airGRiwrm: an extension of the airGR R-package for handling Integrated Water Resources Management modeling

David Dorchies, Olivier Delaigue, Guillaume Thirel

▶ **To cite this version:**

David Dorchies, Olivier Delaigue, Guillaume Thirel. airGRiwrm: an extension of the airGR R-package for handling Integrated Water Resources Management modeling. EGU General Assembly 2021 - vEGU21: Gather Online, Apr 2021, Vienna, Austria. pp.EGU21-2190, 10.5194/egusphere-egu21-2190 . hal-03330881

HAL Id: hal-03330881
https://hal.inrae.fr/hal-03330881

Submitted on 20 Apr 2023

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Distributed under a Creative Commons Attribution 4.0 International License

RÉPUBLIQUE FRANÇAISE
Liberté
Égalité
Fraternité

INRAE

airGRiwrm R package

an extension of the airGR R-package for handling Integrated Water Resources Management modeling

David Dorchies[1], Olivier Delaigue[2], and Guillaume Thirel[2]
1. G-EAU, Univ Montpellier, AgroParisTech, CIRAD, IRD, INRAE, Institut Agro, Montpellier, France
2. Université Paris-Saclay, INRAE, HYCAR research unit, Hydrology Research Group, Antony, France
EGU General Assembly 2021

The airGR packages constellation

airGR is an R-package for running GR rainfall-runoff models

The latest version has semi-distributed hydrological modelling capabilities that are exploited by the **airGRiwrm** package.

See EGU21-1371 in the same session

airGRiwrm package main features

Handle large network of airGR semi-distributed hydrological models with a minimum of complexity

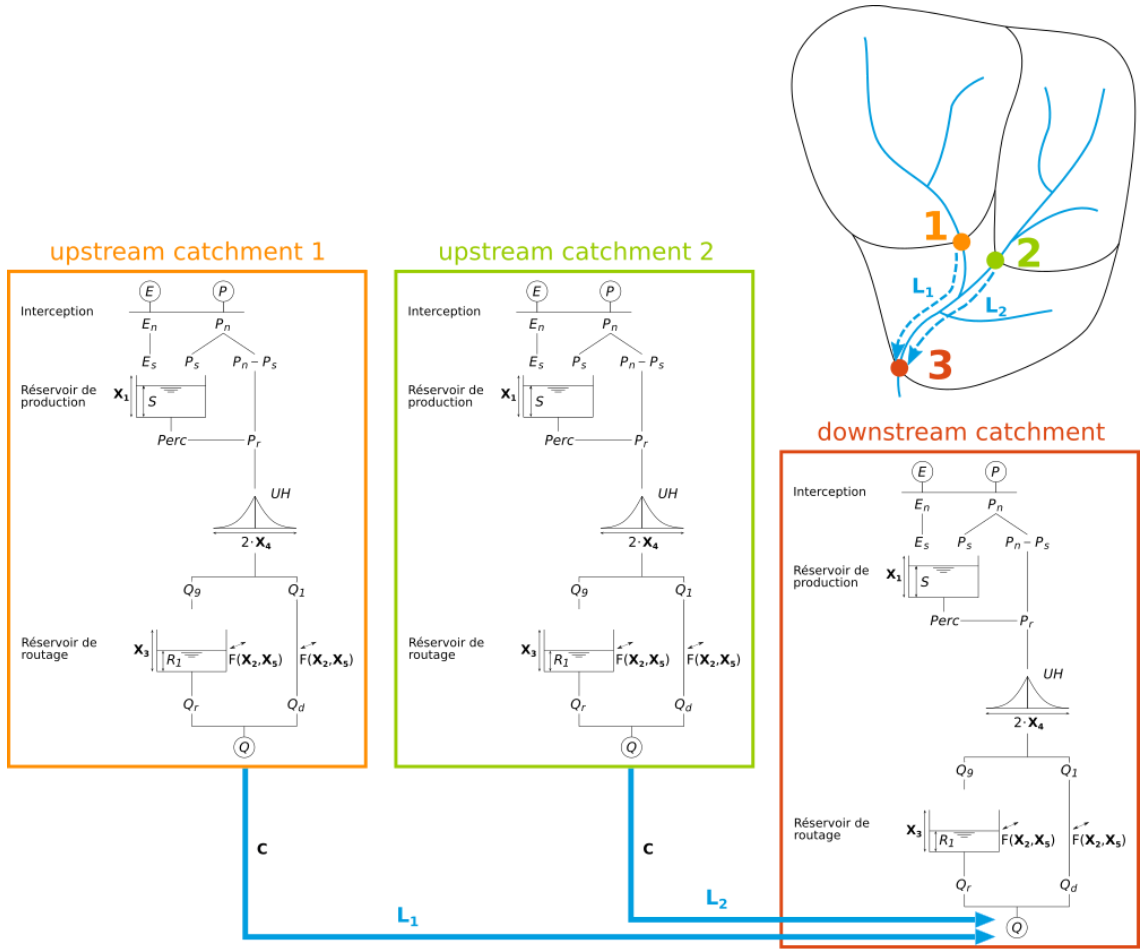
Easily integrate withdrawal and release flows in the network

