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 (MADEP) and data submitted to the national WQX database. Although the initial conception of **MassWateR** was to address

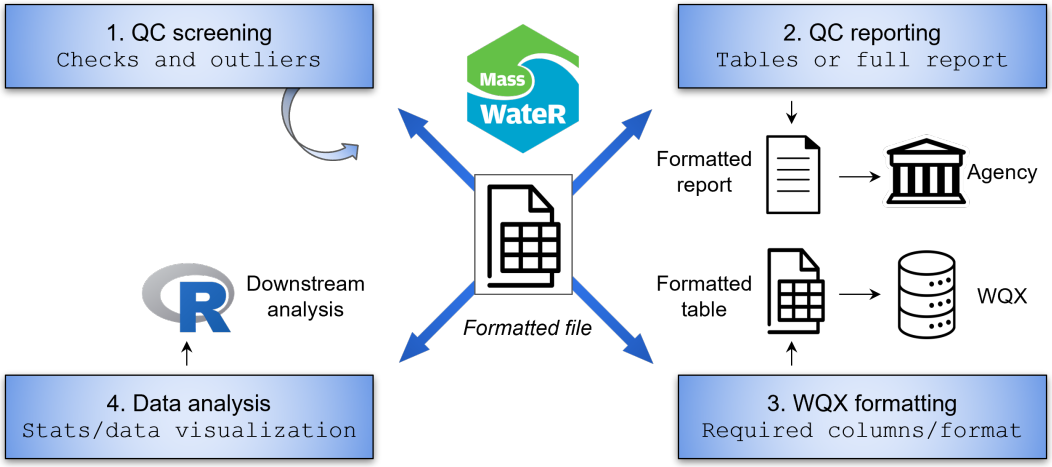


Figure 1: Workflow demonstrating how a user could engage with the **MassWateR** package. A user can apply one to any of the four steps depending on their need. The first step, QC screening, is often iterative as a user can modify parts of the raw data based on input checks or outliers. The second step 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.

regional 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 1, 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 1: File requirements for using MassWateR. Check marks indicate which file is required for each part of the MassWateR workflow.

Formatted file	Description	QC screening	QC reporting	WQX formatting	Data analysis
Results	Water quality results organized by sample location and date	✓	✓	✓	✓
DQO accuracy	Summary of data quality objectives that describe quality control accuracy for data in the results file	✓	✓	✓	✓
DQO frequency and completeness	Summary of data quality objectives that describe quality control frequency and completeness measures for data in the results file	✓	✓		
Sites	A site metadata file, including location names, latitude, longitude, and additional grouping factors for sites in the results file	✓		✓	✓
WQX metadata	A wqx metadata file required for generating output to facilitate data upload to WQX			✓	

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 (e.g., `readMWR`).

- `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

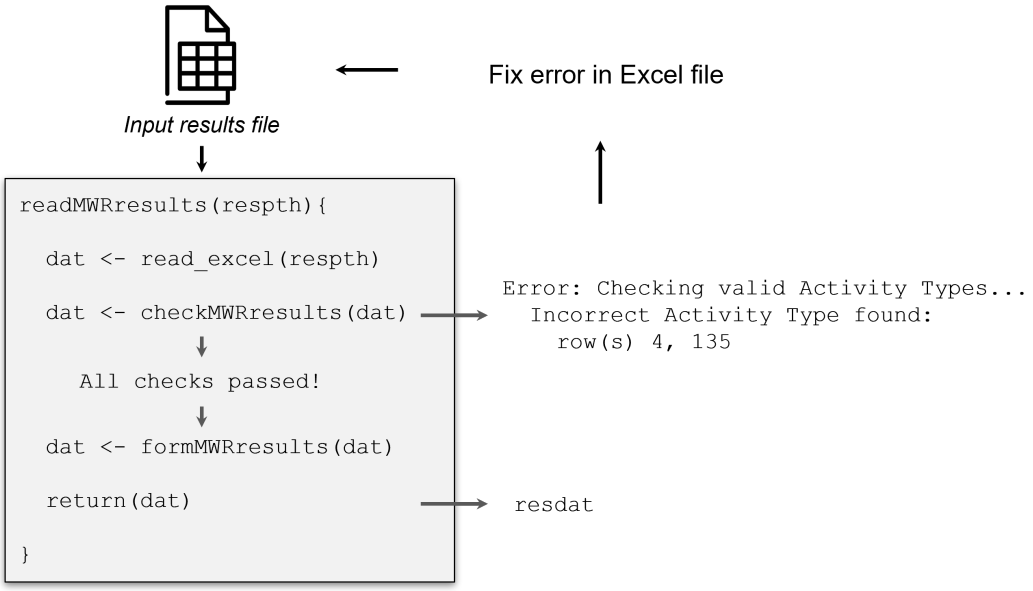


Figure 2: Pseudocode demonstrating the iterative process of importing a required data file for MassWateR. All read functions import an Excel file and the imported file is then passed to a check function. The function exits if an error is encountered, allowing the user to manually fix the identified error 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.

