MassWateR: Improving quality control, analysis, and sharing of water quality data

by Marcus W. Beck, Benjamen Wetherill, Jillian Carr, and Pamela DiBona

Abstract An abstract of less than 150 words.

Introduction

Water quality measurements provide the foundation of environmental monitoring programs designed to protect or restore aquatic resources. In the United States, water quality monitoring programs are broadly guided by the federal Clean Water Act with the singular goal of restoring and maintaining the chemical, physical, and biological integrity of the nation's surface waters. Similarly, the Water Framework Directive provides a framework for the protection of aquatic resources in member states of the European Union. Numeric standards that define critical thresholds for protecting recreational, aquatic life, industrial, navigation and consumptive uses of the resource are often established, that, if exceeded based on water quality measurements, require additional regulatory action to ensure compliance. These standards and other regulatory assessments as applied at the state-level use information from long-term monitoring datasets (Schiff et al. 2016; Tango and Batiuk 2013), or data collected *ad hoc* from multiple assessment endpoints (Stein and Cadien 2009; Behmel et al. 2016; Kumpel et al. 2020), where the former is atypical for most surface water bodies. Many state or regional institutions that assess water quality rely on decentralized data sources, often combining datasets from local watershed groups or participatory science programs rather than a single database that contains adequate coverage for areas of interest (Buytaert et al. 2014; Kelly-Quinn et al. 2022). Use of these monitoring data in a regulatory context is not possible unless standard operating procedures are adopted and the data fulfill quality control requirements.

Monitoring data of sufficient quantity and quality are critical to ensure precise and accurate representation of environmental conditions. A significant bottleneck in the use of monitoring data for environmental assessment of surface waters is the ability to clearly and efficiently indicate that the data fulfill applicable quality control (QC) criteria for regulatory applications or inclusion in a consolidated database (Arndt et al. 2022). Common QC checks for *in situ* field measurements or concentrations measured in the laboratory may include 1) comparison of the precision between replicate samples (duplicates), 2) comparison of a sample to a known concentration (spikes or instrument checks), and 3) precision of the measurement from an empty or blank sample (blanks) (Wilde and U.S. Geological Survey 2002). An adequate number of QC samples must also be included in the dataset as a measure of "completeness". These checks are often compiled in a single report for review by appropriate regulatory agencies. For example, precision of duplicate samples for a given parameter must not vary 5% and at least 10% of the data should be dedicated to these checks as a measure of completeness. For local monitoring groups that lack the resources to develop robust and repeatable workflows, QC reports are often prepared manually before submitting the data for review. This process is time-consuming and prone to errors, often limiting the amount of useful information that is used for regulatory assessments or submitted to formal databases.

The R statistical programming language offers a valuable software platform for developing tools to improve the QC of water quality data. The use of R with document generation systems offered through packages like **knitr** (Xie 2015) and **rmarkdown** (Allaire et al. 2023) can be leveraged to generate QC reports that follow a standard format for review by regulatory agencies. These tools can also be used to format water quality data for submission to state or national water quality databases, such as the Water Quality Exchange (WQX) database maintained by the US Environmental Protection Agency (USEPA). This database is the largest source of monitoring data in the United States that includes information on hydrologic conditions and chemical, physical, and biological measurements from surface waters. Further, many environmental resource managers have the need to analyze status and trends in monitoring data and R packages such as **ggplot2** (Wickham 2016) offer useful approaches to visualize numerous water quality records in a single graph. Integrating this functionality into a single package is expected to have wide ranging utility for anyone collecting surface water data and is likely to improve the quality and insights obtained from these data.

This paper describes the **MassWateR** package developed to improve how environmental professionals perform quality control, analysis, and sharing of monitoring data for surface waters. The regional focus of the package is for monitoring data collected in Massachusetts, USA, with QC reports submitted to the Massachusetts Department of Environmental Protection and data submitted to the national WQX database. Although the initial conception of **MassWateR** was to address regional

