Hydrological modelling & teaching with *airGR* & *airGRteaching*

Guillaume Thirel, guillaume.thirel@irstea.fr
Olivier Delaigue
EGU - Vienna, 11 April 2018

Introduction

Introduction

Hydrological modelling packages in 

- topmodel
- dynatopmodel
- TUWmodel
- Ecohydmod
- hydromad (not on CRAN)
- etc.

… and of course **airGR** & **airGRteaching**

airGR(teaching) Models & Tools

Models

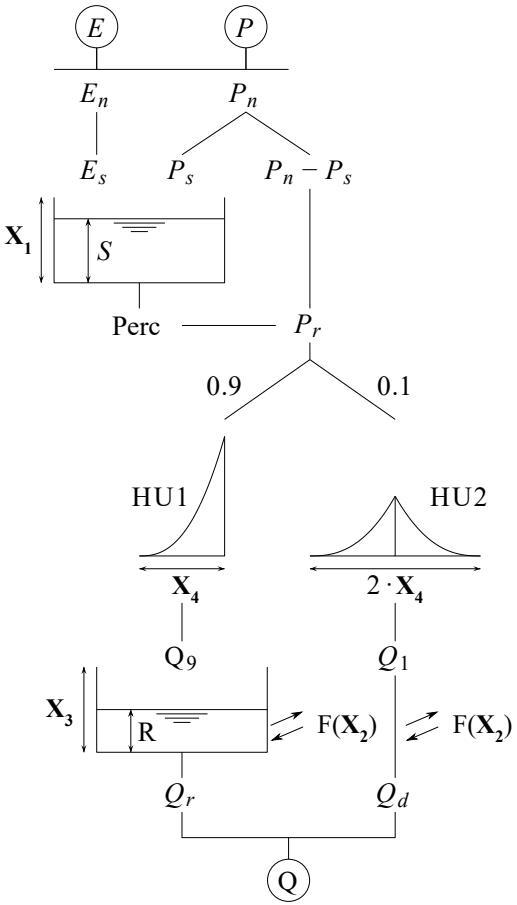
GR hydrological models

Rainfall-runoff lumped models developed since the 90's at IRSTEA

- annual: GR1A
- monthly: GR2M
- daily: **GR4J**, GR5J, GR6J
- hourly: GR4H

Snow model

- CemaNeige



More about GR models: poster A.9
[EGU2018-15734]
Thursday, "History of Hydrology" session

Tools

Error criteria

- RMSE
- Nash-Sutcliffe
- KGE
- KGE'

Calibration algorithm

- Housemade global then local optimisation

Dataset input

Dataset preparation

data.frame of 2 basins from 1985 to 2005

```
summary(dfObs)
```

```
##       Date                  P                  E          
##  Min.   :1985-01-01   Min.   : 0.00000   Min.   :0.0000  
##  1st Qu.:1990-01-01   1st Qu.: 0.04297   1st Qu.:0.4354  
##  Median :1995-01-01   Median : 0.90565   Median :1.0481  
##  Mean   :1995-01-01   Mean   : 3.28253   Mean   :1.2721  
##  3rd Qu.:2000-01-01   3rd Qu.: 4.19317   3rd Qu.:1.9221  
##  Max.   :2004-12-31   Max.   :63.58900   Max.   :5.7799  
##                                                          
##       Qmm                 T              Basin       
##  Min.   : 0.03988   Min.   :-13.668   12001:7305  
##  1st Qu.: 0.62357   1st Qu.:  4.254   50002:7305  
##  Median : 1.42465   Median :  7.986               
##  Mean   : 2.25734   Mean   :  7.814               
##  3rd Qu.: 2.76480   3rd Qu.: 11.480               
##  Max.   :38.97774   Max.   : 24.174               
##  NA's   :1
```

aiGRteaching

airGRteaching overview

Coding can be a hard task for some students

- slows down the understanding of the concepts presented during courses
- limited time to teach programming

airGRteaching contains only very basic functions

Attaching the package

```
library(airGRteaching)
```

```
## Loading required package: airGR
```

airGRteaching data preparation

Needs only a data.frame (or independent vectors)

