



RÉPUBLIQUE
FRANÇAISE

*Liberté
Égalité
Fraternité*

INRAE

airGRiwrM R package

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

David Dorchies¹, Olivier Delaigue², and Guillaume Thirel²

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

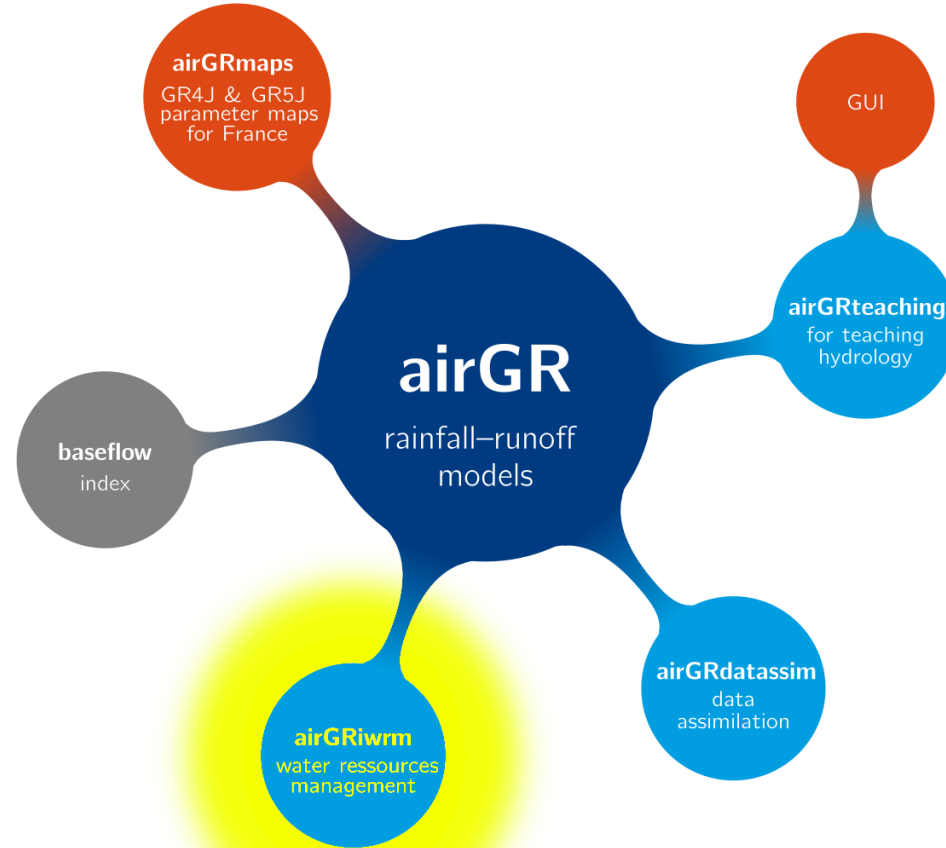
The airGR packages constellation



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 **airGRiwrn** package.

