

# Package ‘airGRdatassim’

April 4, 2025

**Type** Package

**Title** Ensemble-Based Data Assimilation with GR Hydrological Models

**Version** 0.1.4

**Date** 2025-04-04

**Depends** airGR (>= 1.6.9.27)

**Imports** graphics, grDevices, stats, utils

**Suggests** knitr, markdown, rmarkdown

**Description** Add-on to the 'airGR' package which provides the tools to assimilate observed discharges in daily GR hydrological models. The package consists in two functions allowing to perform the assimilation of observed discharges via the Ensemble Kalman filter or the Particle filter as described in Piazzini et al. (2021) <[doi:10.1029/2020WR028390](https://doi.org/10.1029/2020WR028390)>.

**License** GPL-2

**BugReports** <https://gitlab.irstea.fr/HYCAR-Hydro/airgrdatassim/-/issues>

**NeedsCompilation** no

**Encoding** UTF-8

**VignetteBuilder** knitr

**Author** Gaia Piazzini [aut] (<<https://orcid.org/0000-0002-4167-0794>>),  
Olivier Delaigue [aut, cre] (<<https://orcid.org/0000-0002-7668-8468>>),  
Guillaume Thirel [ctb] (<<https://orcid.org/0000-0002-1444-1830>>),  
Maxime Logez [ctb] (<<https://orcid.org/0000-0001-9843-0495>>)

**Maintainer** Olivier Delaigue <[airGR@inrae.fr](mailto:airGR@inrae.fr)>

**Repository** CRAN

**Date/Publication** 2025-04-04 08:00:01 UTC

## Contents

airGRdatassim . . . . .	2
CreateInputsPert . . . . .	3
RunModel_DA . . . . .	6

<b>Index</b>	<b>11</b>
--------------	-----------

## Description

Add-on to the 'airGR' package which provides the tools to assimilate observed discharges in daily GR hydrological models. The package consists in two functions allowing to perform the assimilation of observed discharges via the Ensemble Kalman filter or the Particle filter as described in Piazzì et al. (2021) <doi:10.1029/2020WR028390>. airGRdatassim currently runs on daily hydrological models (GR4J, GR5J and GR6J, with and without the CemaNeige snow model). The package is developed at INRAE-Antony ([Catchment Hydrology research group](#) of the HYCAR Research Unit, France). More information about the efficiency of these data assimilation schemes with GR5J can be found in Piazzì et al. (2021).

### ## — Functions and objects

The airGRdatassim package allows users of GR hydrological models to assimilate discharge observations with the aim of improving streamflow simulations. The package has been designed to allow the choice between two sequential ensemble-based data assimilation (DA) techniques, namely the Ensemble Kalman filter (EnKF) and the Particle filter (PF).

The functions are coded in R and both their names and arguments are consistent with the airGR package.

With the aim of providing a user-friendly package, airGRdatassim relies on two main functions:

- [CreateInputsPert](#) generates the probabilistic model inputs to perform ensemble-based DA when accounting for the uncertainty in meteorological forcings;
- [RunModel\\_DA](#) performs streamflow ensemble simulations with the assimilation of observed discharges through the EnKF or the PF schemes.

Consistently with the airGR package, both structure and class of function arguments are specifically defined to prevent from mis-use and to ensure the flexibility of functions. Advanced users wishing to apply the package to their own models will need to comply with these imposed structures and refer to the package source codes to get all the specification requirements.

### ## — Models

DA schemes are designed to be coupled with GR daily hydrological models, which are implemented in the airGR package. These models can be called within the airGRdatassim package using the following airGR functions (use the command '?airGR' to get the references of the GR models):

- [RunModel\\_GR4J](#): four-parameter daily lumped hydrological model
- [RunModel\\_GR5J](#): five-parameter daily lumped hydrological model
- [RunModel\\_GR6J](#): six-parameter daily lumped hydrological model
- [RunModel\\_CemaNeigeGR4J](#): combined use of GR4J and CemaNeige
- [RunModel\\_CemaNeigeGR5J](#): combined use of GR5J and CemaNeige

- [RunModel\\_CemaNeigeGR6J](#): combined use of GR6J and CemaNeige

### ## — How to get started

Because `airGRdatassim` is an `airGR`-based package, specific `airGR` functions should be jointly used to ensure the proper use of the `airGRdatassim` tools. Indeed, before performing the DA-based streamflow simulations, the hydrological model needs to be calibrated through the `airGR` calibration function. Therefore, the following steps are recommended:

1. refer to the help for [Calibration\\_Michel](#) in the `airGR` package, run the provided example and then refer to the help for [CreateCalibOptions](#) to understand how a model calibration is prepared/made;
2. refer to the help for [CreateInputsPert](#) to understand how the probabilistic model inputs are generated, if the uncertainty in meteorological forcings is taken into account;
3. refer to the help for [RunModel\\_DA](#) to understand how to perform the DA-based streamflow simulations;
4. refer to the help for [ErrorCrit\\_NSE](#) and [CreateInputsCrit](#) in the `airGR` package to understand how the computation of an error criterion is prepared/made.

For more information and to get started with the package, you can refer to the vignette: `vignette("get_started", package = "airGRdatassim")`.

### ## — References

- Piazzzi, G., Thirel, G., Perrin, C. and Delaigue, O. (2021). Sequential data assimilation for streamflow forecasting: assessing the sensitivity to uncertainties and updated variables of a conceptual hydrological model at basin scale. *Water Resources Research*, 57, doi:[10.1029/2020WR028390](https://doi.org/10.1029/2020WR028390).

---

CreateInputsPert

*Generation of ensemble model inputs for data assimilation*

---

## Description

Function which perturbs the model inputs to generate probabilistic meteorological forcings required to perform ensemble-based data assimilation.

## Usage

```
CreateInputsPert(FUN_MOD, DatesR, Precip = NULL,
                 PotEvap = NULL, TempMean = NULL,
                 ZInputs = NULL, HypsoData = NULL, NLayers = 5,
                 NbMbr = 50, Seed = NULL)
```

```
## S3 method for class 'InputsPert'
x[i]
```

```
## S3 method for class 'InputsPert'
plot(x, which = "all", main = NULL,
     ColPrecip = "royalblue", ColPotEvap = "green3",
     ask = prod(par("mfcol")) < length(which) && dev.interactive(), ...)
```

### Arguments

FUN_MOD	[function] hydrological model function (e.g. <a href="#">RunModel_GR5J</a> , <a href="#">RunModel_CemaNeigeGR5J</a> )
DatesR	[POSIXct] vector of dates
Precip	(optional) [numeric] time series of total precipitation to perturb [mm/d]
PotEvap	(optional) [numeric] time series of potential evapotranspiration to perturb [mm/d]
TempMean	(optional) [numeric] time series of mean air temperature [°C] (not perturbed), required to create the CemaNeige module inputs
ZInputs	(optional) [numeric] real giving the mean elevation of the Precip and Temp series (before extrapolation) [m], possibly used to create the CemaNeige module inputs
HypsoData	(optional) [numeric] vector of 101 reals: min, q01 to q99 and max of catchment elevation distribution [m], if not defined a single elevation is used for CemaNeige
NLayers	(optional) [numeric] integer giving the number of elevation layers requested [-], required to create CemaNeige module inputs, default=5
NbMbr	(optional) [numeric] number of ensemble members (minimum of 20 recommended for the EnKF scheme and of 30 for the PF scheme)
Seed	(optional) [numeric] seed of random number generator
x	[InputsPert] containing the vector of dates ( <i>POSIXt</i> ) and the time series of numeric values list perturbed
i	[integer] of the indices to subset a time series or [character] names of the elements to extract
which	(optional) [character] choice of plots (e.g. "Precip" or "PotEvap", default = "all")
main	(optional) [character] an overall title for the plot (see <a href="#">title</a> )
ColPrecip, ColPotEvap	(optional) [character] color to be used for perturbed precipitation and perturbed potential evapotranspiration (in any format that <a href="#">col2rgb</a> accepts)
ask	(optional) [logical] if TRUE, the user is asked before each plot, see <a href="#">par</a> (ask = .)
...	other parameters to be passed through to plotting functions

### Details

The function generates an ensemble of precipitation or/and potential evapotranspiration time series, from data provided as function argument/s. The mean air temperature required to create CemaNeige inputs is not perturbed.

Probabilistic forcing/s is/are generated by stochastically perturbing the meteorological variable/s through a multiplicative stochastic noise, according to the methodology proposed by Clark et al. (2008).

The random perturbations are provided through a first-order autoregressive model relying on a fractional error parameter equal to 0.65 and a temporal decorrelation length of 1 day for rainfall and 2 days for potential evapotranspiration.