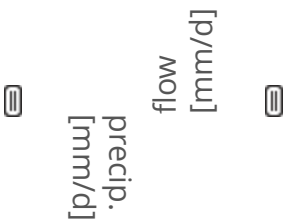
```
B12001 <- dfObs[dfObs$Basin == "12001", c("Date", "P", "E", "Qmm", "T")]


PREP <- PrepGR(ObsDF = B12001,
               HydroModel = "GR4J", CemaNeige = TRUE)
```

airGRteaching graphical functions

`dyplot()`: dynamic chart to explore data

`plot()`: static chart for student reports

```
dyplot(PREP)
```



airGRteaching calibration

Model calibration on a given period, with the KGE' criterion

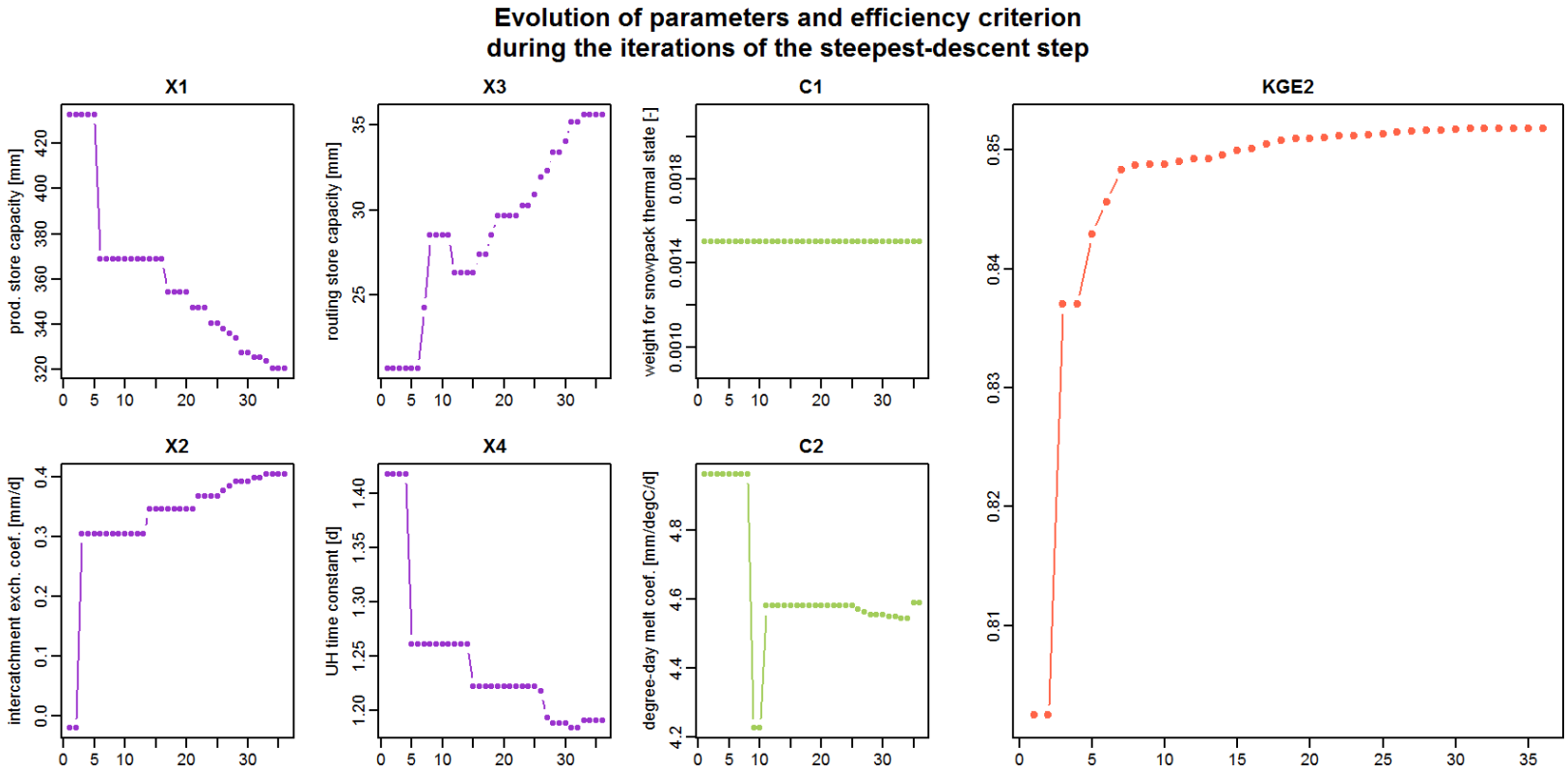
```
CAL <- CalGR(PrepGR = PREP, CalCrit = "KGE2",
             WupPer = c("1989-01-01", "1989-12-31"),
             CalPer = c("1990-01-01", "1993-12-31"))
```

```
## Grid-Screening in progress (0% 20% 40% 60% 80% 100%)
##   Screening completed (729 runs)
##       Param =  432.681 ,   -0.020 ,   20.697 ,    1.417 ,    0.002 ,    4.961
##       Crit KGE'[Q]      = 0.8025
## Steepest-descent local search in progress
##   Calibration completed (36 iterations, 1138 runs)
##       Param =  320.368 ,    0.405 ,   35.589 ,    1.191 ,    0.002 ,    4.589
##       Crit KGE'[Q]      = 0.8518
```

