epanetReader : a package for reading EPANET files into R

Bradley J. Eck

May 2016

Please cite this as: Eck, B. (2016) epanetReader: A Package for Reading EPANET Files into R. World Environmental and Water Resources Congress 2016: pp. 487-496. doi: 10.1061/9780784479865.051

Abstract

The EPANET package for modeling water distribution systems is widely used and several methods are available to interact with network data. These methods include commercial and freely available graphical interfaces, geographic information systems, and the programmer's toolkit. Another approach is to use a technical computing platform such as the R environment to visualize and analyze simulation data. R is open source and freely available. epanetReader is an add-on package for reading files in EPANET's '.inp' and '.rpt' formats into R. Basic summary information and plotting capability for the data contained in the input and report files is provided so that three lines of executable code are needed to (1) read a network file; (2) see the number of each type of element; and, (3) view a map of the network. Once information from EPANET exists within R, other add-on packages for graphics and animation can be used to interpret and visualize simulation results. epanetReader is freely available from GitHub or the comprehensive R archive network (CRAN). This paper provides further detail on the installation and usage of epanetReader, including several examples.

Introduction

The United States Environmental Protection Agency software for modeling piping networks, EPANET, has become a de-facto standard for water systems and enjoys use in many countries. Several methods are available to interact with network data, including commercial and freely available graphical interfaces, geographic information systems, and the EPANET programmer's toolkit. Another approach is to use a technical computing platform such as the R environment to visualize and analyze simulation data.

R is a freely available open source software environment for statistical computing and graphics (R Core Team 2013). Since initially released in the late 1990s, R has become a popular tool for statistics, finance, environmental analysis, and many other applications. Reasons for the popularity include availability of R as open-source software and a system for managing and distributing add-on packages. The R packaging system defines a format for providing additional functionality to an R installation. Packages passing a series of checks may be distributed through the Comprehensive R Archive Network (CRAN), a worldwide group of servers. Packages on CRAN are installable over the internet from within an R session. With a defined format and built-in distribution mechanism, R packages make a convenient way to share functionality with the research community.

At this writing, CRAN has over 7000 different R packages including several focused on environmental and water resources applications. The United States Geological Survey (USGS) distributes the dataRetrieval package (Hirsch and Cicco 2015) for retrieving hydrologic and water quality data from the web. (Turner and Galelli 2016) developed the reservoir package for the analysis, design, and operation of water supply storages. This paper describes the usage of epanetReader, a package for working with data from EPANET simulations.

The remainder of the paper gives a short guide to epanetReader intended for those familiar with EPANET but not R. Readers using R for the first time can find a variety of resources online or in print. The website

statmethods.net and book R in Action (Kabacoff 2015) are good starting points. For the purposes of running the examples shown herein, it is assumed that the current version of R is installed. Sections are devoted to package installation, water network data, hydraulic simulations, and multi-species water quality simulations. Next, a unique visualization of simulation results as table of small data lines called sparklines is presented. The paper closes with information on contributing to the work and applications involving other R packages.

Package Installation

Once R is installed, the epanetReader package can be downloaded and installed from CRAN or from GitHub.com. The stable released version of the package is on CRAN and installable from inside an R session. Packages distributed on CRAN pass a series of tests to ensure that packages will install on different systems, that all of the user functions are documented, and that code examples in the documentation actually run. For users getting started with R or any package, the version on CRAN is the best place to start.

```
# download the package from CRAN and install it
install.packages("epanetReader")
```

The development version of epanetReader is online at www.github.com/bradleyjeck/epanetReader. This version may contain features, documentation, or bug fixes not yet published to CRAN. The master branch of the project is intended to be deployable and so does not contain any known bugs. But, as the development version, it may not have passed all the checks required for a CRAN submission. In case your project needs a feature in the development version, it can be installed using the devtools package.

```
# download the package from Github and install it
devtools::install_github("bradleyjeck/epanetReader")
```

After installation, packages must be loaded into the current R session. After loading, the functions, data, and documentation contained in a package become available to the user. Every function in R has documentation that can be accessed by typing a question mark and the function name.

```
# load the package
#library(epanetReader)
```

```
# access the built-in documentation
?epanetReader
```