Calibration and simulation with both influenced and naturalised flows

Run user control algorithms for simulating flows with automatic regulation

Semi-distributed model in airGR

In airGR, GR hydrological model simulations are routed from upstream basins to downstream basins thanks to a lag model



How airGRiwrm works

It extends **airGR** package functions for handling a network of semi-distributed GR models instead of a single global GR model

```
library(airGR)
library(airGRiwrm)
```

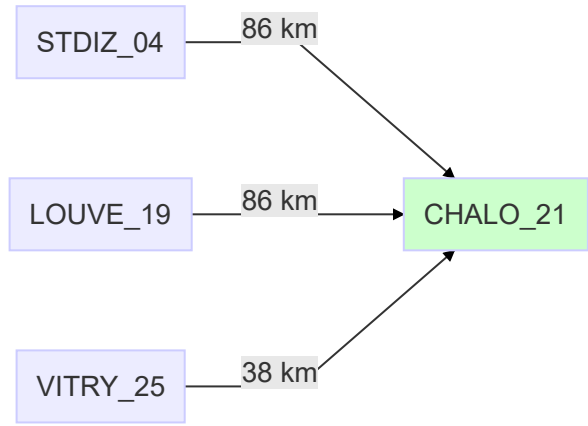
```
##
## Attaching package: 'airGRiwrm'

## The following objects are masked from 'package:airGR':
##
##     Calibration, CreateCalibOptions, CreateInputsCrit,
##     CreateInputsModel, CreateRunOptions, RunModel
```

Description of the network

All required information for running a semi-distributed model network are set in a formated `data.frame` called a `GRiwrm` object which can be displayed as a scheme of the network.

```
DiagramGRiwrm(griwrm)
```



	id	down	length	model	area
4	STDIZ_04	CHALO_21	85.570	RunModel_GR4J	2347.53
19	LOUVE_19	CHALO_21	86.165	RunModel_GR4J	461.74
25	VITRY_25	CHALO_21	38.047	RunModel_GR4J	2109.14
21	CHALO_21	NA	NA	RunModel_GR4J	6291.55

Hydroclimatic input data handling

Inputs are represented by matrices with one named column by sub-basin instead of vectors as in **airGR**

```
head(P, 10)
```

```
##       STDIZ_04 LOUVE_19 VITRY_25 CHALO_21
##  [1,]      0.7      0.4      0.9      0.6
##  [2,]      0.0      0.0      0.0      0.0
##  [3,]      0.0      0.0      0.0      0.0
##  [4,]      0.0      0.0      0.0      0.0
##  [5,]      0.0      0.1      0.0      0.0
##  [6,]      6.1      4.7      1.6      3.8
##  [7,]     17.1     23.6     17.0     16.4
##  [8,]      2.2      3.1      6.1      3.2
##  [9,]     10.8     10.8      6.9      7.8
## [10,]      8.6      9.7      4.1      5.5
```

