Tutorial for using hydroPSO to calibrate TUWmodel

Study area: Trancura River Basin, Chile

Mauricio Zambrano-Bigiarini Oscar M. Baez-Villanueva

February 29th, 2020

Contents

1 Installation

Installing the latest stable version of `hydroPSO` and `TUWmodel`.

```
install.packages("hydroPSO")
install.packages("TUWmodel")
```

Alternatively, you can also try the under-development version of `hydroPSO` (from Github):

```
if (!require(devtools)) install.packages("devtools")
library(devtools)
install_github("hzambran/hydroPSO")
```

Installing the latest stable version of `hydroTSM` and `hydroGOF`. These packages will be used to evaluate the performance of the simulatied streamflow.

```
install.packages("hydroTSM")
install.packages("hydroGOF")
```

2 hydroPSO

`hydroPSO` is a multi-OS and model-independent package based on the Particle Swarm Optimisation technique (PSO; Kennedy and Eberhart 1995) designed to allow the user to perform *i*) model calibration; *ii*) sensitivity analysis; and *iii*) an assessment of the results. This package is fully compatible with calibration tools employing PEST-like files and allows parallelisation.

PSO is a population-based stochastic optimisation technique used to explore a delimited search space with a *swarm* of particles to find the best set of parameters required to maximise a defined objective function. The search space is explored based on individual and neighbourhood-based best-known particle positions starting with a random initialisation of the particles' position and velocities within the parameter space. The position and velocity of each particle are updated taking into consideration its actual values and the location of the best-known optimum in the proximities. The positions and velocities of the particles evolve in time until a user-defined criterion is met (e.g., tolerances are lower than a threshold, or a maximum number of iterations). For a detailed description of the `hydroPSO` package please refer to Zambrano-Bigiarini and Rojas (2013).

3 TUWmodel

`TUWmodel` is a hydrologic model developed by the Technical University of Vienna (Viglione and Parajka 2019) that works at the daily or hourly temporal scales and follows the structure of the HBV model (Bergström 1995). The simulation of the hydrological cycle includes the accumulation of snow, the change of humidity in the soil profile, and the surface flow in the drainage network. It was validated over 320 basins in Austria (Parajka, Merz, and Blöschl 2007) and has been used in several studies (e.g., Ceola et al. 2015; Sleziak et al. 2016, 2018, 2020; Parajka et al. 2016; Nijzink et al. 2016, 2018; Zessner et al. 2017; Cisty and Soldanova 2018; Melsen et al. 2018). Sleziak et al. (2016) evaluated `TUWmodel` in several basins, and reported that it shows good performance in watersheds with snow cover. The parameters used by `TUWmodel` to represent the different hydrological processes of a basin are briefly described in Table 1. The range of values used to calibrate each of the parameters was taken from Viglione and Parajka (2019), which slightly modified the ranges proposed by Parajka, Merz, and Blöschl (2007).

ID	**Description**	**Units**	**Process**	**Range**
SCF	Snow correction factor	-	Snow	0.9 - 1.5
DDF	Degree-day factor	mm/°C/day	Snow	0.0 - 5.0
Tr	Threshold temperature above which precipitation is rain	°C	Snow	1.0 - 3.0
Ts	Threshold temperature below which precipitation is snow	°C	Snow	-3.0 - 1.0
Tm	Threshold temperature above which melt starts	°C	Snow	-2.0 - 2.0
LPrat	Parameter related to the limit for potential evaporation	-	Evaporation	0.0 - 1.0
FC	Field capacity	mm	Infiltration	0.0 - 600
BETA	Non-linear parameter for runoff production	-	Infiltration	0.0 - 20
cperc	Constant percolation rate	mm/day	Infiltration	0.0 - 8.0
k0	Storage coefficient for very fast response	day	Runoff	0.0 - 2.0
k1	Storage coefficient for fast response	day	Runoff	2.0 - 30
k2	Storage coefficient for slow response	day	Runoff	30 - 250
lsuz	Threshold storage state	mm	Runoff	1.0 - 100
bmax	Maximum base at low flows	day	Runoff	0.0 - 30
croute	Free scaling parameter	day^2/mm	Runoff	0.0 - 50

Table 1: Parameters used by *TUWmodel* to represent the different hydrological processes.

4 Model set up

The tutorial here presented explains the use of `hydroPSO` to calibrate the `TUWmodel` parameters rather than serving as a full hydrological analysis. Therefore, the figures generated in this tutorial are not analysed.

4.1 Loading packages

The *hydroPSO* and *TUWmodel* packages contain the data and functions used in this analysis. The *hydroTSM* and *hydroGOF* packages are also loaded to analyse the results and use the modified Kling-Gupta efficiency (KGE'; Gupta et al. 2009; Kling, Fuchs, and Paulin 2012) as the goodness-of-fit.

```
library(TUWmodel)
library(hydroPSO)
library(hydroTSM)
library(hydroGOF)
```

Time series of precipitation (P [mm/day]), air temperature (Temp [°C]), potential evaporation (PET [mm/day]), and streamflow (Q [m^3/s]) from 1979–2016 are provided in the *hydroPSO* package. These data belong to Trancura River Basin, which is located in the Araucan?a Region in Southern Chile.

```
data.drty.in     <- "./data"
Figures.drty.out <- paste0(model.drty, "/Figures")

# If the output directory selected to store the figures does not exists, it is created
if (!file.exists(Figures.drty.out)) dir.create(Figures.drty.out, recursive=TRUE)
```

Setting the calibration (1979–1997) and verification (1998–2016) periods.

```
### Calibration period
Cal.Ini <- "1979-01-01"
Cal.Fin <- "1997-12-31"

### Verification period
Ver.Ini <- "1998-01-01"
Ver.Fin <- "2016-12-31"
```

Loading the input data.

```
inputdata.fname <- "P_Tmean_PET_1979_2016.csv"
Qobs.fname      <- "Qobs_m3s_1979_2016.csv"
```

Storing the catchment area in m^2.

```
area <- 1415025887
```

Name and code of the catchment.

```
qobs.stationname <- "Trancura river"
qobs.ID          <- "09414001"
```

Observed discharge values in [m^3/s] from 01-Jan-1979 to 31-Dec-2016.

```
fname <- paste0(data.drty.in, "/", Qobs.fname)
x     <- read.csv(fname)
dates <- as.Date(x[,1])
qobs  <- zoo(x[,2], dates)
plot(qobs, xlab="Date", ylab=expression(paste("Q, [", m^3/s, "]")), main="Q Obs")
```

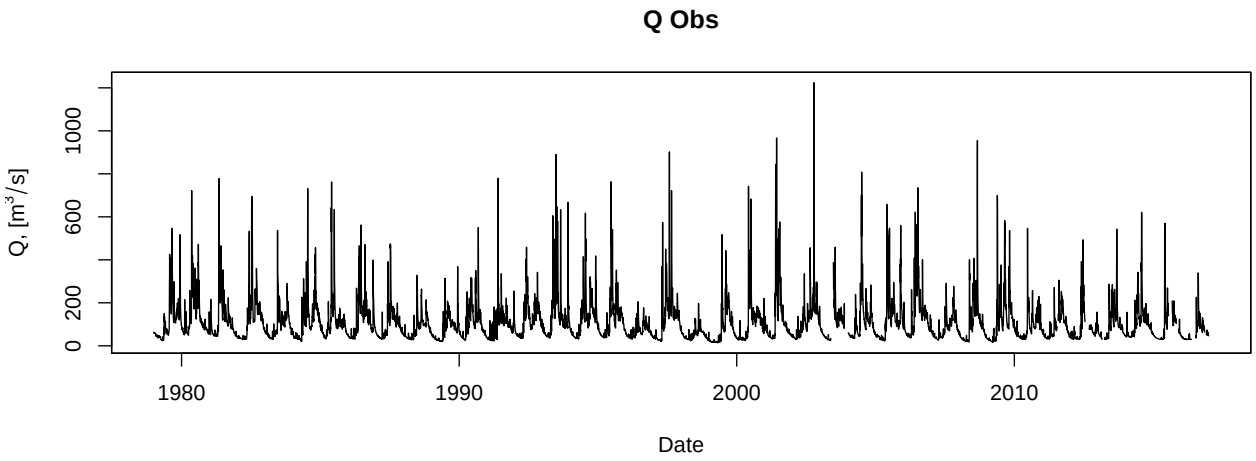

Figure 1: Daily sreamflow of Trancura River Basin in cubic meters per second for 1979–2016.

Converting the observed discharge from m^3/s to mm/day.

```
# 1 day = 86400 s
qobs <- (qobs * 1000 * 86400) / area
plot(qobs, xlab="Date", ylab="Q, [mm]", main="Q Obs")
```

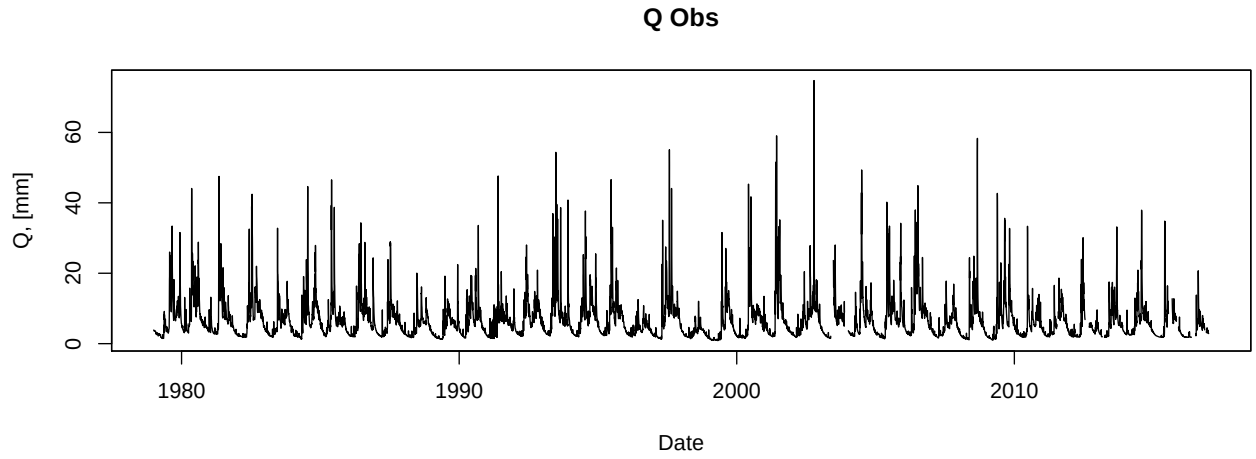

Figure 2: Daily sreamflow of Trancura River Basin in milimeters for 1979–2016.