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)

# 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)
```

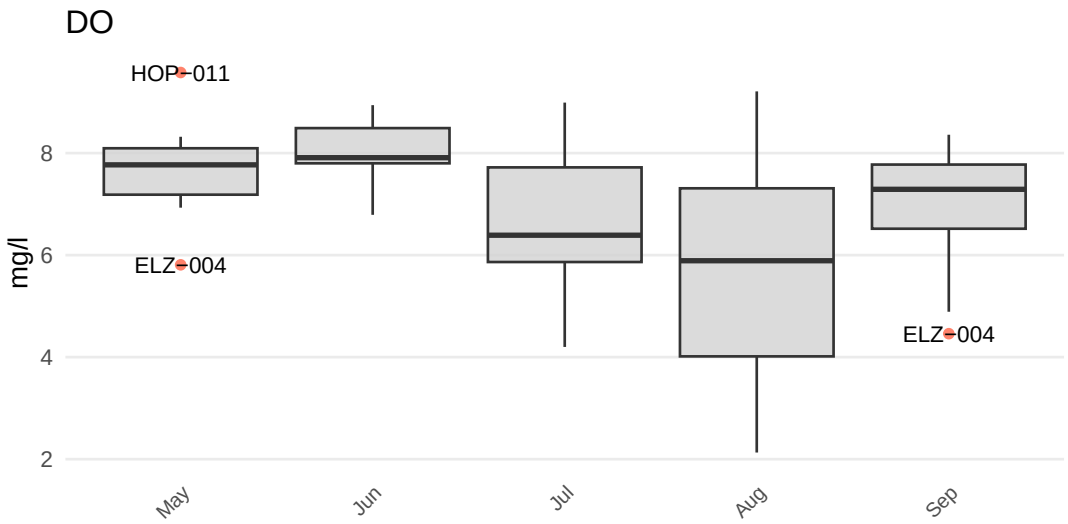
The object `fsetls` can then be used as input to downstream functions.

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.

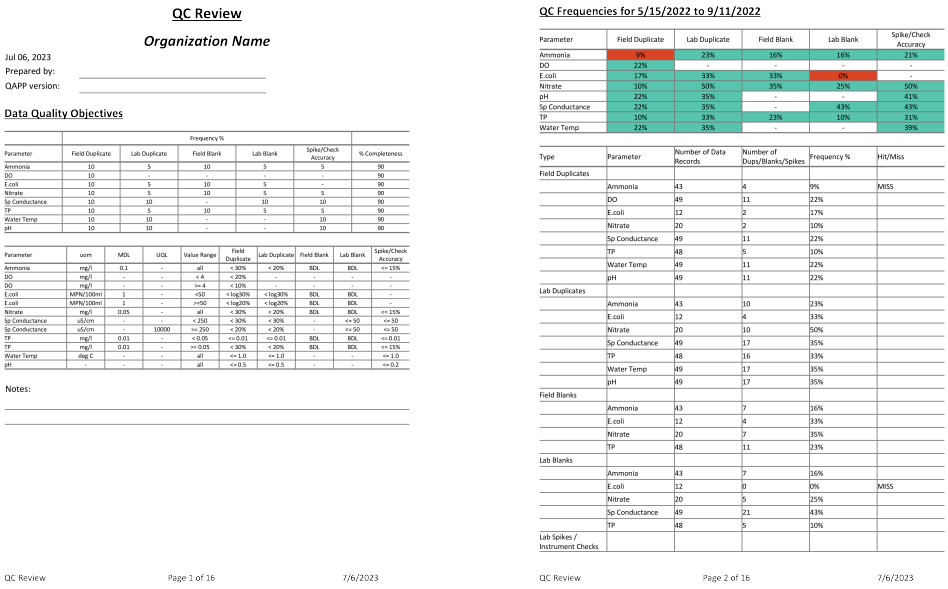
```
# create word output
anlzMWRoutlierall(fset = fsetls, group = 'month', format = 'word',
  output_dir = getwd())

# 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



QC Review

Organization Name

Jul 06, 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 7/6/2023

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 7/6/2023

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 marked as 'MISS' failed the data quality objectives. Users can edit the Word file as needed, e.g., entering the organization name or adding notes.