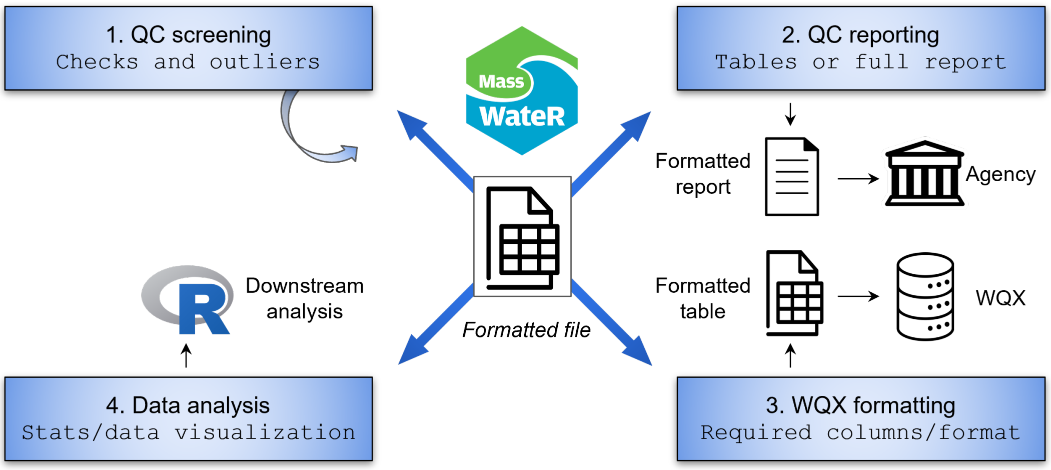


Figure 1: Workflow demonstrating how a user could engage with the MassWateR package. A user can apply one to any of the four steps depending on their need. The first step, QC screening, is often iterative as a user can modify parts of the raw data based on input checks or outliers. The second step can be used to create a QC report for submission to a regulatory agency. The third step can create a formatted table for WQX submission. The fourth step is data analysis and visualization, using MassWateR functions and downstream analysis with additional R packages and functions. All steps require a formatted input file. WQX: Water Quality Exchange; QC: Quality Control.

needs in Massachusetts, there is nothing specific in the package that prevents its use outside of the state as the QC checks and analyses follow routine methods for data collected elsewhere. As such, this paper is written with emphasis on how the tools are broadly applicable to anyone interested in improving efficiency and reproducibility of QC checks, in addition to analysis of water quality data and submission to WQX as the largest source of water quality monitoring data in the US.

Requirements for using MassWateR

To our knowledge, there are no existing R packages on CRAN that can be used to facilitate QC of water quality data, nor are any available that facilitate submission to existing databases. However, there are several that can be used to retrieve and analysis data from existing sources (see the CRAN *Hydrology* Task View). In particular, the dataRetrieval package (De Cicco et al. 2022) has been used widely to retrieve data from the USEPA Water Quality Portal (WQP), which is the counterpart of the WQX system for accessing data submitted using the latter. This package leverages a robust API to query existing water quality data in standardized format provided by the WQP. As such, data retrieval using existing web services is much simpler than data submission to a similar resource, as data formatting requirements do not apply when retrieving data. Developing a robust tool that can facilitate the upload of data to WQX, in addition to streamlining QC processes, would further the value of packages like dataRetrieval by increasing the amount of data that can be accessed through the WQP. The MassWateR package was developed to provide this benefit.

Users can engage with MassWateR to achieve different goals. This design was purposeful based on likely differences in needs among the user community. Although increasing data submission and facilitating QC reporting was the primary goal, we also assumed that users may not want to do both. That is, QC reporting is not a requirement to submit to WQX, whereas state institutions require this reporting for regulatory assessment. Users may also simply have a need to understand trends or to summarize their data, while also wanting to extend these analyses beyond MassWateR using additional R packages. Figure 1 demonstrates how a user may apply the functions in MassWateR once the required data are imported. The functions allow a user to engage with their data several ways, including 1) screening data for quality control, 2) summarizing quality control results into a single report or separate tables, 3) creating graphics for analysis and reports to stakeholders, and 4) formatting data for upload to WQX.

No matter the user need, all data inputs to MassWateR must follow a strict format. Developing a workflow to accommodate data inputs from the dozens of potential users from different organizations that use different data formats would have been impractical. As such, the only limitation to using the package is to adhere to the formatting requirements for all input files. Several resources are provided on the package web page to assist potential users in this effort. These resources included several training activities that were conducted during package development and templates demonstrating the appropriate format and rationale.

The required data files for using **MassWateR** are shown in table xx, including the files that apply to the workflow steps in Figure 1. The files are each imported into R using specific `read` functions with relevant checks, explained in the next section. These checks verify multiple requirements outlined in the template files, with informative errors or warnings returned to the console to prompt the user on the required action to remedy a formatting issue. The largest input file required for all parts of the workflow in Figure 1 is the results file. This file includes all water quality monitoring data to be used with the package. As such, the formatting requirements are the most burdensome for potential users and additional functions are available to assist with formatting.

table xx

The following sections describe the basic approach to using functions in **MassWateR** for any of the processes in the workflow. The naming convention for the functions is meant to provide users with an intuitive format for understanding what each function does and the step of the workflow for which the function applies. Although there are some exceptions to this nomenclature, the general format includes a prefix for each function as follows. Each prefix also includes `MWR` to avoid namespace conflicts with other packages.