Processing with a network of SD models

```
# Preparation of GRiwrmInputsModel object
IMnat <- CreateInputsModel(
  griwrm,
  DatesR = DatesR,
  Precip = P,
  PotEvap = E,
  Qobs = Q
)
```

We use the classical functions and procedures of **airGR** to build objects embedding the whole network data...

```
## CreateInputsModel.GRiwrm: Treating sub-basin STDIZ_04...

## CreateInputsModel.GRiwrm: Treating sub-basin LOUVE_19...

## CreateInputsModel.GRiwrm: Treating sub-basin VITRY_25...

## CreateInputsModel.GRiwrm: Treating sub-basin CHALO_21...
```

Processing with a network of SD models

```
# Calibration period selection
I_Run <- 366:length(DatesR)

# Set up GRiwrmRunOptions object
RO <- CreateRunOptions(
  InputsModel = IM,
  IndPeriod_Run = I_Run
)

# Calibration criterion:
# the GRiwrmInputsCrit object
IC <- CreateInputsCrit(
  InputsModel = IM,
  FUN_CRIT = airGR::ErrorCrit_NSE,
  RunOptions = RO,
  Obs = Q[I_Run,]
)
```

```
# Set up GRiwrmCalibOptions object
CO <- CreateCalibOptions(IM)

# Calibration of the network of
# models from upstream to downstream
OC <- Calibration(
  InputsModel = IM,
  RunOptions = RO,
  InputsCrit = IC,
  CalibOptions = CO,
  useUpstreamQsim = TRUE
)
```

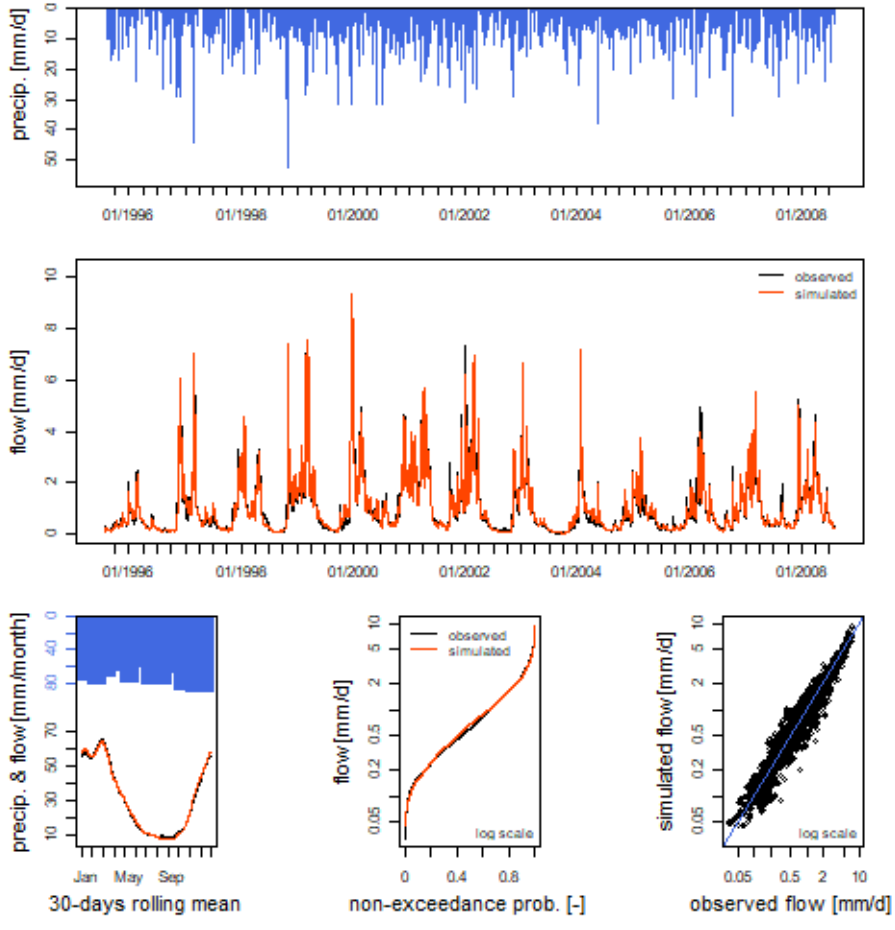
```
Calibration.GRiwrmInputsModel: Treating sub-basin STDIZ_04...
Grid-Screening in progress (0% 20% 40% 60% 80% 100%)
     Screening completed (81 runs)
         Param =  247.151,   -0.020,   83.096,    2.384
         Crit. NSE[Q]       = 0.8732
Steepest-descent local search in progress
     Calibration completed (26 iterations, 273 runs)
         Param =  208.513,   -0.130,   74.440,    3.506
         Crit. NSE[Q]       = 0.9236
```