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 = getwd())
```

The QC report is built using several functions that can be used individually as needed. In particular, the `tabMWRacc()`, `tabMWRfre()`, and `tabMWRcom()` create **flextable** (Gohel and Skintzos 2023) table objects that can be viewed in RStudio and are compatible with Microsoft Word output. These functions are useful for understanding how the QC checks are created for the separate components of the QC report. For example, the `tabMWRacc()` function evaluates accuracy checks for field duplicates, lab duplicates, field blanks, lab blanks, and lab spikes/instrument checks for QC records in the results file based on DQOs in the accuracy file. The function can return a summary of all checks as follows:

```
tabMWRacc(fset = fsetls, type = "summary")
```

Type	Parameter	Number of QC Checks	Number of Misses	% Acceptance
Field Duplicates				
	Ammonia	4	1	75 %
	DO	11	0	100 %
	E.coli	2	0	100 %
	Nitrate	2	0	100 %
	pH	11	0	100 %
	Sp Conductance	11	0	100 %
	TP	5	1	80 %

Type	Parameter	Number of QC Checks	Number of Misses	% Acceptance
	Water Temp	11	0	100 %
Lab Duplicates				
	Ammonia	10	0	100 %
	E.coli	4	0	100 %
	Nitrate	10	0	100 %
	pH	17	1	94 %
	Sp Conductance	17	0	100 %
	TP	16	0	100 %
	Water Temp	17	0	100 %
Field Blanks				
	Ammonia	7	0	100 %
	E.coli	4	0	100 %
	Nitrate	7	0	100 %
	TP	11	1	91 %
Lab Blanks				
	Ammonia	7	1	86 %
	E.coli	0	-	-
	Nitrate	5	0	100 %
	Sp Conductance	21	1	95 %
	TP	5	0	100 %
Lab Spikes / Instrument Checks				
	Ammonia	9	0	100 %
	Nitrate	10	1	90 %
	pH	20	1	95 %
	Sp Conductance	21	0	100 %
	TP	15	0	100 %
	Water Temp	19	1	95 %

The table shows the types and amounts of QC checks applied to each parameter and which of those checks did not satisfy the DQOs in the accuracy file. For example, there were four field duplicate records for ammonia in the results file and only one of those records included a duplicate value outside of the acceptable range, i.e., there was a 75% acceptance rate for the four field duplicate records. This same table can be viewed relative to the applicable frequency rate for the amount of passing checks considered to be appropriate for the QC records as identified in the frequency and completeness DQO file.

