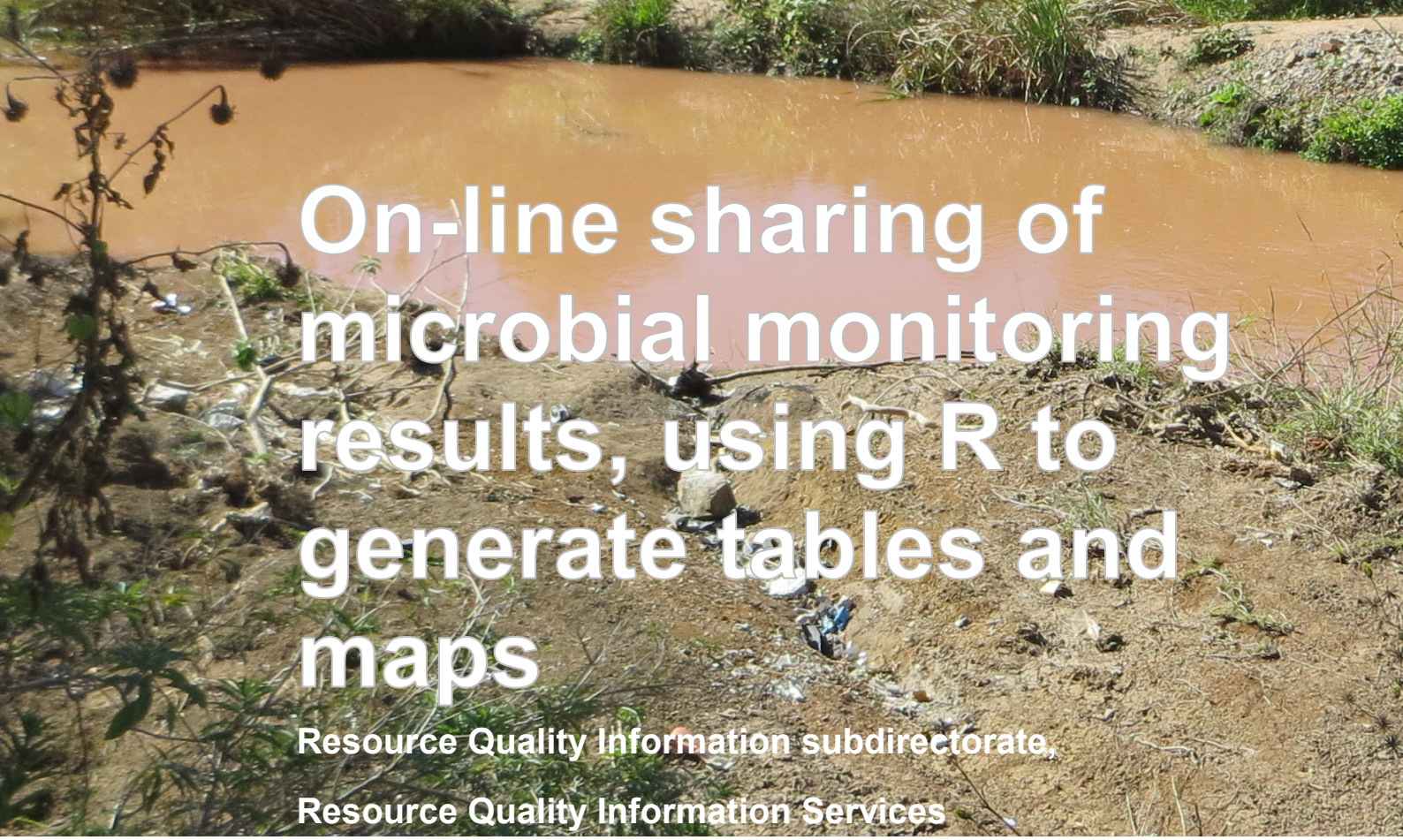
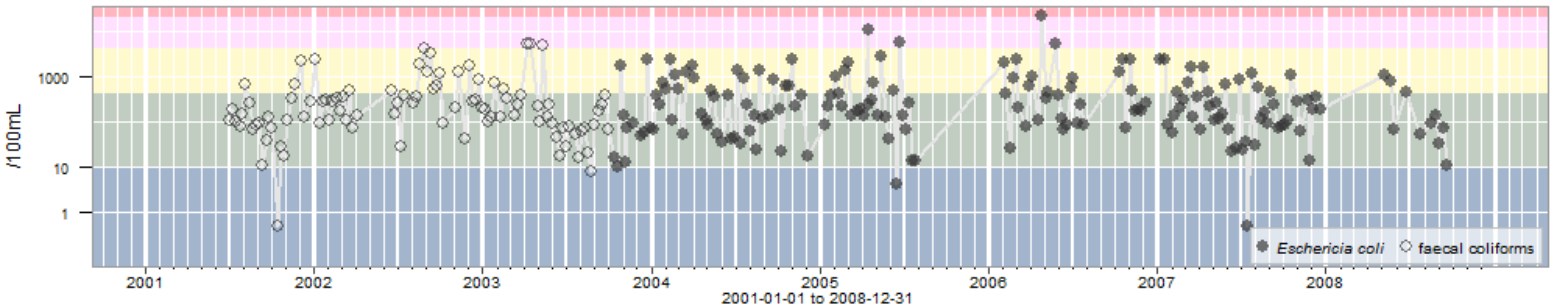


