#### 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 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 ((Martinelli et al., 2016)). 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 et al., n.d.).

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 be used to guide restoration actions (Zedler, 2017). Under this framework, decisions related to restoration actions are made iteratively based on stating, testing, and updating hypotheses based on observed outcomes (Figure 1). In a restoration context, this information can be used to inform the restoration actions such type of substrate to use in an oyster restoration project or monitoring program design as the project is ongoing, increasing efficiency by maximizing return on investment from restoration dollars. Doing so requires a data management plan designed to improve restoration actions by maximizing learning from previous and ongoing restoration efforts (Tompkins & Adger, 2004).

One example restoration effort funded by NFWF (National Fish and Wildlife Foundation) 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 this project to efficiently operate in an adaptive management framework, we developed a system which captures data as it is collected, guides the data to analyses, documents data and analyses decisions via version control, and archives and makes these data available for long-term reproducible exploration. 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 errors in analyses.

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

**“Living data”**

Living data” are defined as data which are continuously collected and updated (Yenni et al., n.d.). 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 living 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 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 system 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 & Schultz, 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 framework useable for analysis, can quickly lead to a breakdown in the adaptive learning process either in terms of slowing the analyses and limiting their utility for timely decision making, or worse, erroneously informing the decision-making process because of errors in data management or analyses.

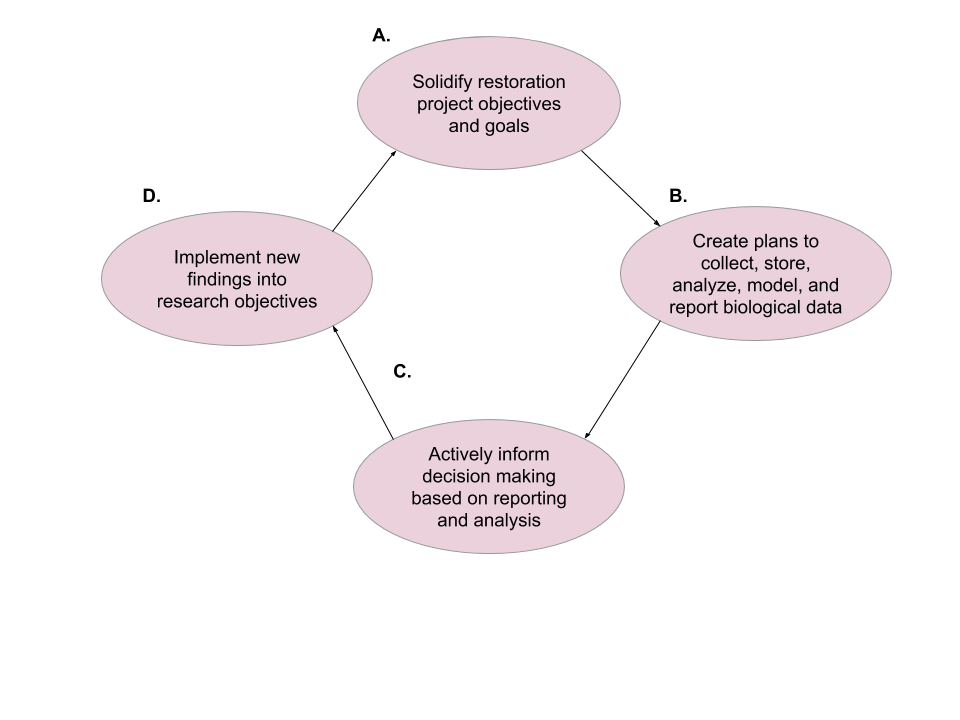


Figure 1- The adaptive management process for ecological restoration projects.

**Version control**

Version control software is a type of software that helps to manage documents, scripts of computer code, and other developmental information documents that are shared and iteratively updated 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 of project code and decision making (Ram, 2013). Version control can be incorporated into a data workflow using software such as Github (https://github.com).