Network Data

EPANET can read and store network information in formatted text files. By convention, these files use the extension '.inp'. epanetReader provides the read.inp() function for reading text files in Epanet's .inp format into R. The function has only one argument: the path of the file to read. Calling the function creates an R object containing the information in the file. The code below reads an .inp file created from the Net2 example distributed with EPANET.

```
# read the file Net2.inp into R and store the
# data in a variable called n2
n2 <- read.inp("Net2.inp")
```

```
## Warning in PATTERNS(allLines): patterns have integer IDs, see ?epanet.inp
```

In this case, reading example network 2 produces a warning. Users interested in data contained in the Patterns section of the .inp file can consult the documentation for further information. The data are stored in a variable for further analysis and manipulation. Once the file is read, the summary() function provides an overview of the network. The summary shows that this network has 35 junctions, 40 pipes and one tank.

```
summary(n2)
```

```
## $Title
## [1] "EPANET Example Network 2"
## [2] "Example of modeling a 55-hour fluoride tracer study."
## [3] "Measured fluoride data is contained in the file Net2-FL.dat"
## [4] "and should be registered with the project to produce a"
## [5] "Calibration Report (select Calibration Data from the Project"
## [6] "menu)."
##
## $entryCounts
##             Number
## Junctions       35
## Tanks            1
## Pipes           40
## Coordinates     36
##
## attr(,"class")
## [1] "summary.epanet.inp"
```

Files in the .inp format are divided into sections specifying information on network elements and simulation parameters. These sections correspond to named parts of the object created by reading the file. In R, these sections are accessed using the dollar sign operator. Built-in functions like summary() also work on the individual parts of the object.

```
# access the Tanks table of the Net2 example
n2$Tanks
```

```
##   ID Elevation InitLevel MinLevel MaxLevel Diameter MinVol VolCurve
## 1 26       235      56.7       50       70       50      0       NA
```

```
# see a summary of the junctions table in the Net2 example
summary(n2$Junctions)
```

```
##       ID              Elevation         Demand        Pattern
##  Length:35          Min.   : 50.0   Min.   :-694.40   2   : 1
##  Class :character   1st Qu.:110.0   1st Qu.:   2.50   NA's:34
##  Mode  :character   Median :150.0   Median :   8.00
##                     Mean   :148.3   Mean   : -10.62
##                     3rd Qu.:190.0   3rd Qu.:  15.00
##                     Max.   :230.0   Max.   :  34.78
```

The summary of the junctions table shows that elevations average 148.3 and range from 50 to 230. A list of the parts of the variable n2 created by reading the file Net2.inp are available using the names() function.

```
names(n2)
```

```
##  [1] "Title"       "Junctions"   "Tanks"       "Reservoirs"  "Pipes"
##  [6] "Pumps"       "Valves"      "Demands"     "Patterns"    "Curves"
## [11] "Energy"      "Status"      "Times"       "Options"     "Coordinates"
```

Users may note that the presence of some names here that did not appear in the summary such as Pumps and Reservoirs. Further interrogation will show the values for these sections are NULL because the file did not contain information on these elements. It may also be noted that some sections contained in the .inp file are not reported by the names() function. These include the Quality and Reactions sections of Net2.inp. As of this writing, the functionality for reading several sections of .inp files is not yet implemented.

In addition to reading data, the package also provides some basic plotting functionality. A simple map of the

