e.g., text_value) could be included in the Data table, with a missing (NA) value in the value column if the value has no numeric equivalent.

- A missing value in the value column of the Data table should be explicit rather than implicit (Wickham 2014). If a value was not measured for a location–param–*x* columns combination, there should not be a row with that location–param–*x* columns combination in the Data table. This may be desired of there is no natural value of the proper type for the value column, but information still exists in other columns (e.g., text_value as described above). Depending on the underlying data structure, a missing value may be represented by NA (R), NaN (R, Python, and others), or NULL (database).

- The implementation of additional columns to each of the required tables is specific to each implementation of the MUData structure and could be either multiple columns in each table or a single column in each table containing key–value pair information in a format such as JSON or hstore. To facilitate viewing or editing in spreadsheet software, these additional columns should always have the ability to be displayed as columns. This adds the constraint that missing values cannot have a meaning in these columns other than that the column is not applicable to that row. If this is an issue, the column should never be included in JSON object–hstore conversion.

- The MUData structure is specified such that most of the information in the data set is contained in the Data table. Other tables are useful for providing metadata in an organized, complete, and computer-readable format; however, reviewing or plotting the Data table should be able to provide a reasonable idea of what the data contain.

This structure can also be further modified with additional tables following the guidelines above. For example, many spatiotemporal studies split time ranges into zones, the start and end of which could be encoded in a table with columns data set, location, name, x_start, and x_end. In the case of the example data, it may be desirable to provide a lookup table for "flag" values, which could be encoded as a table with columns data set, param, flag, and description. Extensions to the format, much like namespace extensions to Extensible Markup Language (XML), should not interfere with plotting–manipulation routines that only require the basic MUData structure described here.

Implementations

A MUData structure is fundamentally a group of tables that are related, a format that most naturally lends itself to a database implementation (Codd 1971, 1990). A database implementation is certainly possible; however, formats that can be directly read by commonly distributed software such as spreadsheet software are preferable. One such way is a folder with five CSV files: data.csv, locations.csv, params.csv, datasets.csv, and columns.csv. These files are readable by spreadsheet software as well as all major programming languages, with the disadvantage that CSV may not support complex data values. The collection of files can be combined into a single archive (e.g., ZIP archive) for compression and (or) distribution. A similar (but less useful to non-spreadsheet users) approach would be a spreadsheet file with separate "sheets" corresponding to each table.

The MUData structure is particularly amenable to a database implementation, because it contains a fixed number of tables, each of which contain a fixed number of columns. Non-required columns can be condensed to key–value pair information which can be stored in a single column using formats such as JSON or hstore. This implementation would be useful for a large database or where a database is already used to store location information (e.g., Postgis) and would be viewable in both spreadsheet and GIS software with a database connection properly configured.

In addition, we have created an R package implementing the MUData structure, with tools to create, combine, subset, plot, summarize, and write to disk data encoded in the MUData format (R Core Team 2013; Dunnington 2017a). Currently, the package reads and writes a CSV and a JSON implementation of the MUData structure as described above. An example of Canadian climate data distribution in this format can be found in the "rclimatca" R package (Dunnington 2017b). The usage of these packages is outside the scope of this paper; for details readers are referred to package documentation that is available (with the packages) on the Comprehensive R Archive Network.

Discussion

From our experience working with large amounts of multiparameter spatiotemporal data, the MUData format serves two purposes. First, it serves as a method to combine data from multiple initial sources. At a small scale, this can be a number of spreadsheets from several instruments and (or) laboratories that have processed samples for a given study. At a large scale, this can be combining results from multiple databases, such as the Environment Canada HYDAT database (wateroffice.ec.gc.ca/) or Environment Canada historical climate data (climate.weather.gc.ca/). Regardless of the initial format in which these data are provided, the MUData format offers a template to which most data can be converted with minimal data loss and minimal metadata loss. This ensures that data from multiple sources can be analysed consistently, repeatably, and completely, without ignoring data flags, detection limits, and uncertainty information that may be present in the original data.

Second, the MUData format serves as an effective format to communicate study results. The table-based nature of the format ensures that it is universally viewable, eliminating the need for specialized software to view the contents of a file. Because metadata, data flags, and uncertainty information can

be included in the format, data can be communicated within the proper context. We have used this format to effectively communicate the results of multicore paleolimnological studies and archive thesis data containing paleolimnological data and water quality monitoring results such that it can be effectively used in the future.

