Package ‘TUWmodel’

June 7, 2019

Type Package

Title Lumped/Semi-Distributed Hydrological Model for Education Purposes

Version 1.0-2

Date 2019-06-05

Depends R (>= 3.0.0), stats

Description
The model, developed at the Vienna University of Technology, is a lumped conceptual rainfall-runoff model, following the structure of the HBV model.
The model can also be run in a semi-distributed fashion.
The model runs on a daily or shorter time step and consists of a snow routine, a soil moisture routine and a flow routing routine.
See Parajka, J., R. Merz, G. Bloeschl (2007) <DOI:10.1002/hyp.6253> Uncertainty and multiple objective calibration in regional water balance modelling: case study in 320 Austrian catchments, Hydrological Processes, 21, 435-446.

NeedsCompilation yes

License GPL (>= 2)

Author Alberto Viglione [aut, cre],
Juraj Parajka [aut]

Maintainer Alberto Viglione <viglione@hydro.tuwien.ac.at>

Repository CRAN

Date/Publication 2019-06-07 18:51:39 UTC

R topics documented:

example_TUWmodel *Data-sample*

Description

Data for examples in the package

Usage

```
data(example_TUWmodel)
```

Format

The data are time series for the river Vils at Vils (Laende):

1. Q_Vils vector of observed daily discharges [mm/day];
2. P_Vils matrix of observed daily precipitation [mm/day] for 6 zones;
3. T_Vils matrix of observed daily temperatures [degC] for 6 zones;
4. PET_Vils matrix of potential evapotranspiration [mm/day] for 6 zones;
5. SWE_Vils matrix of observed snow water equivalent [mm] for 6 zones;
6. areas_Vils areas of the 6 zones.

Examples

```
data(example_TUWmodel)

ls()

t <- as.Date(rownames(P_Vils))
plot(t, apply(P_Vils, 1, weighted.mean, w=areas_Vils),
     type="S", xlab="", ylab="Precipitation [mm/day]")

plot(t, apply(PET_Vils, 1, weighted.mean, w=areas_Vils),
     type="l", xlab="", ylab="Potential evapotranspiration [mm/day]")

plot(t, T_Vils[,1], type="l", xlab="", ylab="Temperature [degC]")
 lines(t, T_Vils[,6], col=3)

plot(t, SWE_Vils[,6], col=3, type="l", xlab="", ylab="Snow water equivalent [mm]")
 lines(t, SWE_Vils[,1], col=1)

plot(as.Date(names(Q_Vils)), Q_Vils, type="l", xlab="", ylab="Runoff [mm/day]")
```

TUWmodel — *Lumped hydrological model developed at the Vienna University of Technology for education purposes*

Description

TUWmodel is a lumped conceptual rainfall-runoff model, following the structure of the HBV model. The model runs on a daily or shorter timestep and consists of a snow routine, a soil moisture routine and a flow routing routine. See Parajka, J., R. Merz, G. Bloeschl (2007) Uncertainty and multiple objective calibration in regional water balance modelling: case study in 320 Austrian catchments, Hydrological Processes, 21, 435-446.

Usage

```
TUWmodel (prec, airt, ep, area=1,
          param=c(1.2,1.2,2,-2,0,0.9,100,3.3,0.5,9,105,50,2,10,26.5),
          incon=c(50,0,2.5,2.5), itsteps=NULL)
```

Arguments

`prec` vector/matrix of precipitation input [mm/timestep] (`ncol` = number of zones)

`airt` vector/matrix of air temperatures [degC]

`ep` vector/matrix of potential evapotranspiration [mm/timestep]

`area` if more zones, vector of the percentage of area for each zone (or proportional to it, i.e., if the sum is different from 1, it will be rescaled to be 1)

`param` vector/matrix of parameters (`ncol` = number of zones):

1. `SCF` snow correction factor [-] (e.g., 0.9-1.5);
2. `DDF` degree day factor [mm/degC/timestep] (e.g., 0.0-5.0 mm/degC/day);
3. `Tr` threshold temperature above which precipitation is rain [degC] (e.g., 1.0-3.0 degC);
4. `Ts` threshold temperature below which precipitation is snow [degC] (e.g., -3.0-1.0 degC);
5. `Tm` threshold temperature above which melt starts [degC] (e.g., -2.0-2.0 degC);
6. `LPrat` parameter related to the limit for potential evaporation [-] (e.g., 0.0-1.0);
7. `FC` field capacity, i.e., max soil moisture storage [mm] (e.g., 0-600 mm);
8. `BETA` the non linear parameter for runoff production [-] (e.g., 0.0-20.0);
9. `k0` storage coefficient for very fast response [timestep] (e.g., 0.0-2.0 days);
10. `k1` storage coefficient for fast response [timestep] (e.g., 2.0-30.0 days);
11. `k2` storage coefficient for slow response [timestep] (e.g., 30.0-250.0 days);
12. `lsuz` threshold storage state, i.e., the very fast response start if exceeded [mm] (e.g., 1.0-100.0 mm);