In order to ensure reproducible results, *Seed* can be set to fix the randomness in the generation of perturbations.

For further details, see the references section.

Nota: The function can be applied when using GR4J, GR5J and GR6J models (i.e. daily model time step), with or without the CemaNeige module.

### Value

[list] of class 'InputsModel' containing the ensembles of perturbed model inputs required to perform data assimilation:

DatesR	[POSIXlt] vector of dates
Precip	[numeric] matrix (dim(NbTime, NbMbr)) of ensemblist time series of perturbed total precipitation [mm/d] (with NbTime the length of the period; generated only if the precipitation time series is provided as a function argument)
PotEvap	[numeric] matrix (dim(NbTime, NbMbr)) of ensemblist time series of perturbed potential evapotranspiration [mm/d] (generated only if the time series of potential evapotranspiration is provided as a function argument)
NbMbr	[integer] atomic vector of number of ensemble members

On the graphical outputs:

- solid line: medians of the input values
- polygon: minima and maxima of the input values

### Author(s)

Gaia Piazzzi, Olivier Delaigue

### References

- Clark, M. P., Rupp, D. E., Woods, R. A., Zheng, X., Ibbitt, R. P., Slater, A. G. et al. (2008). Hydrological data assimilation with the ensemble Kalman filter: Use of streamflow observations to update states in a distributed hydrological model. *Advances in Water Resources*, 31(10), 1309-1324, doi:10.1016/j.advwatres.2008.06.005
- Piazzzi, G., Thirel, G., Perrin, C. and Delaigue, O. (2021). Sequential data assimilation for streamflow forecasting: assessing the sensitivity to uncertainties and updated variables of a conceptual hydrological model at basin scale. *Water Resources Research*, 57, doi:10.1029/2020WR028390.

**See Also**[RunModel\\_DA](#)**Examples**

```

library(airGRdatassim)

## loading catchment data
data(L0123001, package = "airGR")

## preparation of the InputsModel object
InputsModel <- CreateInputsModel(FUN_MOD = RunModel_GR5J, DatesR = BasinObs$DatesR,
                                Precip = BasinObs$P, PotEvap = BasinObs$E)

## run period selection
IndRun <- seq(which(format(BasinObs$DatesR, format = "%Y-%m-%d")=="2006-01-01"),
              which(format(BasinObs$DatesR, format = "%Y-%m-%d")=="2006-01-31"))

## preparation of perturbed meteorological ensemble
InputsPert <- CreateInputsPert(FUN_MOD = RunModel_GR5J,
                              DatesR = BasinObs$DatesR,
                              Precip = BasinObs$P,
                              PotEvap = BasinObs$E,
                              NbMbr = 100L)

str(InputsPert)

## results preview
oldpar <- par(mfrow = c(2, 1))
plot(InputsPert)
par(oldpar)

## results preview on a subset on one perturbed variable
oldpar <- par(mfrow = c(1, 1))
plot(InputsPert[IndRun], which = "PotEvap")
par(oldpar)

```

---

**RunModel\_DA***Run discharge simulations with ensemble-based data assimilation*

---

**Description**

Function which performs discharge ensemble simulations with the assimilation of observed discharges through the Ensemble Kalman filter (EnKF) or the Particle filter (PF) schemes. More information about the efficiency of these data assimilation schemes with GR5J can be found in Piazzoli et al. (2021).

**Usage**

```

RunModel_DA(InputsModel, InputsPert = NULL, Qobs = NULL,
             IndRun,
             FUN_MOD, Param,
             DaMethod = c("EnKF", "PF", "none"), NbMbr = NULL,
             StateEnKF = NULL, StatePert = NULL,
             Seed = NULL)

## S3 method for class 'OutputsModelDA'
x[i]

## S3 method for class 'OutputsModelDA'
plot(x, Qobs = NULL, main = NULL,
      ColSim = "orangered", ColObs = par("fg"), ...)