airGRteaching calibration

Plotting calibration process

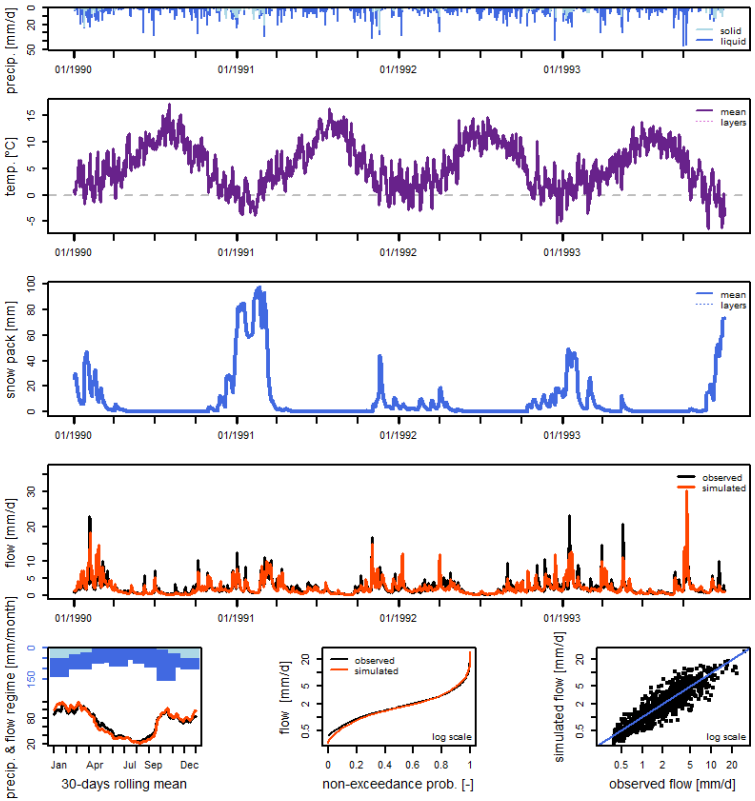
```
plot(CAL, which = "iter")
```



airGRteaching calibration

Plotting performances

```
plot(CAL, which = "perf")
```



airGRteaching simulation

To check if the calibration is correct, we can perform an evaluation on an independent period (Klemes split-sample test)

```
SIM <- SimGR(PrepGR = PREP, CalGR = CAL, EffCrit = "KGE2",
             WupPer = c("1993-01-01", "1993-12-31"),
             SimPer = c("1994-01-01", "1998-12-31"))
```

```
## Crit. KGE'[Q] = 0.8679
```

```
##  SubCrit. KGE'[Q] cor(sim, obs, "pearson") = 0.8941
##  SubCrit. KGE'[Q] cv(sim)/cv(obs)          = 1.0469
##  SubCrit. KGE'[Q] mean(sim)/mean(obs)      = 0.9365
```