The MUData format draws from a multitude of existing tools that have been built around linked tables. The linked structure of the MUData structure resembles a spatial database, which is often implemented in software such as Postgis. In particular, the Locations table is essentially an attribute table that could contain a geometry column (Postgis) or be linked to a geometry file (ESRI Shapefile). Many analysis and plotting routines require data in a flat, wide format (e.g., correlation matrix as implemented in R, most plotting as implemented in spreadsheet software; Wickham 2014), to which the Data table is easily converted using pivot tables or the R package "tidyr" (Wickham and RStudio 2017). Analysis and plotting routines that readily handle data in a flat, long format (Wickham 2011; e.g., pivot tables in spreadsheet software, the "ggplot2" package, and the split–apply–combine analysis approach; Wickham et al. 2016) can be used directly on the Data table. Database join operations effectively copy metadata from the Locations, Params, and Data Sets tables into other tables where the information is relevant for plotting or analysis. In short, converting data to and from the MUData format in the context of existing data formats and widely distributed software is easily accomplished.

The MUData format does not, by definition, require the distribution of metadata or uncertainty information alongside environmental data; however, its structure and implementations are intended to encourage such activity by simplifying this task. Tables that list location information and parameters measured are common in publications, so in many cases these data may already exist in a tabular form. The metadata requirements of the LiPD and EML formats (Fegraus et al. 2005; Emile-Geay and Eshleman 2013), such as reference (e.g., DOI), parameter (e.g., units), and location (e.g., coordinates) information, can be accommodated by the Locations, Params, Data Sets, and Columns tables. Currently, these metadata specifications are implemented in JSON and XML (for LiPD and Ecological Metadata Language (EML), respectively), which are difficult to edit outside the context of dedicated software tools. By implementing a metadata specification in a table-based format, users are able to edit and create metadata using widely applied and widely available spreadsheet software, increasing the likelihood that metadata are distributed with data outside the context of data repositories for which they may be required. We hope that this encourages efficient ad-hoc collaboration in addition to efficient use of large-scale aggregations of environmental data that currently exist.

The type of data to which the MUData structure can be applied includes all data that contain multiple parameters in discrete locations where parameters are measured along a common axis. This includes climate data collected from fixed climate stations (where parameters may be temperature or precipitation and the common axis is calendar date), paleolimnological data (where parameters may be diatom abundance or element concentration and the common axis is depth or age, if an age-depth model is available), water quality data when sampled at discrete locations (where parameters may be pH or water temperature and the common axis is calendar date), among other examples.

The MUData format is most useful when all values are numeric; a mix of numeric and non-numeric values in a single column causes many programmatic implementations to interpret all values as non-numeric, which decreases the usability of the structure in a programmatic context. Similarly, when the number of locations approaches the number of points in time, storing these data in two tables as opposed to one begins to inhibit usability rather than increase it. In these situations we hope the concepts and references included in this paper can also be applied to promote comparability and documentation of these data sets.

Conclusions

There is a wealth of information contained in previously collected multiparameter, spatiotemporal data sets (Carpenter et al. 2009; Peters 2010; Marx 2013; Specht et al. 2015; McKay and Emile-Geay 2016). These data and data collected in the future are most useful when compared with each other and analyzed together (Carpenter et al. 2009; Peters 2010; Marx 2013; Specht et al. 2015). Currently, these data are stored in a variety of formats, some of which limit the degree to which uncertainty information and metadata can be distributed alongside the data. The MUData format attempts to ameliorate these issues by introducing a flexible, efficient, and usable format that is able to distribute metadata and uncertainty information alongside the data itself. The format can be implemented as spreadsheet files, CSV files, JSON files, databases, or objects in a software environment such as R or Python (Dunnington 2017a). The format can be applied as a storage mechanism, an input format, or an output format for researchers, data repositories, and software packages seeking to collect and compare multiparameter spatiotemporal data.

Acknowledgements

We would like to acknowledge funding from the Centre for Water Resources Studies at Dalhousie University, Acadia University, and the National Sciences and Engineering Research Council of Canada. We would also like to acknowledge the many students whose data we have helped organize; it is these conversations that formed many of the ideas found in this paper.

Author contributions

DWD and ISS conceived and designed the study. ISS contributed resources. DWD and ISS drafted or revised the manuscript.

Competing interests

The authors have declared that no competing interests exist.

Data accessibility statement

All relevant data are within the paper and available from Environment Canada (historical climate data available from climate.weather.gc.ca/).