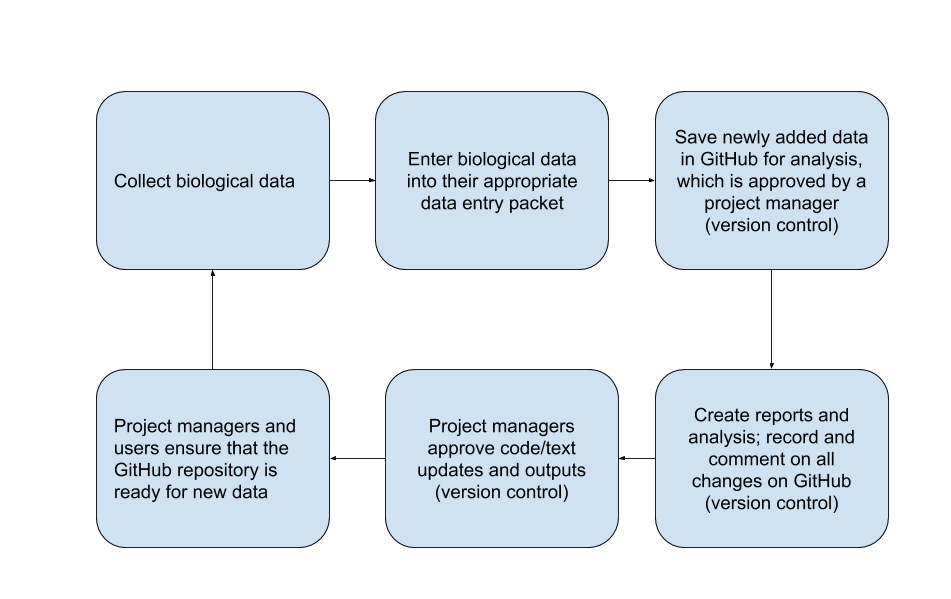


Figure 2 – Generalized version control workflow for the LCR project, detailed workflow information can be found here (zenodo link for Github workflow).

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

The LCR project collects data on multiple parameters to measure ecosystem response to oyster reef restoration. One response metric are observations of water conductivity and temperature collected hourly from autonomous sensors. These types of data are measured and recorded by the sensor and are output in a standard format than can be interpreted for analyses directly by a computer. A second metric are counts 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 fieldwork during winter low tide events. These data are collected by people and then must be entered into a computer as a standard data form before these data can be analyzed. We created a data management workflow to efficiently process and analyze data from both of these data streams. These data streams are then consolidated, and used to actively inform decision-making for the project 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 program R (Lefcheck, 2016) 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**

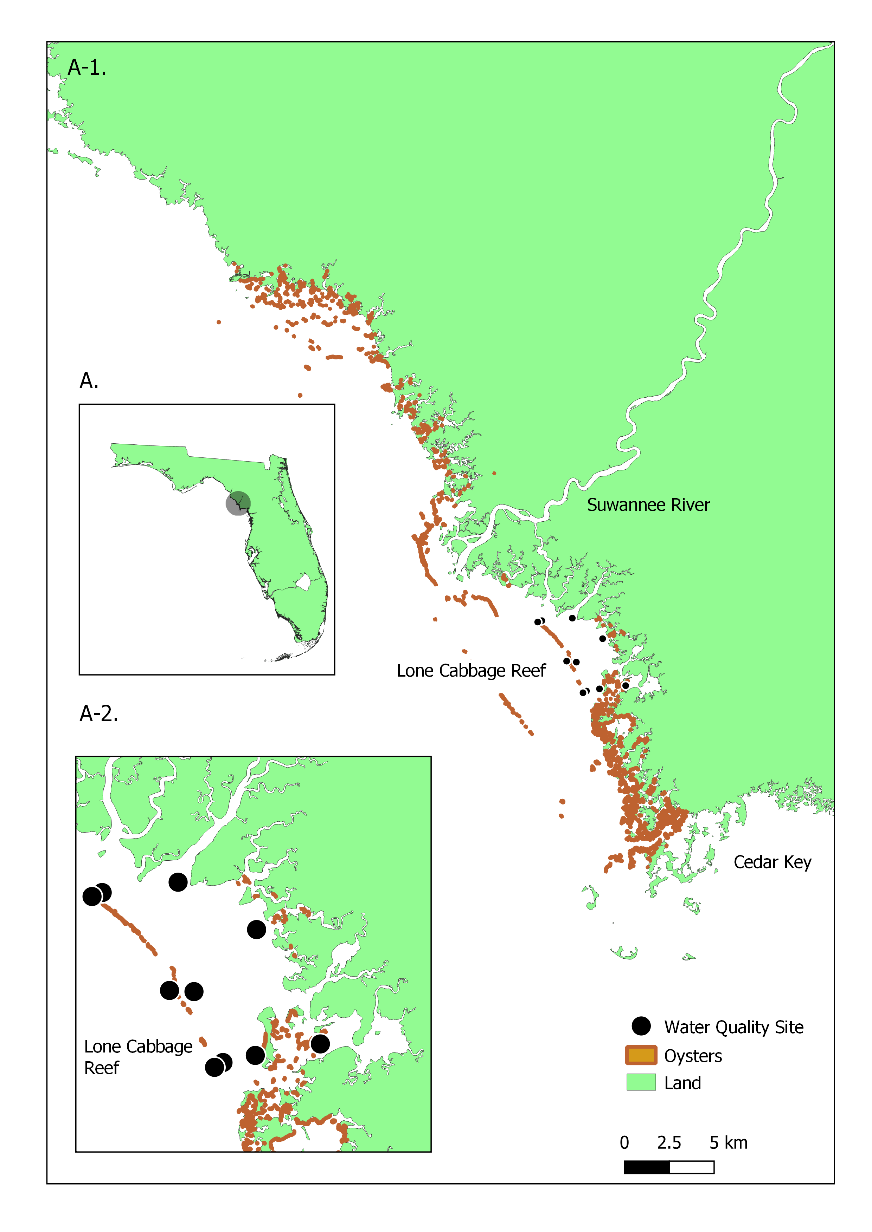
The LCR project generates data from autonomous sensors, and human observed counts/measurements. Several types of data are collected at various frequencies (seasonally, bi-monthly) and each data type requires a specific plan to monitor data workflow from collection to analyses.