On-line sharing of microbial monitoring results, using R to generate tables and maps

Resource Quality Information subdirectorate,

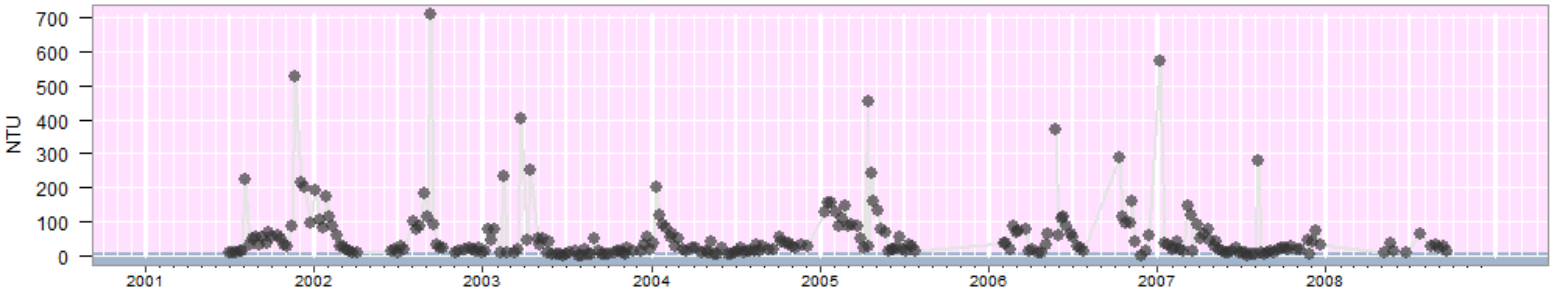
Resource Quality Information Services

Full record for monitoring site 88583,
showing high risk guidelines for indicators of pathogenic bacteria.



2001-01-01 to 2008-12-31
Znah-Newl Nahoon River at Newlands Road Bridge
<=10 low-med risk; >10 high risk when drinking untreated water
>400 high risk from full or partial contact; >4000 high risk when irrigating crops that are eaten raw
>20000 high risk when drinking after only limited treatment

Turbidity



Data from the national water quality database (WMS). Plotted at Resource Quality Information Services, Department of Water and Sanitation, on 2018-12-10 14:42:56 using NMMP_TimeLine.R v1.5 under R version 3.4.4 (2018-03-15)

water & sanitation

Department:
Water and Sanitation
REPUBLIC OF SOUTH AFRICA

DOCUMENT STATUS

WORKING TITLE:	On-line sharing of microbial monitoring results, using R to generate tables and maps
AUTHOR:	Michael Silberbauer
REPORT STATUS:	**First version**
RQS REPORT NUMBER:	N/0000/00/NMMP/2018
DATE:	March 2019

Executive summary

This document is aimed at those who need to maintain and further develop the Internet applications for disseminating information on the DWS website. It is specifically meant for the RQIS webmasters and the National Microbial Monitoring Programme (NMMP) staff.