Input time series of P, Temp, and PET, from 01-Jan-1979 to 31-Dec-2016.

```
fname <- paste0(data.drty.in, "/", inputdata.fname)
x     <- read.csv(fname)

dates <- as.Date(x[,1])
P     <- x[, 2]
Temp  <- x[, 3]
PET   <- x[, 4]
```

Transforming numeric time series into zoo objects.

```
P    <- zoo(P, dates)
Temp <- zoo(Temp, dates)
PET  <- zoo(PET, dates)
```

Subsetting the inputs and observed discharge values to the calibration period. This is recommended when you are not interested in uncertainty analysis during the verification period.

```
P.cal    <- window(P, start=Cal.Ini, end=Cal.Fin)
Temp.cal <- window(Temp, start=Cal.Ini, end=Cal.Fin)
PET.cal  <- window(PET, start=Cal.Ini, end=Cal.Fin)
qobs.cal <- window(qobs, start=Cal.Ini, end=Cal.Fin)
```

Plotting the P, Temp, and PET values.

```
plot(P.cal, xlab="Date", ylab="P [mm/day]", main="Precipitation")
```

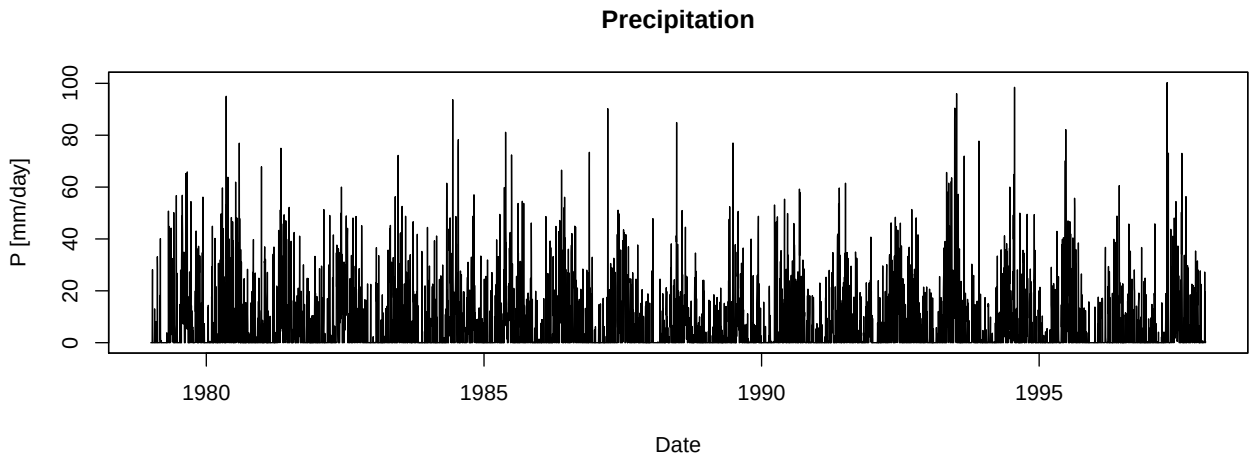

Figure 3: Daily averaged precipitation of Trancura River Basin for the calibration period.

```
plot(Temp.cal, xlab="Date", ylab="Temp [degC]", main="Temperature")
```

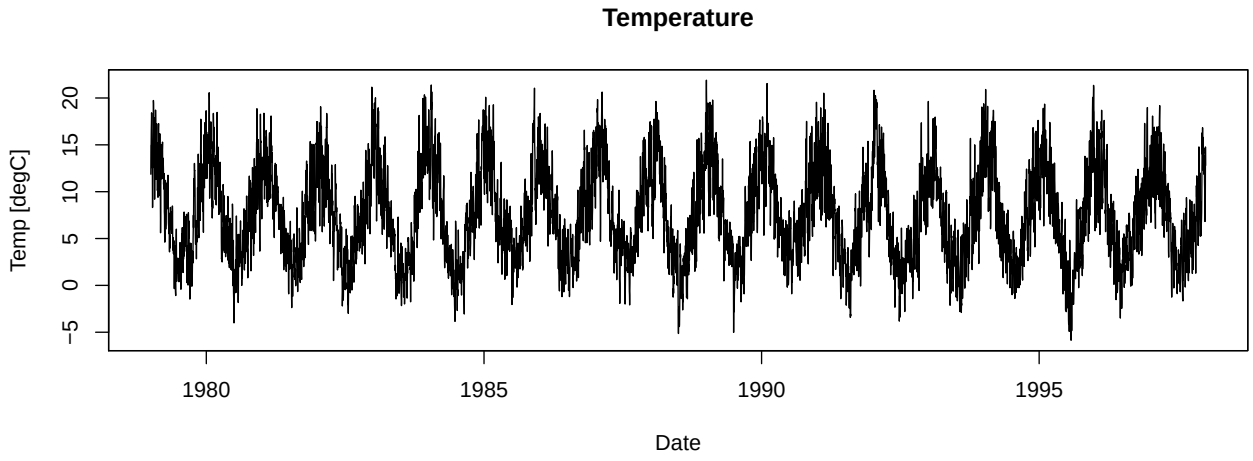

Figure 4: Daily averaged temperature of Trancura River Basin for the calibration period.

```
plot(PET.cal, xlab="Date", ylab="PET [mm/day]", main="Potential evaporation")
```

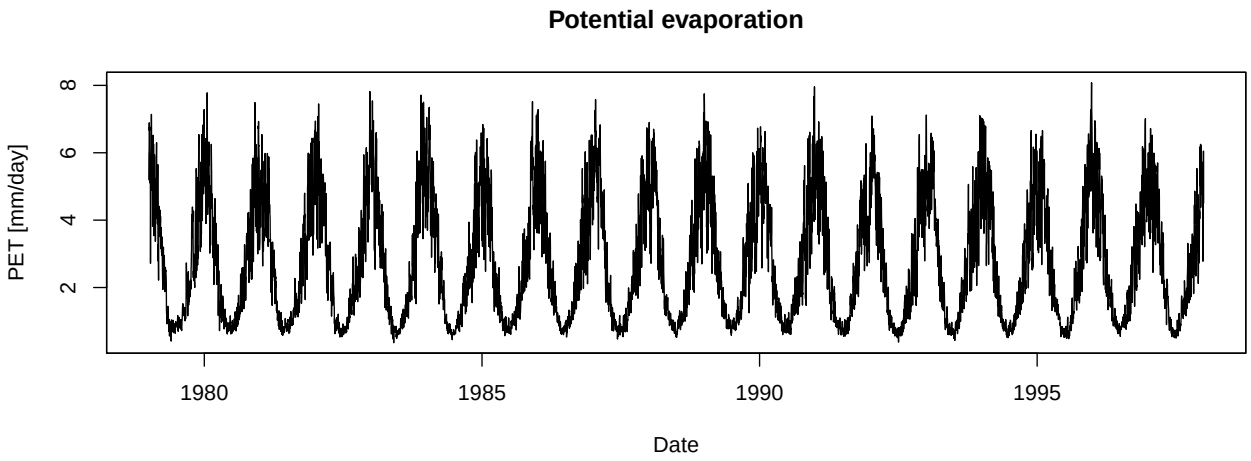

Figure 5: Daily averaged potential evaporation of Trancura River Basin for the calibration period.

Storing the dates of the input time series used to drive the simulations.

```
dates.cal <- time(P.cal)
```

Transforming the input time series into numeric, in order to allow the correct behaviour of TUWmodel (TUWmodel does not accept zoo objects as input).

```
P.cal    <- as.numeric(P.cal)
Temp.cal <- as.numeric(Temp.cal)
PET.cal  <- as.numeric(PET.cal)
```

4.2 Model parameters

Setting the lower and upper boundaries for model parameters as described in Viglione and Parajka (2019).

```
names <- c("SCF", "DDF", "Tr", "Ts", "Tm", "LPrat", "FC", "Beta",
           "k0", "k1", "k2", "lsuz", "cperc", "bmax", "croute")
lower <- c(0.9,   0.0,   1.0,  -3.0, -2.0,   0.0,     0,     0,
             0,   2,    30,     1,      0,      0,        0)
upper <- c(1.5,   5.0,   3.0,   1.0,  2.0,   1.0,    600,    20,
             2,   30,   250,   100,     8,      30,       50)

names(lower) <- names
names(upper) <- names
```

4.3 Running `TUWmodel`

4.3.1 Running `TUWmodel` with averaged parameter values

For this example, the `TUWmodel` parameters are set to the mid point between the lower and upper bound.

```
params <- lower + (upper - lower) / 2
qsim   <- TUWmodel(prec=P.cal,  airt=Temp.cal, ep=PET.cal, param=params)
```

Extracting simulated values.

```
qsim <- as.numeric(qsim$q)
```

Converting simulated and observed values from numeric into zoo objects.

```
qsim.cal <- zoo(qsim, dates.cal)
qobs.cal <- zoo(qobs, dates.cal)
```

Evaluating the simulated period.

```
ggof(sim=qsim.cal, obs=qobs.cal, ylab="Q, [mm]", cex=0.5)
```

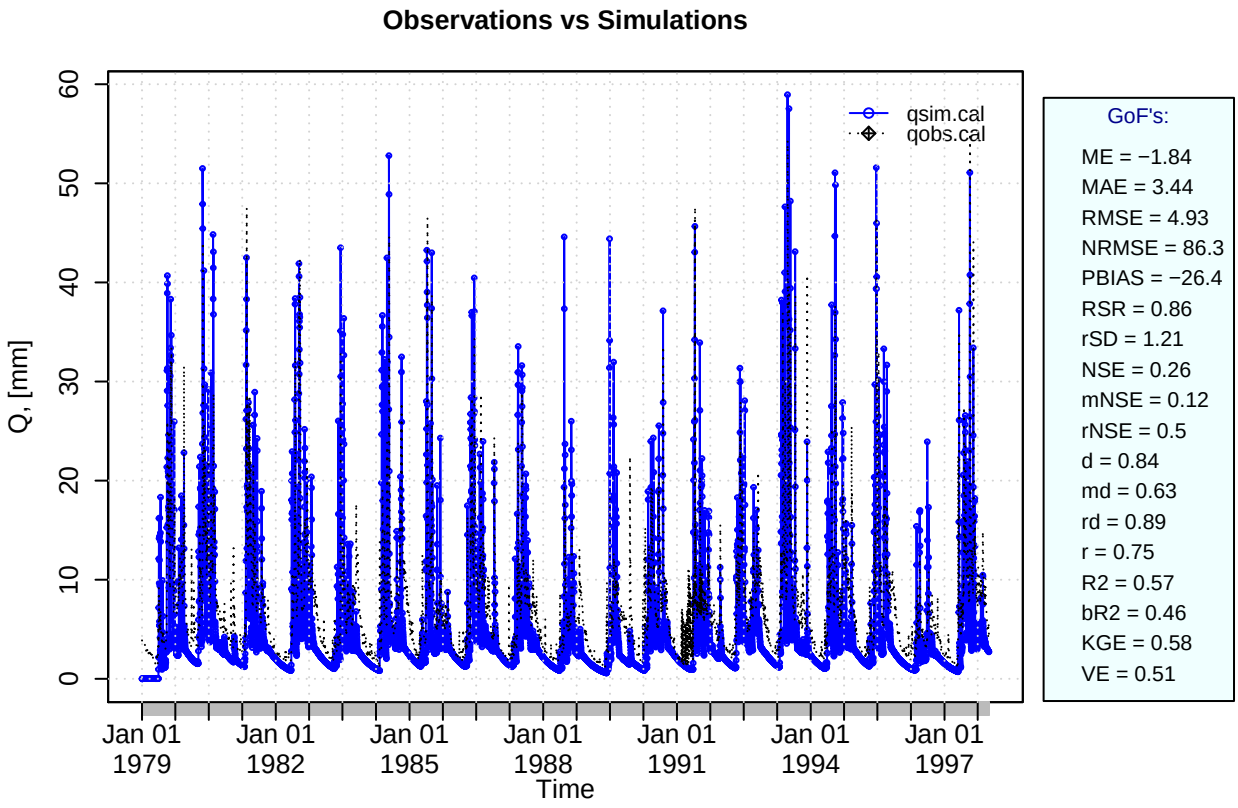

Figure 6: Evaluation of *qsim* for the calibration period using several performance indices.