Figure 3- Water quality location map. A) Map of Florida identifying general Lone Cabbage Reef area; A-1) Map scale 1:23,100 of Florida coastline between the mouth of the Suwannee River and Cedar Key, Fl; A-2) Map scale 1:9,000 of Lone Cabbage Reef with water quality sites identifies (black circles). The oyster shapefile used in this map is a from a University of Florida sampling effort in 2001.

*LCR project naming conventions*

A critical component of our data management plan was to create a naming convention standard for every sampling location. Every oyster transect sampling location is identified by its location, which we abbreviate (e.g, Lone Cabbage= LC, Horseshoe Beach = HB), and then a number added to the end to identify the location as separate from other sampling locations. Each sampling location is then recorded in a master sampling list to ensure that we can go back to the same location and to also reference the same location in the exact way throughout multiple sampling events. Even though the oyster transect sampling locations are randomized when selected, it is an integral part of our sampling workflow to establish the naming convention of the sampling location prior to the sampling event.

Another aspect of our naming convention standards, which directly relates to data management, are the way we name project files. We use a standard of referencing the date the file was created and what the file is so that every project member will be able to decipher the subject matter of the file without having to view it’s content. For example, our water quality sensor download files are named in a specific format YYYYMMDD\_sitelocation\_sensortype.file (e.g., 20200902\_wq7\_diver.MON). The file name is important to identify the date and site in a clear fashion especially when we are investigating sensor readings which may be corrupt or uncalibrated. This file naming format has saved time for project team members because all files are uniform and consistent in their naming, making it easier for each team member to follow the naming convention guidelines.

*Water quality data from autonomous sensors*

We collect hourly water quality observations from 11 different sites around Lone Cabbage reef (Figure 3). These observations are downloaded from autonomous sensors approximately every two weeks. 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 (most number of observations) and require strict data management protocols (Box 3) both in terms of launching and maintaining the sensors and in importing the data files to maintain database integrity.

*Oyster counts and measurements from field sampling by people*

The lowest lunar tides of the year in the area around Lone Cabbage Reef occur during winter, and these low-tide events de-water oyster reefs which allow teams of people to count and measure oysters to document status and trends of oyster populations (Moore et al. 2020). These count data are recorded in the field on datasheets and then entered into a computer by people through a dual data entry system where each data record is entered independently into the computer. The dual entry system also includes data validation drop downs that ensure that the user is only entering appropriate data. For example, if a user is entering the height of an oyster to be 1000 cm the data validation will reject this entry because there is a range of acceptable oyster heights that it will allow. Any mismatch or errors in the dual data entry worksheets will then be reconciled by the research coordinator. This double data entry system was created to reduce the chance of data entry errors and human introduced errors (Box 4 3A).