```
tabMWRacc(fset = fsetls, type = "percent")
```

Parameter	Field Duplicate	Lab Duplicate	Field Blank	Lab Blank	Spike/Check Accuracy
Ammonia	75%	100%	100%	86%	100%
DO	100%	-	-	-	-
E.coli	100%	100%	100%	-	-
Nitrate	100%	100%	100%	100%	90%
pH	100%	94%	-	-	95%

Parameter	Field Duplicate	Lab Duplicate	Field Blank	Lab Blank	Spike/Check Accuracy
Sp Conductance	100%	100%	-	95%	100%
TP	80%	100%	91%	100%	100%
Water Temp	100%	100%	-	-	95%

The cells in red showed that four of the checks for three of the parameters did not include a sufficient amount of QC records that passed the relevant accuracy checks. All other parameters and checks in green had enough QC records to satisfy the DQOs. Empty cells include checks where no QC records were available in the results file, e.g., no lab duplicate for dissolved oxygen. The empty cells are typically for QC checks that do not readily apply to a parameter. For example, dissolved oxygen is measured in the field with monitoring equipment with no lab processing, such that lab QC checks are not relevant.

Detailed information about the QC checks for an individual parameter can be obtained by changing the arguments to any of the `tab` functions. For example, the results for every field blank check and every parameter can be obtained as follows by changing the `type` and `accchk` arguments. This is the same information that is summarized using `type = "summary"` or `type = "percent"`.

Table 4: Field Blanks

Parameter	Date	Site	Result	Threshold	Hit/Miss
Ammonia					
	2022-05-15		BDL	0.1 mg/l	
	2022-06-12		BDL	0.1 mg/l	
	2022-07-17		BDL	0.1 mg/l	
	2022-07-17		BDL	0.1 mg/l	
	2022-08-14		BDL	0.1 mg/l	
	2022-08-14		BDL	0.1 mg/l	
	2022-09-11		BDL	0.1 mg/l	
E.coli					
	2022-06-13		BDL	1 MPN/100ml	
	2022-07-18		BDL	1 MPN/100ml	
	2022-08-01		BDL	1 MPN/100ml	
	2022-08-29		BDL	1 MPN/100ml	
Nitrate					
	2022-05-15		BDL	0.05 mg/l	
	2022-06-12		BDL	0.05 mg/l	
	2022-06-12		BDL	0.05 mg/l	
	2022-07-17		BDL	0.05 mg/l	
	2022-07-17		BDL	0.05 mg/l	
	2022-08-14		BDL	0.05 mg/l	
	2022-09-11		BDL	0.05 mg/l	
TP					
	2022-05-15		BDL	0.01 mg/l	
	2022-05-15		BDL	0.01 mg/l	
	2022-06-12		BDL	0.01 mg/l	
	2022-06-12		BDL	0.01 mg/l	
	2022-07-17		BDL	0.01 mg/l	

Table 4: Field Blanks

Parameter	Date	Site	Result	Threshold	Hit/Miss
	2022-07-17		BDL	0.01 mg/l	
	2022-07-17		0.01 mg/l	0.01 mg/l	MISS
	2022-08-14		BDL	0.01 mg/l	
	2022-08-14		BDL	0.01 mg/l	
	2022-09-11		BDL	0.01 mg/l	
	2022-09-11		BDL	0.01 mg/l	

The example shows that all but one field blank passed the checks, where most were below the detection limit (BDL) of the laboratory equipment. Only one sample sample for total phosphorus (TP) was at or above the threshold on July 17th.

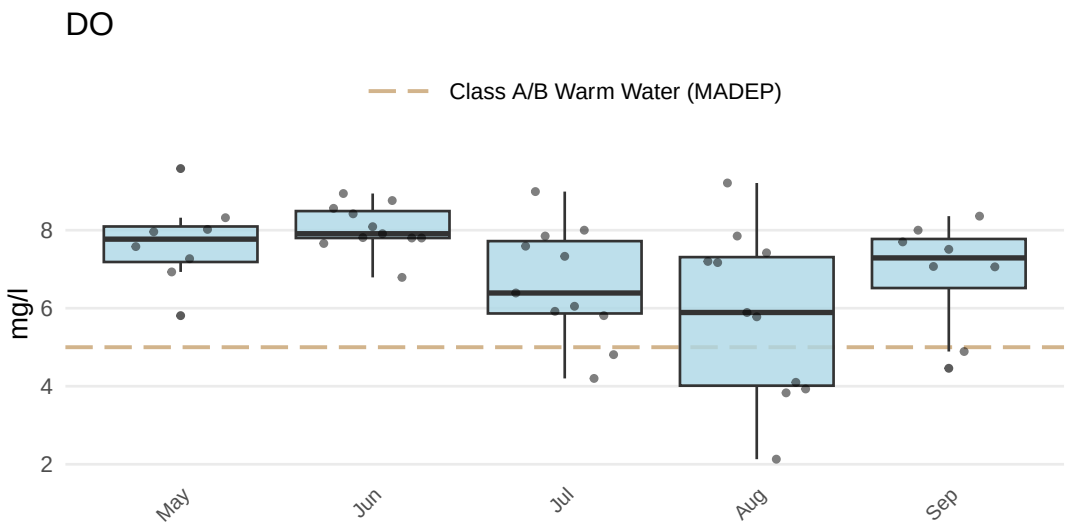
Additional functions for creating QC tables include `tabMWRfre()` for assessing if a sufficient number of QC samples are available and `tabMWRcom()` for assessing completeness checks that compare the number of regular samples (field measurements or lab samples) to the number of qualified samples. As with all the QC functions, these can be used individually within an R session to quickly view QC results or the `qcMWRreview()` function can be used to create a complete report that combines results from the individual functions. The summarized report can then be submitted to an appropriate regulatory agency for review to ensure that any submitted datasets fulfill appropriate data quality objectives.

Analysis

The analysis functions in **MassWateR** provide a simple approach to quickly evaluate data in the results file. Although several base R functions and supporting packages can be used to develop individual assessments, it was recognized that many users will not be comfortable developing their own custom visualization routines. As such, the analysis functions were designed for a rapid overview of the results that required minimal decisions by the user to create the output. Several default settings described below allow further customization of the output as desired by the user. The four primary analysis functions can be used to analyze seasonal trends, trends by date, date by site, and spatially using maps.

Seasonal trends can be evaluated using the `anlzMWRseason()` function that summarizes results for a single parameter using boxplots or barplots with seasonal groups assigned to months or weeks of the year. Boxplots or barplots can also include jittered points of the observations on top or only the jittered points can be shown. Below demonstrates a jittered boxplot of dissolved oxygen observations by month.