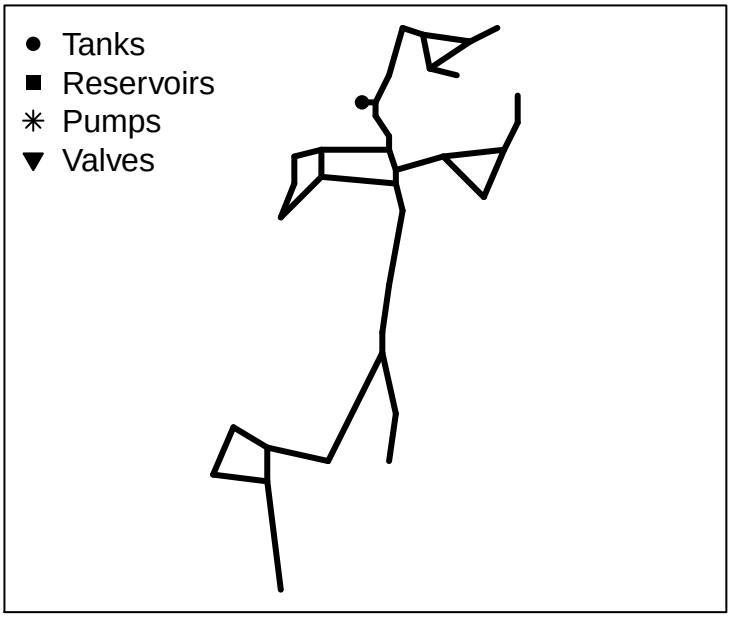

Figure 1: Plot of example network 2

network is available through the plot function. Due to the layout of the Net2 example, the legend fits better in the top left corner of the plot (Figure 1). A title for the plot may also be added.

```
plot(n2,legend.locn = "topleft")
title( main = "Net2 Example")
```

Hydraulic Simulations

Results of a hydraulic simulation can be stored in text file using EPANET's command line or graphical interface. By convention, text files reporting the results of a simulation use the extension '.rpt'. Output of a simulation is controlled by the Report section of the .inp file. epanetReader is designed to read output formatted as a single page, without new page symbols. The results most often of interest pertain to links and nodes of the network. These results must be requested in order to appear in the output. To read results using epanetReader, the report section of the .inp file should include the following lines.

```
[REPORT]
Page 0
Links All
Nodes All
```

Continuing the Net2 example, the package reads results of a hydraulic simulation in a manner similar to reading the network file.

```
# read the results of a hydraulic simulation for example network 2
n2r <- read.rpt("Net2.rpt")
```

After reading the data, a high-level summary is useful to understand some basic details of the simulation. Results were generated for 56 time steps. Over the simulation period, fluoride levels in the tank varied between 0.73 and 1.0 while levels at demand nodes were as low as 0.07. Looking at the pipes, at least half had very low velocity and almost negligible head loss.

```
summary(n2r)
```

```
## Contains node results for  56 time steps
##
## Summary of Junction Results:
##      Demand            Pressure         Fluoride
##  Min.   :-694.400   Min.   : 26.54   Min.   :0.0700
##  1st Qu.:   2.430   1st Qu.: 46.70   1st Qu.:0.1375
##  Median :   6.825   Median : 63.31   Median :0.7700
##  Mean   :  -1.057   Mean   : 64.64   Mean   :0.6106
##  3rd Qu.:  13.445   3rd Qu.: 81.04   3rd Qu.:1.0000
##  Max.   :  85.560   Max.   :117.84   Max.   :1.0500
##
## Summary of Tank Results:
##      Demand          Pressure        Fluoride
##  Min.   :-794.0   Min.   :24.35   Min.   :0.7300
##  1st Qu.:-313.1   1st Qu.:25.14   1st Qu.:0.7850
##  Median : 151.6   Median :26.29   Median :0.9000
##  Mean   :  37.0   Mean   :26.32   Mean   :0.8848
##  3rd Qu.: 353.8   3rd Qu.:27.41   3rd Qu.:0.9925
##  Max.   : 594.3   Max.   :28.22   Max.   :1.0000
##
## Contains link results for  56 time steps
##
## Summary of Pipe Results:
##       Flow            Velocity         Headloss
##  Min.   :-794.04   Min.   :0.0000   Min.   :0.0000
##  1st Qu.:   0.64   1st Qu.:0.0300   1st Qu.:0.0000
##  Median :   9.90   Median :0.1100   Median :0.0100
##  Mean   :  85.16   Mean   :0.4022   Mean   :0.2767
##  3rd Qu.:  77.84   3rd Qu.:0.5900   3rd Qu.:0.3300
##  Max.   : 694.40   Max.   :2.2500   Max.   :2.6900
```