- `read`: Read input files
- `check`: Check input files for formatting issues, used internally to the read functions
- `form`: Format input files for downstream functions, used internally to the read functions once the checks have passed
- `anlz`: Analyze imported data
- `tab`: Create formatted tables for QC analysis
- `qc`: QC functions for summarizing QC results, used internally to the table functions
- `util`: Various utility functions that accomplish routine tasks, but may be useful as standalone functions

Additionally, functions may often include a suffix that describes the relevant file used as input or otherwise evaluated in a downstream function.

- `results`: The results input file
- `acc`: The data quality objectives file for accuracy
- `frecom`: The data quality objectives file for frequency and completeness
- `sites`: The site metadata file
- `wqx`: The WQX metadata file

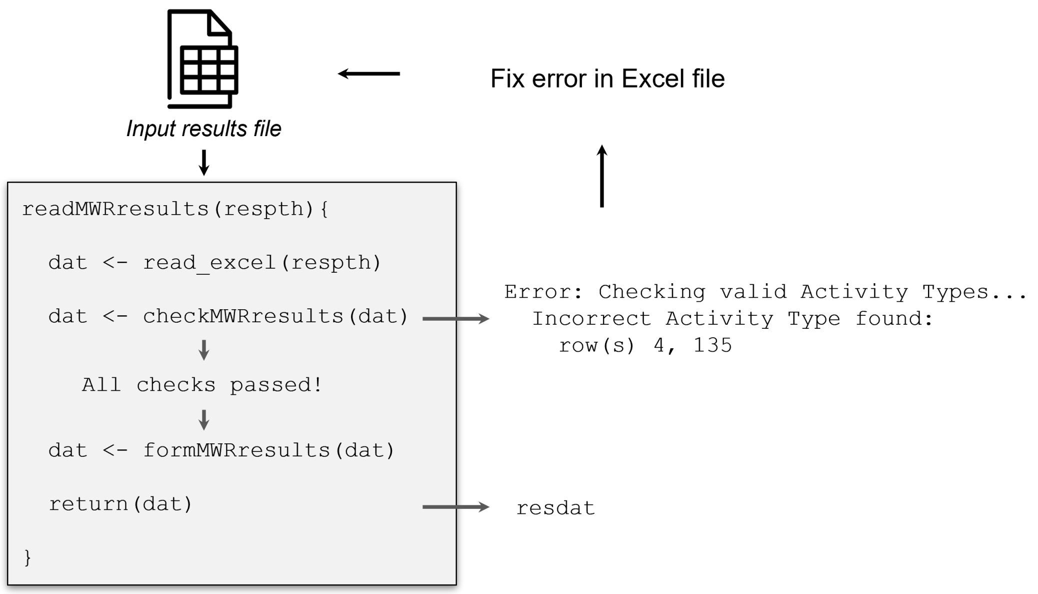
MassWateR functions

Read and check files

The primary task of the `read` functions is to ensure all imported files follow the required format for the package. Excel files are the expected format for all inputs and the `read` functions use the `read_excel` function from the **readxl** package (Wickham and Bryan 2023). The `read` functions do very little other than import the file - once the file is imported it is immediately passed to one of the relevant `check` functions inside the `read` function. There are several checks for each type of input file, with the number of checks increasing based on the complexity of the input file. Each check is printed to the R console on completion, whereas an error is returned at the first instance of a failed check, at which point the function is exited. The error will typically indicate which parts of the input data need to be changed to rectify the issue, often indicating a specific cell in the Excel file that requires attention. As such, the workflow is intended to be iterative, where a user imports a file, receives an error, manually changes the input file in Excel, then imports the data again until all checks pass (Figure 2). Again, this design was purposeful as many monitoring agencies and groups collect data differently and a standard input format for the package was the best option to accommodate potential users. This may also encourage future standardization among groups for how data are maintained to ease formatting challenges to using **MassWateR**. A user only needs to format their data once to use the package.