```
anlzMWRseason(fset = fsetls, param = "DO", thresh = "fresh",
  group = "month", type = "jitterbox")
```

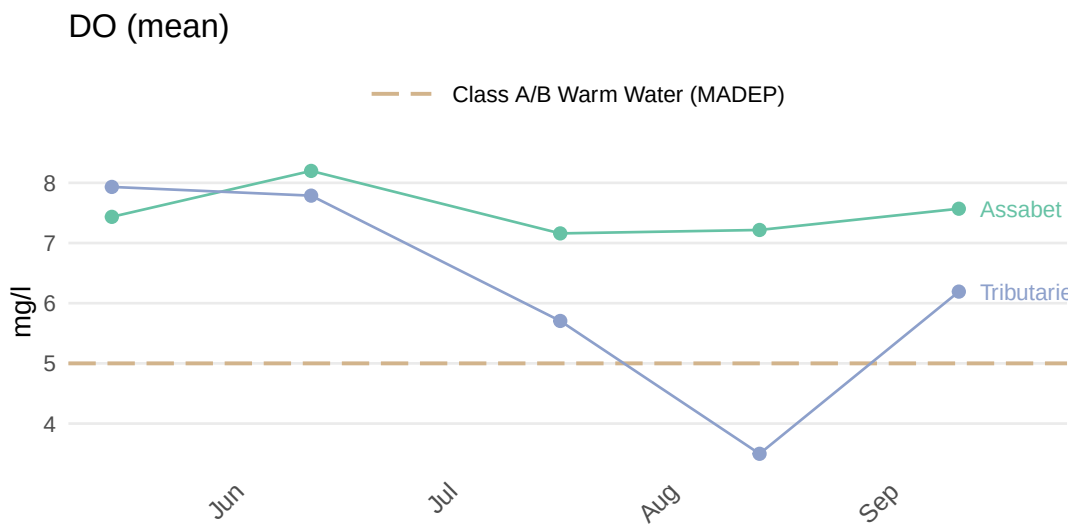


The arguments specify different options available for the plot, including which parameter to plot in the results file (`param`), the type of threshold lines to show (`thresh`), the grouping, and the type of plot (jittered points over boxplots). The `thresh` argument in particular is used to plot relevant thresholds of interest based on the parameter and if the samples are from freshwater or marine sites. The `thresholdMWR` data object included with the package includes thresholds for ten common parameters that are either regulatory standards or recommended safe limits for different designated uses in freshwater or marine environments. Each threshold includes a source that is displayed in the legend above the plot. In the above example, the threshold line applies to class A or B freshwater environments that support warmwater fisheries, as defined by MADEP. Threshold lines can be suppressed (`thresh = "none"`) or user-specific thresholds can de added using additional **ggplot2** functionality (i.e., with `geom_hline()`).

The y-axis scaling of the plot is also determined automatically from the data inputs. The DQO file for accuracy includes information on the distribution of each parameter, i.e., parameters with "log" in any of the columns are plotted on log10-scale, otherwise arithmetic. This behavior is controlled by the `yscl` argument, where the default is `"auto"` that indicates information on scaling is obtained automatically from the DQO file. Setting `yscl = "linear"` or `yscl = "log"` will set the axis as linear or log10-scale, respectively, regardless of the information in the DQO file.

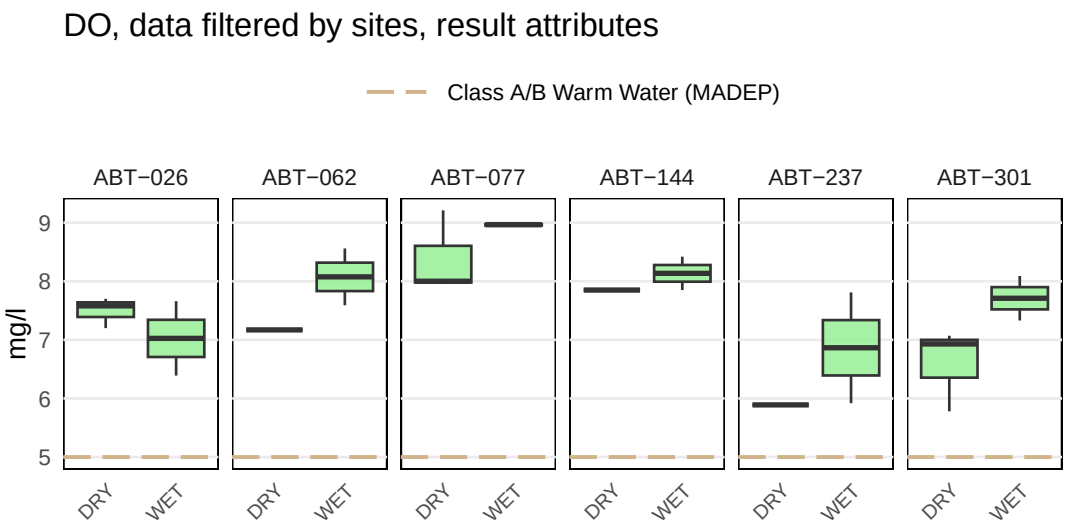
The `anlzMWRdate()` function plots results continuous in time on the date the samples or measurements were collected. This information can be use evaluate how samples have changed at individual sites over time or as aggregate samples across sites. In the former case, individual points and lines are used for each site with appropriate labels. For the latter case, sites are aggregated and appropriate summary statistics are shown as the mean with 95% confidence intervals. Sites can be aggregated by sample dates or by "location groups" in the site metadata file. For example, all sites along the same river or tributary can be aggregated. The following demonstrates an aggregation of sites by the mainstem of the Assabet river and its tributaries using the `group = "locgroup"` argument. A user can specify any desired grouping in the `Location Group` column of the site metadata file.