In 2011, while updating the NMMP Excel spreadsheet used for calculating the microbial programme results (DWAF 2002), it became clear that a programmatic method would be more efficient. This led to the development of an R script to mimic the microbial results layout specified in the implementation manual for the NMMP, including the maps and tables, with simple HTML navigation, first by year, then by two-month period, then by region. Tables are colour-coded according to risk and have links to time-series plots.

The earlier conversion of the national chemical water quality monitoring reporting system to R and ODBC had shown that the Internet is a useful medium for communicating results from monitoring programmes. The distribution of edited spreadsheets by email was a slow, manual process, and the new Internet system provided a quicker turnaround time. The design of the system requires that an automated run of the R script regenerates the entire NMMP website once a week, incorporating any data updates and corrections. NMMP staff can perform manual updates at any time, if required.

The test results showed that the method is a practical means for data and information distribution. It is applicable to other data types and has subsequently led to the development of the NEMP web data system. It also prompted the development of an interactive NMMP map using leaflet.

Introduction

The successful conversion of the National Chemical Monitoring Programme reporting system to R and ODBC suggested that the Internet would be a suitable medium for communicating other monitoring data. The NMMP implementation manual sets out an Excel-based reporting system, which requires that the national coordinator email spreadsheets to regional managers every two months (DWAF 2002). The R script described here creates HTML files that mimic the microbial results layout specified in the NMMP implementation manual, including the maps and tables, with a simple navigation path, first by year, then by two-month period, then by region. The table is colour-coded according to user risk and has links to time-series plots.

Methods

Trial versions 0.1 to 2.04 of `NMMP_WMA_web.R` were coded in R during June and July 2011. This document describes software as it was at version 7.8 of `NMMP_WMA_web.R` in August 2017. It will likely undergo further changes because of enhancements and debugging.

The script comprises 1 main routine and 9 functions. The rest of this section deals with the details of the code.

Dependencies

The script requires several R packages (Table 1). It also requires other locally-written scripts in order to operate properly (Table 2).

Table 1. R packages required for NMMP_WMA_web.R, e.g. `library(RODBC)`

package	purpose	description	authors
RODBC	ODBC Database Access	An ODBC database interface	Ripley and Lapsley 2016
maptools	Tools for Reading and Handling Spatial Objects	Set of tools for manipulating and reading geographic data, in particular 'ESRI Shapefiles'	Bivand and Lewin-Koh 2018
rgdal	Bindings for the 'Geospatial' Data Abstraction Library	Provides bindings to the 'Geospatial' Data Abstraction Library ('GDAL') (>= 1.6.3) and access to projection/transformation operations from the 'PROJ.4' library	Bivand *et al.* 2018
stringr	Simple, Consistent Wrappers for Common String Operations	A consistent, simple and easy to use set of wrappers around the 'stringi' package. All function and argument names (and positions) are consistent, all functions deal with "NA"'s and zero length vectors in the same way, and the output from one function is easy to feed into the input of another	Wickham 2018

Table 2. Local scripts required for NMMP_WMA_web.R, e.g. source ("C:/data/program/R/toTitle.R")

script	purpose	description
toTitle.R	Changes ALL CAPITALS to Proper Case, except in stop words	A script that converts text in capitals to a more legible mixed case, with provision for special case such as alphanumeric codes, chemical symbols and map codes
NMMP_TimeLine.R	Time series plots of microbial data	Constructs a time series plot for all records at each monitoring point, showing *Escherichia coli* and turbidity.