4.3.2 Running TUWmodel with the default parameter set

In this example, the `TUWmodel` will run with the default parameters.

```
params <- c(1.02, 1.70, 2,0, -0.336, 0.934,
            121, 2.52, 0.473, 9.06, 142, 50.1, 2.38, 10,25)

qsim   <- TUWmodel(prec=P.cal, airt=Temp.cal, ep=PET.cal, param=params )
```

Extracting simulated values.

```
qsim <- as.numeric(qsim$q)
```

Converting simulated and observed values from numeric into zoo objects.

```
qsim.cal <- zoo(qsim, dates.cal)
qobs.cal <- zoo(qobs, dates.cal)
```

Evaluating the simulated period

```
ggof( sim=qsim.cal, obs=qobs.cal, ylab="Q, [mm]", cex=0.5)
```

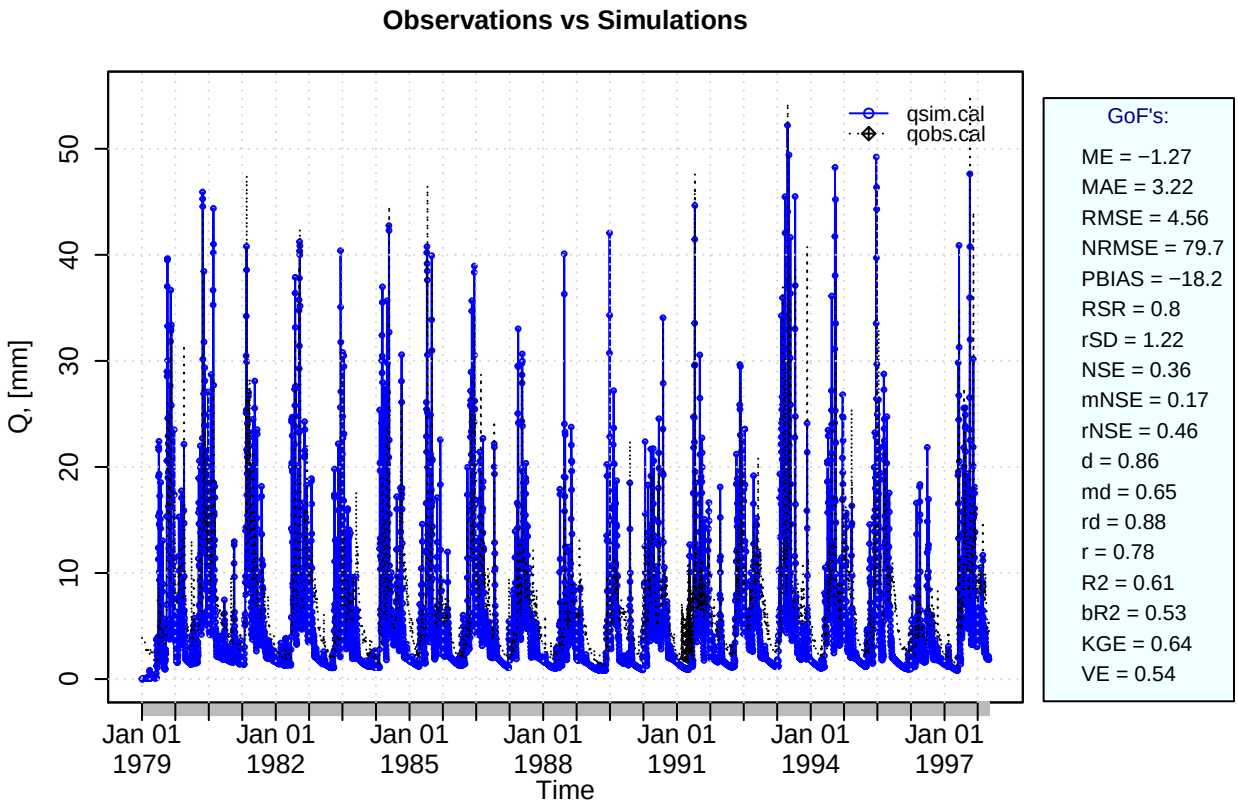

Figure 7: Evaluation of *qsim* for the calibration period using several performance indices.

5 Calibrating TUWmodel with hydroPSO

Subsetting the inputs and observed discharge values to the desired calibration period.

```
qobs.cal <- zoo(qobs, dates)
qobs.cal <- window(qobs.cal, start=Cal.Ini, end=Cal.Fin)
```

Building the function that will be used by the *hydroPSO* function.

```
TUWhydromod.CAL <- function(x) {

   simLump  <- TUWmodel(param=x, prec=P.cal, airt=Temp.cal, ep=PET.cal)

   qsim.cal <- zoo(as.numeric(simLump$q), dates.cal)

   return( KGE(sim=qsim.cal, obs=qobs.cal, method="2012") )

} # 'TUWhydromod.CAL' end
```

Running the *hydroPSO* optimisation (this may take few minutes!).

```
out <- hydroPSO(fn=TUWhydromod.CAL, lower=lower, upper=upper, method="spso2011",
                control=list(write2disk=TRUE, MinMax="max", npart=80,
                             normalise=TRUE, maxit=100, REPORT=10,
                             parallel="none", reltol=1E-10)
                )
```

Automatic plotting of the results into the graphic device.

```
plot_results()
```

Saving the best model parameters obtained by the optimisation.

```
best.param <- out$par
```

Running `TUWmodel` with the "optimum" parameter set (only for the calibration period).

```
simLump <- TUWmodel(param=best.param, prec=P.cal, airt=Temp.cal, ep=PET.cal)
```

Transforming the numeric output of the model into a zoo object.

```
qsim.cal <- zoo(as.numeric(simLump$q), dates.cal)
```

Evaluating the simulated period.

```
ggof(sim=qsim.cal, obs=qobs.cal, ylab="Q, [mm]", cex=0.5)
```

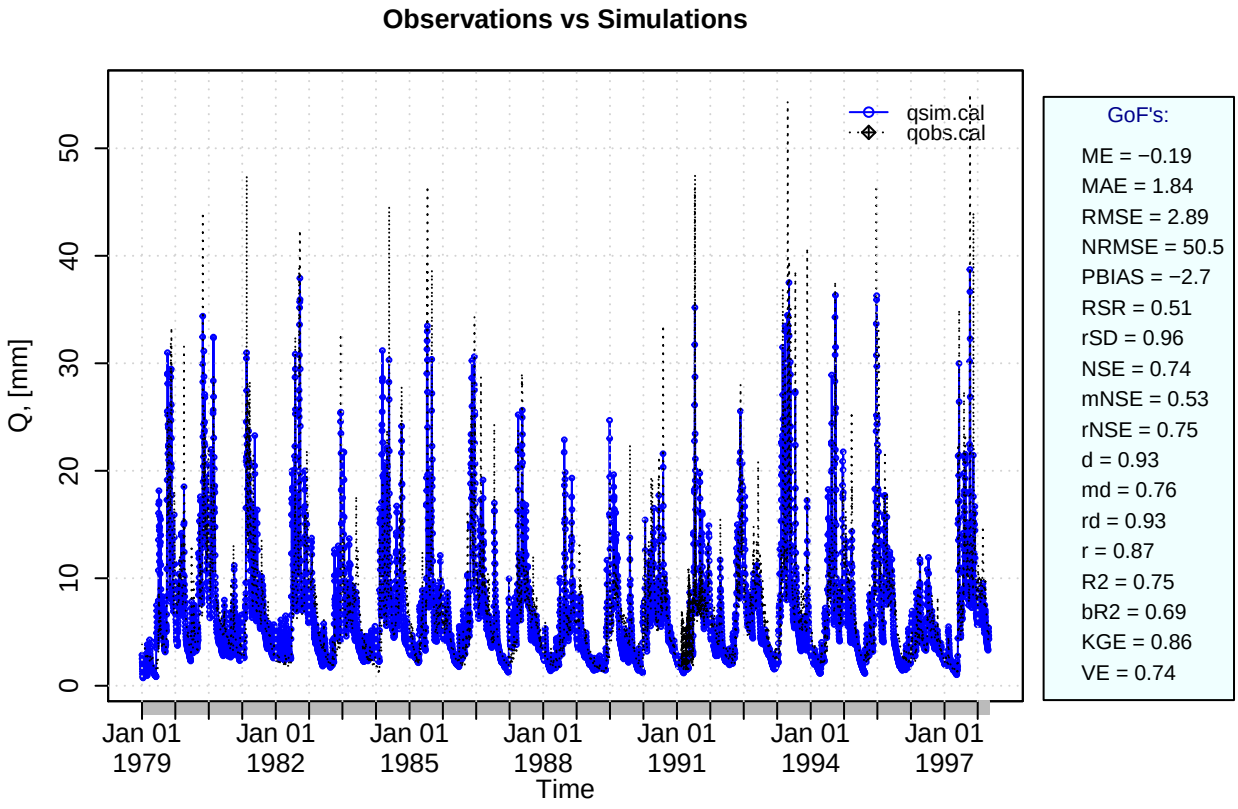

Figure 8: Evaluation of *qsim* for the calibration period using several performance indices.

5.1 Calibration results

Customised title for the figures.

```
main <- paste0("TUWmodel: ", qobs.stationname, " (", qobs.ID, ")")
```

Graphical comparison of daily and monthly simulated (*qsim.cal*) and observed (*qobs.cal*) values for the calibration period.

```
ggof(sim=qsim.cal, obs=qobs.cal, ftype="dm",
     FUN=mean, main=main, ylab="Q, [mm]", cex=0.5)
```

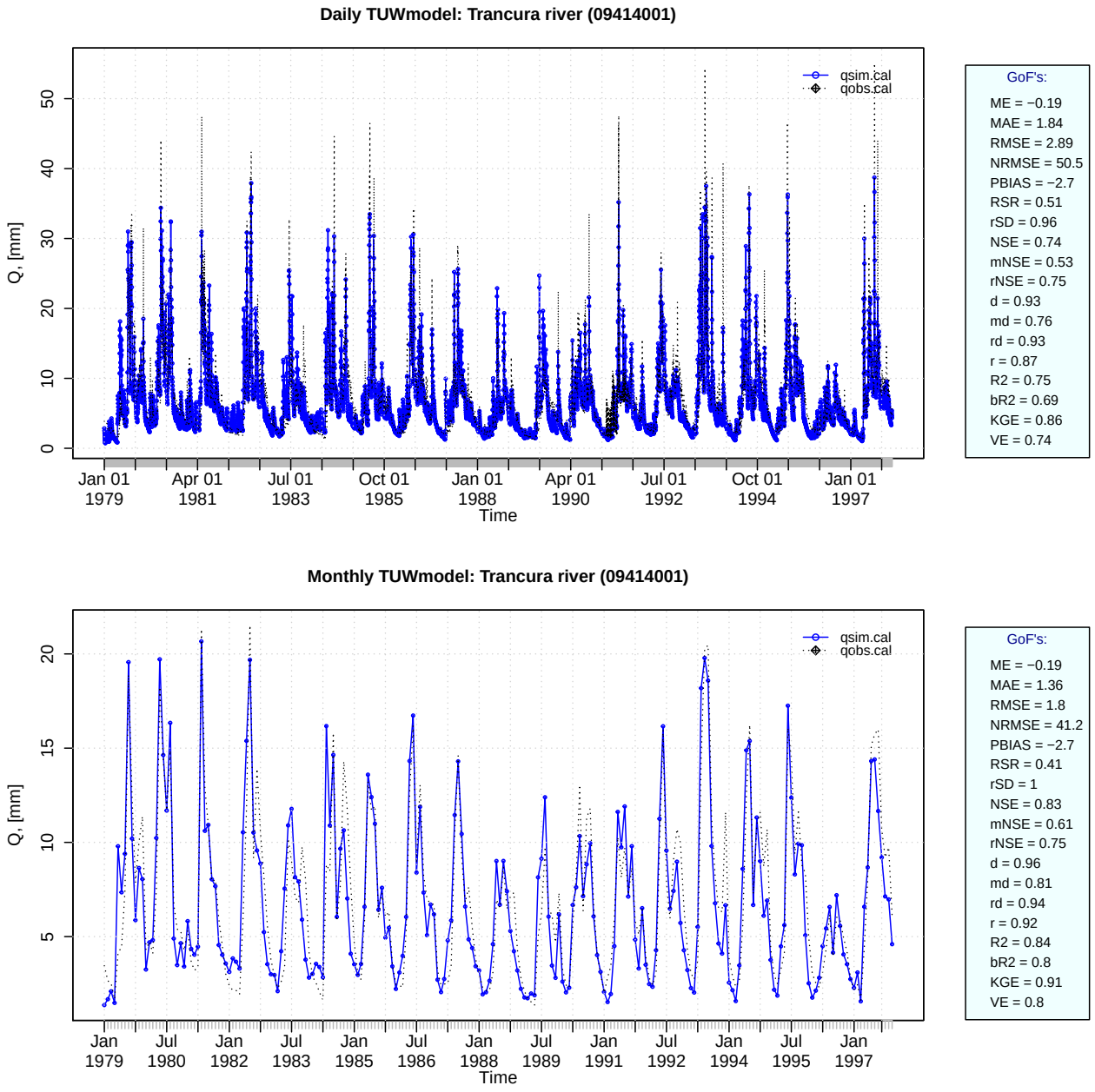

Figure 9: Daily and monthly evaluation of *qsim* for the calibration period using several performance indices.