```
anlzMWRdate(fset = fsetls, param = 'DO', group = 'locgroup',
  thresh = 'fresh')
```



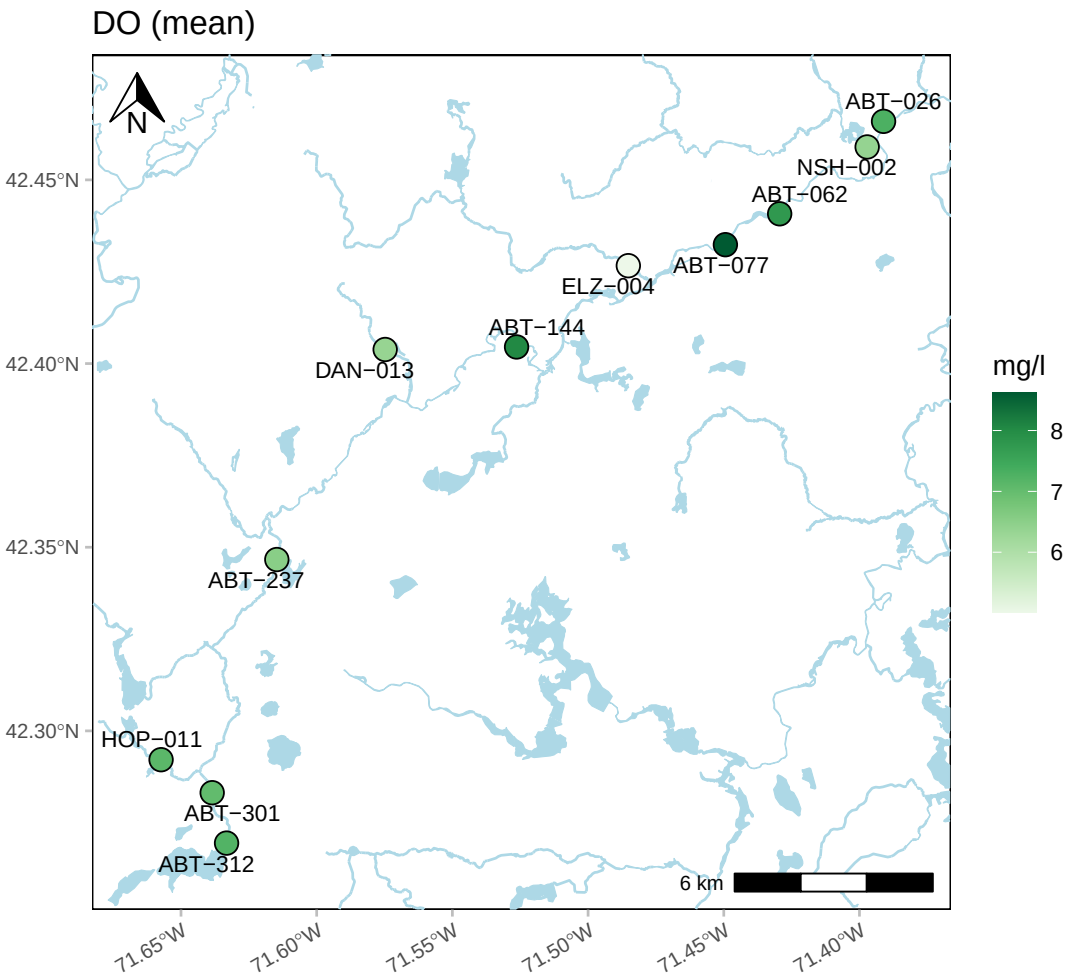
The `anlzMWRsite()` function can be used to view distributions of a parameter across sites. This function summarizes results for a single parameter using boxplots or barplots separately for each site on the x-axis. Boxplots or barplots can also include jittered points of the observations on top or only the jittered points can be shown. Results at each site can also be grouped by the `Result Attribute` column in the results file, where a user can enter arbitrary grouping criteria for samples. For example, sites can be grouped by wet or dry conditions if this information is included in the results file. Individual sites can also be specified with the `site` argument.