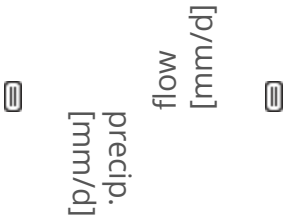
Or with a known parameter set

```
PARAM <- c(X1 = 200, X2 = 2, X3 = 300, X4 = 2, C1 = 0.5, C2 = 4)
SIM <- SimGR(PrepGR = PREP, Param = PARAM, EffCrit = "KGE2",
             WupPer = c("1993-01-01", "1993-12-31"),
             SimPer = c("1994-01-01", "1998-12-31"))
```

airGRteaching simulation

Plotting the resulting discharge time series

```
dyplot(SIM)
```



airGRteaching Shiny interface

List of 2 basins data.frame

```
colSelect <- c("Date", "P", "E", "Qmm", "T")
listObs <- list(B12001 = dfObs[dfObs$Basin == "12001", colSelect],
                B50002 = dfObs[dfObs$Basin == "50002", colSelect])
```

Launching the Shiny interface for 2 basins

```
ShinyGR(ObsDF = listObs,
        SimPer = list(c("1994-01-01", "1998-12-31"),
                      c("1994-01-01", "1998-12-31")))
```

Only daily models available

Possibility to put the GUI on the Web with a Shiny server

airGR

airGR overview

airGR allows for more advanced or systematic tests

Use of:

- bootstrap technique for calibration
- additional calibration algorithms

Possibility to set:

- some model parameters
- initial reservoir levels
- internal states
- etc.

Attaching the package

```
library(airGR)
```

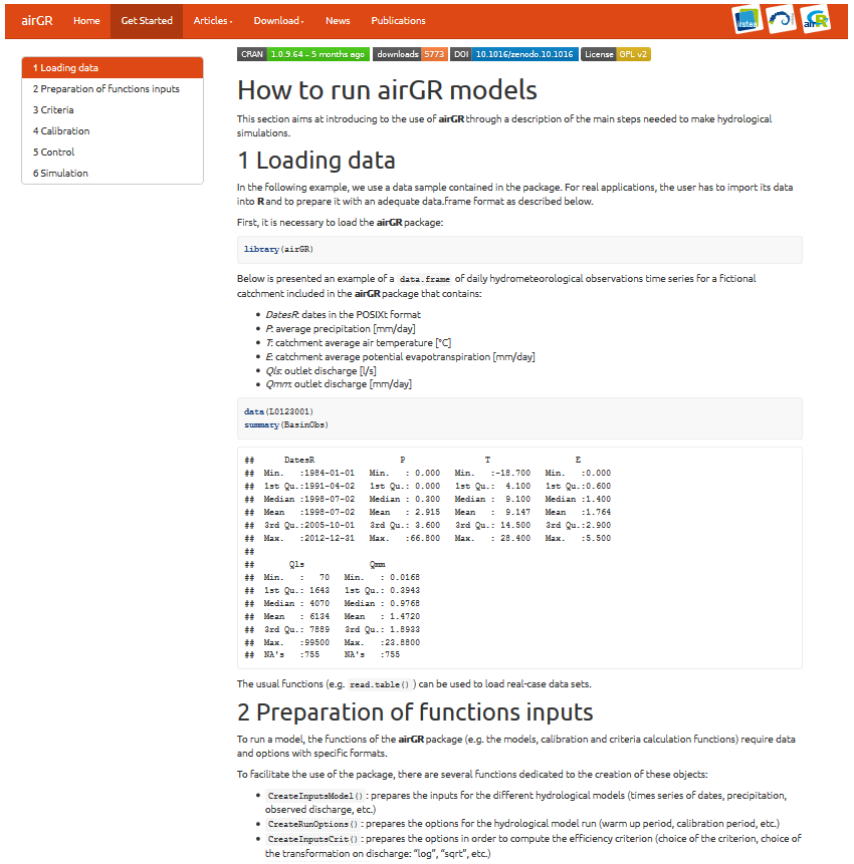
Steps

Functions

- `CreateInputsModel()`: model inputs
- `CreateRunOptions()`: run options
- `CreateInputsCrit()`: calibration criterion
- `CreateCalibOptions()`: calibration options
- `Calibration_Michel()`: run calibration
- `RunModel_GR4J()`: run simulation