See [EGU21-1371](#) in the same session



airGRiwr package main features

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

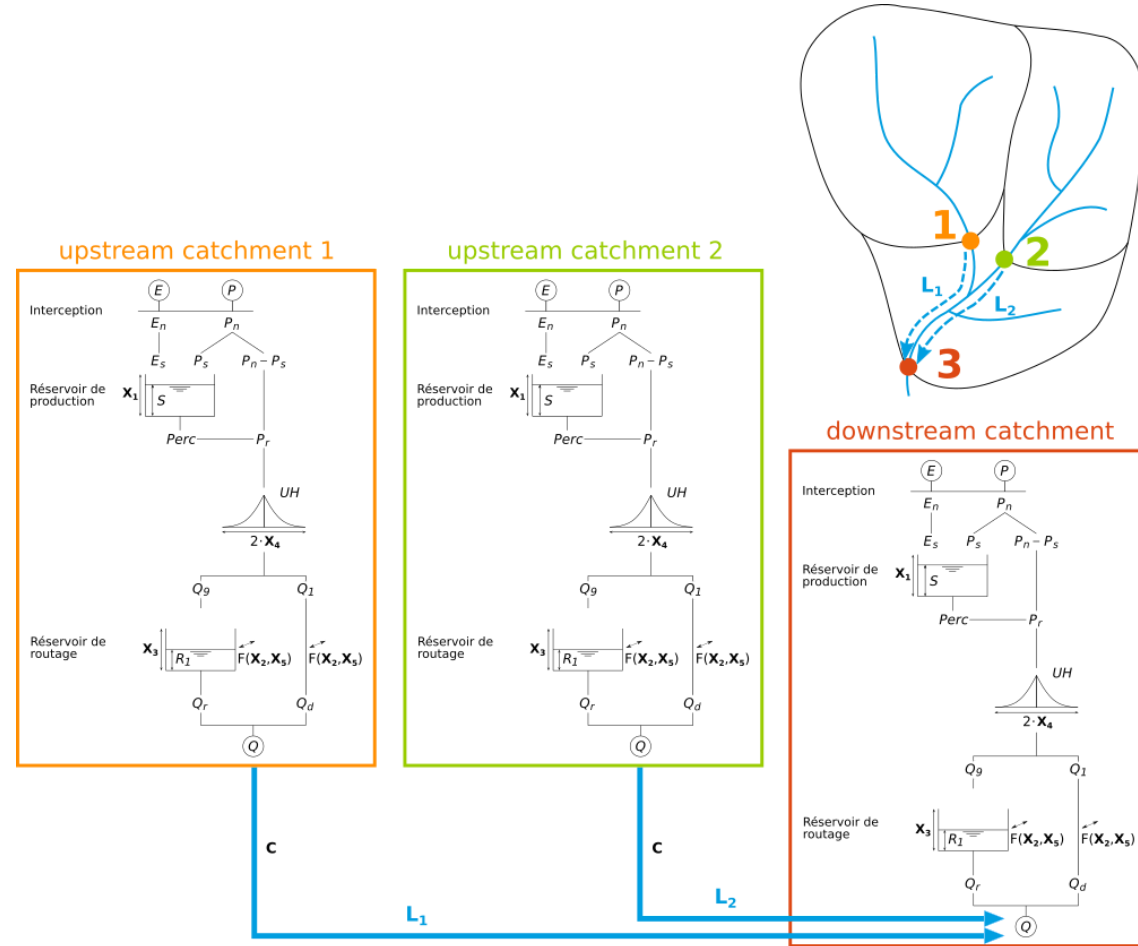
Easily integrate withdrawal and release flows in the network

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 airGRiwrn 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(airGRiwrn)
```

```
##
## Attaching package: 'airGRiwrn'

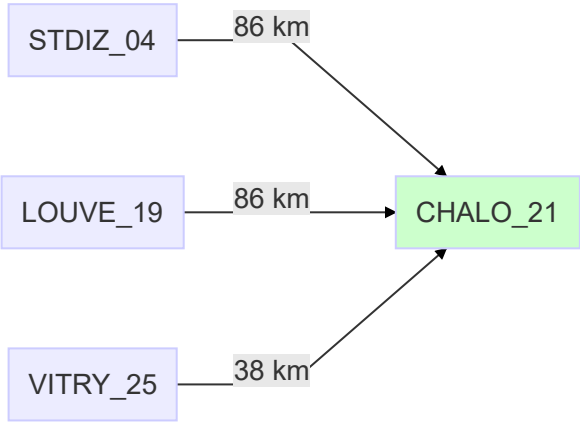
## The following objects are masked from 'package:airGR':
##
##      Calibration, CreateCalibOptions, CreateInputsCrit,
##      CreateInputsModel, CreateRunOptions, RunModel
```

Description of the network



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

```
DiagramGRiwr(griwr)
```



	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 GRiwrInputsModel object
IMnat <- CreateInputsModel(
  griwrn,
  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.GRiwrn: Treating sub-basin STDIZ_04...
```

```
## CreateInputsModel.GRiwrn: Treating sub-basin LOUVE_19...
```

```
## CreateInputsModel.GRiwrn: Treating sub-basin VITRY_25...
```

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

Processing with a network of SD models

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

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

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

```
# Set up GRIwrnCalibOptions 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.GRIwrnInputsModel: 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
```