script	purpose	description
NMMP_miniplots.R	Compact summary graph	Constructs a very compact summary of water quality based on coliforms.
NMMP_KML_balloons.R	Google Earth files for microbial sites	Constructs KML files for individual sites and for all sites in each WMA-9 and WMA-19, for viewing in Google Earth. Compresses KML to KMZ for quicker download.
NMMP_R2leaflet.R	Map inventory using leaflet mapper in NMMP_R2leaflet_mapper.R(Grant 2013)	Constructs an interactive map with microbial monitoring sites. Users can click on a site for a miniplot, and click on the miniplot for a TimeLine plot.

Main routine

The main routine connects with the database, reads in geographical layers, sets up guideline levels, assigns colour codes, obtains the site information and sample data for all sites, and generates the output graphics and text files. The input data from WMS are counts of faecal coliforms and *Escherichia coli*, turbidity, pH and temperature. The geographical layers include monitoring site locations, primary and tertiary drainage regions, water management areas (19 and 9), rivers, dams and towns.

Database settings

The input is from an Informix water quality database, the Water Management System (WMS), accessed through the R package, RODBC (Ripley and Lapsley 2016). For the database connection to function, first set up an ODBC link on the user's computer (Appendix 1). The following commands activate the RODBC package and open the database:

```
library(RODBC)
odbcCloseAll()
db <- "wmsdb"
channel <- odbcConnect(db)
db_unlock <- "set isolation dirty read"
sqlQuery(channel, db_unlock)
```

The last two lines are there on the recommendation of the database administrator, to avoid read errors.

Constants

The parameters for a batch run are set by assigning values to constants (Table 3).

Table 3. The default parameters in NMMP_WMA_web.R

Variable	Default value	Description
auto.xtras	TRUE	run the auxiliary scripts for plots, miniplots, KMLs and leaflet map
readGIS	TRUE	set to FALSE to skip reading GIS data repeatedly during testing
readWMS	TRUE	set to FALSE to skip reading WMS repeatedly during testing
yr1	1990	start year
yr2	current year	year extracted from today's date

Variable	Default value	Description
wdr0	C:/tmp/nmmp/web/	root directory for NMMP output
glcol	"#0000CD", "#2E8B57", "#CDAD00", "#FF0000", "#8B008B", "#C71585"	guideline colour coding
glcolpastel	"#C1CDC1", "#FFFACD", "#FFE1FF", "#FFB6C1"	guideline colour background shading
glclassn	"low-med risk", "high risk when drinking untreated water", "high risk from full or partial contact", "high risk when irrigating crops that are eaten raw", "high risk when drinking after only limited treatment"	risk labels
glrange	10, 400, 4000, 20000	risk cut-off values
gl.risk*	(1, 10) (2000, 20000) (1000, 4000) (200, 400)	guideline risk ranges
gsize	1.0, 1.5, 2.1, 2.5, 3.0	symbol and text sizes

GIS data input

The readOGR() function inputs all geographical data layers required for producing maps (Bivand *et al.* 2018).