A correctly formatted input file would be imported as follows, with the messages in the console indicating the checks that are performed and that all checks were successful. Below demonstrates what would be shown for the results file using an example dataset included with the package. A total of fifteen checks are applied to the results file.

```
library(MassWateR)
```



```
readMWRresults(respth){

  dat <- read_excel(respth)

  dat <- checkMWRresults(dat)

     All checks passed!

  dat <- formMWRresults(dat)

  return(dat)

}
```

```
Error: Checking valid Activity Types...
  Incorrect Activity Type found:
    row(s) 4, 135
```

```
resdat
```

Figure 2: Pseudocode demonstrating the iterative process of importing a required data file for MassWateR. All read functions import an Excel file and the imported file is then passed to a check function. The function exits if an error is encountered, allowing the user to manually fix the identified error and then import again. After all checks are passed, a formatting function is applied to correct minor issues (e.g., standardize date format as YYYY-MM-DD) and the final data object is returned.

```
# import results data
respth <- system.file("extdata/ExampleResults.xlsx", package = "MassWateR")
resdat <- readMWRresults(respth)

#> Running checks on results data...

#>  Checking column names... OK

#>  Checking all required columns are present... OK

#>  Checking valid Activity Types... OK

#>  Checking Activity Start Date formats... OK

#>  Checking depth data present... OK

#>  Checking for non-numeric values in Activity Depth/Height Measure... OK

#>  Checking Activity Depth/Height Unit... OK

#>  Checking Activity Relative Depth Name formats... OK

#>  Checking values in Activity Depth/Height Measure > 1 m / 3.3 ft... OK

#>  Checking Characteristic Name formats... OK

#>  Checking Result Values... OK

#>  Checking QC Reference Values... OK

#>  Checking for missing entries for Result Unit... OK

#>  Checking if more than one unit per Characteristic Name... OK
#>  Checking acceptable units for each entry in Characteristic Name... OK

#>
#> All checks passed!
```

The following shows a typical error that might be returned if a check fails importing the results file. The `resdat` object is an imported results file that has passed all checks, but incorrect entries are added to the `chk` object to demonstrate the error that is returned if a user would have attempted to import this file. The `checkMWRresults()` function is run inside the `readMWRresults()` function and runs the checks (e.g., the column names are correct, all required columns are present), but then stops when invalid activity types in the `Activity Type` column are found. In this example, a user would need to change the entries in rows 4 and 135 of the `Activity Type` column in their Excel file to fix the issue and import the file again as in Figure 2. The online vignette specifies the valid entries.

```
chk <- resdat
chk[4, 2] <- "Sample"
chk[135, 2] <- "Field"
checkMWRresults(chk)

#> Running checks on results data...

#>  Checking column names... OK

#>  Checking all required columns are present... OK

#> Error:   Checking valid Activity Types...
#>  Incorrect Activity Type found: Sample, Field in row(s) 4, 135
```

The `readMWRresultsview()` function is also available to assist with troubleshooting formatting issues for the results file. This function exports an Excel file that shows the unique values that are found in each column to allow a user to quickly see potential incorrect entries. The output is similar to running `apply(resdat, 2, unique)` in the console, but includes an external file that users may be more comfortable evaluating to troubleshoot formatting problems.

After all file format checks are fixed, a standard approach for using **MassWateR** is to import all required files and save them as a list of named data frame objects that can be used by nearly all the package functions. This prevents the need to identify which input datasets are needed for each function, although the latter approach could used because arguments for individual input files are also provided for all functions. In the latter case, a path or data object can be used as input for each file. For the former approach, the beginning of a script for using the package could appear as follows. Example files included with the package are imported, whereas a user will specify paths to their own files.

```
library(MassWateR)

# import results data
respth <- system.file("extdata/ExampleResults.xlsx", package = "MassWateR")
resdat <- readMWRresults(respth)

# import accuracy data
accpth <- system.file("extdata/ExampleDQOAccuracy.xlsx", package = "MassWateR")
accdat <- readMWRacc(accpth)

# import frequency and completeness data
frecompth <- system.file("extdata/ExampleDQOFrequencyCompleteness.xlsx", package = "MassWateR")
frecomdat <- readMWRfrecom(frecompth)

# import site data
sitpth <- system.file("extdata/ExampleSites.xlsx", package = "MassWateR")
sitdat <- readMWRsites(sitpth)

# import WQX meta data
wqxpth <- system.file("extdata/ExampleWQX.xlsx", package = "MassWateR")
wqxdat <- readMWRwqx(wqxpth)

# a list of input data frames
fsetls <- list(res = resdat, acc = accdat, frecom = frecomdat, sit = sitdat, wqx = wqxdat)
```

