#### Establishing a progressive data management workflow for biological data to inform adaptive management decisions

**Abstract**

Data management and data collection techniques have been advancing at a rapid pace over the past decade. A constant struggle in biology is to keep up with these advancements and the lack of reproducibility and accessibility to these data make these challenges apparent. These challenges hinder the ability to make rapid and informed decision making in ecological efforts. We customized a modern workflow for continuous and discrete long-term ecological data to assist in adaptive decision making that focuses on tackling many of the data management concerns with these types of data. We accomplish this by 1) standardizing field datasheets; 2) performing quality assurance and control (QA/QC); 3) creating scripts to analyze data and inform decision making; and 4) use a version control workflow to track changes to data, scripts and documents. The workflow uses open source software and tools to create a modern-day data management structure, which could be implemented in many research efforts.

#### Introduction and Background

Traditional field biology programs, many of which are designed to monitor animal populations and their environments, have experienced a substantial evolution in data collection, management, and storage technology in recent years. Changes include new sensor technology, data collection methods, and data observing platforms that are being used in large-scale monitoring programs including SECOORA (Southeast Coastal Ocean Observing Regional Association) and NEON (National Ecological Observing Network). As an example, advancements in sensor technology have allowed for significant changes in water quality monitoring such as transitioning from discrete single location and single point in time sample collections to real-time continuous observations at multiple locations. While the scale and technological capacity of many monitoring programs has increased these monitoring programs are still most often conceived, planned, and used by personnel trained as biologists and not data scientists. The lack of training in basic data management, curation, and workflow of data generated from these types of data collection platforms was demonstrated in a recent NSF (National Science Foundation) survey (Lowndes et al., 2017) which highlighted that of the 704 scientists who participated in the survey, “data skills” (e.g. multi-step workflows, ability to store, share and publish data) was identified as the largest unmet need (Barone, Williams, and Micklos, 2017; Lowndes et al., 2017).

The US Gulf of Mexico region is undergoing a large restoration effort to reverse observed declines in key ecosystem components including seagrass, fish communities, and oyster reefs using funding from the consolidated Deepwater Horizon settlements (see https://www.nfwf.org/gulf/Pages/home.aspx as an example). These restoration projects vary in spatial scale and funding, but, like other restoration efforts, these projects will have data collection and evaluation efforts that occur frequently throughout the project. These projects explicitly require an adaptive management plan to guide the restoration process. Extensive data management plans are mandated with the overall purpose of creating opportunities to improve future restoration actions by maximizing learning from previous and ongoing restoration efforts.

Our project is a large-scale restoration effort in the eastern part of the Gulf of Mexico funded by NFWF-GEBF on Lone Cabbage Reef (LCR). Our project’s primary goal is to restore specific historical oyster reefs so that they may be resilient to changing sea level and river discharge. This project generates data from multiple sources including continuous autonomous water quality data via sensors and observations of oyster populations by field biologists. These data are generated at different time frequencies with sensor data obtained at hourly time intervals from multiple spatial locations and biological data collected at discrete time intervals from multiple spatial locations. For both cases, there is a need to prepare data, meet data quality standards, and complete routine analyses of data to ensure data collected are useful for project objectives and questions. For adaptive management to work efficiently in our project, we developed a system which captures data as it is collected, guides the data to analyses, version control and data storage. Our data management system is essential to improve data quality by reducing the likelihood of data collection and analysis errors.