*Water quality measured by field-crews*

During water quality service trips, we also collect water quality measurements using a hand-held YSI (Yellow Springs Instrument) device 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 appear to be 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

Data collected in the field are stored in a relational database. 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 for water quality management (Box 3) addresses goals and special concerns identified through whiteboarding but requires open source computational tools, some level of knowledge of computational tools (e.g., MySQL and R) and version control (e.g., GitHub), which are tools essential for basic data management. While this example is specific to the LCR project, we feel that the workflow developed could\ 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:

1. Datasheets are standardized and include pre-populated fields including the location and 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 weather 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 measurements are manually entered into our MySQL relational database in its appropriate table.

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 4- 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 (Figure 5). 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.

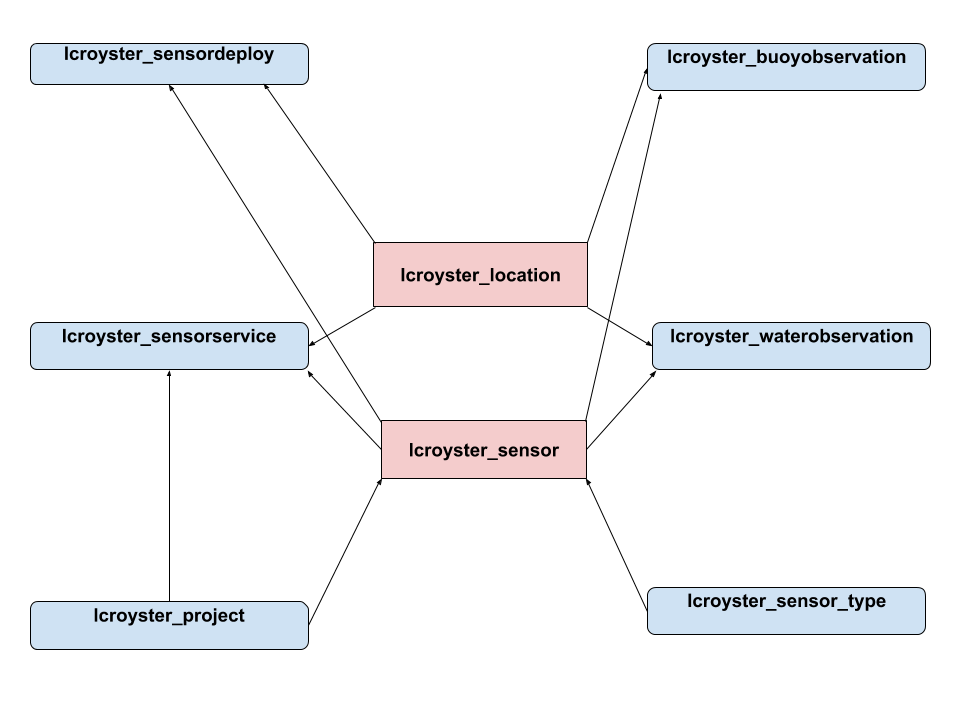


Figure 5 - Diagram of how the tables in our MySQL relational database are connected by a sensor’s location (lcroyster\_location) and serial number (lcroyster\_sensor). More information on the description of the tables and the data they archive can be found here (zenodo link for MySQL).

**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 and 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 are used 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 6- Data workflow for oyster measurements.

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

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

We developed standardized datasheets for recording information by hand from field observations. These datasheets were designed to (1) clearly detail format of information to be recorded, (2) minimize errors, (3) allow for easy transcription from field observation, to paper, to entry into the computer. As the oyster counts and measurements from the field transects are entered into the data entry form built in Microsoft Excel. Within Excel, data validation checks are automatically applied. These data validation checks provide initial assessments that every new manually entered observation is restricted and limited to what is applicable for that column. As an example, restrictions include oyster height measurement ranges, site location names, and acceptable date ranges for surveys. For some types of data entry, such as site names, we use drop down menus such that the person entering the data must choose the name of the site from a predefined list, instead of typing the name. This reduces the changes of entering a name incorrectly, but would still allow for an incorrect name to be selected. For other types of data entered, such as oyster heights, we pre-define a minimum and maximum range that is expected of any given oyster height such that if a value entered for a height falls outside the accepted range, then, a warning is issued and must be addressed. We also require that two people separately enter oyster observations, in two separate Microsoft Excel tabs (Box 4 3.A). An additional Microsoft Excel tab compares the two entry sheets to determine whether the separate 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. The process of a dual-entry workflow is known to significantly reduce data entry errors (Barchard & Pace, 2011).