```
anlzMWRsite(fset = fsetls, param = "DO", thresh = "fresh", type = "box",
  site = c("ABT-026", "ABT-062", "ABT-077", "ABT-144", "ABT-237", "ABT-301"),
  resultatt = c('DRY', 'WET'), byresultatt = TRUE)
```



Finally, maps of results at each site can be created with the `anlzMWRmap()` function. Parameter values at each site are summarized with colors indicating relative values and the maps can include various spatial information as reference. For example, hydrologic lines and waterbodies from the National Hydrograpy Dataset (NHD) can be shown with varying level of detail and base maps can be included from the **ggmap** package (Kahle and Wickham 2013). NHD data were included to provide more specific information on hydrologic features of interest because of insufficient detail in basemaps. These datasets are available from an external source and clipped to an approximate bounding box for the selected stations. Currently, only flowlines and waterbodies for Massachusetts are included, although examples are provided in a package vignette for including custom shapefiles. The below example shows mean dissolved oxygen concentrations using the medium level of detail for the NHD maps (`addwater = "medium"`).

```
anlzMWRmap(fset = fsetls, param = "DO", addwater = "medium")
```



Samples and measurements across dates can be aggregated differently using the `sumfun` argument. If `sumfun = "auto"` (default), the mean is used where the distribution is determined automatically from the DQO file for accuracy, i.e., parameters with `"log"` in any of the columns are summarized with the geometric mean, otherwise arithmetic. Any other valid R summary function will be applied if passed to `sumfun` (`"mean"`, `"geomean"`, `"median"`, `"min"`, `"max"`). This argument also applies to other analysis functions where the data can be aggregated across dates or locations.

Data submission

The last part of the **MassWateR** workflow is preparation of data for submission to WQX. A single function, `tabMWRwqx()`, is provided to format data inputs using a template designed for data submission. This function will export a single Excel workbook with three sheets, named "Project", "Locations", and "Results", all of which are required for data submission. The output is populated with as much content as possible based on information in the input files. The remainder of the information not included in the output will need to be manually entered before uploading the data to WQX. All required columns are present, but individual rows must be verified for completeness and accuracy by the user before uploading the data.

The workbook can be created as follows by including the required files and specifying an output directory where the Excel file is saved. Once the function is done running, a message indicating success and where the file is located is returned.