**#### Box 1. Terminology**

**“Living data”**

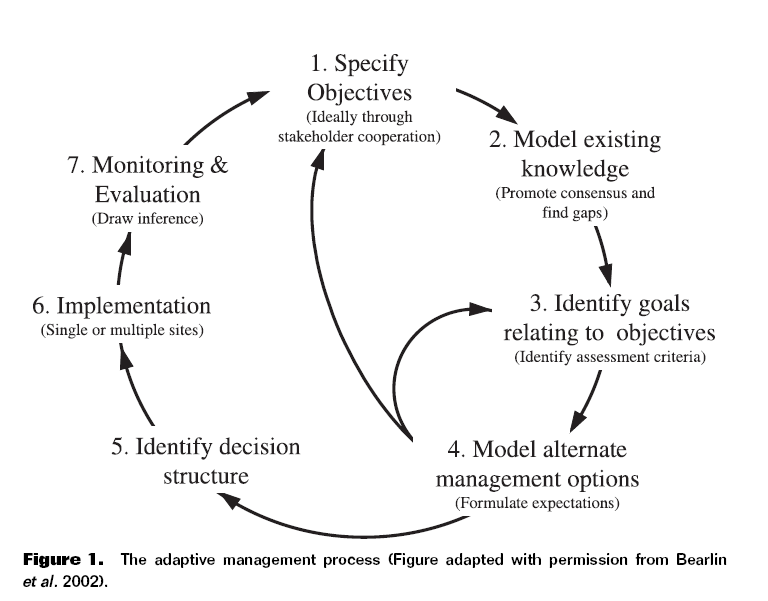
“Living data” are defined as data which are continuously collected and updated. These types of data are critical to this type of adaptive learning to inform restoration and management actions (Yenni et al., 2018). These informed adaptations during a restoration project can be small such as shifting the location of an autonomous sensor, to larger changes including restoration practices or revamping of sampling programs because of low statistical power. Living data are challenging to work with from a data management perspective because the data (by design) change as new data are collected and usually frequently. In a restoration or management context as these data are being collected, they must be processed, and analyses of these data to be completed to help draw inferences on how the system of interest is responding to the restoration action. This idea of iteratively integrating new data, analyses, and comparing these outcomes with previously stated objectives is not new and is a central aspect of the “adaptive management” process for natural resources first described in the 1970’s (Holling 1978; Walters 1986).

### Version control

Version control is defined as a software that allows for the saving and management of changes in content, documents, and other developmental information over time. The key purpose of using version control software is to document and confirm that changes in content are intended and planned. The advantages of using version control (1) a version control system saves all versions of a file, (2) version control records who made what changes to specific files and makes the user write detailed notes about what they changed (3) allows these changes to be undone if needed, (4) can facilitate reproducibility and transparency (Ram, 2013). Version control can be incorporated into a data workflow using software such as Github (<https://github.com>).

**Adaptive management**

By design, an adaptive management program requires rapid feedback loops between data collection, analyses, and interpretation to drive the process of updating knowledge, examining management and restoration options, making decisions and implementing actions (Nie and Shultz, 2012). This process is repeated (Figure 1) to improve management actions such as identifying the best restoration approach. To carry out a restoration project adaptively, data used in these continuous efforts meet quality assurance/quality control (QA/QC) protocols to identify and correct inconsistencies and errors in field or sensor observations before these data are used in an analysis. Errors in these data, or delays in producing the data in a usable framework, can quickly lead to a breakdown in the adaptive learning process either in terms of slowing the analyses limiting their utility for timely decision making, or worse, erroneously informing the decision-making process because of errors in data management or analyses.

Figure 1- The adaptive management process (Schreiber et al., 2004, adapted from Bearlin et al., 2002).

**#### End of Box 1**

Though there are only a few team members collecting and analyzing data, we have tackled some challenges to create and implement our current data management workflow. Our study involves water quality observations collected every hour every day and humanly observed oyster counts, and measurements surveyed during summer and winter. We created a data management workflow to efficiently process and analyze data to actively inform decision-making on efforts such as the amount of sampling trips needed to optimize oyster density estimates. We use software and tools that are open sourced, widely available and familiar to most field biologists such as R programming and Microsoft Excel. This paper explains our approach with the purpose of creating a guide for conservation efforts and emphasizing the necessity for establishing a data management workflow.

**#### Box 2. The LCR project data types**

The LCR project generates data from autonomous sensors, and humanly observed counts/measurements. The project is funded by NFWF-GEBF, over an 8-year span, to reconstruct the Lone Cabbage oyster reef and monitor the surrounding water quality. Several types of data are collected at various frequencies (seasonally, bi-monthly) and each data type requires individualized attention.

*Highest-frequency, autonomous water quality observations*

We collect hourly water quality observations from 11 different sites around Lone Cabbage reef. These observations are downloaded from the field bi-monthly. Regular maintenance of these sensors and their protective housing to ensure continuous stream of data. These “living data” are the most intensive and require strict data management protocols (Box 3).