Graphical comparison of monthly and annual simulated and observed values for the calibration period.

```
ggof(sim=qsim.cal, obs=qobs.cal, ftype="ma", pt.style="bar",
     FUN=mean, main=main, ylab="Q, [mm]")
```

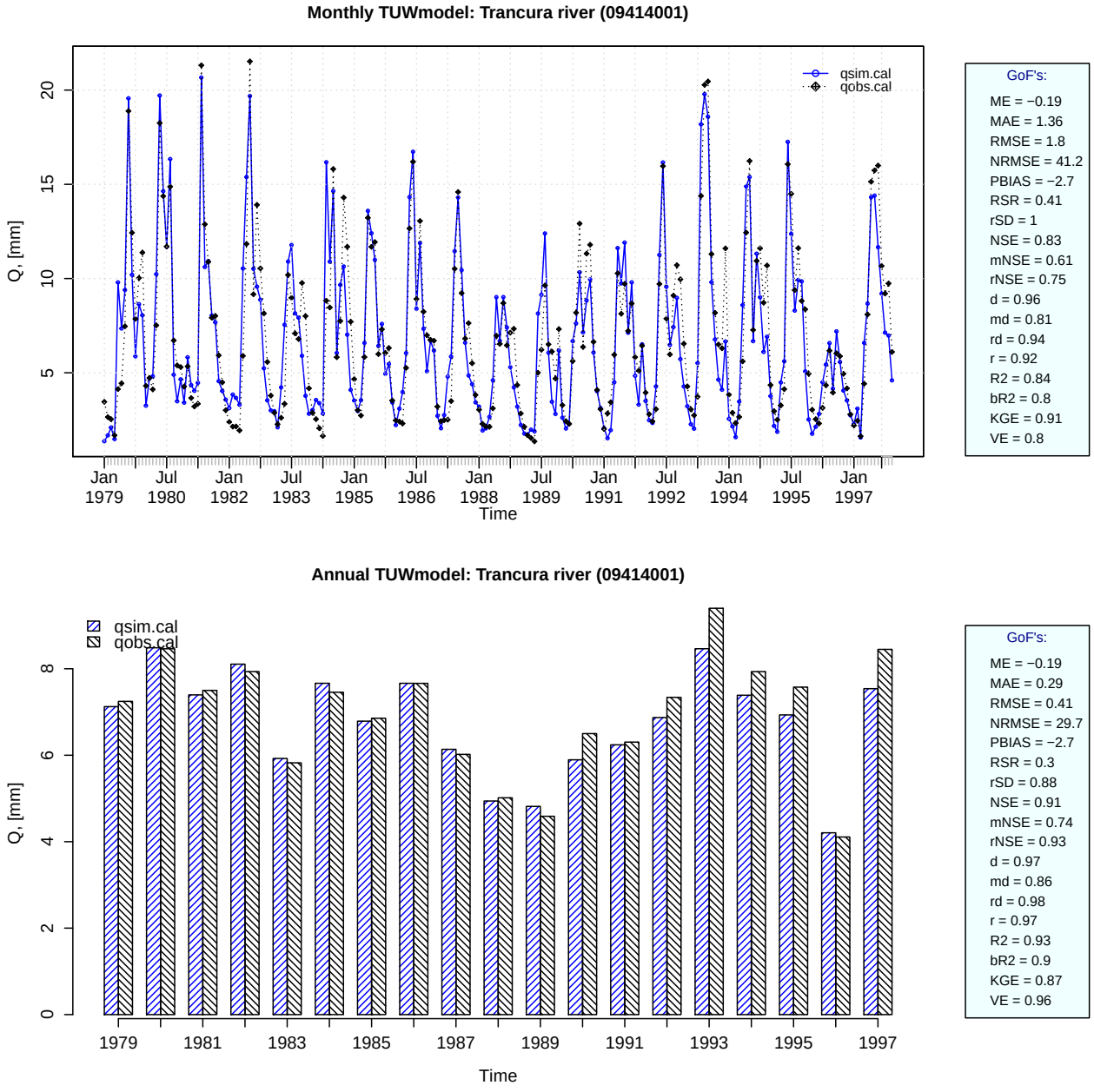

Figure 10: Monthly and annual evaluation of *qsim* for the calibration period using several performance indices.

Graphical comparison of seasonal simulated and observed values (one plot for each weather season).

```
ggof(sim=qsim.cal, obs=qobs.cal, ftype="seasonal", ylab="Q, [mm]",
     season.names=c("Summer", "Autumn", "Winter", "Spring"),
     FUN=mean, main=main, cex.main=2)
```

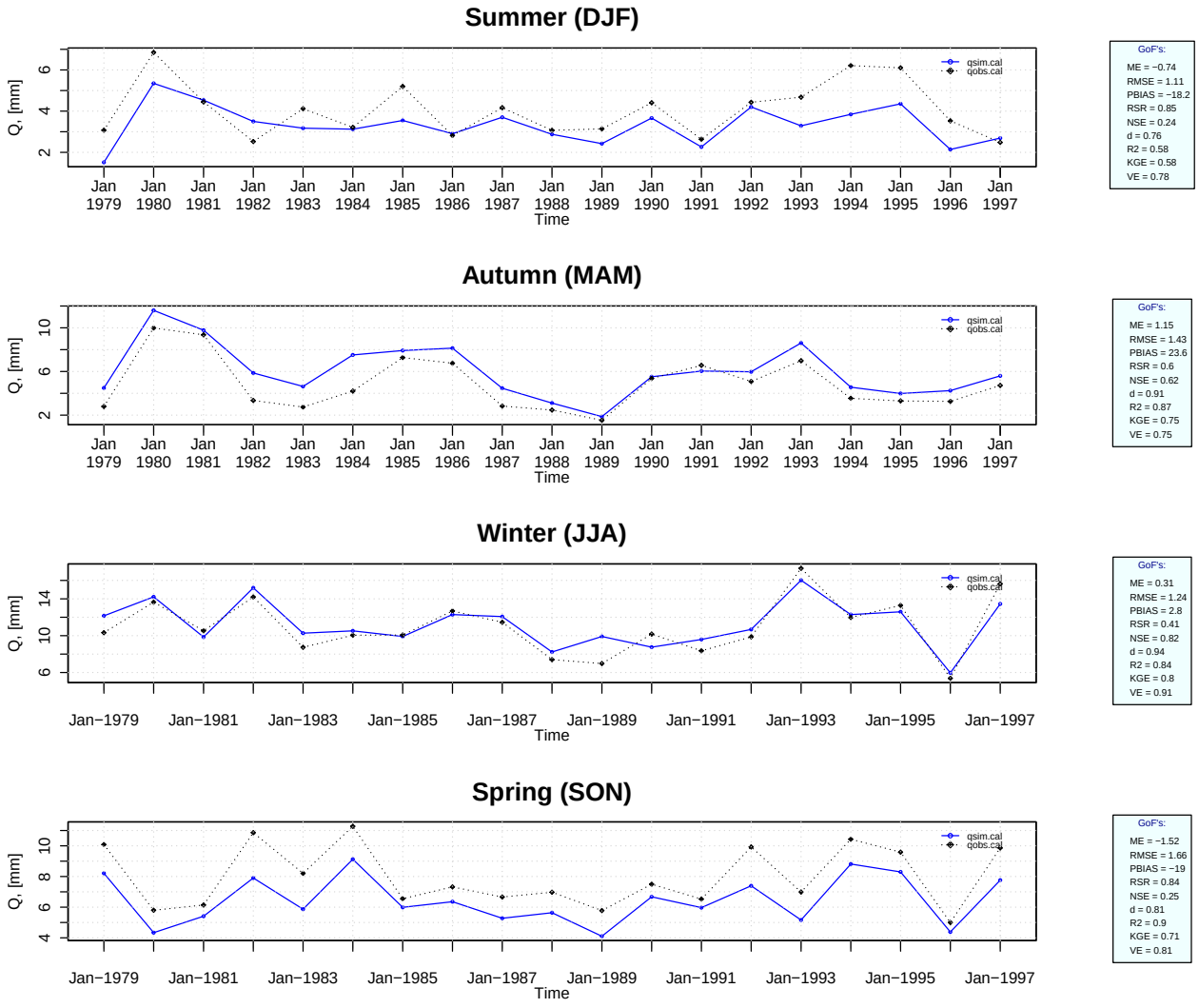

Figure 11: Seasonal evaluation of *qsim* for the calibration period using several performance indices.

Computing and plotting the daily residuals for the calibration period.

```
residuals <- qsim.cal - qobs.cal
```

Daily, monthly and annual plots, boxplots and histograms of the residuals.

```
hydroplot(residuals, FUN=mean, main=main, var.unit="mm")
```