Tutorial

https://webgr.irstea.fr/airGR-website



Options

Some model parameters settings

```
CalibOptions <- CreateCalibOptions(FUN_MOD = RunModel_CemaNeigeGR4J,
                                   FUN_CALIB = Calibration_Michel,
                                   FixedParam = c(NA, NA, NA, NA, 0.5, 4))
```

Use of optimisation algorithms from other R packages

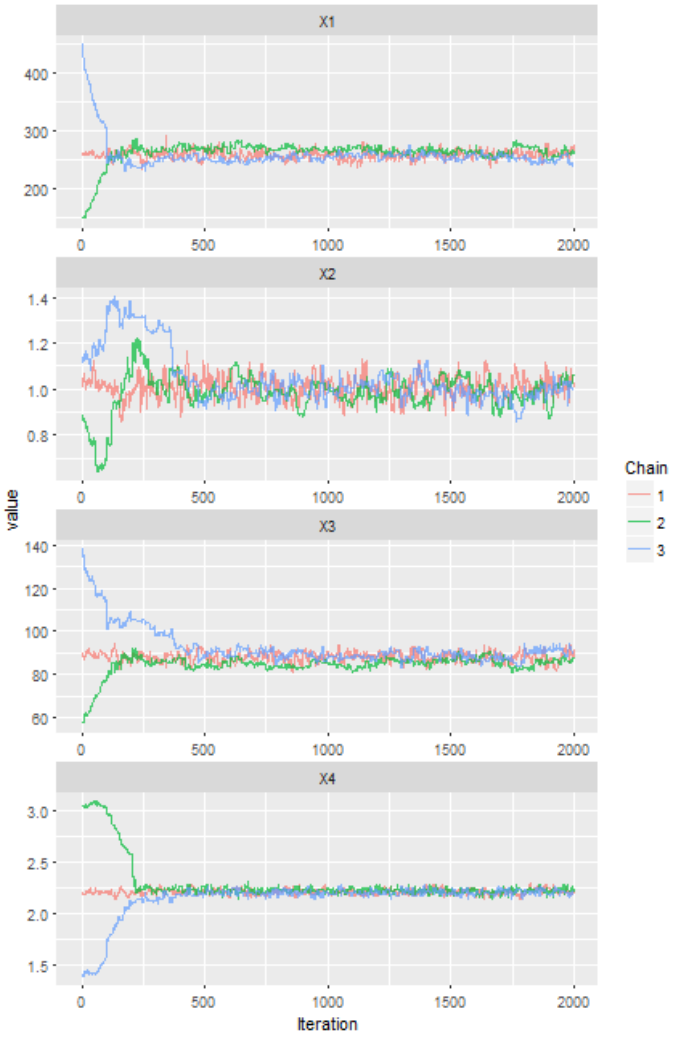
See the "V02.1_param_optim" vignette for implementation

```
optDE <- DEoptim::DEoptim(fn = OptimGR4J,
                          lower = lowerGR4J, upper = upperGR4J)
optPSO <- hydroPSO::hydroPSO(fn = OptimGR4J,
                             lower = lowerGR4J, upper = upperGR4J)
```

Options

Use of optimisation algorithms from other R packages

Markov Chain Monte Carlo calibration



Conclusion

Final words

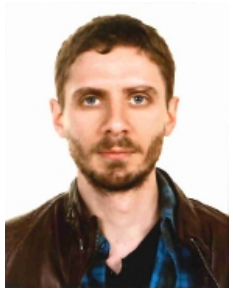
airGRteaching can be used to :

- play with the models parameters or understand the models components
- help someone very bad at coding to make hydrological modelling

airGR can be used to:

- make more advanced hydrological modelling

Contact: airGR@irstea.fr



Meet Olivier Delaigue and myself at the airGRteaching poster
X.142 from 1:30 pm today