Latest developments of the airGR rainfall-runoff modelling R package: new calibration procedures and other features

O. Delaigue, G. Thirel, F. Bourgin, L. Coron

To cite this version:


HAL Id: hal-02607857
https://hal.inrae.fr/hal-02607857
Submitted on 16 May 2020

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.
GR is a family of lumped hydrological models designed for flow simulation at various time steps. The models are freely available in an R package called airGR (Coron et al., 2017a, 2017b). The models can easily be implemented on a set of catchments with limited data requirements.

### GR hydrological models
- Designed with the objective to be as efficient as possible for flow simulation at various time steps (from hourly to interannual)
- Warranted complexity structures and limited data requirements
- Can be applied on a wide range of conditions, including snowy catchments (CemaNeige snow routine included)

### How to use other R packages to perform parameters estimation
- Definition of the necessary function
  - transformation of parameters to real space (available in airGR)
  - computation of the value of the performance criterion (e.g. RMSE)

```r
OptimGR4J <- function(Param_Optim) {
  Param_Optim_Vre <- airGR::TransfoParam_GR4J(ParamIn = Param_Optim,
                                             Direction = "TR")
  OutputModel <- airGR::RunModel_GR4J(InputModel = InputCrit,
                                             OutputCrit = airGR::ErrorCrit_RMSE(InputCrit = InputCrit, OutputModel = OutputCritModel))
  return(OutputCrit$CritValue)
}
```
- Definition of the lower and upper bounds of the four GR4J parameters in the transformed parameter space
  - lowerGR4J <- rep(-9.99, times = 4)
  - upperGR4J <- rep(+9.99, times = 4)

### Local optimisation
- Single-start (here) or multi-start approach to test the consistency of the local optimisation

```r
optPORT_ <- function(x) {
  startGR4J <- expand.grid(data.frame(CalibOptions$StartParamDistrib))
  startGR4J <- c(4.1, 3.9, -0.9, -8.7)
  lowerGR4J <- rep(-9.99, times = 4)
  upperGR4J <- rep(+9.99, times = 4)
  lower = lowerGR4J, upper = upperGR4J,
  control = list(write2disk = FALSE)
  optPORT <- hydroPSO::hydroPSO(fn = OptimGR4J,
                                 lower = lowerGR4J, upper = upperGR4J,
                                 control = list(trace = 1))
}
```

### Global optimisation
- Most often used when facing a complex response surface, with multiple local minima

```r
optDE <- DEoptim::DEoptim(fn = OptimGR4J,
                          lower = lowerGR4J, upper = upperGR4J,
                          control = DEoptim::control(FO = 80, trace = 10))
```

### Results

<table>
<thead>
<tr>
<th>Algo</th>
<th>X1</th>
<th>X2</th>
<th>X3</th>
<th>X4</th>
<th>RMSE</th>
<th>CPU</th>
</tr>
</thead>
<tbody>
<tr>
<td>airGR</td>
<td>257.238</td>
<td>1.012</td>
<td>88.235</td>
<td>2.208</td>
<td>0.7852</td>
<td>00.53 s</td>
</tr>
<tr>
<td>PORT</td>
<td>256.808</td>
<td>1.004</td>
<td>88.167</td>
<td>2.205</td>
<td>0.7852</td>
<td>01.79 s</td>
</tr>
<tr>
<td>PSO</td>
<td>256.836</td>
<td>1.006</td>
<td>88.207</td>
<td>2.204</td>
<td>0.7852</td>
<td>29.55 s</td>
</tr>
<tr>
<td>MA-LS</td>
<td>256.808</td>
<td>1.004</td>
<td>88.167</td>
<td>2.205</td>
<td>0.7852</td>
<td>18.42 s</td>
</tr>
</tbody>
</table>

### News since EGU 2017 – airGR 1.0.9.64 vs airGR 1.0.5.12
- The Param.Sets GR4J dataset was added. It contains generalist parameter sets for the GR4J model
- If the calibration period is too short (< 6 months) and by consequence non representative of the catchment behaviour, a local calibration algorithm can give poor results and we recommend to use the generalist parameter sets instead
- Vignettes were added. They explain how to perform parameters estimation with:
  - Differential Evolution calibration algorithm
  - Particle Swarm calibration algorithm
  - MA-LS-Chains calibration algorithm
  - Bayesian MCMC framework
- A new airGRteaching package (Delagade et al., 2018) provides tools to simplify the use of the airGR hydrological package for education, including a ‘Shiny’ interface

### Future developments
- New version of CemaNeige that allows to use satellite snow cover area for calibration (Riboulet et al., accepted)
- Parameters maps on France for GR4J, GR5J & GR6J models for ungauged basins (Poncelet et al., submitted)