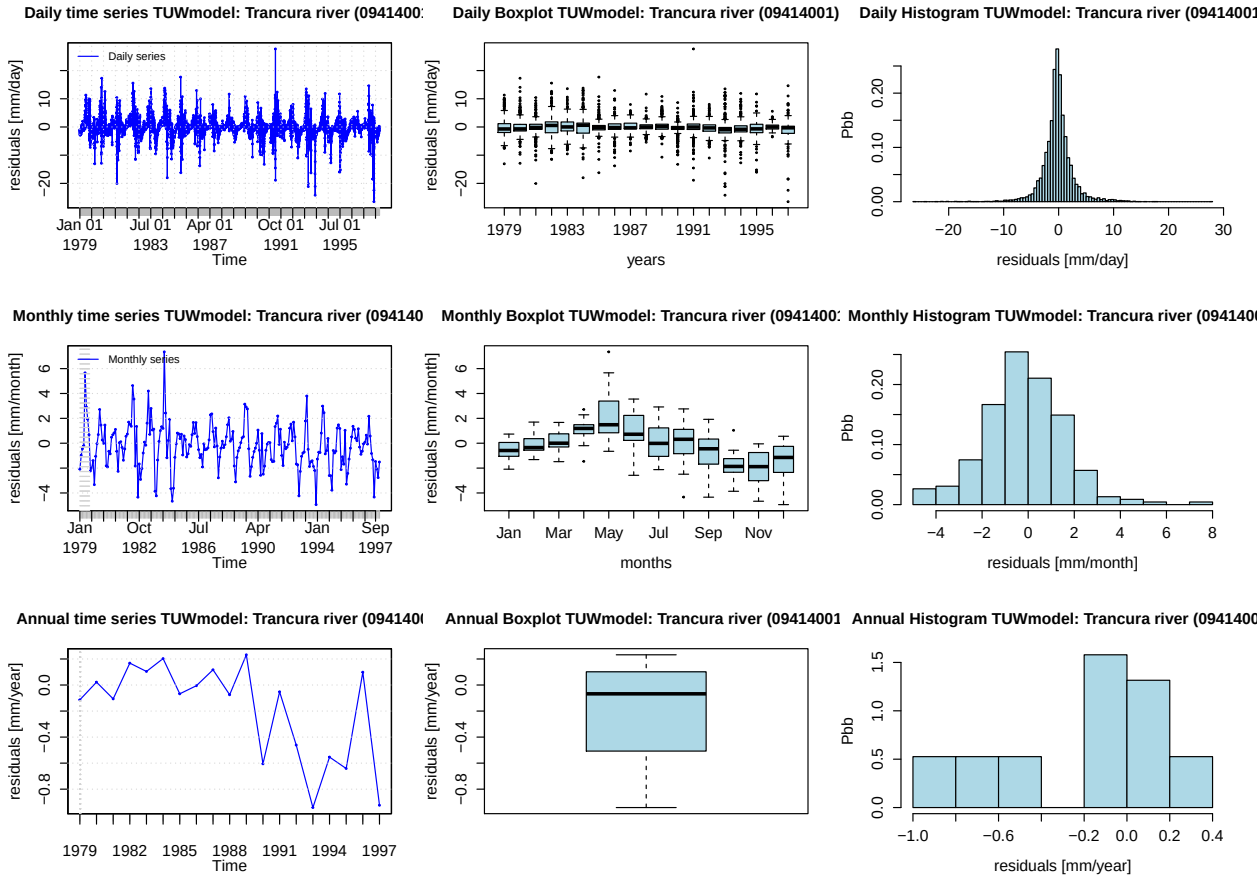

Figure 12: Analysis of the residuals (simulations - observations) for the calibration period at the daily, monthly, and annual temporal scales.

Seasonal plots and boxplots of the residuals.

```
hydroplot(residuals, FUN=mean, pfreq="seasonal",
          season.names=c("Summer", "Autumn", "Winter", "Spring"),
          h=0, main=main, cex.main=2)
```

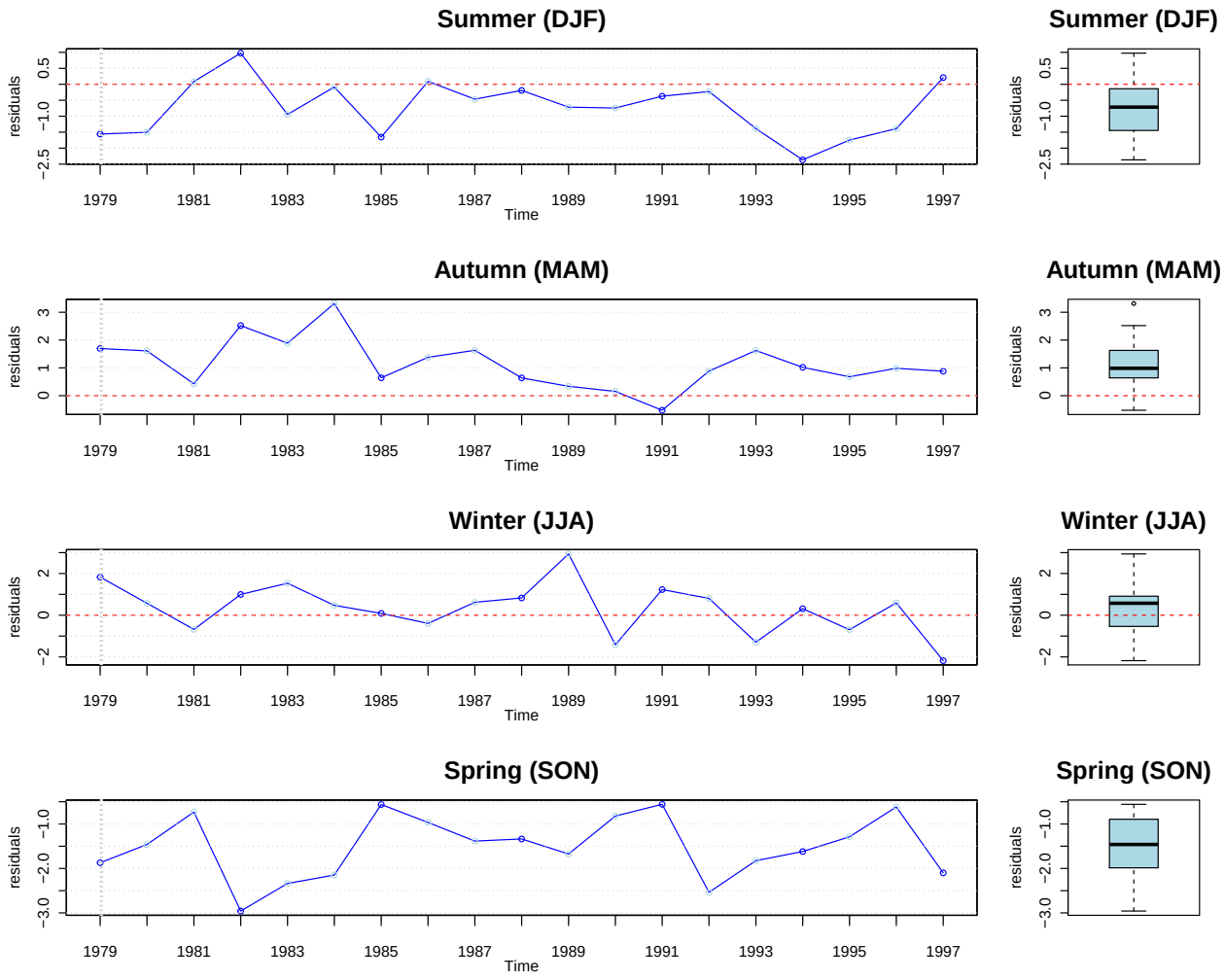

Figure 13: Analysis of the residuals (simulations - observations) for the calibration period at the seasonal scale.

5.2 Uncertainty analysis of results for calibration

Reading all the results of `hydroPSO`.

```
PSO.drty <- paste0(model.drty, "/PSO.out/")
res      <- read_results(drty.out=PSO.drty, MinMax="max", beh.thr=0.5)
```

```
## [                                                  ]
## [         Reading output files ...                 ]
## [                                                  ]
##
## [ Reading the file 'Particles.txt' ... ]
## [ Total number of parameter sets: 8000 ]
## [ Number of behavioural parameter sets: 6849 ]
##
```

```
## [ Reading the file 'Velocities.txt' ... ]
## [ Total number of parameter sets: 8000 ]
## [ Number of behavioural parameter sets: 6849 ]
##
## [ Reading the file 'Model_out.txt' ... ]
## [ Total number of parameter sets: 8000 ]
## [ Number of model outputs for each parameter set: 1 ]
## [ Number of behavioural model outputs           :  ]
##
## [ Reading the file 'ConvergenceMeasures.txt' ... ]
## [ Total number of iterations: 100 ]
## [ Number of iterations with Gbest >= 0.5: 100 ]
##
## [ Reading the file 'Particles_GofPerIter.txt' ... ]
## [ Number of particles : 80 ]
## [ Number of iterations: 100 ]
```

Assignments of the best parameter set ('best.param') for `TUWmodel`, the full set of parameters ('params'), and the goodness-of-fit ('gofs').

```
best.param <- res[["best.param"]]
params     <- res[["params"]]
gofs       <- res[["gofs"]]
```

Defining a wrapper function to run `TUWmodel` during the calibration period with each beahavioural parameter set. It returns a matrix with dimension [nrow(params), length(qsim.cal)].

```
TUWmodel2 <- function(x) {

  simLump  <- TUWmodel(param=x, prec=P.cal, airt=Temp.cal, ep=PET.cal)

  as.numeric(simLump$q)

} # 'TUWmodel2' end
```

Running TUWmodel for each one of the behavioural parameter sets.

```
qsim <- t( apply(X=params, MARGIN=1, FUN=TUWmodel2) )
```

Computing weighted quantiles of each column of a matrix containing streamflows simulated by different (behavioural) parameter sets. The weight applied to each streamflow (row) is the goodness-of-fit ('gofs') obtained by the parameter set used to compute those streamflows.

```
q025.q50.q975 <- wquantile(qsim, weights=gofs, byrow=FALSE, probs=c(0.025, 0.5, 0.975),
                           normwt=TRUE, verbose=FALSE)
```

```
q025 <- zoo(q025.q50.q975[,1], dates.cal)
q975 <- zoo(q025.q50.q975[,3], dates.cal)
```

Plotting the "best" simulated streamflows and the Prediction Uncertainty at 95% (95 PPU).

```
main.uncert  <- paste0("Uncertainty bounds TUWmodel: ",
                       qobs.stationname, " (", qobs.ID, ")")
qobs.col     <- "black"
best.sim.col <- "blue"
bands.col    <- "lightblue"

  plot(qobs.cal, xaxt="n", xlab="", type="n", main=main.uncert, ylab="Q, [mm]")
  drawTimeAxis(qobs)
  plotbandsonly(lband=q025, uband=q975)
  lines(qobs.cal, col=qobs.col, lwd=0.3)      # qobs
  lines(qsim.cal, col=best.sim.col, lwd=0.3) # best simulation
  grid()

  legend("topleft", legend=c("qobs", "best.sim", "95PPU"), lty=c(1, 1, NA),
         pch=c(NA, NA, 15), col=c(qobs.col, best.sim.col, bands.col), bty="n", cex = 0.7)
```

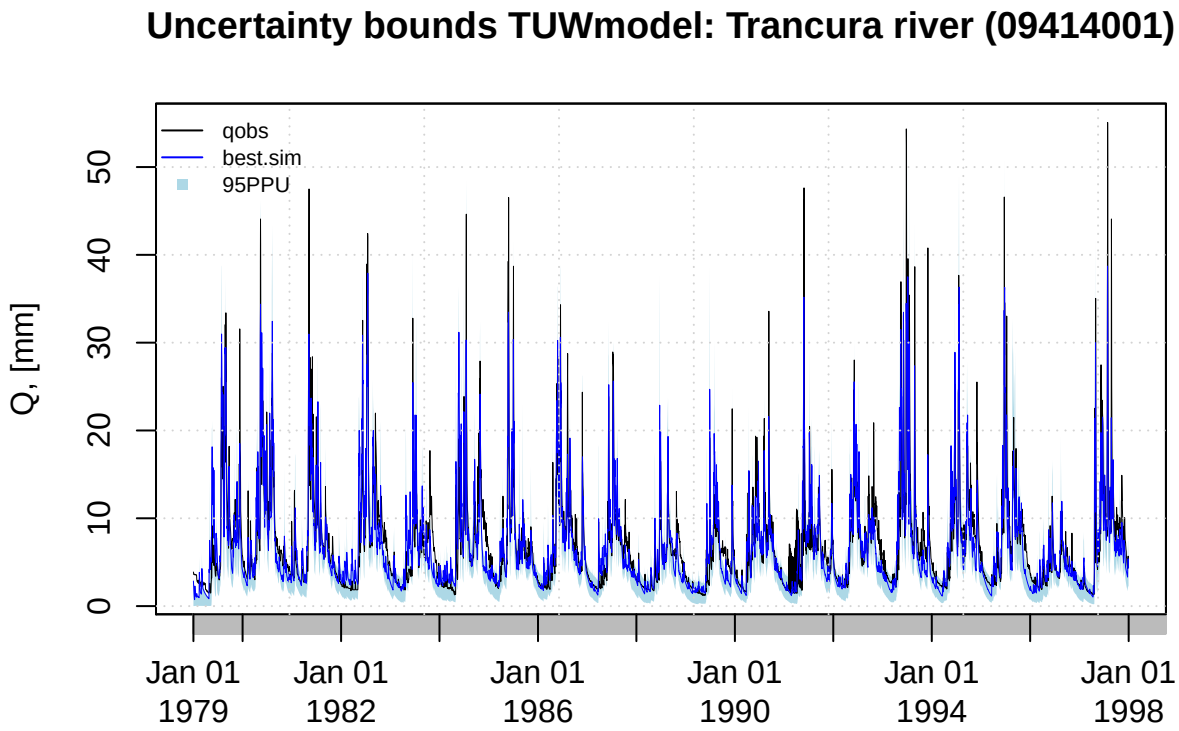

Figure 14: Uncertainty analysis for the calibration period.