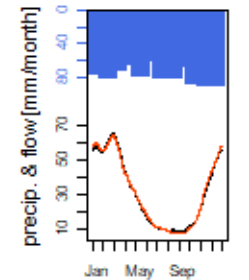
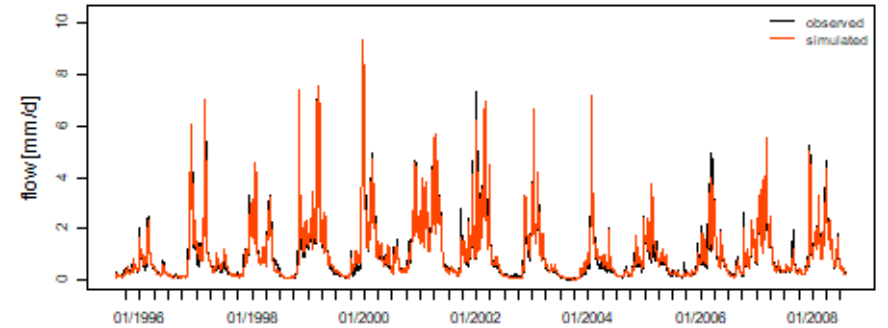
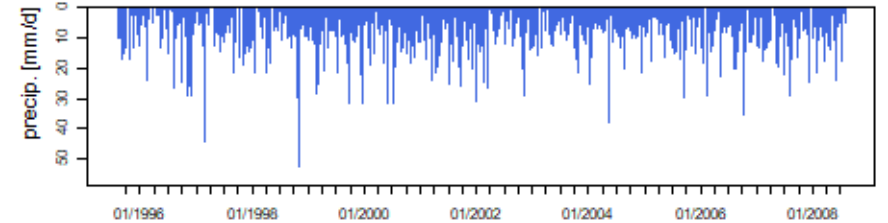
Simulation run and outputs



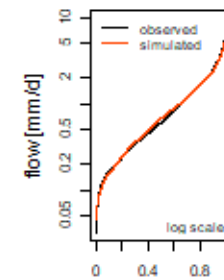
```
# Make a list with sub-basin  
# calibrated parameters  
Param <- supply(  
  names(OC),  
  function(x) {OC[[x]]$Param}  
)
```

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

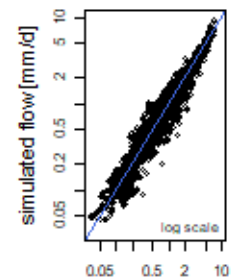
```
# Plot downstream sub-basin  
# with airGR::plot.OutputsModel  
plot(OM[[4]], Q[I_Run, 4])
```



30-days rolling mean



non-exceedance prob. [-]



observed flow [mm/d]

Flow plotting of all network nodes

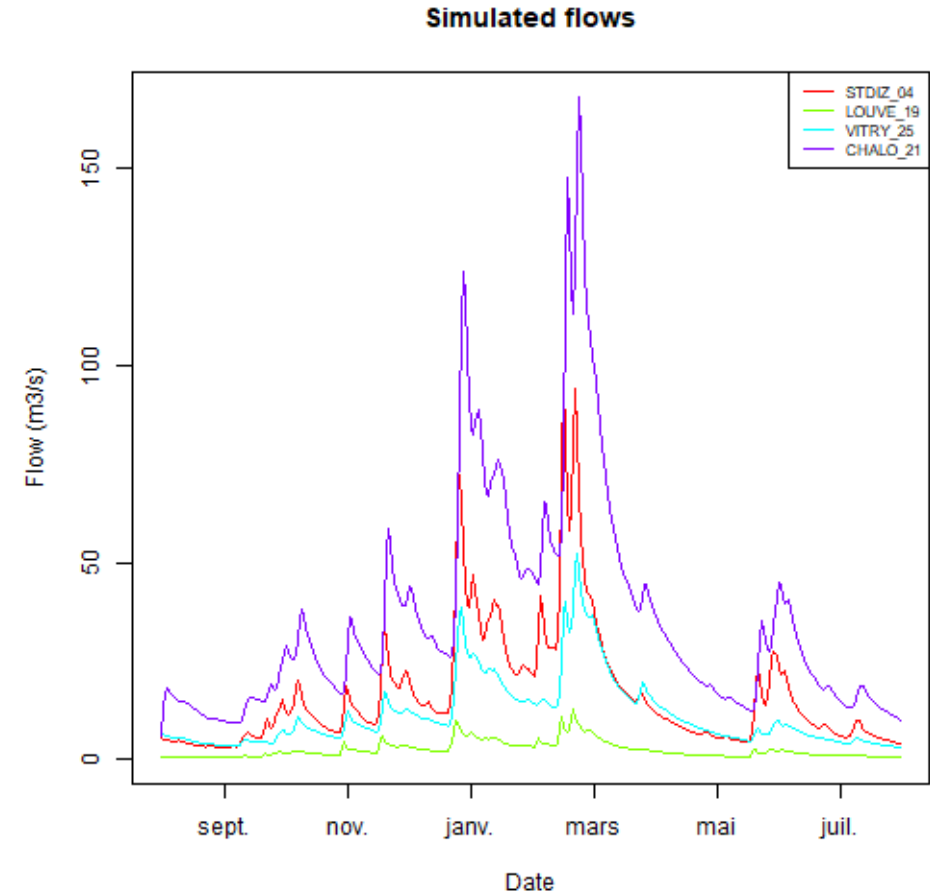


The `GRIwrmOutputsModel` has an attribute `"Qm3s"` containing a `data.frame` with simulated flows for all the nodes in m^3/s .