The following code reads the required spatial data from shapefiles:

```
if (readGIS) {
  shp.dir <- "C:/data_large/av/drainage"
  mpt.dir <- "C:/data_large/av"
  print (paste("Define and read geographical data from", shp.dir))
  write(paste0(Sys.time(), ": Defining and reading geographical data from ",
 shp.dir), file=fil.log, append=TRUE)
  stn <- readOGR(dsn=mpt.dir, layer="nms_wms_geo", stringsAsFactors=FALSE)
  stn$FEAT_ID <- as.numeric(stn$FEAT_ID)
  pri <- readOGR(dsn=shp.dir, layer="hca_1_simpler", stringsAsFactors=FALSE)
  prip <- readOGR(dsn=shp.dir, layer="hca_1geo", stringsAsFactors=FALSE)
  ter <- readOGR(dsn=shp.dir, layer="hca3geo", stringsAsFactors=FALSE)
  rv <- readOGR(dsn=shp.dir, layer="wriall500_SimplifyLine_500m",
GDAL1_integer64_policy=TRUE, stringsAsFactors=FALSE)
  lak <- readOGR(dsn=shp.dir, layer="wla500g", stringsAsFactors=FALSE)
  lak$FEAT_ID <- as.numeric(lak$FEAT_ID)
  town <- readOGR(dsn=shp.dir, layer="smu_500g_simpl_pr_100m",
stringsAsFactors=FALSE)
  twn <- town[!is.na(town$NAMETAG), ]
  twn_big <- twn[twn$AREA>quantile(twn$AREA, 97 / 100, type=1, na.rm=TRUE), ]
  twn_big <- twn_big[ !duplicated(twn_big$NAMETAG), ]
} else {
  print("Assuming GIS data already in memory - set readGIS and readWMS to TRUE if
not")
  write(paste0(Sys.time(), ": Assuming GIS data already in memory - set readGIS and
readWMS to TRUE if not"), file=fil.log, append=TRUE)
}
```

The files and file paths referenced must be accessible on the computer where the script is run.

A locality map appears at the bottom of each bimonthly report page. The script invokes two packages for handling spatial data, namely `rgdal` (Bivand *et al.* 2018) and `maptools` (Bivand and Lewin-Koh 2018).

A separate script, `NMMP_R2leaflet.R`, creates map interfaces to the NMMP data and to all microbial data on WMS.

Layout parameters

A set of constants defines colour coding, range labels and page settings (Table 3).

Time variables

During program testing, timing variables are helpful for detecting parts of the code that require optimising in order to speed up the process:

```
time.start <- format(Sys.time(),"%Y-%m-%d %H:%M:%S") # (t1)
[…]
time.end <- format(Sys.time(),"%Y-%m-%d %H:%M:%S") # (t2)
difftime(time.end, time.start)
```

The variables yr1 and yr2 specify the range of data to be used. The second variable is usually specified as the current year:

```
yr2 <- as.numeric(substr(Sys.time(), 1, 4))
```

Site selection

A normal run of the script will process a list of all available sites in the NMMP programme.

```
programme.n <- 135
q <- paste("SELECT UNIQUE mon_feature_id ",
      "FROM programme_sample WHERE mon_program_id = ", programme.n, sep="")
sites.nmmp <- sqlQuery(channel, q)
```

Microbial data input

The script reads in the *E. coli,* faecal coliforms, turbidity, pH and temperature data for NMMP sites, using a structured query submitted via the open RODBC channel, e.g.:

```
qs <- paste(
    "SELECT mon_feature_id, sample_begin_date, sample_begin_time,",
    " sample_begin_depth, mon_variable_id,",
    " result_num_value FROM released_result",
    " WHERE mon_variable_id IN (", paste(unlist(varidc), collapse=","), ")",
    " AND mon_feature_id IN (", sites.list, ") ",
    sep="")
  df.coli <- sqlQuery(channel, qs)
```

The Department of Water and Sanitation originally had 19 water management areas, and later consolidated these into 9 regions. The script provides output in both formats.

```
versions.wma <- c(2004, 2012)
for(version.wma in versions.wma) {…}
```