Simulation run and outputs

```
# Make a list with sub-basin
# calibrated parameters
Param <- sapply(
  names(OC),
  function(x) {OC[[x]]$Param}
)

# Run the network of models
OM <- RunModel(IM,
               RunOptions = RO,
               Param = Param)
```

```
# Plot downstream sub-basin
# with airGR::plot.OutputsModel
plot(OM[[4]], Q[I_Run, 4])
```



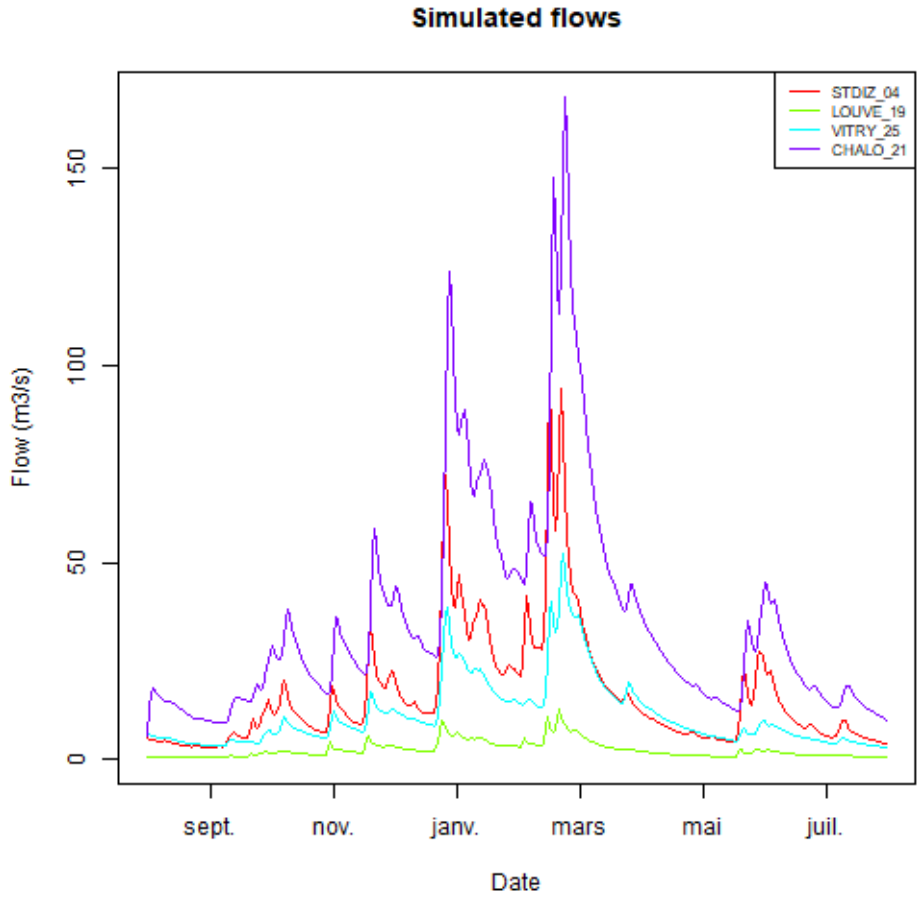
Flow plotting of all network nodes

The `GRIwrmOutputsModel` has an attribute *"Qm3s"* containing a `data.frame` with simulated flows for all the nodes in m^3/s.

