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

**Abstract**

Advances in technology such as expanded remote sensing and animal tracking platforms have triggered rapid expansion of data available for ecologists and natural resource scientists to understand how plants, animals, and their environments interact and respond to anthropogenic change. This increase in new data creates both opportunities for learning and also challenges for managing these data and creating data workflows that lead to reproducible results. We customized a modern data workflow for continuous and discrete long-term ecological data to assist in adaptive decision making related to a large habitat restoration project. This workflow focuses on the data management concerns commonly encountered with large restoration efforts such as simultaneously managing data from autonomous sensors and field observations to inform ongoing restoration efforts. To promote reproducibility in our workflows and reduce data collection errors, we incorporated specific standards into our program including (1) standardizing field datasheets linked to an electronic data entry platform; (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 and is an example 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, but, like other restoration efforts, these projects have data collection and evaluation efforts that occur frequently throughout the project. Several of the restoration programs in this funding program require basic adaptive management concepts to be used to guide restoration actions. Under this framework, decisions related to restoration actions are made iteratively based on stating, testing, and updating hypotheses based on observed outcomes. In a restoration context, this information can be used to inform the restoration as the project is ongoing to maximize intended benefit. Doing so requires a data management plan designed to improve restoration actions by maximizing learning from previous and ongoing restoration efforts (Tompkins and Adger, 2004).

One example restoration effort funded by NFWF as part of the consolidated Deepwater Horizon settlements is the Lone Cabbage Reef (LCR) oyster reef project in the northeastern Gulf of Mexico. The primary goal of this project is to restore specific oyster reefs to historic levels 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 steps 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 capture and process data to meet standards and then complete routine analyses of these data to ensure they are useful for informing project objectives and questions. This is critical because this project by design uses adaptive management principles to inform the restoration through an interactive process of collecting data, analyzing these data, and informing restoration actions from these analyses. 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. Here we describe this data management system and the structure and decisions made in implementing the system to improve data quality and reduce 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 (Yenni et al., 2018). These types of data are critical to adaptive learning to inform restoration and management actions. Examples of learning as part of a restoration project includes small changes like shifting the location of an autonomous sensor, to larger changes such as revamping of sampling programs because of low statistical power. Living data can inform these decisions, but these data are challenging to work with from a data management perspective because the data (by design) change as new data are collected. In a restoration 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 as originally described (Holling 1978; Walters 1986).

**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. Data used must 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.

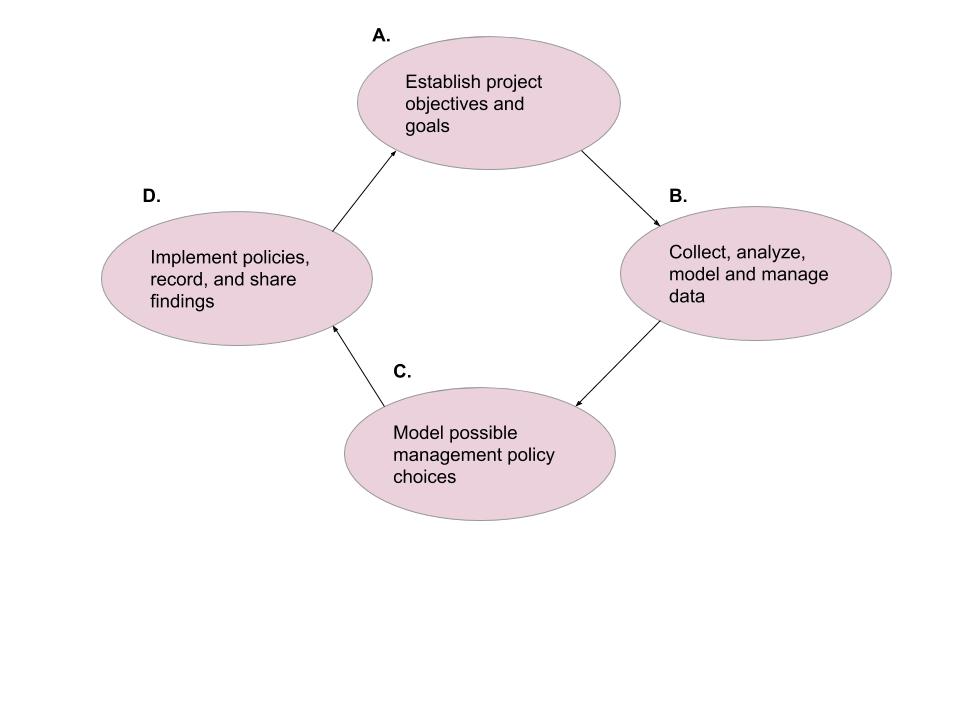


Figure 1- The adaptive management process (A) establish restoration project goals and objectives prior to data collection; (B) collect, analyze, model and store data; (C) use collected and analyzed data to create possible alternative management choices; and (D) implement these updated policies, , record and communicate findings.

**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>).

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

The Lone Cabbage restoration (LCR) project collects data on several parameters to measure ecosystem response to the restoration. One response metric is water quality observations collected hourly from autonomous sensors. A second metric are accounts of oysters at locations where restoration has been done (restored sites) and sites where restoration has not been done (wild oyster bars). Oyster counts are made by people conducting the field work during winter low tide events. 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 source, widely available and familiar to many field biologists such as R programming and Microsoft Excel. This paper documents this workflow and provides an example for use in other restoration and conservation projects.