Then, for each year in yr1 to yr2, the script assembles the starting and ending dates for the six bimonthly (i.e. two-monthly) periods. Then, for each period, it checks for the presence of monitoring sites and *E. coli* or faecal coliform data. If data are not available, the HTML heading code for a no-data condition is written; otherwise the script constructs an HTML table line-by-line.

```
# select E. coli only:

ecolis.a <- ccolis[ccolis$mon_variable_id %in% varide, ]

ecolis.a <- na.omit(ecolis.a)

print(paste(paste(unlist(ids), collapse=" + "), "Number of E. coli samples =",
nrow(ecolis.a)))

write(paste(paste(unlist(ids), collapse=" + "), "\nNumber of E. coli samples =",
nrow(ecolis.a)), file=fil.log, append=TRUE)

# select faecal coliforms only:

fcolis.a <- ccolis[ccolis$mon_variable_id %in% varidf, ]

fcolis.a <- na.omit(fcolis.a)

print(paste(paste(unlist(ids), collapse=" + "), "Number of faecal coliform samples
=", nrow(fcolis.a)))

write(paste(paste(unlist(ids), collapse=" + "), "\nNumber of faecal coliform
samples =", nrow(fcolis.a)), file=fil.log, append=TRUE)

# combine the query results into a single data frame:

efcolis.a <- merge(ecolis.a, fcolis.a,

by=c("mon_feature_id", "sample_begin_date", "sample_begin_time",
"sample_begin_depth"), all=TRUE)

names(efcolis.a)[6] <- "ecoli"

names(efcolis.a)[8] <- "fcoli"
```

A combined field, “ccoli”, is populated with either *E. coli* data, or if that is not available, faecal coliform data. If neither data type is available, the field is blank.

Plotting time series data

The NMMP implementation manual specifies 2-month plots of coliforms and turbidity. The plot file contains the feature ID and date range. This structure needs to be precise if cross-references pointing at the file are to work.

```
fil.pl <- paste(wdir, "coli_",  featid, "_", "NMMP", "_", Date1, "_", Date2,
"_plot.png", sep="")
```

A separate routine generates time-series plots of all microbial and turbidity data at each monitoring site. The names also need to be consistent for cross-referencing:

```
fil.fpng <- paste(wdr0,"Chart_nmmp_", featid, ".png",sep="")
```

Text file

Advanced users of the water quality data may prefer to make their own time series plots, so the `NMMP_R2leaflet.R` script exports zipped comma-separated-value (CSV) files. The current default is

pkzip – if it is not available, R's built-in `zip` or any alternative compression program may work. This code is from `NMMP_R2leaflet.R`:

```
nmmp.data <- chemRead(featids, "micro") # highly inefficient!

nmmp.data <- nmmp.data[, -which(names(nmmp.data) =="NO3_NO2_N_Calc_Water")]

nmmp.data <- nmmp.data[!is.na(nmmp.data$E_COLI_Susp_Water) |
!is.na(nmmp.data$FC_Susp_Water), ]

nmmp.data <- nmmp.data[order(nmmp.data$mon_feature_id, nmmp.data$date_time,
nmmp.data$sample_begin_depth), ]

write.csv(nmmp.data, file=nmmp.data.file, row.names=FALSE, na="")

nmmp.zip.file <- paste0(dir.base, gsub('[[:punct:][:space:]]', '_', map.type),
"_rawdata.zip")

if (file.exists(nmmp.zip.file))

shell(paste("cd", dirname(nmmp.zip.file), "& del /f", basename(nmmp.zip.file)))

shell(paste("pkzipc -add -silent=banner -move", nmmp.zip.file, nmmp.data.file))

shell(paste("pkzipc -add -silent=banner -move", nmmp.zip.file, nmmp.list.file))

shell(paste("pkzipc -add -silent=banner -move", nmmp.zip.file, nmmp.meta.file))
```

Time-series plots

Plotting uses the R `layout(1:2)` method to arrange plots of pathogens and turbidity on the same page (Figure 1). Where no data are available for either variable during the selected period the script produces an empty plot for the variable (e.g. the "No turbidity data…" in Figure 2). If no data are available at all, no plot is produced.

