Environmental Modelling & Software 84 (2016) 149–154

Contents lists available at ScienceDirect

Environmental Modelling & Software

journal homepage: www.elsevier.com/locate/envsoft

Short communication

An R package for reading EPANET files

Bradley J. Eck

IBM Research, Dublin, Ireland

ARTICLE INFO

Article history:
Received 4 March 2016
Received in revised form
19 May 2016
Accepted 28 June 2016

Keywords:
Water networks
R
EPANET
Hydraulic modeling

ABSTRACT

The EPANET software for modeling piping networks is widely used for the design and analysis of water systems. This short communication describes epanetReader, an R package for reading EPANET files. The package reads network and simulation data in text-based file formats into R and provides summary and plotting functionality. epanetReader also introduces sparklines as a visualization of water network simulations. The package is available through the Comprehensive R Archive Network and GitHub.com.

Software availability

epanetReader is available on the world wide web at: https://cran.r-project.org/package=epanetReader and https://github.com/bradleyjeck/epanetReader
License: MIT
System requirements: R version 3.0.0 or higher
Installation: using install.packages() function

1. Introduction

Researchers and practitioners show continued interest in simulating the behavior of water piping networks. One popular tool for these simulations is EPANET, the United States Environmental Protection Agency's model for water movement and quality in pressurized pipe networks. EPANET enjoys wide use in academic and commercial contexts. The user manual for EPANET version 2 (Rossman, 2000) has nearly 2000 citations on Google Scholar. Several commercial software packages use the EPANET computational engine or provide compatibility with its files. EPANET has thus become a de-facto standard for water network analysis.

The version of EPANET on the US EPA website includes three interfaces for interacting with simulation data. A graphical interface aimed at users of Microsoft Windows™ allows users to create a network, run an analysis, and view results. An application programming interface known as the programmer's toolkit provides a library of functions to manipulate network data, customize simulations, and access results. EPANET also has a command-line interface that takes as arguments a file for simulation inputs and a file name for reporting simulation results. Also available through a command line and application programming interface is the multi-species extension, EPANET-MSX (Shang et al., 2011), for modeling multiple interacting chemical species.

The existing interfaces have several limitations with regard to analysis and visualization. The graphical interface can plot results for one network element or simulation parameter at a time. Results may be exported in tabular form, but it is not currently possible to export results for groups of elements such as tanks or pumps. The programmer's toolkit, which comprises over 50 functions, allows more flexibility. However, analyzing or visualizing EPANET data using the toolkit requires writing and compiling a new program that calls toolkit functions. The command line interface provides core simulation capability but not functionality for subsequent analysis or visualization. These pieces of functionality—statistical summaries, graphics, customized queries—and many others are available on a technical computing platform such as the R environment.

R is a freely available and 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 many scientific applications. Tippmann (2015) reports that nearly 1% of articles indexed in Scopus during 2014 explicitly cited R or an R package. One reason for this popularity is R's system for managing and distributing add-on packages. Extensions to the base

E-mail address: bradley.eck@ie.ibm.com.

http://dx.doi.org/10.1016/j.envsoft.2016.06.027
1364-8152/

distribution of R may be provided in a package that conforms to a specified format. Packages that comply with formatting and other requirements are eligible for distribution through the Comprehensive R Archive Network (CRAN). Packages on CRAN install over the web from inside an R session. With a clear structure and convenient distribution channel, R packages allow research communities to build and share tools.

As of February 2016, CRAN has nearly 8000 different R packages including several focused on water resources applications. The United States Geological Survey distributes the dataRetrieval package (Hirsch and Cicco, 2015) for retrieving hydrologic and water quality data from the web. The wq package by Jassby and Cloern (2015) provides functions to process and explore data, particularly water quality data, from environmental monitoring programs. Turner and Galelli (2016) developed the reservoir package to design, analyze, and operate water supply storages. The epanetReader package described herein adds to the list of R packages for water resources modeling.