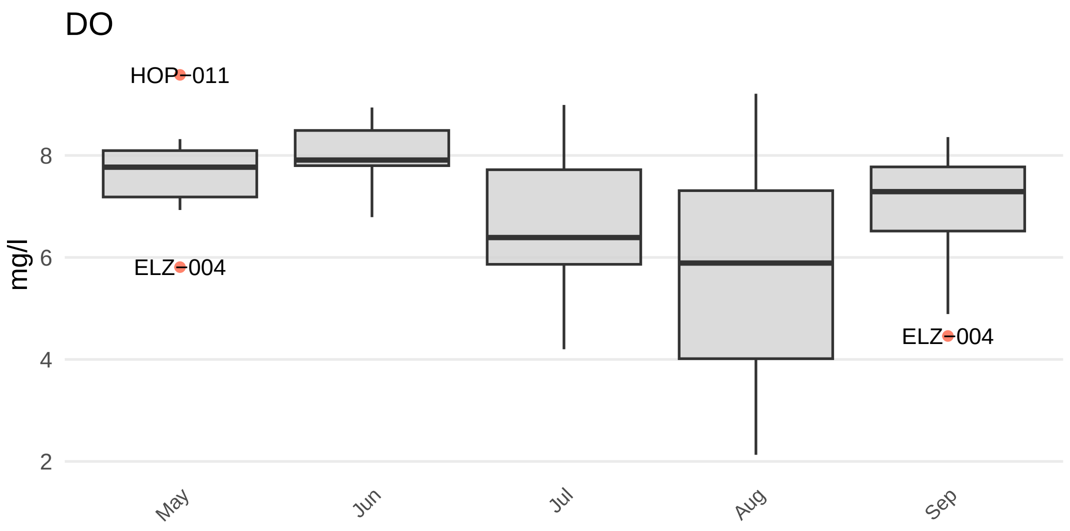
The object `fsetls` can then be used as input to downstream functions.

A final note about the data inputs is that many types of water quality measurements can be included for analysis, although the package is currently limited to working with discrete samples as compared to data from continuous monitoring equipment. The `paramsMWR` dataset included with **MassWateR** provides a complete list of the parameters that can be used with the package. On data import, this list is referenced to ensure that only relevant parameters are included and that appropriate units of measurement are provided. The list includes 43 different parameters, each with multiple valid units of measurement. Additionally, only one unit of measurement is allowed per parameter, which was an intentional design so that tedious functions for converting between dozens of units of measurement did not need to be created during package development. It is not an unreasonable expectation for users to provide only one unit of measurement per parameter. Laboratories typically have a standard reporting format based on the same methodology or instrument used to measure concentrations of water quality parameters.

An additional set of functions can be used to check for outliers in the results file. Outliers can be caused by several reasons, including data entry errors, laboratory processing errors, or anomalous environmental events. Identifying and potentially removing or correcting outliers is a critical part of quality control. The functions in **MassWateR** simply identify potential outliers to provide an opportunity for users to address these values. The decision on how to address outliers remains with the user and no automated tools are provided in the package to correct outliers.

The `anlzMWRoutlier()` function uses the results file and data quality objectives for accuracy to plot potential outliers by month. The accuracy file is used to automatically identify the y-axis scaling (as arithmetic or log) and to replace concentrations with appropriate values for those labelled as beyond detection. Outliers are defined using the standard definition of 1.5 times the interquartile range (the 25th to 75th percentile) of a parameter and can be visually identified as points above or below the whiskers in the boxplots. The station name for a point is also shown. The `param` specifies the water quality parameter to evaluate and the `group` argument specifies how the boxplots are grouped (by year, month, week, or station). In the following example, three stations are identified as potential outliers for the respective month.

```
anlzMWRoutlier(fset = fsetls, param = "DO", group = "month")
```



The outliers in the above plot can also be viewed as tabular output using `outliers = TRUE` to aid in their identification.

```
anlzMWRoutlier(fset = fsetls, param = "DO", group = "month", outliers = TRUE)
```

```
#> # A tibble: 3 x 6
#>   `Monitoring Location ID` `Activity Start Date` `Activity Start Time`
#>   <chr>                    <date>                <chr>
#> 1 ELZ-004                  2022-05-15            06:50
#> 2 HOP-011                  2022-05-15            06:55
#> 3 ELZ-004                  2022-09-11            07:20
#> # i 3 more variables: `Characteristic Name` <chr>, `Result Value` <dbl>,
#> #   `Result Unit` <chr>
```

A user can also evaluate outliers for every water quality parameter in the results file. The anlzMWRoutlierall() function creates a Word document with images of boxplots for every parameter created with anlzMWRoutlier(). Images for each parameter can also be created as standalone files. The param argument does not need to be specified because all parameters are identified in the results file and processed accordingly. The following shows how to use the function by creating a single Word file or individual image files in the working directory.

```
# create word output
anlzMWRoutlierall(fset = fsetls, group = 'month', format = 'word', output_dir = getwd())

# create png output
anlzMWRoutlierall(fset = fsetls, group = 'month', format = 'png', output_dir = getwd())
```

As in Figure 2, a user can identify outliers from the results, modify the file in Excel, and import the file again for further QC reporting or analysis.