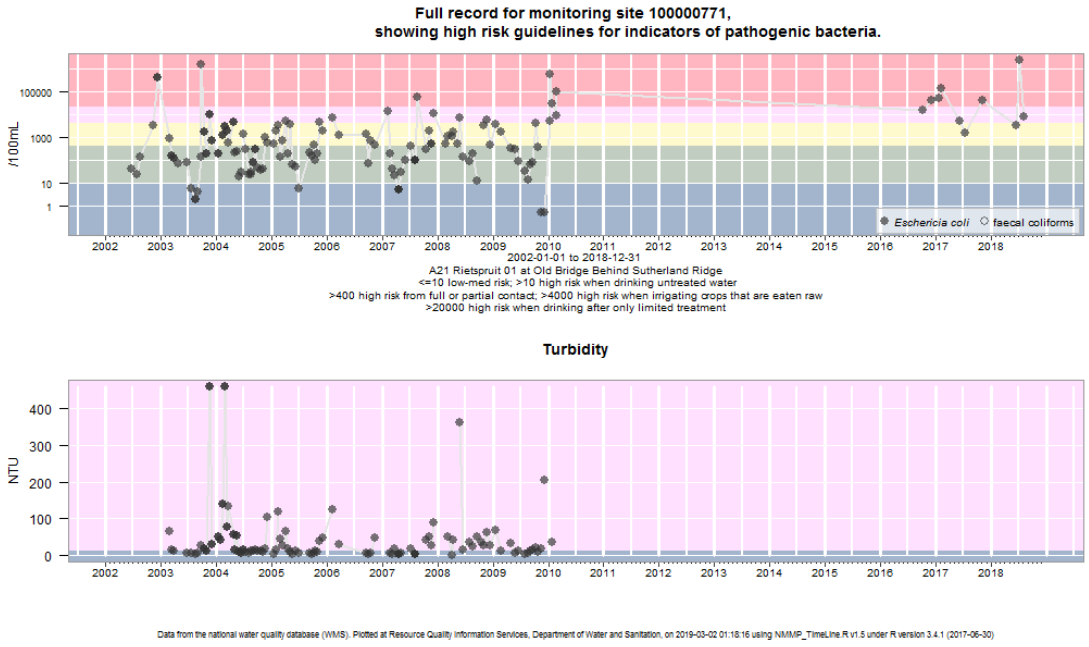


Figure 1. A time series plot of *E. coli* and turbidity at Sutherland Ridge, Gauteng, from 2003 to 2018. Note the termination of turbidity measurements in 2010 and the gap in pathogen data from 2010 until 2016.