```

**Arguments**

InputsModel	[list] object of class <i>InputsModel</i> containing the data required to run the model (see <a href="#">CreateInputsModel</a> for details)
InputsPert	(optional) [list] object of class <i>InputsModel</i> containing the ensembles of perturbed data required to evaluate the model probabilistic outputs, if the uncertainty in meteorological forcings is taken into account. The variables data proposed in <i>InputsPert</i> will erase the variables data given in <i>InputsModel</i>
Qobs	(optional) [numeric] time series of observed flow [mm/d]
IndRun	[numeric] index of period to be used for the model run [-]
FUN_MOD	[function] daily hydrological model functions (GR4J, GR5J or GR6J; e.g. <a href="#">RunModel_GR5J</a> , <a href="#">RunModel_CemaNeigeGR5J</a> )
Param	[numeric] vector of model parameters (number of parameters depends on the used hydrological model, from 4 parameters in GR4J up to 10 parameters in GR6J with CemaNeige)
DaMethod	[character] name of the data assimilation technique: <ul style="list-style-type: none"> <li>"EnKF" discharge assimilation via Ensemble Kalman filter</li> <li>"PF" discharge assimilation via Particle filter</li> <li>"none" discharge assimilation is not performed (i.e. open-loop simulation)</li> </ul>
NbMbr	(optional) [numeric] number of ensemble members (minimum of 20 recommended for the EnKF scheme and of 30 for the PF scheme; by default=NULL: 50 if <i>InputsPert</i> is not set or <i>InputsPert\$NbMbr</i> otherwise)
StateEnKF	[character] vector of the names of state variables to be updated via EnKF: <ul style="list-style-type: none"> <li>"Prod" level of the production store [mm]</li> <li>"Rout" level of the routing store [mm]</li> <li>"UH1" unit hydrograph 1 levels [mm] (not defined for the GR5J model)</li> <li>"UH2" unit hydrograph 2 levels [mm]</li> </ul>

StatePert	[character] vector of the names of state variables to be perturbed via EnKF or PF:
"Prod"	level of the production store [mm]
"Rout"	level of the routing store [mm]
"UH1"	unit hydrograph 1 levels [mm] (not defined for the GR5J model)
"UH2"	unit hydrograph 2 levels [mm]
Seed	(optional) [numeric] seed of random number generator
x	[OutputsModelDA] containing the vector of dates ( <i>POSIXt</i> ) and the time series of numeric values DA model outputs
i	[integer] of the indices to subset a time series or [character] names of the elements to extract
main	(optional) [character] an overall title for the plot (see <a href="#">title</a> )
ColSim, ColObs	(optional) [character] color to be used for simulated flow and observed flow (in any format that <a href="#">col2rgb</a> accepts)
...	other parameters to be passed through to plotting functions

## Details

Discharge observations are sequentially assimilated at each time step (if *DaMethod* != "none") via the EnKF or PF schemes. Because of the sequential approach, the analysis states resulting from the assimilation procedure at time step  $t$  are used as initial states at the following prediction time step  $t + 1$ .

When accounting for the uncertainty in model inputs (see *InputsPert*), the ensemble discharge simulations are driven by perfect meteorological forecasts, which are generated according to the methodology proposed by Clark et al. (2008) (see [CreateInputsPert](#) for details).

It is possible to enable/disable the update of specific state variables via EnKF by defining/not defining their names in *StateEnKF*. Please note that *StateEnKF* is therefore required only when using the EnKF.

Both the PF and EnKF can account for uncertainty in model states by perturbing the state variables specified in *StatePert* (Salamon and Feyen, 2009). It is noteworthy that, when using the EnKF, the perturbation is allowed only for the state variables included in *StateEnKF*. If the state uncertainty is taken into account, the initial states at the prediction time step  $t + 1$  are the perturbed analysis states resulting from the assimilation and perturbation procedures at time step  $t$ .

In order to ensure reproducible results, *Seed* can be set to fix the randomness in the generation of perturbations.

For further details and guidelines on the choice of the DA technique, see the references section.

Nota: The function can be applied when using GR4J, GR5J and GR6J models (i.e. daily model time step), with or without the CemaNeige module (see [airGR](#) package).



**Value**

[list] 'OutputsModel' containing the outputs organised as follows:

DatesR	[POSIXlt] series of dates (length of IndRun)
QsimEns	[numeric] matrix (dim(NbTime, NbMbr)) of ensemble discharge simulations
EnsStateBkg	[numeric] array (dim(NbTime, NbMbr, Nstate)) of ensemble values of background model states (before the filter update)
EnsStateA	[numeric] array (dim(NbTime, NbMbr, Nstate)) of ensemble values of analysis model states (after the filter update)
NbTime	[integer] atomic vector of length of IndRun
NbMbr	[integer] atomic vector of number of ensemble members
NbState	[integer] atomic vector of number of of model states

On the graphical outputs:

- solid line: medians of the input values
- polygon: minima and maxima of the input values