P-factor is the percent of observations that are within the given uncertainty bounds.

```
pfactor(x=qsim.cal, lband=q025, uband=q975, na.rm=TRUE)
```

```
## [1] 0.9979827
```

R-factor is the average width of the given uncertainty bounds divided by the standard deviation of the observations.

```
rfactor(x=qsim.cal, lband=q025, uband=q975, na.rm=TRUE)
```

```
## [1] 0.9341486
```

6 Analysis for verification period

Subsetting the inputs and observed discharge values to the desired verification period.

```
P.val    <- window(P, start=Ver.Ini, end=Ver.Fin)
Temp.val <- window(Temp, start=Ver.Ini, end=Ver.Fin)
PET.val  <- window(PET, start=Ver.Ini, end=Ver.Fin)
qobs.val <- window(qobs, start=Ver.Ini, end=Ver.Fin)
```

Storing the dates of the input time series used to drive the simulations.

```
dates.val <- time(P.val)
```

Transforming the input ts into numeric to allow the correct behaviour of `TUWmodel` (it does not support zoo bojects as input).

```
P.val    <- as.numeric(P.val)
Temp.val <- as.numeric(Temp.val)
PET.val  <- as.numeric(PET.val)
```

6.1 Running TUWmodel with the "optimum" parameter set

Now, we will run `TUWmodel` for the entire period of calibration and verification:

Reading the results of the calibration with `hydroPSO` to extract the values of the "best" parameter set.

```
res <- read_results(MinMax="max", beh.thr=0.5, verbose=TRUE)
```

Getting the numeric values of the "best" parameter set.

```
best.param <- res[["best.param"]] # best parameter set
```

Getting the streamflow simulated with the "best" parameter set.

```
simLump <- TUWmodel(param=best.param, prec=P.val, airt=Temp.val, ep=PET.val)
```

Transforming the simulated streamflow from numeric into zoo bojects.

```
qsim <- zoo(as.numeric(simLump$q), dates.val)
```

Subsetting the simulated and the observed streamflow to the verification period.

```
qsim.val <- window(qsim, start=Ver.Ini, end=Ver.Fin)
qobs.val <- window(qobs, start=Ver.Ini, end=Ver.Fin)
```

6.2 Verification results

Graphical comparison of daily and monthly simulated (*qsim.cal*) and observed (*qobs.cal*) values for the verification period.

```
ggof(sim=qsim.val, obs=qobs.val, ftype="dm",
     FUN=mean, main=main, ylab="Q, [mm]", cex=0.5)
```

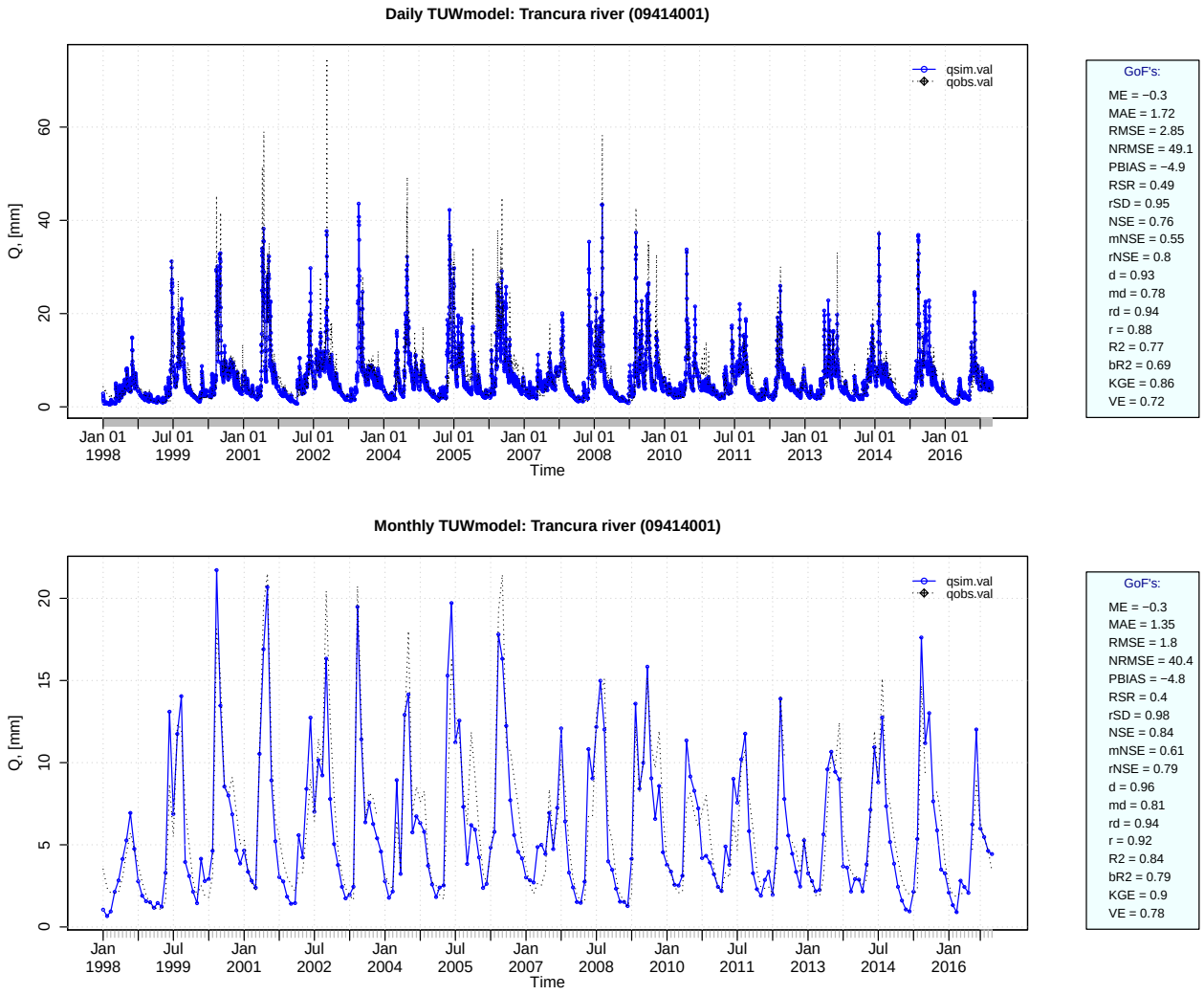

Figure 15: Daily and monthly evaluation of *qsim* for the verification period using several performance indices.

Graphical comparison of monthly and annual simulated and observed values for the verification period.

```
ggof(sim=qsim.val, obs =qobs.val, ftype="ma", pt.style="bar",
     FUN=mean, main=main, ylab="Q, [mm]")
```

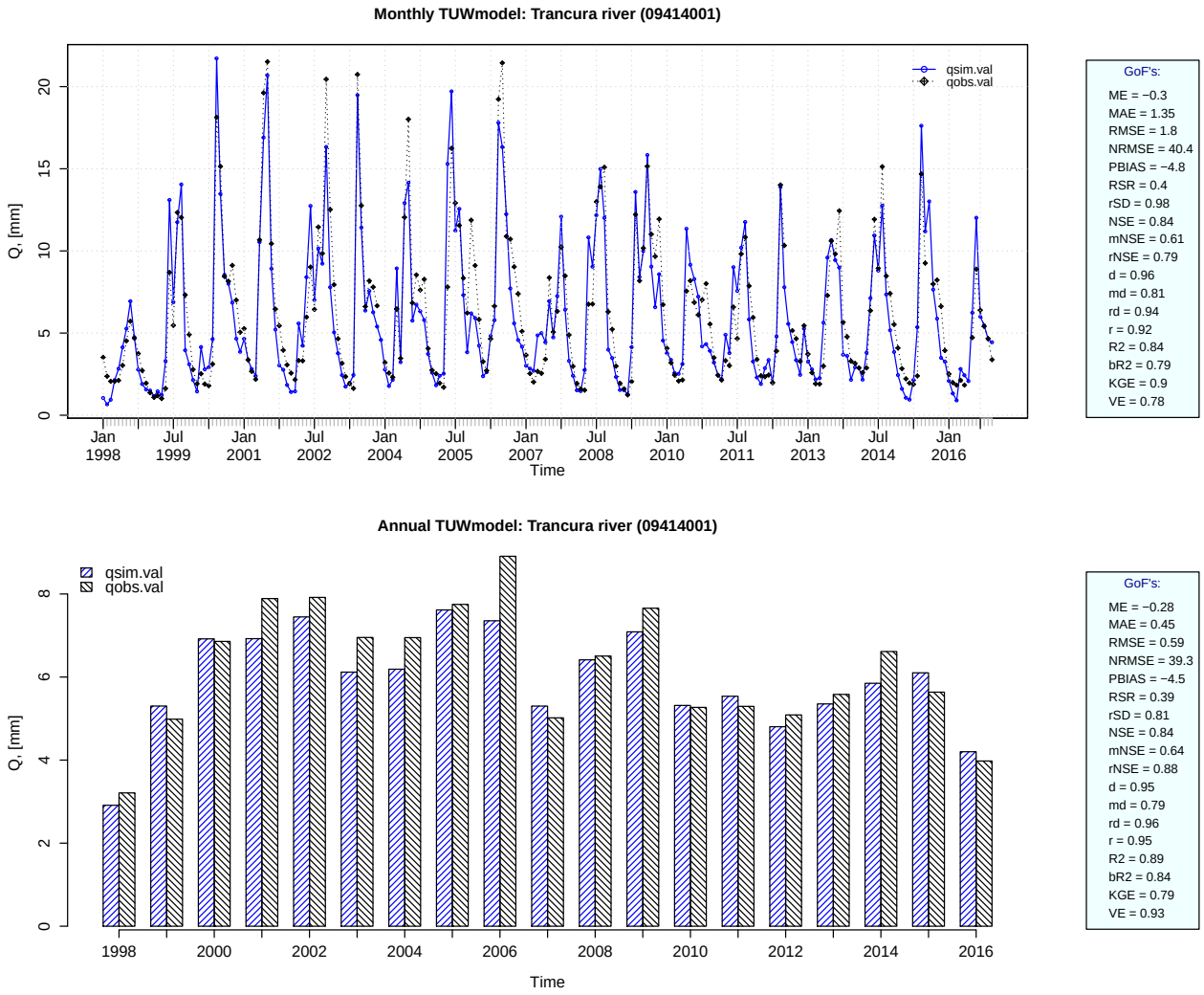

Figure 16: Monthly and annual evaluation of *qsim* for the verification period using several performance indices.