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

Reconciled oyster observations are ultimately stored in our master data repository on GitHub and team members are required to follow the same workflow as defined for water quality data to upload this info to GitHub (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 with remote storage as a backup to the GitHub servers.

**Regularly updated data and adaptive management**

We have developed scripts of computer code that use these data to create summaries and routine analyses to provide information to (1) adaptively update sampling efforts and (2) meet reporting requirements of funding agency. Because of data workflow is standardized, each time new data are collected, then these standard computer scripts can be run to inform ongoing research efforts. As an example, we routinely use a type of power analyses to guide field sampling efforts during winter oyster sampling. Prior to the field sampling season, data from previous years are used to develop preliminary sampling guides in terms of location and number of transect samples to take. As field collections begin and new data are collected and entered into the computer, we re-run the power analyses to then update the sampling effort for that field season based on observed oyster density and variability from within season sampling. This allows us to allocate effort to locations where they provide the most information to meet project objectives.

#### Discussion

Establishing a data management workflow is receiving more attention in ecological efforts. Thus, creating a data management workflow from the beginning of the research initiative makes data management an easier endeavor to maintain than trying to reconcile and document the aspects of the study after a manuscript has been prepared (Archmiller et al., 2020) .Data and scripts without proper initial data management workflows can lead to an increased effort and time to properly archive and clean, and though it is possible for post-reconciliation in theory it is rarely followed in practice (Nelson & Grubesic, 2018) . Our data management addresses many of the challenges with “living data” such as reducing human introduced error, permanent data storage, and version control for text and scripts. 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 employ a modern workflow.

Our data management workflow may not work for every ecological project, however many elements discussed in this paper should still be applicable. The concept of creating a data management workflow prior to an conservation venture is one of our main talking points. Much of project planning time is allocated to the ecological question asked and how to set up the sampling design. However, the planning should continue and expand to how the collected data will be managed and to train team members on the workflow. Knowing the frequency of data collected, where it will be stored, how it will be entered, is necessary to ensure data integrity. The data collected and analyzed will ultimately guide ecological efforts and inform funding agencies of the progress. The principles of securing and validating data should also be considered of high importance for monitoring efforts as well.

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/>). 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/> ). Using GitHub offers much desired flexibility in code development through “pull requests” (Rahman & Roy, 2014) and version control (Blischak et al., 2016). GitHub consistently updates their software features making it a reliable resource for many projects.

Some initial difficulties to our workflow may arise in teaching team members how to use the workflow and to ensure that they are following workflow processes. It is important to communicate effectively with team members to guarantee they are collecting and maintaining data within the workflow procedures. Another disadvantage to our current workflow is that it can only handle only certain types and a limited amount of storage space. 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 outweighs their restrictions.

Adaptive management is described as a process which continually improves policies and practices based on data outcomes (Pahl-Wostl, 2007). Due to the recent advancements of technology, one would assume that adaptive management should be widely employed among ecological programs, however adaptive management is infrequently implemented (Weimer 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 feed-back loop for adaptive management.

**References**

Archmiller, A. A., Johnson, A. D., Nolan, J., Edwards, M., Elliott, L. H., Ferguson, J. M., Iannarilli, F., Vélez, J., Vitense, K., Johnson, D. H., & Fieberg, J. (2020). Computational Reproducibility in The Wildlife Society’s Flagship Journals. *Journal of Wildlife Management*, *84*(5), 1012–1017. https://doi.org/10.1002/jwmg.21855

Barchard, K. A., & Pace, L. A. (2011). Preventing human error: The impact of data entry methods on data accuracy and statistical results. *Computers in Human Behavior*, *27*(5), 1834–1839. https://doi.org/10.1016/j.chb.2011.04.004

Barone, L., Williams, J., & Micklos, D. (2017). *Unmet Needs for Analyzing Biological Big Data: A Survey of 704 NSF Principal Investigators*. https://doi.org/10.1101/108555

Blischak, J. D., Davenport, E. R., & Wilson, G. (2016). A Quick Introduction to Version Control with Git and GitHub. *PLoS Computational Biology*, *12*(1). <https://doi.org/10.1371/journal.pcbi.1004668>

Holling, C. S. (1978). Adaptive environmental assessment and management. John Wiley & Sons.