```
tabMWRwqx(fset = fsetls, output_dir = getwd())
#> Excel workbook created successfully! File located at /tmp/RtmpBrk9nw/wqxtab.xlsx
```

Additional templates and instructions for data submission are available on the package website. It is assumed that users are familiar with the WQX data submission portal if they are already using the package. **MassWateR** eliminates the need to format the data by hand and it is expected to increase the amount of data made available on the WQP (via WQX upload) as the user base for the package increases.

Building a community and future work

To ensure that **MassWateR** is a known resource for potential users and to encourage its use for QC reporting, analysis, and data submission, a community of practice was established during package development and following its initial release on CRAN. This included several beta testing and training workshops to gather feedback on anticipated data analysis workflows and to educate potential users on appropriate use of the package. Many users have not previously been exposed to R for data analysis and a substantial portion of the trainings included an introduction to the software, as well as use of **MassWateR**. Emphasis was placed on simple use of the core functions, as opposed to developing custom workflows that combined core R functions with **MassWateR**. However, users were encouraged to learn how to extend the use of the package by leveraging additional R packages, such as **ggplot2** to create additional visualizations for modifying the existing analysis plots in **MassWateR**. For example, a vignette was included on the package website to demonstrate how these plots can be modified for custom output using **ggplot2**. Further, a community forum was created as a resource for users to post questions about using the package and for others to view the discussion if similar issues were encountered. This approach followed the model used by other popular web forums (e.g., StackOverflow) for troubleshooting software issues by minimizing duplicative issues through sharing solutions in a public forum.

As noted above, **MassWateR** was developed to meet specific needs of water quality professionals in Massachusetts, but the principles for QC reporting, analysis, and WQX data submission are largely universal and the package can be used outside of the state as long as the following minor limitations are addressed. First, some of the parameter thresholds applied to the analysis plots are unique to Massachusetts, whereas others apply more broadly, such as those defined using standards from the USEPA. Users can simply omit the thresholds if they do not apply using `thresh = "none"` or add custom thresholds using standard **ggplot2** functions. Second, the NHD waterbodies used by `anlzMWRmap()` are specific to watersheds in Massachusetts. Users can omit these layers from the plot using `addwater = NULL`, add a base map using the `maptype` argument, or add custom waterbody layers as simple features objects (Pebesma and Bivand 2023) using the `geom_sf()` function from **ggplot2**. Third, the QC report created by `qcMWRreview()` uses a format vetted by MADEP. This format may not meet the requirements of other state organization, although the reporting principles are generally universal and the resulting Word document can easily be manually modified. Finally, minor components of the Excel file for WQX submission created by `tabMWRwqx()` are specific to Massachusetts. This includes the timezone for the `WQX Activity Start Time Zone` field and some default entries for sample collection methods that include `"MassWateR"` in the text. Each of these can also be modified by hand in the output file. Future enhancements and additions to **MassWateR** will likely include automated tools to address these minor issues using the package outside of Massachusetts.

Additional future work to improve the functionality of **MassWateR** is also expected as the user base increases and the functions mature with additional bug fixes or minor enhancements. Specifically, the inclusion of historical data for some of the analysis functions could provide additional context on status and trends for monitoring data at specific locations. This enhancement would require the extraction of existing data included in the WQP, which would not be significantly challenging given the robust web retrieval tools already available. Existing R packages leverage these tools (e.g., **dataRetrieval**) and a similar approach could be used by **MassWateR**. A second and more challenging enhancement would be the ability to work with continuous monitoring data collected at high temporal resolution with equipment deployed *in situ*. These data present additional challenges not encountered with routine samples collected at longer time intervals, including increased data volume and additional QC needs (Wagner et al. 2006; Horsburgh et al. 2015). For the latter, automated tools are needed for detecting and handling QC issues common with monitoring equipment deployed in the field for long duration, such as instrument drift, biofouling, or missing data. More complex methods for detecting outliers in continuous monitoring data beyond the existing tools in **MassWateR** will also be needed (Leigh et al. 2019).

Summary

The **MassWateR** package represents an important set of functions that are expected to significantly improve how resource managers develop QC assessments and apply exploratory analyses to water quality monitoring data. The package can also expedite the preparation of data for submission to the largest water quality database in the United States, which will likely contribute to the amount of data made available through the WQP. These activities are critically needed to ensure that monitoring data are of sufficient quality and quantity for use in regulatory applications or routine assessment of status and trends of environmental resources. As mentioned throughout this paper, **MassWateR** was developed to address specific needs for resource management professionals in Massachusetts, although

the workflow in Figure 1 can easily be applied to data collected elsewhere. Future development of the package will not only make the package functions more generalizable to other locations, but also provide additional features for working with continuous and historical monitoring data. The community of practice developed for **MassWateR** is expected to grow and the package will be supported by the authors as the user base increases.

Acknowledgments

This work was supported by an Exchange Network grant from the US Environmental Protection Agency awarded to the Massachusetts Bay Estuary Partnership. We thank the early users of **MassWateR** that contributed ideas on improving the package during initial testing.

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.

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.

Gohel, David, and Panagiotis Skintzos. 2023. *Flextable: Functions for Tabular Reporting*. https://CRAN.R-project.org/package=flextable.

Horsburgh, Jeffery S, Stephanie L Reeder, Amber Spackman Jones, and Jacob Meline. 2015. "Open Source Software for Visualization and Quality Control of Continuous Hydrologic and Water Quality Sensor Data." *Environmental Modelling & Software* 70: 32–44. https://doi.org/10.1016/j.envsoft.2015.04.002.

Kahle, David, and Hadley Wickham. 2013. "Ggmap: Spatial Visualization with Ggplot2." *The R Journal* 5 (1): 144–61. https://journal.r-project.org/archive/2013-1/kahle-wickham.pdf.

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.

Leigh, Catherine, Omar Alsibai, Rob J Hyndman, Sevvandi Kandanaarachchi, Olivia C King, James M McGree, Catherine Neelamraju, et al. 2019. "A Framework for Automated Anomaly Detection in High Frequency Water-Quality Data from in Situ Sensors." *Science of the Total Environment* 664: 885–98. https://doi.org/10.1016/j.scitotenv.2019.02.085.

Pebesma, Edzer, and Roger Bivand. 2023. *Spatial Data Science: With applications in R*. Chapman and Hall/CRC. https://r-spatial.org/book/.

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.

Wagner, Richard J, Robert W Boulger Jr, Carolyn J Oblinger, and Brett A Smith. 2006. "Guidelines and Standard Procedures for Continuous Water-Quality Monitors: Station Operation, Record Computation, and Data Reporting." Techniques and Methods 1-D3. Version 1.0. Reston, VA: U.S. Geological Survey. https://doi.org/10.3133/tm1D3.

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