Graphical comparison of seasonal simulated and observed values for the verification period (one plot for each weather season).

```
ggof(sim=qsim.val, obs=qobs.val, ftype="seasonal", ylab="Q, [mm]",
     season.names=c("Summer", "Autumn", "Winter", "Spring"),
     FUN=mean, main=main, leg.cex=5)
```

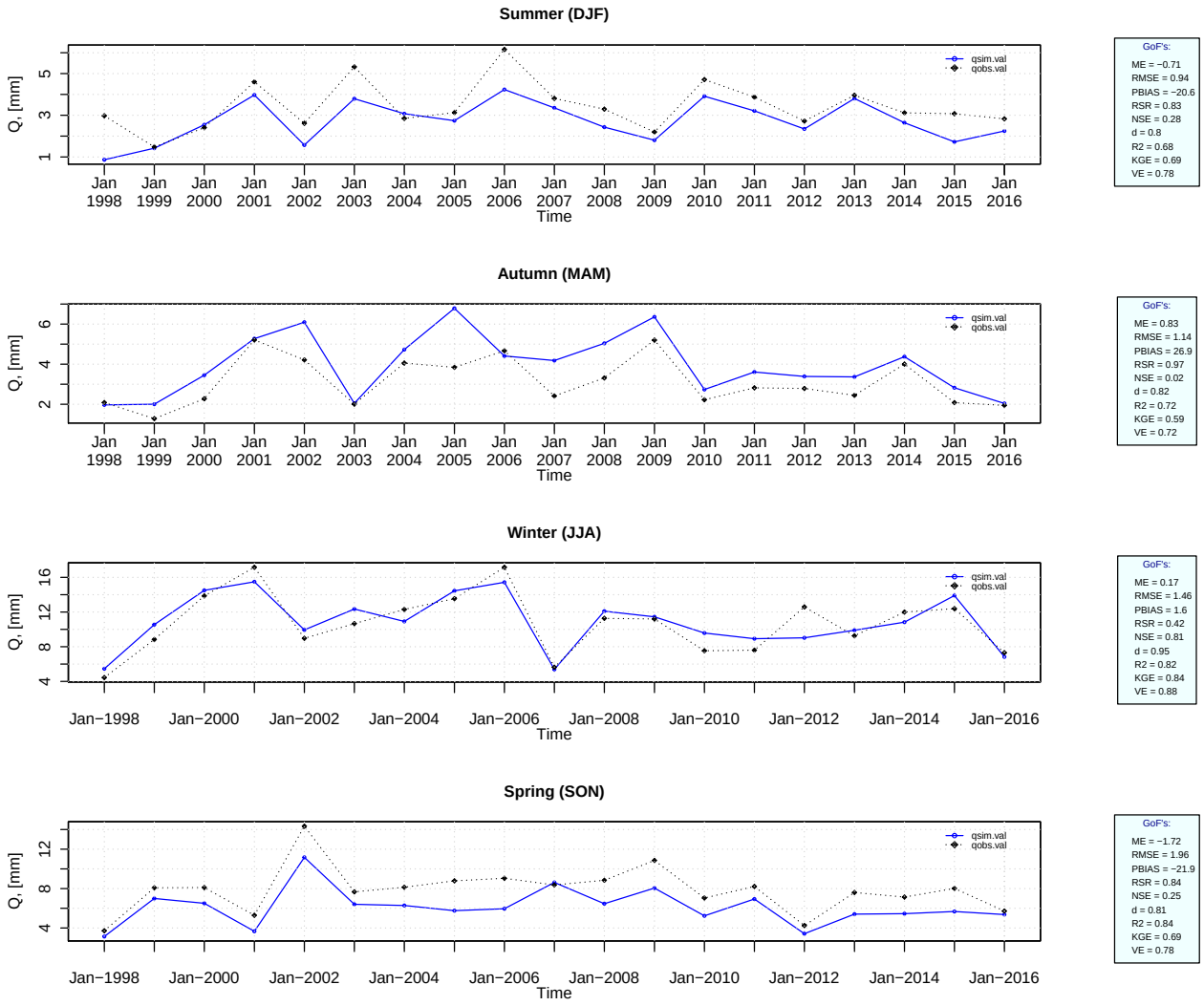

Figure 17: Seasonal evaluation of *qsim* for the verification period using several performance indices.

Computing and plotting the daily residuals.

```
residuals <- qsim.val - qobs.val
```

Daily, monthly and annual plots, boxplots and histograms of the residuals for the verification period.

```
hydroplot(residuals, FUN=mean, main=main, cex.main=1, var.unit="mm")
```

Seasonal plots and boxplots of the residuals for the verification period.

```
hydroplot(residuals, FUN=mean, pfreq="seasonal",
          season.names=c("Summer", "Autumn", "Winter", "Spring"),
          h=0, main=main, cex.main=2)
```

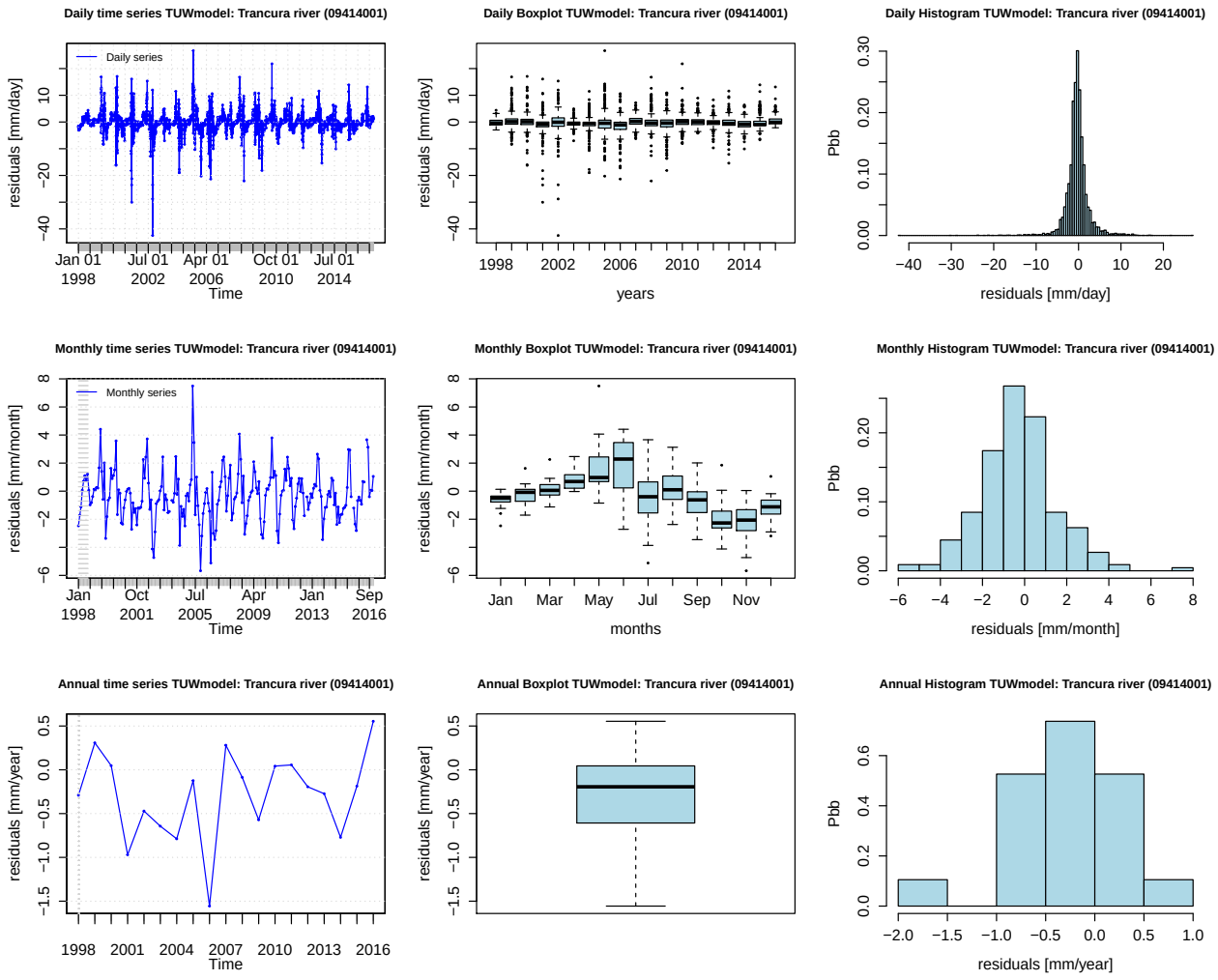

Figure 18: Analysis of the residuals (simulations - observations) for the verification period at the daily, monthly, and annual temporal scales.

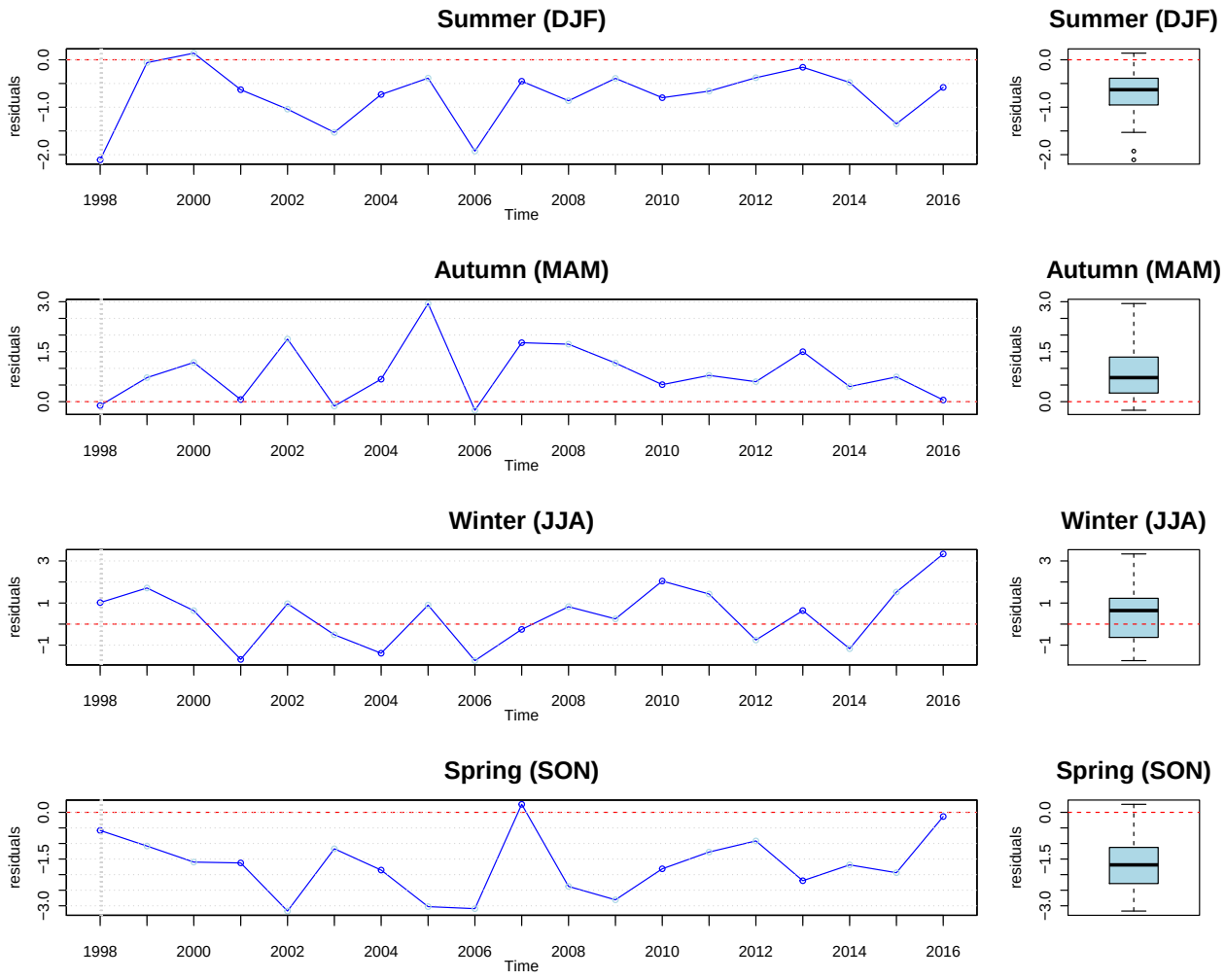

Figure 19: Analysis of the residuals (simulations - observations) for the verification period at the seasonal scale.