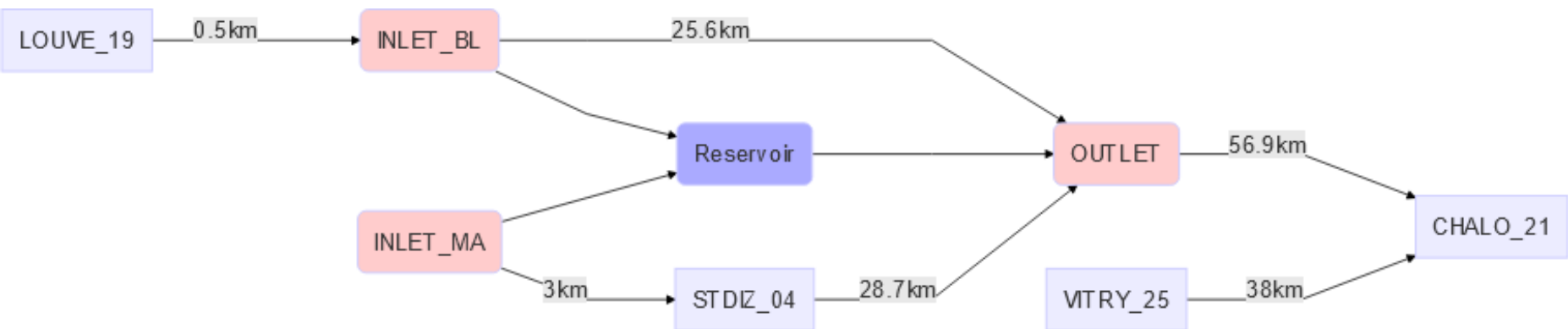
```
Qm3s <- attr(OM, "Qm3s")
```

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

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

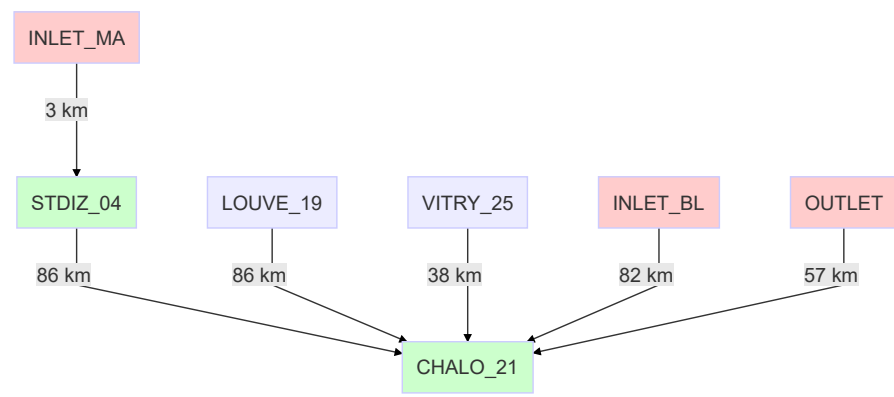


Anthropic influence integration



Scheme of the real network

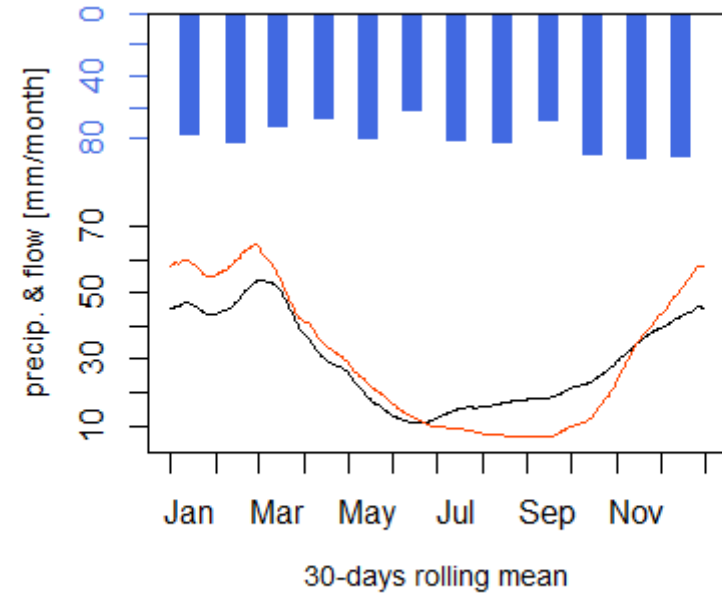
airGRiwrn easily integrates direct flow injections or withdrawals in the network by using the matrix of observed flows instead of an hydrological model in network nodes.



Network representation in airGRiwrn →

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.

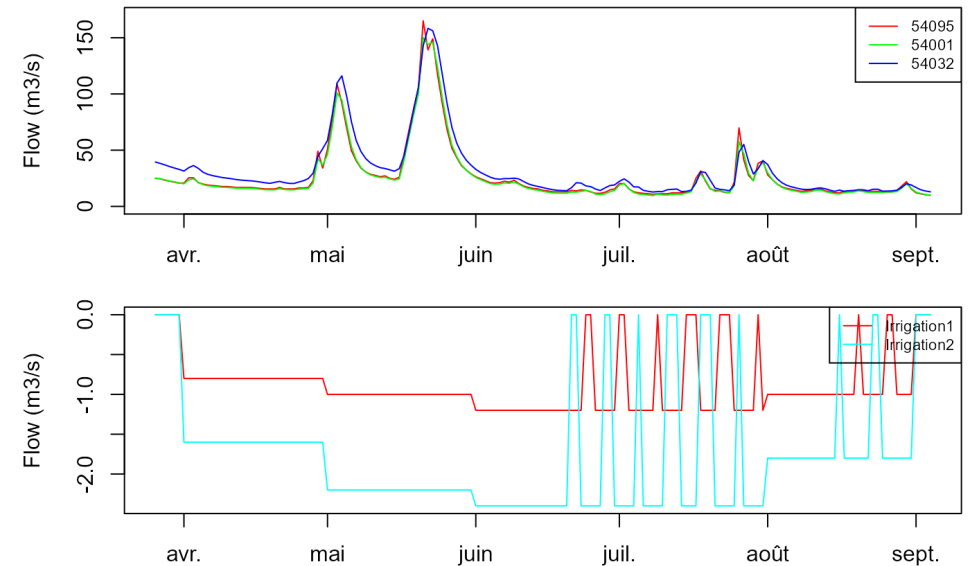


Regime of observed influenced flows (black) and naturalised simulated flows (orange)

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://airgriworm.g-eau.fr>



airGRiworm: airGR based Integrated Water Resource Management R package

airGRiworm 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 **airGRiworm** is under development and is only available on Gitlab:

```
remotes::install_gitlab("in-wop/airGRiworm", 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/airGRiworm>

License

[Full license](#)

[AGPL-3](#)

Developers

David Dorchies
Author, maintainer 

[All authors...](#)

Source code, bug tracker...



<https://gitlab.irstea.fr/in-wop/airGRiwrn>

The screenshot shows the GitLab interface for the 'airGRiwrn' project. The left sidebar contains navigation links: Project overview, Details, Activity, Releases, Repository, Issues (8), Merge Requests (0), CI/CD, Security & Compliance, and Operations. The main content area displays the project name 'airGRiwrn' with its ID (1747) and statistics: 203 Commits, 2 Branches, 3 Tags, 1.9 MB Files, 33.1 MB Storage, and 3 Releases. A description states: 'R package, 'airGR' based Integrated Water Resource Management Model. This R package works on top of on the 'airGR' package. It aims to model the catchment area with a semi-distributed model integrating human infrastructures and their management.' Below the description, there are buttons for 'dev', 'airGRiwrn / +', 'History', 'Find file', 'Web IDE', 'Clone', and a commit hash '62ed0d27'. A commit message is visible: 'fix(RunModel.Supervisor): downstream node flow not handled by getDataLocation' by Dorchies David, authored 2 days ago. At the bottom, there are buttons for 'README', 'GNU AGPLv3', 'CI/CD configuration', 'Add CHANGELOG', 'Add CONTRIBUTING', and 'Auto DevOps enabled'.

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