*Medium-frequency, oyster counts and measurements*

Every year we survey oysters along transects and measure their lengths, at randomized locations, during the sampling periods of summer and winter. These data are recorded in the field on datasheets and translated into a computer through a dual-entry system. These data can be intensive as they require dual-entry and checking/reconciliation. Our oyster observation workflow reduces the chance for human introduced errors (Box 4 3A).

*Low-frequency, water quality YSI measurements*

During water quality service trips, we also collect YSI measurements to provide a supplemental check on our autonomous sensor observations. These measurements are recorded once during the water quality service trip for each site location. These observations are the least intensive as their frequency is low, and they are manually entered in the MySQL database (Box 3).

Though these data types may seem relatively similar, they are collected and handled differently to ensure data integrity. Addressing the variety of concerns which have been discovered through regularly updating these data types may also address many data management challenges which researchers may confront.

**#### End of Box 2**

### Establishing a modern data workflow

Prior to the start of the LCR project there was a need to create a workflow to manage and store water quality observations. This need required the expertise of the University of Florida Academic Research Consulting & Services (ARCS, <http://arcs.uflib.ufl.edu/> ) to provide guidance on how to create our MySQL relational database and implement this workflow. The justificaiton to create a database and workflow was to track multiple autonomus sensors through space and time regardless of where they are located. By doing this we were able to address our greatest concern, which was loosing the ability to track observations at a particular site. The workflow we have developed (Box 3) for water quality management requires some level of konwldge of computational tools (e.g., MySQL and R) and verstion control (e.g., GitHub, git). More than ever, data management is recognized as a core skill for biologists and ecologists (Hampton et al., 2017). This manuscript illustrates our project’s data management workflow for the purpose of providing guidance where others might be able to implement this system into their own conservation efforts. Boxes 3 and 4 describe our workflow implementation.

**#### Box 3. Water quality workflow**

For more information about our project’s MySQL import process go to (zenodo link for MySQL).

1. Datasheets are standardized and include pre-populated fields including the location, date to minimize error.

2. Water quality hourly sensor observations are downloaded in the field, unto a field laptop, and the technicians physically write the observation counts and any notes pertaining to the download. YSI measurements are also recorded at each location and physically written into the same datasheet.

3.A. Files are uploaded into a secure University of Florida internal server and a trigger starts the Python import process into our MySQL relational database, permanent storage. YSI measurements are manually entered into our MySQL relational database. The Python import process includes QA/QC procedures such as duplicate observation flagging.

3.B. QA/QC R scripts pull and process the water quality observations to check for flatlined or out of bound measurements.

3.C. Processed data, edited scripts, and documents are then stored and updated in GitHub. Standardized GitHub workflows are used during collaborative projects to ensure proper version control utility (zenodo link for GitHub workflow).

Figure 2- Data workflow for water quality observations.

**### End of Box 3**

**Adding water quality measurements to our permanent MySQL relational datatabase and version control**

Our MySQL relational database permanently stores water quality observations and can be accessed by any team member through a username and password. We store water quality data in specific tables where the sensor serial number and location must be pre-defined prior to importing the sensor observations (Box 3). These pre-definitions allow us to track which sensors are in which location at a specific time, which addresses our most challenging concern. The MySQL database relates to multiple tables through foreign keys and in our project the tables are related through sensor serial number and site location. We use R scripts to pull these unedited observations and process additional QA/QC procedures. These processed observations and their accompanying scripts are then updated using version control in the project GitHub master data repository (<https://github.com/LCRoysterproject>). This repository includes an up to date `master` branch which is protected from any unintended or incorrect updates using GitHub repository restrictions. To submit any changes to the `master` branch (referred to as a “pull request”), it is mandatory to have the changes/edits reviewed by another member of our team to ensure data integrity (zenodo link for GitHub workflow). Every “pull request” requires a thorough message describing each change, in the event an update to the `master` branch has to be investigated. Version control allows for team members to view a previous iteration of the `master data` branch and go back to that iteration if needed (Perez-Riverol et al., 2016). This workflow protects the `master branch` from possibly merging accidental or incorrect changes, giving a layer of needed security to the data.