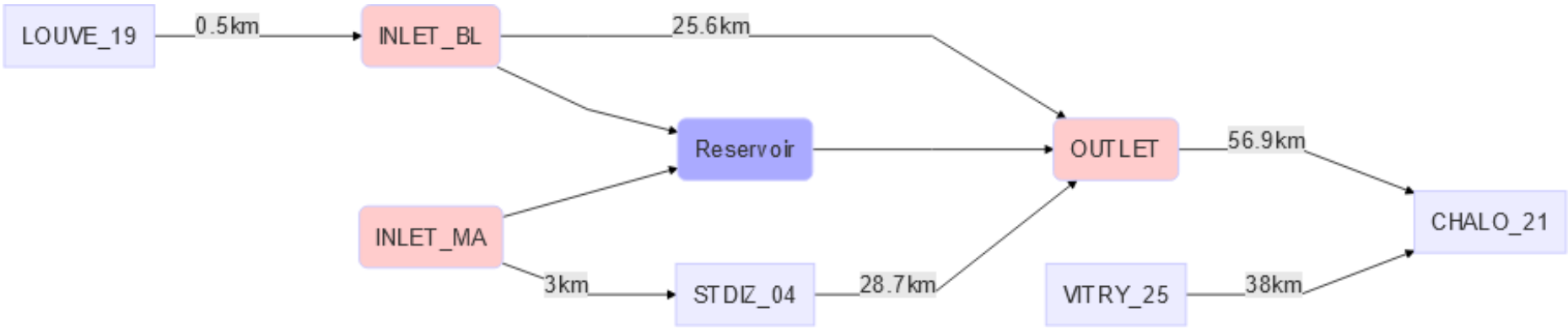
```
Qm3s <- attr(OM, "Qm3s")
```

A dedicated plot function displays simulated flow time series for all the nodes.

```
plot(Qm3s[1:365,])
```