13. cperc constant percolation rate [mm/timestep] (e.g., 0.0-8.0 mm/day);
14. bmax maximum base at low flows [timestep] (e.g., 0.0-30.0 days);
15. croute free scaling parameter [timestep^2/mm] (e.g., 0.0-50.0 days^2/mm);

incon
: vector/matrix of initial conditions for the model (ncol = number of zones): SSM0 soil moisture [mm]; SWE0 snow water equivalent [mm]; SUZ0 initial value for fast (upper zone) response storage [mm]; SLZ0 initial value for slow (lower zone) response storage [mm]

itsteps
: length of the output (if NULL all the time series are used)

Details

More details about the model structure are given in the Appendix of

Parajka, J., R. Merz, G. Bloeschl (2007) Uncertainty and multiple objective calibration in regional water balance modelling: case study in 320 Austrian catchments, Hydrological Processes, 21, 435-446, doi:10.1002/hyp.6253.

An example of using TUWmodel, including R scripts for automatic calibration, can be found in the Supplement of

Ceola, S., et al. (2015) Virtual laboratories: new opportunities for collaborative water science, Hydrology and Earth System Sciences, 19(4), 2101-2117, doi:10.5194/hess-19-2101-2015.

Value

TUWmodel gives a vector of simulated runoff as q [mm/timestep], and the following vector/matrices:

1. qzones simulated runoff for each zone [mm/timestep];
2. q0 surface runoff [mm/timestep];
3. q1 subsurface runoff [mm/timestep];
4. q2 baseflow [mm/timestep];
5. rain liquid precipitation [mm/timestep];
6. snow solid precipitation [mm/timestep];
7. melt snowmelt [mm/timestep];
8. moist soil moisture [mm];
9. swe snow water equivalent [mm];
10. eta actual evapo-transpiration [mm/timestep];
11. suz upper storage zone [mm];
12. slz lower storage zone [mm];

Examples

```
## Load the data
data(example_TUWmodel)


## Simulate runoff and plot observed vs simulated series
## Lumped case (weighted means of the inputs)
```

```
simLump <- TUWmodel(prec=apply(P_Vils, 1, weighted.mean, w=areas_Vils),
                    airt=apply(T_Vils, 1, weighted.mean, w=areas_Vils),
                    ep=apply(PET_Vils, 1, weighted.mean, w=areas_Vils),
                    area=sum(areas_Vils),
             param=c(1.02,1.70,2,0,-0.336,
                     0.934,121,2.52,
                     0.473,9.06,142,
                     50.1,2.38,10,25))

plot(as.Date(names(Q_Vils)), Q_Vils, type="l", xlab="", ylab="Discharges [mm/day]")
 lines(as.Date(rownames(T_Vils)), simLump$q, col=2)
legend("topleft", legend=c("Observations","Simulations"), col=c(1,2), lty=1, bty="n")

plot(as.Date(rownames(SWE_Vils)), apply(SWE_Vils, 1, weighted.mean, w=areas_Vils),
     type="l", xlab="", ylab="Snow Water Equivalent [mm]")
 lines(as.Date(rownames(T_Vils)), simLump$swe, col=2)

## Distribute input case (6 zones)
simDist <- TUWmodel(prec=P_Vils, airt=T_Vils, ep=PET_Vils, area=areas_Vils/sum(areas_Vils),
             param=c(1.02,1.70,2,0,-0.336,
                     0.934,121,2.52,
                     0.473,9.06,142,
                     50.1,2.38,10,25))

plot(as.Date(names(Q_Vils)), Q_Vils, type="l", xlab="", ylab="Discharges [mm/day]")
 lines(as.Date(rownames(T_Vils)), simDist$q, col=2)
legend("topleft", legend=c("Observations","Simulations"), col=c(1,2), lty=1, bty="n")

plot(as.Date(rownames(SWE_Vils)), apply(SWE_Vils, 1, weighted.mean, w=areas_Vils),
     type="l", xlab="", ylab="Snow Water Equivalent [mm]")
 lines(as.Date(rownames(T_Vils)), apply(simDist$swe, 1, weighted.mean, w=areas_Vils), col=2)

## Distributed input and parameters case
parametri <- matrix(rep(c(1.02,1.70,2,0,-0.336,
                          0.934,121,2.52,
                          0.473,9.06,142,
                          50.1,2.38,10,25), 6), ncol=6)
parametri[2,] <- c(1.4, 1.7, 1.9, 2.2, 2.4, 3.0)
simDist2 <- TUWmodel(prec=P_Vils,
                    airt=T_Vils,
                    ep=PET_Vils,
                    area=areas_Vils/sum(areas_Vils),
             param=parametri)

plot(as.Date(names(Q_Vils)), Q_Vils, type="l", xlab="", ylab="Discharges [mm/day]")
 lines(as.Date(rownames(T_Vils)), simDist2$q, col=2)
legend("topleft", legend=c("Observations","Simulations"), col=c(1,2), lty=1, bty="n")
```

Index