Quality Control reporting

The quality control functions in **MassWateR** are designed to create a single report that compares the water quality data in the results file (resdat) to data quality objectives in the accuracy (accdat) and frequency and completeness (frecomdat) files. In general, the QC checks for accuracy evaluate if laboratory and field duplicates, blanks, or spikes are within acceptable ranges. The QC checks for frequency and completeness evaluate if a sufficient number of records in the results file satisfy the accuracy checks and that sufficient QC data have been collected. The values in the accdat and frecomdat inputs are collectively described as data quality objectives (DQOs), such that the QC samples in resdat must satisfy these objectives to be considered accurate and precise data for use in regulatory or other assessments by water quality professionals.

The qcMWRreview() function creates a single QC report as a Word document that evaluates all data in the results file using the DQOs in the accdat and frecomdat input files. This file includes several tables created by individual **MassWateR** functions, where each describe relevant evaluations for the QC assessment. The file is created as follows, which typically requires less than a minute to complete and is followed by a message in the console indicating the report was successfully created and the path where the Word file is located. A user can then further edit the Word document as needed. The first two pages of the QC report are shown in Figure 3.

```
qcMWRreview(fset = fsetls, output_dir = '~/Desktop/)
```

Analysis

Data submission

Building a community and future work

All developers hope that their package is used by the intended audience. To ensure this for **MassWateR**, a community of practice was...

Building the community of practice...

MassWater in other locations, expansion elsewhere

Inclusion of historical data

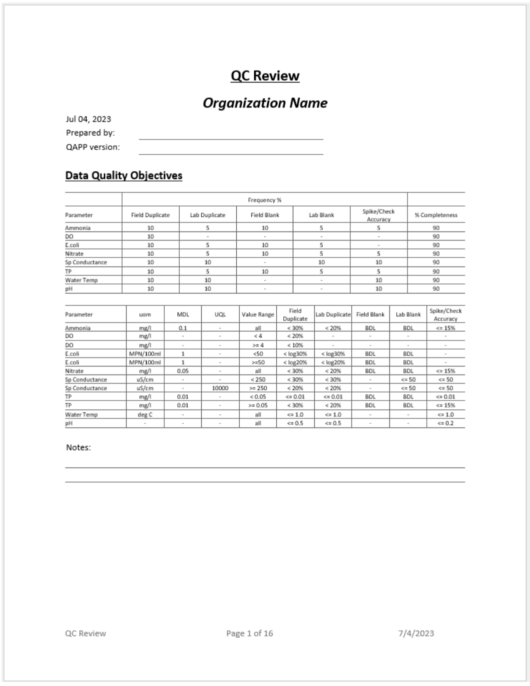
Inclusion for continuous monitoring data, consider outlier anlaysis, drift, biofouling, etc.

Summary

Acknowledgments

References

Allaire, JJ, Yihui Xie, Christophe Dervieux, Jonathan McPherson, Javier Luraschi, Kevin Ushey, Aron Atkins, et al. 2023. *Rmarkdown: Dynamic Documents for r*. https://github.com/rstudio/rmarkdown.



QC Review

Organization Name

Jul 04, 2023

Prepared by: ______________________

QAPP version: ______________________

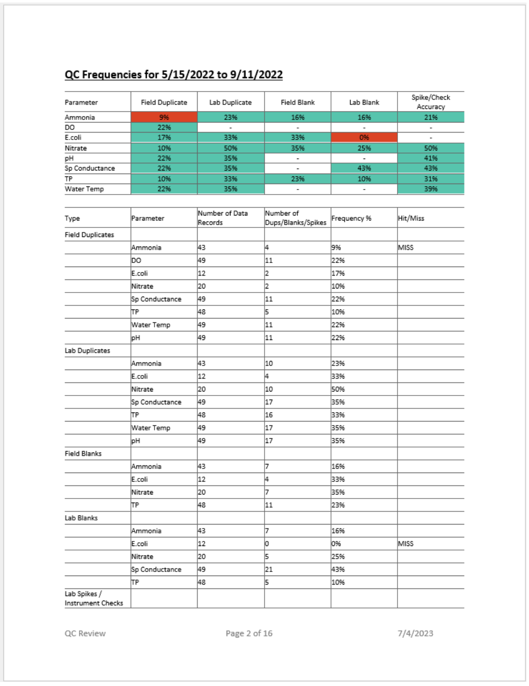
Data Quality Objectives

	Frequency %					
Parameter	Field Duplicate	Lab Duplicate	Field Blank	Lab Blank	Spike/Check Accuracy	% Completeness
Ammonia	10	5	10	5	5	90
DO	10	-	-	-	-	90
E.coli	10	5	10	5	-	90
Nitrate	10	5	10	5	5	90
Sp Conductance	10	10	-	10	10	90
TP	10	5	10	5	5	90
Water Temp	10	10	-	-	10	90
pH	10	10	-	-	10	90