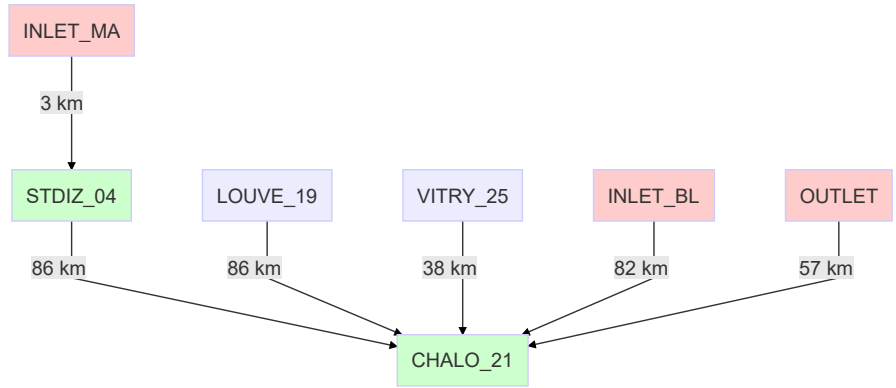
Anthropic influence integration



Scheme of the real network

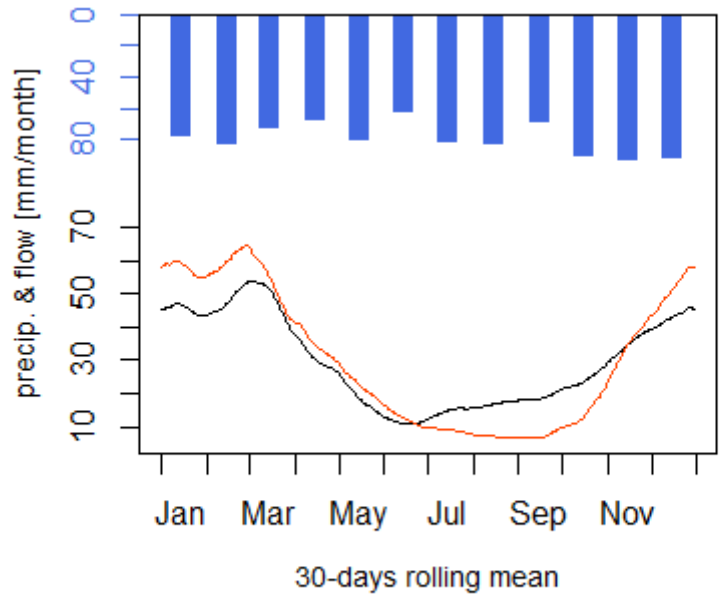
airGRiwrm easily integrates direct flow injections or withdrawals in the network by using the matrix of observed flows instead of an hydrological model in network nodes.

*Network representation in **airGRiwrm** →*



Naturalised vs influenced flows

Integrating human withdrawals and releases in the network allow to calibrate the model with influenced observed flows and then to run the model without the influences in order to compute naturalised flows by simulation.

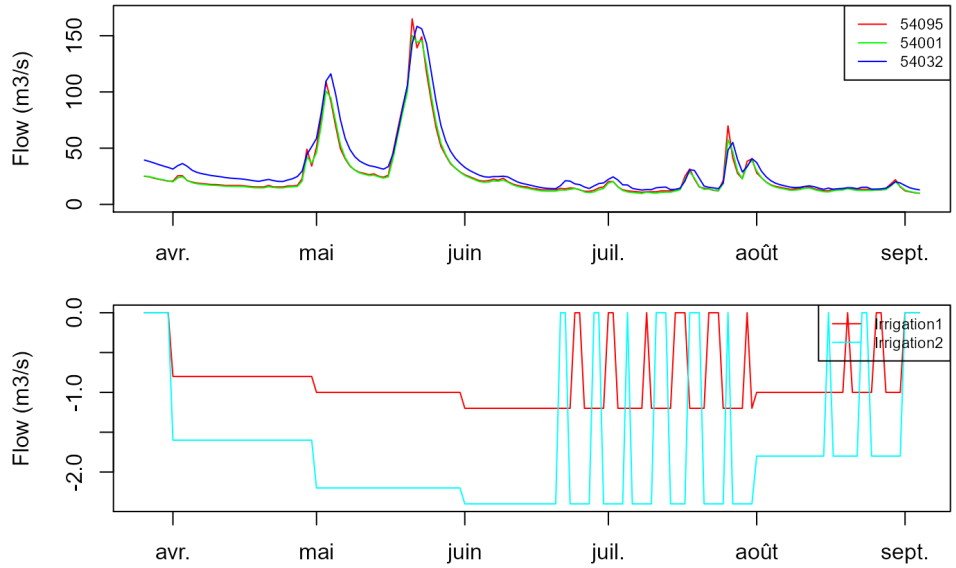


Regime of observed influenced flows (black) and naturalised simulated flows (orange)

Regulation algorithm integration

A supervisor provides simulation outputs during simulation to controllers that execute a logic of control which apply regulated flows in the network.