6.3 Uncertainty analysis of verification results

Reading all the results of `hydroPSO`.

```
res <- read_results(drty.out=PSO.drty, MinMax="max", beh.thr=0.5)
```

Assignments of the best parameter set ('best.param') for `TUWmodel`, the full set of parameters ('params') and the goodness-of-fit ('gofs').

```
best.param <- res[["best.param"]]
params     <- res[["params"]]
gofs       <- res[["gofs"]]
```

Defining a wrapper function to run `TUWmodel` during the verification period with each beahavioural parameter set. It returns a matrix with dimension [nrow(params), length(qsim.val)].

```
TUWmodel2 <- function(x) {

  simLump  <- TUWmodel(param=x, prec=P.cal, airt=Temp.cal, ep=PET.cal)
```

```
  as.numeric(simLump$q)

} # 'TUWmodel2' end
```

Running `TUWmodel` for each one of the behavioural parameter sets.

```
qsim <- t( apply(X=params, MARGIN=1, FUN=TUWmodel2) )
```

Computing weighted quantiles of each column of a matrix containing streamflows simulated by different (behavioural) parameter sets. The weight applied to each streamflow (row) is the goodness-of-fit ('gofs') obtained by the parameter set used to compute those streamflows.

```
q025.q50.q975 <- wquantile(qsim, weights=gofs, byrow=FALSE, probs=c(0.025, 0.5, 0.975),
                           normwt=TRUE, verbose=FALSE)

q025 <- zoo(q025.q50.q975[,1], dates.cal)
q975 <- zoo(q025.q50.q975[,3], dates.cal)
```

Plotting the "*best*" simulated streamflows and the 95 PPU.

```
main         <- paste0("Uncertainty bounds TUWmodel: ",
                       qobs.stationname, " (", qobs.ID, ")")
qobs.col     <- "black"
best.sim.col <- "blue"
bands.col    <- "lightblue"

  plot(qobs.cal, xaxt="n", xlab="", type="n",
       main=main, ylab="Q, [mm]")
  drawTimeAxis(qobs)
  plotbandsonly(lband=q025, uband=q975)
  lines(qobs.cal, col=qobs.col, lwd=0.3)     # qobs
  lines(qsim.cal, col=best.sim.col, lwd=0.3) # best simulation
  grid()
  legend("topleft", legend=c("qobs", "best.sim", "95PPU"), lty=c(1, 1, NA),
         pch=c(NA, NA, 15), col=c(qobs.col, best.sim.col, bands.col), bty="n", cex = 0.7)
```

P-factor is the percent of observations that are within the given uncertainty bounds.

```
pfactor(x=qsim.cal, lband=q025, uband=q975, na.rm=TRUE)
```

```
## [1] 0.9979827
```

R-factor is the average width of the given uncertainty bounds divided by the standard deviation of the observations.

```
rfactor(x=qsim.cal, lband=q025, uband=q975, na.rm=TRUE)
```

```
## [1] 0.9341486
```

7 Software details

This tutorial was built under:

```
## [1] "x86_64-pc-linux-gnu (64-bit)"
```

```
## [1] "R version 3.6.2 (2019-12-12)"
```

```
## [1] "hydroTSM 0.5-1"
```

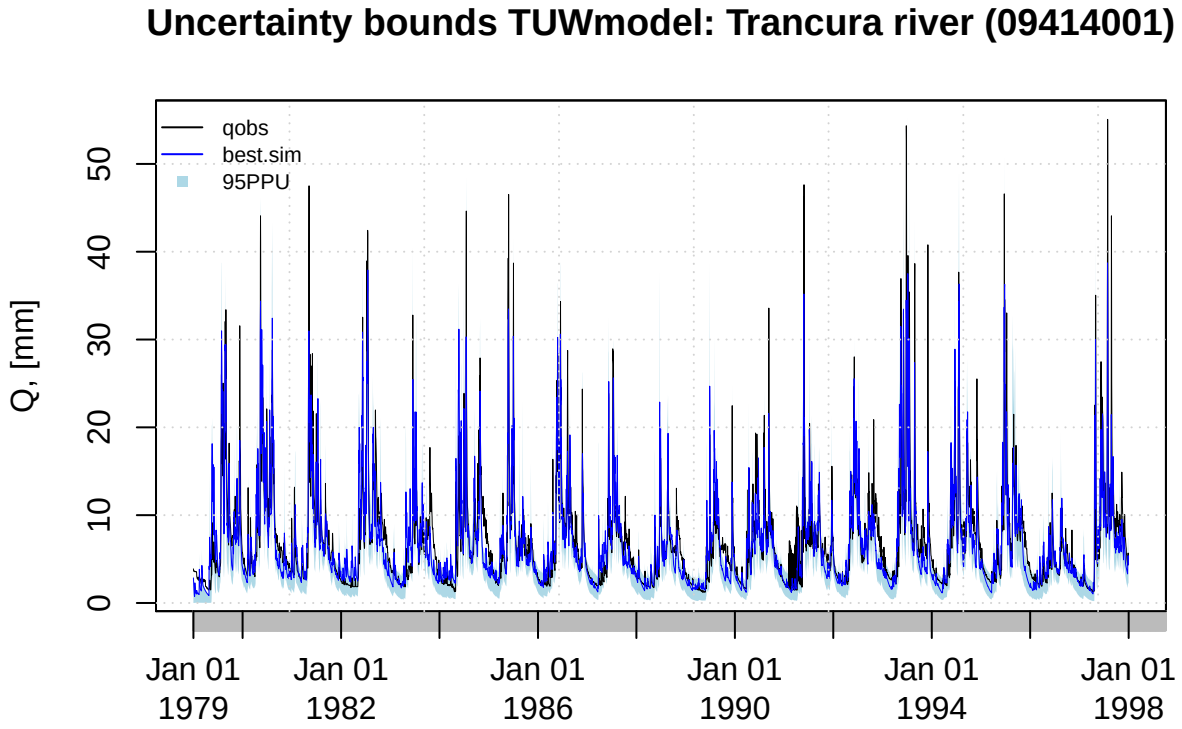

Figure 20: Uncertainty analysis for the verification period.

References

Bergström, S. 1995. "The Hbv Model." *Computer Models of Watershed Hydrology*. Water Resources Publications.

Ceola, Serena, Berit Arheimer, E Baratti, G Blöschl, Réne Capell, Attilio Castellarin, Jim Freer, et al. 2015. "Virtual Laboratories: New Opportunities for Collaborative Water Science." *Hydrology and Earth System Sciences* 19 (4): 2101–17.

Cisty, Milan, and Veronika Soldanova. 2018. "Flow Prediction Versus Flow Simulation Using Machine Learning Algorithms." In *International Conference on Machine Learning and Data Mining in Pattern Recognition*, 369–82. Springer.

Gupta, Hoshin V, Harald Kling, Koray K Yilmaz, and Guillermo F Martinez. 2009. "Decomposition of the Mean Squared Error and Nse Performance Criteria: Implications for Improving Hydrological Modelling." *Journal of Hydrology* 377 (1-2). Elsevier: 80–91.

Kennedy, James, and Russell Eberhart. 1995. "Particle Swarm Optimization." In *Proceedings of Icnn'95-International Conference on Neural Networks*, 4:1942–8. IEEE.

Kling, Harald, Martin Fuchs, and Maria Paulin. 2012. "Runoff Conditions in the Upper Danube Basin Under an Ensemble of Climate Change Scenarios." *Journal of Hydrology* 424. Elsevier: 264–77.

Melsen, L. A., N. Addor, N. Mizukami, A. J. Newman, P. J. J. F. Torfs, M. P. Clark, R. Uijlenhoet, and A. J. Teuling. 2018. "Mapping (Dis)agreement in Hydrologic Projections." *Hydrology and Earth System Sciences* 22 (3): 1775–91. https://doi.org/10.5194/hess-22-1775-2018.

Nijzink, RC, Susana Almeida, IG Pechlivanidis, Rene Capell, D Gustafssons, Berit Arheimer, Juraj Parajka, et al. 2018. "Constraining Conceptual Hydrological Models with Multiple Information Sources." *Water Resources Research* 54 (10). Wiley Online Library: 8332–62.

Nijzink, Remko, Christopher Hutton, Ilias Pechlivanidis, René Capell, Berit Arheimer, Jim Freer, Dawei Han, et al. 2016. "The Evolution of Root-Zone Moisture Capacities After Deforestation: A Step Towards Hydrological Predictions Under Change?" *Hydrology and Earth System Sciences* 20 (12): 4775–99.

Parajka, J., R. Merz, and G. Blöschl. 2007. "Uncertainty and Multiple Objective Calibration in Regional Water Balance Modelling: Case Study in 320 Austrian Catchments." *Hydrological Processes* 21 (4): 435–46. https://doi.org/10.1002/hyp.6253.

Parajka, Juraj, Alfred Paul Blaschke, Günter Blöschl, Klaus Haslinger, Gerold Hepp, Gregor Laaha, Wolfgang Schöner, Helene Trautvetter, Alberto Viglione, and Matthias Zessner. 2016. "Uncertainty Contributions to Low-Flow Projections in Austria." *Hydrology and Earth System Sciences* 20 (5). Copernicus GmbH: 2085–2101.

Sleziak, Patrik, Ján Szolgay, Kamila Hlavčová, Doris Duethmann, Juraj Parajka, and Michal Danko. 2018. "Factors Controlling Alterations in the Performance of a Runoff Model in Changing Climate Conditions." *Journal of Hydrology and Hydromechanics* 66 (4). Sciendo: 381–92.

Sleziak, P, J Szolgay, K Hlavčová, M Danko, and J Parajka. 2020. "The Effect of the Snow Weighting on the Temporal Stability of Hydrologic Model Efficiency and Parameters." *Journal of Hydrology*. Elsevier, 124639.

Sleziak, P, J Szolgay, K Hlavčová, and J Parajka. 2016. "The Impact of the Variability of Precipitation and Temperatures on the Efficiency of a Conceptual Rainfall-Runoff Model." *Slovak Journal of Civil Engineering* 24 (4). De Gruyter Open: 1–7.

Viglione, Alberto, and Juraj Parajka. 2019. *TUWmodel: Lumped/Semi-Distributed Hydrological Model for Education Purposes*. https://CRAN.R-project.org/package=TUWmodel.

Zambrano-Bigiarini, Mauricio, and Rodrigo Rojas. 2013. "A Model-Independent Particle Swarm Optimisation Software for Model Calibration." *Environmental Modelling & Software* 43. Elsevier: 5–25.

Zessner, Matthias, Martin Schönhart, Juraj Parajka, Helene Trautvetter, Hermine Mitter, Mathias Kirchner, Gerold Hepp, Alfred Paul Blaschke, Birgit Strenn, and Erwin Schmid. 2017. "A Novel Integrated Modelling Framework to Assess the Impacts of Climate and Socio-Economic Drivers on Land Use and Water Quality." *Science of the Total Environment* 579. Elsevier: 1137–51.