**Automated data checks through Python and R scripts**

Water quality observations are imported into our MySQL relational database through custom Python scripting. The Python import process provides QA/QC procedures such as flagging duplicate water quality observations. If observations are flagged through the Python import process an inspection of these observations take place to find out if they are labeled as a duplicate. All unique observations are imported into our MySQL relational database, where they will be additionally reviewed via R programming scripts. The R scripts check for out of range measurements and additional scripts remove flatlined water quality measurements (usually due to ocean fouling). Additionally, water quality visualizations help check for data integrity. There R scripts are not automated, but they do provide a way to provide an additional layer of data integrity.

**#### Box 4. Oyster observation workflow**

For more information about our dual-entry system using a structure data packet visit here (zenodo link for data packet). Note, many of the lab processes are similar to those in the water quality workflow (Box 3).

1. Datasheets are standardized include pre-populated fields including the location, date to minimize error.

2. Team members walking along transects will count and measure oyster heights, while recording exact GPS coordinates. These observations are physically written these data sheets.

3.A. In the lab the team member who went out into the field, usually but not always, will enter their observations in a data packet that includes a dual entry system and data validation tools to ensure that the data entered into the packet is within range and standardized (e.g., site location, capitalization, appropriate oyster height range, etc) .

3.B. Use R programming to create internal reporting to estimate densities and power analysis which influence the number sampling trips needed for the season.

3.C. Processed data, scripts, and documents are then stored in GitHub. Standardized GitHub workflows are used during collaborative projects to ensure proper version control utility (zenodo link for Github workflow).

Figure 3- Data workflow for oyster observations.

**#### End of Box 4**

**Pre-populating data sheets**

Some of the challenges we have faced as an on-going project is to keep field data collection concise and accurate. Our oyster restoration project had two previous sampling events (e.g., Epochs 1 and 2), where field biologists also gathered oyster counts and measurements and transcribed them to data sheets. At the time data sheet structures frequently changed, for one reason or another, and some versions sometimes included or not included pertinent fields to the sampling event. Currently, in Epoch 3, we use one data sheet structure designed to fully contain all the fields and their units needed for the trip. Since we know exactly what information we should be collecting in the field, it is a straightforward process to pre-populate fields prior to collecting observations, to ensure that at least our randomly selected sampling locations and dates are correct. Minimizing how much human introduced errors prior to field collection has greatly reduced the effort needed to reconcile these fields in the future.

**Data validation on newly added oyster data**

Oyster counts and measurements are subject to data validation features through Microsoft Excel. Data validation ensures that every new observation manually entered is restricted and limited to what is applicable for that field. Such restrictions include oyster height measurement ranges, site location names, and acceptable dates for surveys. Two people, normally the technicians who surveyed the oysters, separately enter oyster observations with the data validation restrictions, in two separate Microsoft Excel tabs (Box 4 3.A). An additional Microsoft Excel tab will conclude whether the two separately entered versions are identical. If the dual-entry versions are not the same a “check” notification will appear on the Excel cells (e.g., the cell column and row number) that do not match. The flagged cells will then be reconciled by a third team member, who will investigate the discrepancy using the original data sheets. Dual-entry workflows are widely known to significantly reduce data entry errors (Bachard and Pace, 2011).

**Adding oyster observations to a central storage and version control**

Similarly, reconciled oyster observations are ultimately stored in our master data repository on GitHub and team members are required to follow the same workflow as previous mentioned (Box 4 3.C). The workflow ensures that every new type of oyster data updated are reviewed prior to merging with the protected `master` branch. It is also important to note that oyster measurements are not stored in MySQL since our MySQL relational database was created specifically for water quality observations. Oyster measurements are additional stored on a University of Florida protected server as back- up.

**Reproducibility and transparency using version control**

Using open-source software such as GitHub has increased our impact and credibility as a project by allowing others to view our data and reproduce our analyses. As a result, we may be able to gain benefits to our project by accepting feedback from other researchers. Gaining feedback from biologist’s harbors scientific collaboration, which is currently not common among of the scientific publishing community (Molloy, 2011). Working collaboratively may also help us to write more efficient scripts, which is a staple in the development of scientific programs (Prlic and Procter, 2012). Using GitHub version control also makes our project more transparent, describing why and how we have made edits/changes to our code and documents. Our ability to share our efforts through an open-source software may also facilitate novel research opportunities (Ram, 2013), while allowing providing guidance to others who may want to implement a similar version control workflow.