In contrast to the treatment of .inp files described in the previous section, the structure of the object created by read.rpt() differs slightly from the structure of the .rpt file. Report files generated by EPANET contain a table of results for nodes and a table of results for links corresponding to each time step for which results were requested. epanetReader combines the tables across time steps so that the object created by reading an .rpt file contains a table for node results and a table for link results. The consolidated tables also contain extra columns beyond what appears in the .rpt file. The time of the result is stored in seconds in addition to the character-based hour:minute:second format. A column is also added distinguish the type of node or link in each row. These choices aim at making simulation results easier to manipulate and visualize.

The tables of node results and link results are visible using the built-in names() function and accessed using the dollar sign operator. The subset() function can be used to investigate parts of the simulation that may be of interest. For example, the summary of results indicated that the maximum demand in a junction was

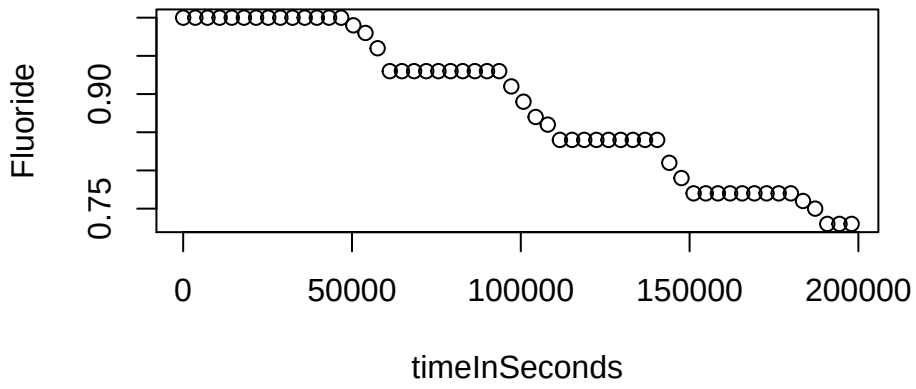

Figure 2: Trajectory of fluoride concentrations in tank 26 for example network 2

just above 85. That information can be used with the subset() function to see that the demand occurred in node 11 ten hours from the start of the simulation and that the fluoride level at that time was 0.69.

```
subset(n2r$nodeResults, nodeType == "Junction" & Demand > 85 )
```

```
##     ID Demand   Head Pressure Fluoride note Timestamp timeInSeconds
## 371 11  85.56 291.83    46.29     0.69       10:00:00         36000
##     nodeType
## 371 Junction
```

Generating graphs of results is also straightforward. The fluoride concentration in the tank can be shown using the subset function together with the plot function (Figure 2). From looking at the tanks section of the .inp file in the previous section, the ID of the only tank in this network is 26.

```
plot(Fluoride ~ timeInSeconds, data = subset(n2r$nodeResults, ID == "26"))
```

The .inp file used to generate an .rpt file can also be used to generate a p-based visualization of simulation results. Figure 3 shows the state of the network ten hours into the simulation. Fluoride concentrations, indicated by circle size, are highest near the tank, and lowest at the ends of the system. Velocities, indicated by line width, are also highest near the tank. The help page ?plot.epanet.rpt provides more detail on the functions for plotting simulation results.

```
plot(n2r, n2, Timestep = "10:00:00", juncQty = "Fluoride", legend1.locn = "bottomright")
```

Multi-Species Water Quality Simulations

EPANET supports reactive transport modeling in pipe systems for a single reacting species. Simulation of multiple interacting species is possible through the use of the multi-speices extension, EPANET-MSX (Shang, Uber, and Rossman 2011) Like EPANET, the multi-species extension can run from the command line and produce simulation results in a formatted text file. epanetReader parses multi-species results into R using the read.msxrpt() function.