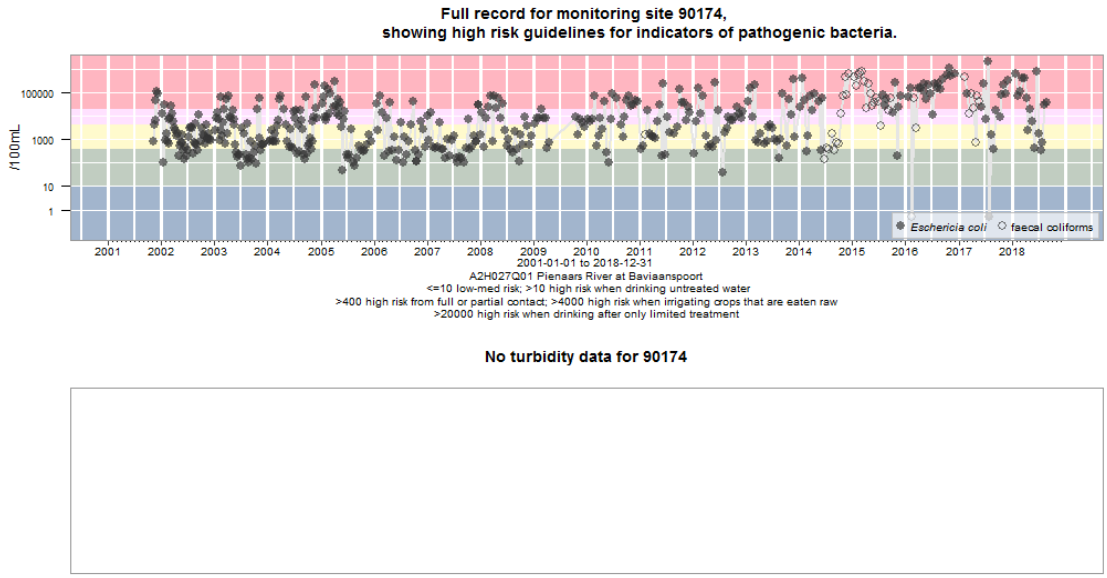


Figure 2. A time series plot of *E. coli* and faecal coliforms at Roodeplaat Dam from 2002 to 2018. No turbidity results are available for this site.

Plots have background shading according to the usability of the water (Figure 1).

Functions

Version 7.8 of `NMMP_WMA_web.R` consists of 9 functions and a main routine.

DateStamp

`DateStamp` creates date stamp and script version text strings for all messages, output and logging.

wmscheck

`wmscheck` reports a SQL query error, then terminates untidily.

HTMLindex

`HTMLindex` creates an index by year to the HTML tables created by `NMMP_WMA_web.R`.

HTMLhead

`HTMLhead` writes the header for each output table.

cap

`cap` makes the first character in a string upper case. Could be replaced by the `str_to_title()` function in `stringr`.

leapyear

`leapyear` tests whether a year is a leap year. Can be replaced by the `leap_year()` function in `lubridate`.

fil.pr

`fil.pr` checks for the previous logical file in a sequence. This is needed for the previous date scrolling arrow in each web page.

fil.nx

`fil.nx` checks for the next logical file in a sequence. This is needed for the next date scrolling arrow in each web page.

classtext

The `classtext` function assembles the text describing the sample class according to the *E. coli* or faecal coliform maximum.

Discussion and conclusion

The generation of the HTML and image files by NMMP_web.R takes about six hours and the upload to the website a further six hours, using the batch file `robocopy_nmmp.bat` (Appendix 2). The entire NMMP website is regenerated using an overnight batch file that runs weekly, incorporating any data updates and corrections. The user needs to check that the batch run has completed without errors before uploading the new information to the website.

This R script has turned out to be a useful means for distributing data and information. It is applicable to other data types and inspired the development of the NEMP web data system. It also prompted the development of interactive maps using `NMMP_R2leaflet.R`. The `robocopy_nmmp.bat` batch job that copies files to the website also updates these maps.

The benefit of the software is that anyone can access the microbial data without having to be part of the NMMP email distribution list. Provided that the Internet is functional, users can check on their monitoring areas without having to wait for the manual data distribution cycle.

A useful development would be to have an interactive map below each table, rather than the current static map. However, this will only be practical once the Internet connection available to DWS users becomes faster and more reliable.

Acknowledgements

Willie Geldenhuys and Deon van Zyl assisted with the formulation of SQL queries. The Resource Quality Information Services directorate and its consultants have performed an enormous amount of work in establishing and maintaining the WMS. Sbongile Mwale-Phungula and Elijah Mogakabe have kept the microbial monitoring programme on track, and maintained the regional laboratory contracts for as long as was possible.

References

Bivand, R., Keitt, T., and Rowlingson, B., 2018. *rgdal: Bindings for the 'Geospatial' Data Abstraction Library*.

Bivand, R. and Lewin-Koh, N., 2018. *maptools: Tools for Reading and Handling Spatial Objects*.

DWAF, 2002. *National Microbial Monitoring Programme for Surface Water Implementation Manual*. Pretoria, South Africa: Department of Water Affairs and Forestry, Technical Report.

Grant, R., 2013. *R function to plot data on interactive JavaScript maps: GitHub robertgrant/R2leaflet*.

Ripley, B. and Lapsley, M., 2016. *RODBC: ODBC Database Access*.

Wickham, H., 2018. *stringr: Simple, Consistent Wrappers for Common String Operations*.

Appendices

Appendix 1 and Appendix 2 are not available in the online version of this document.