**Author(s)**

Gaia Piazzzi, Olivier Delaigue

**References**

- Clark, M. P., Rupp, D. E., Woods, R. A., Zheng, X., Ibbitt, R. P., Slater, A. G. et al. (2008). Hydrological data assimilation with the ensemble Kalman filter: Use of streamflow observations to update states in a distributed hydrological model. *Advances in Water Resources*, 31(10), 1309-1324, doi:[10.1016/j.advwatres.2008.06.005](https://doi.org/10.1016/j.advwatres.2008.06.005)
- Piazzzi, G., Thirel, G., Perrin, C. and Delaigue, O. (2021). Sequential data assimilation for stream-flow forecasting: assessing the sensitivity to uncertainties and updated variables of a conceptual hydrological model at basin scale. *Water Resources Research*, 57, doi:[10.1029/2020WR028390](https://doi.org/10.1029/2020WR028390).
- Salamon, P. and Feyen, L. (2009). Assessing parameter, precipitation, and predictive uncertainty in a distributed hydrological model using sequential data assimilation with the particle filter. *Journal of Hydrology*, 376(3-4), 428-442, doi:[10.1016/j.jhydrol.2009.07.051](https://doi.org/10.1016/j.jhydrol.2009.07.051)

**See Also**

[CreateInputsPert](#)

**Examples**

```
library(airGRdatassim)

## loading catchment data
data(L0123001, package = "airGR")
Param <- c(X1 = 194.243, X2 = -0.088, X3 = 117.740, X4 = 1.680, X5 = 0.000)
```

```
## run period selection
IndRun <- seq(which(format(BasinObs$DatesR, format = "%Y-%m-%d")=="2006-01-01"),
             which(format(BasinObs$DatesR, format = "%Y-%m-%d")=="2006-01-31"))

## preparation of the InputsModel object
InputsModel <- CreateInputsModel(FUN_MOD = RunModel_GR5J, DatesR = BasinObs$DatesR,
                                Precip = BasinObs$P, PotEvap = BasinObs$E)

## number of ensemble members
## minimum of 20 recommended for the EnKF scheme
## minimum of 30 recommended for the PF scheme
NbMbr <- 20L

## preparation of perturbed meteorological ensemble
InputsPert <- CreateInputsPert(FUN_MOD = RunModel_GR5J,
                              DatesR = BasinObs$DatesR,
                              Precip = BasinObs$P, PotEvap = BasinObs$E,
                              NbMbr = NbMbr)

## simulation with DA via EnKF
OutputsModelDA <- RunModel_DA(InputsModel = InputsModel,
                              InputsPert = InputsPert,
                              Qobs = BasinObs$Qmm,
                              IndRun = IndRun,
                              FUN_MOD = RunModel_GR5J, Param = Param,
                              DaMethod = "EnKF", NbMbr = NbMbr,
                              StateEnKF = c("Prod", "Rout"),
                              StatePert = c("Prod", "Rout"))

## results preview
plot(OutputsModelDA, Qobs = BasinObs$Qmm[IndRun])

## results preview on a subset
plot(OutputsModelDA[1:10], Qobs = BasinObs$Qmm[IndRun][1:10])
```

# Index

- \* **GR4J**
  - airGRdatassim, 2
- \* **data assimilation**
  - airGRdatassim, 2
- \* **ensemble**
  - airGRdatassim, 2
- \* **hydrology**
  - airGRdatassim, 2
- \* **model**
  - airGRdatassim, 2

[.InputsPert (CreateInputsPert), 3  
[.OutputsModelDA (RunModel\_DA), 6

airGR, 8  
airGRdatassim, 2  
airGRdatassim-package (airGRdatassim), 2

Calibration\_Michel, 3  
col2rgb, 4, 8  
CreateCalibOptions, 3  
CreateInputsCrit, 3  
CreateInputsModel, 7  
CreateInputsPert, 2, 3, 3, 8, 9

ErrorCrit\_NSE, 3

par, 4  
plot.InputsPert (CreateInputsPert), 3  
plot.OutputsModelDA (RunModel\_DA), 6

RunModel\_CemaNeigeGR4J, 2  
RunModel\_CemaNeigeGR5J, 2, 4, 7  
RunModel\_CemaNeigeGR6J, 3  
RunModel\_DA, 2, 3, 6, 6  
RunModel\_GR4J, 2  
RunModel\_GR5J, 2, 4, 7  
RunModel\_GR6J, 2

title, 4, 8