As with reading .inp and .rpt files, reading results of a multi-species simulation creates an R object. Following

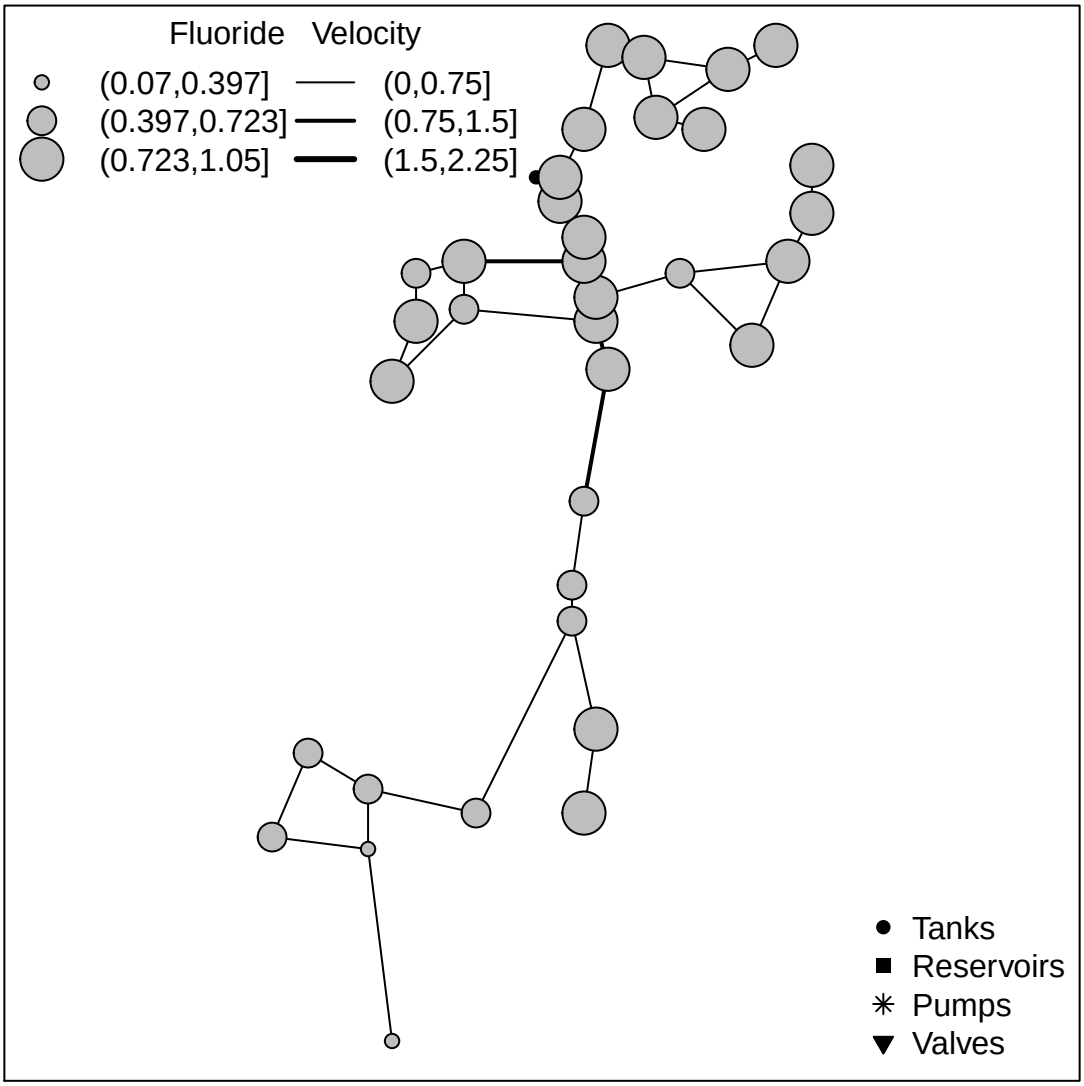

Figure 3: Example network 2 simulation results after ten hours

the approach for hydraulic simulations, the object has a table of node results and a table of link results that combine results for all the time periods. The generic functions including summary(), names(), subset(), and plot() can be used in conjunction with the object created by read.msxrpt().