Lefcheck, J. S. (2016). piecewiseSEM: Piecewise structural equation modelling in r for ecology, evolution, and systematics. *Methods in Ecology and Evolution*, *7*(5), 573–579. https://doi.org/10.1111/2041-210X.12512

Lowndes, J. S. S., Best, B. D., Scarborough, C., Afflerbach, J. C., Frazier, M. R., O’Hara, C. C., Jiang, N., & Halpern, B. S. (2017). Our path to better science in less time using open data science tools. *Nature Ecology and Evolution*, *1*(6). https://doi.org/10.1038/s41559-017-0160

Martinelli, M., Guicciardi, S., Penna, P., Belardinelli, A., Croci, C., Domenichetti, F., Santojanni, A., & Sparnocchia, S. (2016). Evaluation of the oceanographic measurement accuracy of different commercial sensors to be used on fishing gears. *Ocean Engineering*, *111*, 22–33. https://doi.org/10.1016/j.oceaneng.2015.10.037

Nelson, J. R., & Grubesic, T. H. (2018). The implications of oil exploration off the Gulf Coast of Florida. *Journal of Marine Science and Engineering*, *6*(2). https://doi.org/10.3390/jmse6020030

Nie, M. A., & Schultz, C. A. (2012). Decision-Making Triggers in Adaptive Management. *Conservation Biology*, *26*(6), 1137–1144. https://doi.org/10.1111/j.1523-1739.2012.01915.x

Pahl-Wostl, C. (2007). Transitions towards adaptive management of water facing climate and global change. *Water Resources Management*, *21*(1), 49–62. https://doi.org/10.1007/s11269-006-9040-4

Perez-Riverol, Y., Gatto, L., Wang, R., Sachsenberg, T., Uszkoreit, J., Leprevost, F. da V., Fufezan, C., Ternent, T., Eglen, S. J., Katz, D. S., Pollard, T. J., Konovalov, A., Flight, R. M., Blin, K., & Vizcaíno, J. A. (2016). Ten Simple Rules for Taking Advantage of Git and GitHub. In *PLoS Computational Biology* (Vol. 12, Issue 7). Public Library of Science. https://doi.org/10.1371/journal.pcbi.1004947

Rahman, M. M., & Roy, C. K. (2014). An insight into the pull requests of GitHub. *11th Working Conference on Mining Software Repositories, MSR 2014 - Proceedings*, 364–367. https://doi.org/10.1145/2597073.2597121

Ram, K. (2013). *Git can facilitate greater reproducibility and increased transparency in science* (Vol. 8). http://www.scfbm.org/content/8/1/7

Tompkins, E. L., & Adger, W. N. (2004). Does Adaptive Management of Natural Resources Enhance Resilience to Climate Change? *And Society*, *9*(2). <https://doi.org/10.2307/26267677>

Walters, C. J. (1986). Adaptive management of renewable resources. Macmillan Publishers Ltd.

Weimer, T., Williams, B. K., Szaro, R. C., Shapiro, C. D., Adamcik, R., Boatman, M., Bransom, S., Casterson, J., Fay, J., Florence, S., Growitz, D., Hermans, C., Johnson, F. A., Kendall, J., Kubly, D., Mayer, M., Moyer, S., Pattison, M., Peterson, R., … Rodriguez, V. (2007). *Adaptive Management The U.S. Department of the Interior Technical Guide Lead Authors Other Contributors Book Design*.

Yenni, G. M., Christensen, E. M., Bledsoe, E. K., Supp, S. R., Diaz, R. M., White, E. P., & Morgan Ernest, S. K. (2018). *Developing a modern data workflow for living data*. https://doi.org/10.1101/344804

Zedler, J. B. (2017). What’s New in Adaptive Management and Restoration of Coasts and Estuaries? *Estuaries and Coasts*, *40*(1). https://doi.org/10.1007/s12237-016-0162-5