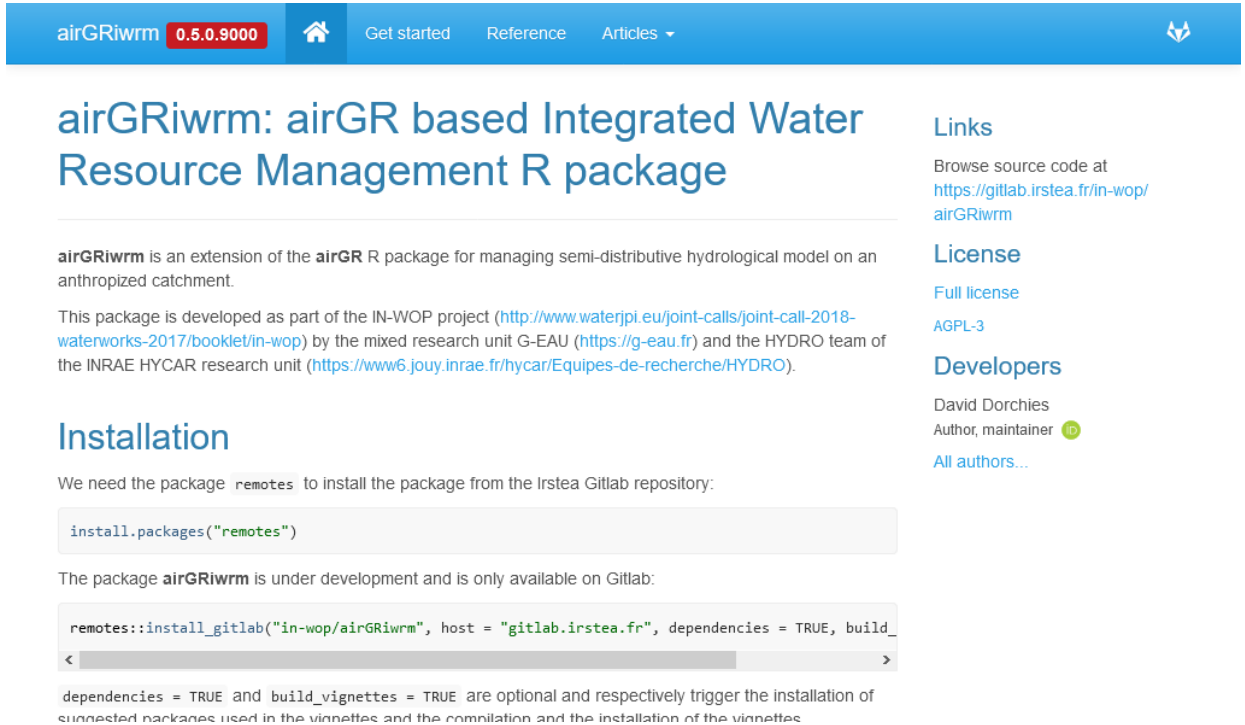
The user can write control logic representing withdrawal restriction, reservoir operations... with a complex algorithm



Results of the simulation of regulated withdrawal restrictions

For complete documentation

https://airgriwrm.g-eau.fr



airGRiwrm 0.5.0.9000 | Get started | Reference | Articles

airGRiwrm: airGR based Integrated Water Resource Management R package

airGRiwrm is an extension of the **airGR** R package for managing semi-distributive hydrological model on an anthropized catchment.

This package is developed as part of the IN-WOP project (http://www.waterjpi.eu/joint-calls/joint-call-2018-waterworks-2017/booklet/in-wop) by the mixed research unit G-EAU (https://g-eau.fr) and the HYDRO team of the INRAE HYCAR research unit (https://www6.jouy.inrae.fr/hycar/Equipes-de-recherche/HYDRO).

Installation

We need the package `remotes` to install the package from the Irstea Gitlab repository:

```
install.packages("remotes")
```

The package **airGRiwrm** is under development and is only available on Gitlab:

```
remotes::install_gitlab("in-wop/airGRiwrm", host = "gitlab.irstea.fr", dependencies = TRUE, build_
```

`dependencies = TRUE` and `build_vignettes = TRUE` are optional and respectively trigger the installation of suggested packages used in the vignettes and the compilation and the installation of the vignettes