In multi-species simulations several parameters are usually of interest, often at specific locations in the system. The multi-species extension caters for this need by allowing output only for specific network elements. From a plotting perspective, network maps are not meaningful without all of the network elements. Thus, instead of a map the default plot for multi-species results is a table of sparklines. Sparklines are discussed in the next section.

Visualizing with Sparklines

An interest in visualizing multiple simulation parameters led to an implementation for plotting a table of sparklines in R. A sparkline is small line graph, usually without axes or coordinates (Tufte 2006). The function plotSparklineTable() is included with the epanetReader package but may also be useful to visualize data from other domains. Within epanetReader, a table of sparklines in the default plotting style for multi-species water quality results. This type of visualization can also be used for hydraulic simulation results, as shown below, as well as many other kinds of data.

Figure 4 shows a table of sparklines generated from the hydraulic simulation of Net2. The gray sparkline shows the variation of a value over the time of the simulation. Starting and ending values are written adjacent to black dots marking the start and end of the sparkline. The table shows that the starting concentration of Fluoride was 1 at every node. The pattern of Fluoride fluctuation can be examined and compared with the demand and pressure variation at the node. Such a table shows a large number of data values, in this case about 6000, in a single view.

```
plotSparklineTable( n2r$nodeResults, row.var = "ID",
                    col.vars = c("Pressure", "Demand", "Fluoride" ) )
```

Contributions

epanetReader is released as an open-source project under the MIT license and remains under active development. Since the initial release, several new features have been added including reading multi-species results. Future updates may improve plotting capabilities, add more comprehensive documentation, or implement other features. Contributions from third-party developers or other suggestions for improvement are most welcome. The project page at github.com/bradleyjeck/epanetReader has more information on how to contribute.

Conclusion

This paper has aimed at providing an overview to manipulating water network simulation data in the R environment using the epanetReader package. The package builds on R's extensive functionality for analysis and visualization by providing customized functions for water network data. Objects created by reading EPANET files can be manipulated using the many built-in functions of R. Customized summary and plotting functions provide a starting point for interactive analysis and interpretation of network information and simulation results.

Interpreting simulation results may become easier looking at a table of sparklines. This graphical representation achieves a high data density and enables comparisons across simulation parameters. Although motivated by an interest in water network simulations, the implementation of sparkline tables in epanetReader is suitable for other applications.