Parameter	uom	MDL	UQL	Value Range	Field Duplicate	Lab Duplicate	Field Blank	Lab Blank	Spike/Check Accuracy
Ammonia	mg/l	0.1	-	all	< 30%	< 20%	BDL	BDL	<= 15%
DO	mg/l	-	-	< 4	< 20%	-	-	-	-
DO	mg/l	-	-	>= 4	< 10%	-	-	-	-
E.coli	MPN/100ml	1	-	<50	< log30%	< log30%	BDL	BDL	-
E.coli	MPN/100ml	1	-	>=50	< log20%	< log20%	BDL	BDL	-
Nitrate	mg/l	0.05	-	all	< 30%	< 20%	BDL	BDL	<= 15%
Sp Conductance	uS/cm	-	-	< 250	< 30%	< 30%	-	<= 50	<= 50
Sp Conductance	uS/cm	-	10000	>= 250	< 20%	< 20%	-	<= 50	<= 50
TP	mg/l	0.01	-	< 0.05	<= 0.01	<= 0.01	BDL	BDL	<= 0.01
TP	mg/l	0.01	-	>= 0.05	< 30%	< 20%	BDL	BDL	<= 15%
Water Temp	deg C	-	-	all	<= 1.0	<= 1.0	-	-	<= 1.0
pH	-	-	-	all	<= 0.5	<= 0.5	-	-	<= 0.2

Notes:

QC Review Page 1 of 16 7/4/2023



QC Frequencies for 5/15/2022 to 9/11/2022

Parameter	Field Duplicate	Lab Duplicate	Field Blank	Lab Blank	Spike/Check Accuracy
Ammonia	9%	23%	16%	16%	21%
DO	22%	-	-	-	-
E.coli	17%	33%	33%	0%	-
Nitrate	10%	50%	35%	25%	50%
pH	22%	35%	-	-	41%
Sp Conductance	22%	35%	-	43%	43%
TP	10%	33%	23%	10%	31%
Water Temp	22%	35%	-	-	39%

Type	Parameter	Number of Data Records	Number of Dups/Blanks/Spikes	Frequency %	Hit/Miss
Field Duplicates					
	Ammonia	43	4	9%	MISS
	DO	49	11	22%	
	E.coli	12	2	17%	
	Nitrate	20	2	10%	
	Sp Conductance	49	11	22%	
	TP	48	5	10%	
	Water Temp	49	11	22%	
	pH	49	11	22%	
Lab Duplicates					
	Ammonia	43	10	23%	
	E.coli	12	4	33%	
	Nitrate	20	10	50%	
	Sp Conductance	49	17	35%	
	TP	48	16	33%	
	Water Temp	49	17	35%	
	pH	49	17	35%	
Field Blanks					
	Ammonia	43	7	16%	
	E.coli	12	4	33%	
	Nitrate	20	7	35%	
	TP	48	11	23%	
Lab Blanks					
	Ammonia	43	7	16%	
	E.coli	12	0	0%	MISS
	Nitrate	20	5	25%	
	Sp Conductance	49	21	43%	
	TP	48	5	10%	
Lab Spikes / Instrument Checks					

QC Review Page 2 of 16 7/4/2023

Figure 3: The first two pages of the quality control report that evaluates the results data relative to data quality objectives. The first page shows the data quality objectives for accuracy, frequency, and completeness. The second page shows QC results for frequency and completeness. Parameters shown in red or markded as 'MISS' failed the data quality objectives. Users can edit the Word file as needed, e.g., entering the organization name or adding additional notes.

Arndt, Julia, Julia S Kirchner, Kevin S Jewell, Michael P Schluesener, Arne Wick, Thomas A Ternes, and Lars Duester. 2022. "Making Waves: Time for Chemical Surface Water Quality Monitoring to Catch up with Its Technical Potential." *Water Research* 213: 118168. https://doi.org/10.1016/j.watres.2022.118168.