**Regularly updated data and adaptive management**

Due to efforts in creating and implementing our workflow we have rapid feedback between data collection, analysis and adaptive management. A recent example of adaptive management includes regular internal reporting of oyster densities (from oyster count observations) to influence the amount of sampling needed to optimize our efforts through power analysis. Another example on how we are implementing adaptive management to our project is analyzing if the 11 sensors we currently have deployed, are necessary for our water quality models. This is an ongoing effort, and there are no direct decisions made, however, we do plan to implement our findings on reducing the number of sensors if our analysis proves that to be necessary. Knowing the precise number of sampling trips and sensors needed for our project ensures that efforts and funds are allocated efficiently as required by our NFWF-GEBF grant contract.

#### Discussion

There is an increasing need for well-designed data management workflows in ecological efforts. Many journal publishing companies, and funding agencies are now requiring thought out data management workflows, which describes how data will be collected and what will happen to data after it is collected. Establishing a modern data management workflow prior to the collection or start of a conservation effort should be common practice. Projects that create a well-constructed data management workflow will be able to grapple possible challenges such as what do with with large or “living data”, which is a known difficulty in many science regimes (Marx, 2013). While at the same time, improving a project’s reproducibility and transparency may yield consistent results among collaborators (Ellison, 2010). Our data management workflow we’ve employed in the LCR project answers many of the challenges with “living data”. Our workflow uses a mixture of familiar software (e.g., Microsoft Excel) and versatile software (e.g., R programming, GitHub, MySQL), which encompasses an array of skills needed from a biologist to implement a modern workflow.

There are many advantages to using open-sourced tools (e.g., GitHub, R programming, and MySQL) in a data management workflow. Firstly, these software are free and there is continuous support for these applications online. Secondly, this workflow can be achieved by few biologists using online training programs such as The Carpentries (<https://carpentries.org/>), for example with GitHub integration. Many universities also offer R programming courses which teach the basics of statistical analysis with R (e.g., WIS 4601, Quantitative Ecology, <https://wec.ifas.ufl.edu/undergraduate-students/undergraduate-course-listing/>) and similar data management techniques described in this paper (e.g., WIS 6934, <https://datacarpentry.org/semester-biology/> ). Finally, using GitHub offers much desired flexibility in code development through “pull requests” (Rahman and Roy, 2014) and version control (Blischak et al., 2016).

Some of the disadvantages to our current workflow is that it can only handle only certain types of data and also so much “living data”. Our MySQL database can only store numerical or character information, it cannot store images or completed maps (<https://www.mysqltutorial.org/mysql-data-types.aspx> ). The MySQL database can also be difficult to make fundamental changes to, which we do not want to do at this time, and would require the expertise of ARCS to make any real changes to the functionality of the relational database. GitHub has a repository limit of 1 GB and up to 100 MB for an individual file (<https://help.github.com/en/github/managing-large-files/what-is-my-disk-quota#file-and-repository-size-limitations>), which can make it difficult to store large files without compressing them. However, despite these limitations with MySQL and GitHub, their functionality greatly outweights their restrictions.

Adapative management is also described as a process which continually improves policies and practices based on data outcomes (Pahl-Wostl, 2006). Due to the recent advancements of tehcnoglogy, one would assume that adapative management should be widely employed among ecological programs, however adpative management is infrequenctly implemented (Williams et al., 2007). We have described some of our challenges and our approaches to address these concerns through our data management workflow, hoping that it can provide guidance to future research efforts. Our data management workflow currently does not address some common concerns such as 1) citation and authorship credit, 2) managing for maps (e.g., geodatabases), and 3) data licensing (for collaborative data efforts). Collecting and managing “living data” is becoming the norm in many research programs. Making the effort to train teams and cultivate this new data type will ensure that scientists will be able to effectively manage these data, and these data will ultimately provide a feeback loop for adapative management.