Links

Browse source code at https://gitlab.irstea.fr/in-wop/airGRiwrm

License

Full license

AGPL-3

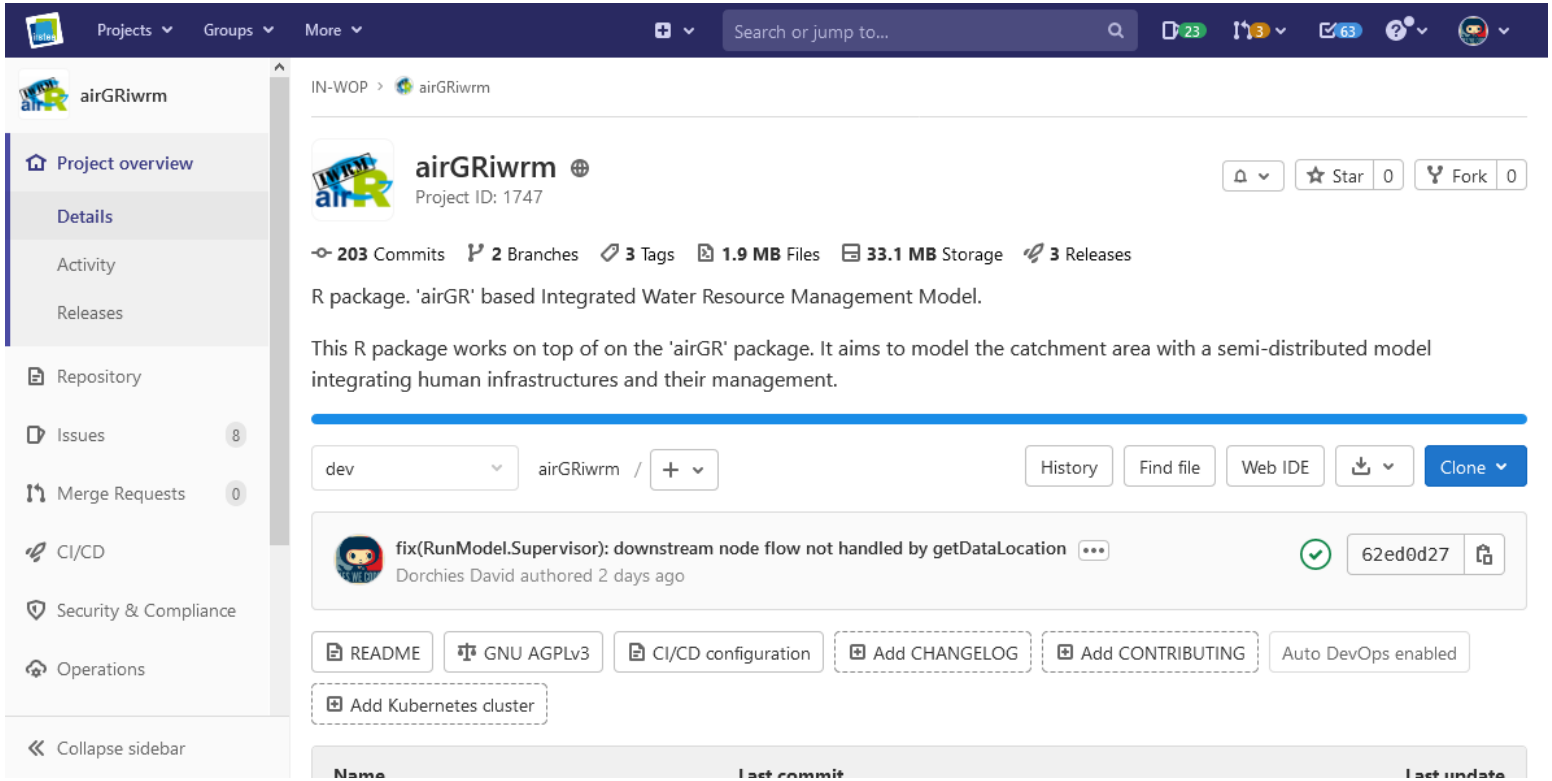
Developers

David Dorchies
Author, maintainer

All authors...

Source code, bug tracker...

https://gitlab.irstea.fr/in-wop/airGRiwrm



Package installation

The package will be submitted on CRAN soon.

In the meantime, to install the package*:

```
install.packages("devtools") ## required

remotes::install_gitlab("in-wop/airgriwrm",
                        host = "gitlab.irstea.fr")
```

* It also requires the priori installation of 'Rtools' on windows platforms for compiling **airGR** Fortran's code

RÉPUBLIQUE
FRANÇAISE
Liberté
Égalité
Fraternité

INRAE