Behmel, Sonja, Mathieu Damour, Ralf Ludwig, and MJ Rodriguez. 2016. "Water Quality Monitoring Strategies—a Review and Future Perspectives." *Science of the Total Environment* 571: 1312–29. https://doi.org/10.1016/j.scitotenv.2016.06.235.

Buytaert, Wouter, Zed Zulkafli, Sam Grainger, Luis Acosta, Tilashwork C Alemie, Johan Bastiaensen, Bert De Bièvre, et al. 2014. "Citizen Science in Hydrology and Water Resources: Opportunities for Knowledge Generation, Ecosystem Service Management, and Sustainable Development." *Frontiers in Earth Science* 2: 26. https://doi.org/10.3389/feart.2014.00026.

De Cicco, Laura A., David Lorenz, Robert M. Hirsch, William Watkins, and Mike Johnson. 2022. *dataRetrieval: R Packages for Discovering and Retrieving Water Data Available from u.s. Federal Hydrologic Web Services* (version 2.7.12). Reston, VA: U.S. Geological Survey; U.S. Geological Survey. https://doi.org/10.5066/P9X4L3GE.

Kelly-Quinn, M, JN Biggs, S Brooks, P Fortuño, S Hegarty, JI Jones, and F Regan. 2022. "Opportunities, Approaches and Challenges to the Engagement of Citizens in Filling Small Water Body Data Gaps." *Hydrobiologia*, 1–21. https://doi.org/10.1007/s10750-022-04973-y.

Kumpel, Emily, Clara MacLeod, Kara Stuart, Alicea Cock-Esteb, Ranjiv Khush, and Rachel Peletz. 2020. "From Data to Decisions: Understanding Information Flows Within Regulatory Water Quality Monitoring Programs." *Npj Clean Water* 3 (1): 38. https://doi.org/10.1038/s41545-020-00084-0.

Schiff, Ken, PR Trowbridge, ET Sherwood, Peter Tango, and Rich A Batiuk. 2016. "Regional Monitoring Programs in the United States: Synthesis of Four Case Studies from Pacific, Atlantic, and Gulf Coasts." *Regional Studies in Marine Science* 4: A1–7. https://doi.org/10.1016/j.rsma.2015.11.007.

Stein, Eric D, and Donald B Cadien. 2009. "Ecosystem Response to Regulatory and Management Actions: The Southern California Experience in Long-Term Monitoring." *Marine Pollution Bulletin* 59 (4-7): 91–100. https://doi.org/10.1016/j.marpolbul.2009.02.025.

Tango, Peter J, and Richard A Batiuk. 2013. "Deriving Chesapeake Bay Water Quality Standards." *JAWRA Journal of the American Water Resources Association* 49 (5): 1007–24. https://doi.org/10.1111/jawr.12108.

Wickham, Hadley. 2016. *Ggplot2: Elegant Graphics for Data Analysis*. Springer-Verlag New York.

https://ggplot2.tidyverse.org.

Wickham, Hadley, and Jennifer Bryan. 2023. *Readxl: Read Excel Files*. https://CRAN.R-project.org/package=readxl.

Wilde, Franceska D., and U.S. Geological Survey. 2002. "Chapter A5. Processing of Water Samples." Techniques of Water-Resources Investigations 09-A5. Version 2.2, Revised February 2009. Reston, VA: U.S. Geological Survey. https://doi.org/10.3133/twri09A5.

Xie, Yihui. 2015. *Dynamic Documents with R and Knitr*. 2nd ed. Boca Raton, Florida: Chapman; Hall/CRC. https://yihui.org/knitr/.

Marcus W. Beck
Tampa Bay Estuary Program
263 13th Ave S
St. Petersburg, Florida, USA 33701
https://tbep.org
ORCiD: 0000-0002-4996-0059
mbeck@tbep.org

Benjamen Wetherill
ACASAK Consulting
Boston, Massachusetts, USA
https://www.acasak.com/
ORCiD: 0000-0002-0912-0225
bwetherill@acasak.co

Jillian Carr
Massachusetts Bays National Estuary Partnership
University of Massachusetts Boston, 100 Morrissey Blvd
Boston, Massachusetts, USA 02125
https://www.mass.gov/orgs/massachusetts-bays-national-estuary-partnership
Jillian.Carr@umb.edu

Pamela DiBona
Massachusetts Bays National Estuary Partnership
University of Massachusetts Boston, 100 Morrissey Blvd
Boston, Massachusetts, USA 02125
https://www.mass.gov/orgs/massachusetts-bays-national-estuary-partnership
pamela.dibona@state.ma.us