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

The LCR project generates data from autonomous sensors, and human observed counts/measurements. Several types of data are collected at various frequencies (seasonally, bi-monthly) and each data type requires a specific plan to monitor data workflow from collection to analyses.

*Water quality data from autonomous sensors*

We collect hourly water quality observations from 11 different sites around Lone Cabbage reef. These observations are downloaded from autonomous sensors. Regular maintenance of these sensors and their protective housing are completed to ensure continuous stream of data by reducing data errors due to biofouling or equipment loss. These “living data” have the highest frequency of occurrence and require strict data management protocols (Box 3).

*Oyster counts and measurements from field sampling by people*

During winter low-tides when oyster bars are de-watered, teams of people collect counts and measure sizes of oysters on a selected group of oyster bars (Moore et al. 2020). These data are recorded in the field on datasheets and then entered into a computer through a dual data entry system where each data record is entered independently into the computer, and then these records reconciled to identify errors in data entry. Our oyster observation workflow reduces the chance for human introduced errors (Box 4 3A).

*Water quality YSI measurements*

During water quality service trips, we collect water quality data using a hand-held YSI (Yellow Springs Instrument) device 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 data type as their frequency is low, and they are manually entered in the MySQL database (Box 3).

While conceptually each of these data types are similar, because of differences in the frequency the data are collected and the collection method (with a machine or by hand), each data stream must be managed differently. 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

Database development efforts started prior to data collection through development of database “blueprints” via white board exercises to clarify (1) database goals, (2) data types and data sources, and (3) relationships among data types within the database. Blueprinting development efforts were led by University of Florida Academic Research Consulting & Services (ARCS, http://arcs.uflib.ufl.edu/ ). A key database need identified in blueprinting was the ability in the database to track observations at a particular site in space, and not focus on tracking observations recorded by an individual sensor, which could change locations over time. The workflow we have developed (Box 3) for water quality management requires open source computational toolssome level of konwldge of computational tools (e.g., MySQL and R) and verstion control (e.g., GitHub) but these tools are essential to basic data management which is increasingly recognized as a core skill for biologists and ecologists (Hampton et al., 2017). This manuscript illustrates the Lone Cabbage Reef data management workflow as an example that could be implemented in similar restoration efforts.

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

Extensive details on the MySQL import process are provided in the project management library (zenodo link for MySQL). An overview is provided here where:

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 to a field laptop while simultaneously notes are made on paper datasheets related to field conditions and equipment status. Water quality observations using the YSI device are also taken and recorded on the same data sheet.

3.A. Water quality sensor files are then uploaded into a secure University of Florida internal server and a trigger starts the Python import process into the MySQL relational database, which permanently stores raw files as an archive. The YSI device measurements are manually entered into our MySQL relational database.

3.B. QA/QC R scripts pull and process the water quality observations to check for flatlined or out of bound measurements (i.e., outside of expected range).

3.C. Processed data, edited scripts, and documents are then stored and updated unto 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 database and version control**

We use dedicated username and password controls to maintain access to the MySQL database to track access and database changes. 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. The MySQL database relates to multiple data tables through foreign keys (e.g, specified MySQL columns) 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 a review takes place to find out why the observations 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. The R scripts are not automated, but they do provide a way to provide quick and efficient checks on the data.

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

Detailed information on the dual data entry system using a structured data packet is available in the data entry documentation for the Lone Cabbage project (zenodo link for data packet). Several of these entry processes are similar to those in the water quality workflow (Box 3) and will only be briefly reviewed here where:

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

2. In the field, counts of oysters are recorded by team members on datasheets by hand.

3.A. In the lab data are entered using a dual entry system and data validation tools to ensure that the data entered are within range and standardized (e.g., site location, capitalization, appropriate oyster height range, etc.).

3.B. Standard R scripts are used to estimate oyster densities (e.g., population abundances) and power analyses are done using these data as they are entered to inform field sampling efforts within the field season.

3.C. Processed data, scripts, and documents are then stored unto 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 measurements.

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

**Pre-populating data sheets**

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. As an example, restrictions include oyster height measurement ranges, site location names, and acceptable dates for surveys. Two people separately enter oyster observations with the data validation restrictions, in two separate Microsoft Excel tabs (Box 4 3.A). An additional Microsoft Excel tab compares the two entry sheets to determine whether 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 known to significantly reduce data entry errors (Barchard and Pace, 2011).

**Datasheets, data entry and validation of oyster data from the field**

Datasheets are necessary for any ecological field data collection. Our standardized datasheets were created with the fact that these data will ultimately be entered electronically. These datasheets are formatted to be read in a similar manner in which the technicians will need to enter the data electronically in Microsoft Excel. Oyster counts and measurements data 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 (Barchard 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 data are also stored in a University of Florida protected server as a back-up.

**Reproducibility and transparency using version control**

Our efforts to implement a modern data workflow has led to rapid feedback between data collection, analysis, and changes to sampling routines as part of our adaptive management approach to the Lone Cabbage Reef restoration. A recent example of how we use 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.

**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 (Randall, 1990). 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 described as a process which continually improves policies and practices based on data outcomes (Pahl-Wostl, 2006). Due to the recent advancements of technology, 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.