An example and exercise on the usage of the R-package SPG

INTRODUCTION

Exercise 1 is aimed at understanding the structure and possibilities of the package SPG. The complete, commented code is prepared in the document `SPG_v1_0_example.R` or can be found in the documentation of the SPG package by typing "?SPG" in the R-console or RStudio. Exercise 1 consists of reading, understanding and running the existing code. Exercise 2 is a very small exercise for which a new sampling setup must be entered.

EXERCISE 1:

1. System under consideration

Figure 1 is an abstract representation of the catchment under investigation. The population size in the entire catchment is 50,000. G1 is the main catchment and drained by a gravity system towards the sampling point S which is at the influent of the local sewage treatment plant (STP). Sub-catchments G2-4 are also drained by gravity systems but their effluents are collected in pump sumps where the wastewater needs to be lifted locally to be transported to the STP. The relevant characteristics are summarized in Table 1. A questionnaire to obtain these pieces of information can be found in Castiglioni et al. (ES&T 2013).

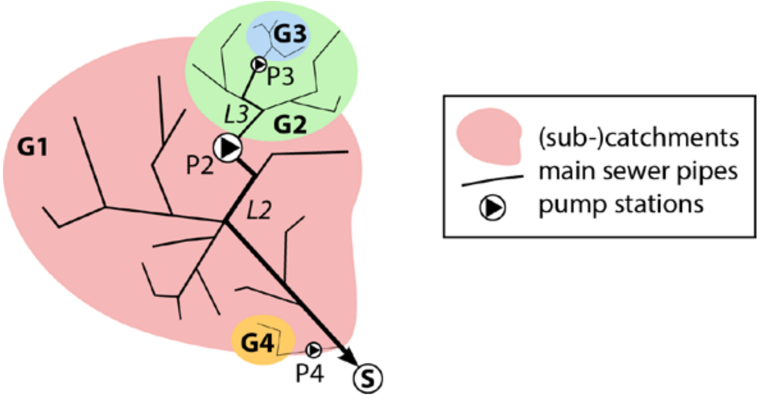

Figure 1. Scheme of the catchment used in the subsequent example.

Table 1. Properties of the catchment under investigation.

Element	Type	Population [-]	Mean flow distance within the catchment [m]	Fraction of consumers of substance of interest (scenario S1)	Expected number of pulses containing the substance of interest (scenario S2)
G1	Gravity system	40,000	3000	0.5%	-
G2	Gravity system	9,000	1000	0.5%	-
G3	Gravity system	1,000	250	0.5%	-
G4	Gravity system	500	100	0.5%	25

Table 2. Properties of the pumps in the catchment under investigation.

Element	Type	Storage capacity in the pump sump [m^3]	Pump rate [L/s]	Distance to the sampling point or the next pump [m]
P2	Pump station	200	100	3200 (L2)
P3	Pump station	10	25	1200 (L3)
P4	Pump station	2	5	100

2. Defining a gravity system

Code on how to define a gravity system:

```
G1.S1 <- def.grav(population=40000,
                     frac.consumers=0.005,
                     pulse.mass=5,
                     diurnal.S.WD=diurnal.variation.Q.S,
                     diurnal.S.WE=diurnal.variation.Q.S,
                     Q.c.day=160,
                     diurnal.Q=diurnal.variation.Q.S,
                     Q.base=0,
                     flow.distance=3000)
```

Note: If you have information on the diurnal variation of flow and the occurrence of substance S, you can enter this as a vector BEFORE calling the function, here: `diurnal.variation.Q.S`

3. Defining a pump

Code on how to define a pump:

```
P2 <- def.pump(V.max=200,
                V.min=1,
                pump.rate=100,
                distance=3200,
                v.flow=1,
                Disp=0.16)
```

Note: The distance that needs to be specified for a pump is the length of the sewer pipe after the pump to the next pump or to the point at which the substance patterns should be evaluated to assess or optimize the sampling setup (e.g. at the influent of the STP).

4. Model the entire sewer system

Code on how to model the sewer system from Figure 1:

```
flow.S1 = G1.S1(sim.dur, temp.res) +

          P2(G2.S1(sim.dur, temp.res) + P3(G3.S1(sim.dur, temp.res))) +

          P4(G4.S1(sim.dur, temp.res))
```

Note: The global variables sim.dur (the number of days that you want the simulation to be ran) and the temporal resolution must be defined BEFORE running these lines of code.

5. Plotting the modelled flows

Code to call the plot for modelled flows:

```
plot(flow.S1)
```

Note: You can select and plot individual days only, e.g. only days 5 and 6 by typing:

```
plot(flow.S1, days=2, from=4)
```

6. Take composite samples and assess the related uncertainty

Code to take samples from modelled flows:

```
samples.S1.15min <- take.samples(flow.S1,
                                 composite.duration=24,
                                 sampling.interval=15)
```

In this example we assume that the STP is equipped to take 24-hour composite samples in a time-proportional mode with a sampling interval of 15 minutes.

To plot the resulting uncertainties you can use the following code:

```
plot(samples.S1.15min, type="relative")    # the errors will be displayed as %
```

and the numeric values can be printed with

```
samples.S1.15min
```

Note: The errors can also be displayed as factors or absolute values:

EXERCISE 2:

The gravity sewer system G4 drains the premise on which a school is located. Imagine you would like to collect 24-hour composite samples to assess the use of ADHD among the students during exams. Unfortunately, you can only access the sewer and install a sampling device before the wastewater enters the pump sump of pump P4. The storage capacity of your sampling device only allows you to take a sample every 30 minutes. You expect about 25 toilet flushes containing the substance of interest. Define a scenario S2 and determine the expected sampling uncertainty with your experimental setup.

Note: The computationally expensive part is the calculation of the patterns. If you model various scenarios you can save them and load them later. In this way you can quickly assess new sampling setups without re-generating the patterns again.

Example:

```
save(flow.S1, file=“flow_S1.RData”)
```

then you can load the data quickly the next time by typing:

```
load(“flow_S1.RData”)
```