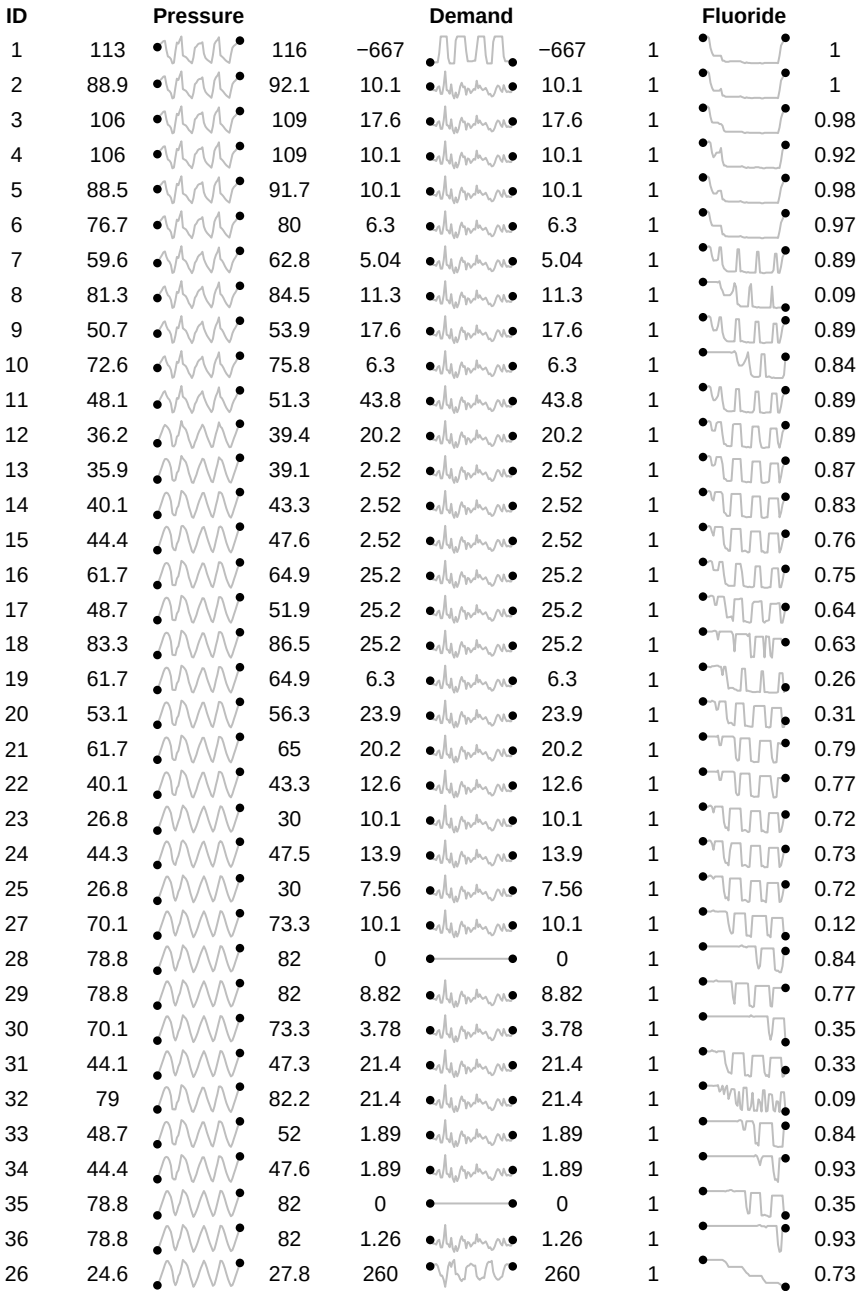

Figure 4: Sparkline plot of hydraulic simulation results for Net 2 example

The examples in this paper have focused on the functionality enabled by epanetReader and the base version of R. There are many other add-on packages developed by others and available on CRAN that add different functionality. Examples include the packages ggplot2 (Wickham 2009) for graphics, ggmap (Kahle and Wickham 2013) for drawing maps, and animation (Xie 2013) for creating videos. epanetReader can be used alone or along with other packages to create unique and insightful results.

Acknowledgements

The suggestions of Sean McKenna and Ernesto Arandia in improving the manuscript are gratefully acknowledged.

References

Hirsch, Robert M., and Laura A. De Cicco. 2015. "User Guide to Exploration and Graphics for RivEr Trends (EGRET) and DataRetrieval: R Packages for Hydrologic Data." In *Techniques and Methods*. Reston, VA: U.S. Geological Survey; U.S. Geological Survey. http://pubs.usgs.gov/tm/04/a10/.

Kabacoff, Robert. 2015. *R in Action, Second Edition*. Manning Publications.

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

R Core Team. 2013. *R: A Language and Environment for Statistical Computing*. Vienna, Austria: R Foundation for Statistical Computing. http://www.R-project.org/.

Shang, F., J.G. Uber, and L.A. Rossman. 2011. *EPANET Multi-Species Extension User's Manual*. Cincinnati, Ohio: US Environmental Protection Agency.

Tufte, Edward. 2006. *Beautiful Evidence*. Cheshire, Connecticut, USA: Graphics Press.

Turner, S.W.D., and S. Galelli. 2016. "Water Supply Sensitivity to Climate Change: An R Package for Implementing Reservoir Storage Analysis in Global and Regional Impact Studies." *Environmental Modelling & Software* 76: 13–19. doi:10.1016/j.envsoft.2015.11.007.

Wickham, Hadley. 2009. *Ggplot2: Elegant Graphics for Data Analysis*. Springer-Verlag New York. http://had.co.nz/ggplot2/book.

Xie, Yihui. 2013. "animation: An R Package for Creating Animations and Demonstrating Statistical Methods." *Journal of Statistical Software* 53 (1): 1–27. http://www.jstatsoft.org/v53/i01/.