References

Anderson E, Arlitt M, Morrey CB III, and Veitch A. 2009. DataSeries: an efficient, flexible data format for structured serial data. ACM SIGOPS Operating Systems Review, 43(1): 70–75. DOI: 10.1145/1496909.1496923

Carpenter SR, Armbrust EV, Arzberger PW, Stuart Chapin F, Elser JJ, Hackett EJ, et al. 2009. Accelerate synthesis in ecology and environmental sciences. BioScience, 59(8): 699–701. DOI: 10.1525/bio.2009.59.8.11

Codd EF. 1971. Normalized data base structure: a brief tutorial. *In* Proceedings of the 1971 ACM SIGFIDET (now SIGMOD) Workshop on Data Description, Access and Control, SIGFIDET '71. ACM, New York, New York. pp. 1–17. DOI: 10.1145/1734714.1734716

Codd EF. 1990. The relational model for database management: version 2. Addison-Wesley Longman Publishing Co., Inc., Boston, Massachusetts.

Dunnington D. 2017a. mudata2: interchange tools for multi-parameter spatiotemporal data [online]: Available from cran.r-project.org/package=mudata2.

Dunnington D. 2017b. rclimateca: fetch climate data from Environment Canada [online]: Available from cran.r-project.org/package=rclimateca.

Emile-Geay J, and Eshleman JA. 2013. Toward a semantic web of paleoclimatology. Geochemistry, Geophysics, Geosystems, 14(2): 457–469. DOI: 10.1002/ggge.20067

Environment Canada. 2017. Historical climate data [online]: Available from climate.weather.gc.ca/.

Fegraus EH, Andelman S, Jones MB, and Schildhauer M. 2005. Maximizing the value of ecological data with structured metadata: an introduction to ecological metadata language (EML) and principles for metadata creation. The Bulletin of the Ecological Society of America, 86(3): 158–168. DOI: 10.1890/0012-9623(2005)86[158:MTVOED]2.0.CO;2

Marx V. 2013. Biology: the big challenges of big data. Nature, 498(7453): 255–260. PMID: 23765498 DOI: 10.1038/498255a

McKay NP, and Emile-Geay J. 2016. Technical note: the Linked Paleo Data framework—a common tongue for paleoclimatology. Climate of the Past, 12(4): 1093–1100. DOI: 10.5194/cp-12-1093-2016

McKinney W. 2016. pandas: Python Data Analysis Library [online]: Available from pandas.pydata.org/.

Pebesma E. 2012. spacetime: spatio-temporal data in R. Journal of Statistical Software, 51(7): 1–30. DOI: 10.18637/jss.v051.i07

Pebesma E, Graeler B, Gottfried T, and Hijmans RJ. 2016. spacetime: classes and methods for spatio-temporal data [online]: Available from cran.r-project.org/package=spacetime.

Peters DPC. 2010. Accessible ecology: synthesis of the long, deep, and broad. Trends in Ecology & Evolution, 25(10): 592–601. PMID: 20728958 DOI: 10.1016/j.tree.2010.07.005

R Core Team. 2013. R: a language and environment for statistical computing. R Foundation for Statistical Computing, Vienna, Austria [online]: Available from R-project.org/.

Reichman OJ, Jones MB, and Schildhauer MP. 2011. Challenges and opportunities of open data in ecology. Science, 331(6018): 703–705. PMID: 21311007 DOI: 10.1126/science.1197962

Specht A, Guru S, Houghton L, Keniger L, Driver P, Ritchie EG, et al. 2015. Data management challenges in analysis and synthesis in the ecosystem sciences. Science of the Total Environment, 534: 144–158. PMID: 25891686 DOI: 10.1016/j.scitotenv.2015.03.092

Strasser C, Cook R, Michener W, and Budden A. 2012. Primer on data management: what you always wanted to know. University of California Office of the President: California Digital Library. DOI: 10.5060/D2251G48

Wickham H. 2011. The split-apply-combine strategy for data analysis. Journal of Statistical Software, 40(1): 1–29. DOI: 10.18637/jss.v040.i01

Wickham H. 2014. Tidy data. Journal of Statistical Software, 59(10): 1–23. DOI: 10.18637/jss.v059.i10

Wickham H, and RStudio. 2017. tidyr: easily tidy data with 'spread()' and 'gather()' functions [online]: Available from cran.r-project.org/package=tidyr.

Wickham H, Chang W, and RStudio. 2016. ggplot2: create elegant data visualisations using the grammar of graphics [online]: Available from cran.r-project.org/package=ggplot2.