The remainder of this paper treats the design and usage of epanetReader. The package defines several R objects that correspond to EPANET files. The design of these objects and some details of their implementation are discussed first. The next section illustrates usage of the package with the base distribution of R. Knowledge of around five R programming commands is sufficient to get started. Examples include a novel visualization of simulation results using sparklines and complement material in a forthcoming conference paper (Eck, 2016). One advantage to working in R is access to other packages and so usage of epanetReader with several other packages is also described. Finally, the paper concludes with some reflections on the work and discussion of future directions.

2. Design

The design of epanetReader aims at making interactive analysis of water networks accessible to new users of R. Although R is growing in popularity for a range of problems in the water domain, it may not be widely used by the community that also uses EPANET. The design aims to fit within the existing R ecosystem by providing a small number of new functions that complement the base distribution.

The primary new functions provided by the package are for parsing text files in EPANET formats into R objects. Three file types are supported: network simulation inputs in .inp format; hydraulic simulation results in .rpt format; and, multi-species simulation results in EPANET-MSX's .rpt format. The package functions read.inp(), read.rpt(), and read.msxrpt() parse the files into R objects. The function names follow the convention of R's existing read.csv() and read.table() functions for formatted text files.

The read.inp() function creates an R object of class epanet.inp. The structure of the object mirrors the structure of an .inp file. An .inp file is a text file of network simulation inputs organized into named sections (Rossman, 2000). Example sections include tanks, junctions, pipes, and pumps. The object is a named list with an entry for each section. List entries use different data types. The junctions table is stored as a data.frame. The title section is stored as a character vector. By adopting the structure of the source file, epanet.inp objects have a design familiar to EPANET users.

The read.rpt() function creates an object of class epanet.rpt. In contrast to epanet.inp objects, the structure of epanet.rpt objects differs somewhat from the structure of the source file. The .rpt files generated by EPANET can include a table of results for network nodes and a table of results for network links at every reporting period for the simulation (Rossman, 2000). Instead of storing each table in the file as an element in a list, all of the node results are combined. Link results are treated in a similar way. Thus, an epanet.rpt object is a list with entries for node results and link results. Combining results across time periods facilitates time series plots and analysis.

To further enable analysis of simulation results, data.frames of node and link results in epanet.rpt objects contain columns beyond those appearing in the file. First, a column is added for the type of node or link. Second, a column is added to store the simulation time stamp. The .rpt file generated by EPANET shows the time for each table of results in a clock format (HH:MM:SS). As results are combined across time steps, this time is stored as an additional column. A third column stores the simulation time in seconds. The time in seconds is computed by conversion from the clock format. These additions were found useful to interrogate and visualize simulation results.

The function read.msxrpt() parses the results of an EPANET-MSX simulation into an R object of class epanet.msxrpt. The object's design follows that of epanet.rpt objects: a list containing a data.frame for node results and data.frame for link results. Results are combined across time-steps. By using the same object design for results from EPANET and EPANET-MSX, the package provides a consistent interface to simulation results.

In addition to functions for reading files, epanetReader provides functions to manipulate and visualize objects created by reading EPANET files and a selection of helper functions (Table 1). A textual summary of an object is obtained by calling the summary() function on the object. Similarly, the plot() function creates a plot of the object. Summary and plot are examples of generic functions where the method invoked depends on the first argument. epanetReader provides implementations of these functions for objects created by reading EPANET files. As noted in the table, the package also provides helper functions to facilitate plotting.

3. Example usage and capability

The primary functionality of the package is to read text files from EPANET into R in a way that is consistent with other functions for reading data and that enables further analysis. The following examples illustrate capabilities the package adds to the base R distribution.

3.1. Installation and data import

In order to use the package in an R session it must be installed and loaded. After installation and loading, the package functions including read.inp(), read.rpt() and read.msxrpt() are available. All of these functions take a single argument, the path to the file, and return a named list. The list contains data from the file and has the class attribute set to the corresponding value. The following commands install and load the package, read the results of simulating example network 1, query the names of elements in the object, and get the object's class attribute (Listing 1).

```
> install.packages("epanetReader")
> library(epanetReader)
> Net1rpt <- read.rpt("Net1.rpt")
> names(Net1rpt)
[1] "nodeResults" "linkResults"
> class(Net1rpt)
[1] "epanet.rpt"
```

Listing 1: Installation and data import