Package ‘dynatopmodel’

January 6, 2016

Type Package

Title Implementation of the Dynamic TOPMODEL Hydrological Model

Version 1.1

Date 2016-01-01

Author Peter Metcalfe, Keith Beven and Jim Freer

Maintainer Peter Metcalfe <p.metcalfe@lancaster.ac.uk>

Description A native R implementation and enhancement of the Dynamic TOPMODEL semi-distributed hydrological model. Includes some pre-processsing and output routines.

Depends R (>= 2.10),

Imports deSolve, rgeos, maptools, rgdal, zoo, xts, sp, raster, lubridate, topmodel, methods, grDevices, stats, utils, graphics, tools

License GPL-2

RoxygenNote 5.0.1

LazyData true

NeedsCompilation no

Repository CRAN

Date/Publication 2016-01-06 17:22:27

R topics documented:

`aggregate_obs` *Resample observation data at a new time interval*

Description

Takes a list of time series and resample to a new interval.

Usage

```
aggregate_obs(obs, dt, is.rate = TRUE)
```

Arguments

`obs`	List of times series (zoo) objects with a POSIXct index.
`dt`	New time interval, hours.
`is.rate`	If TRUE then these are rates i.e m/hr. Otherwise they are absolute values across the interval and are scaled before return by a factor equal to the ratio of the old interval to the new interval.

Details

Time series of observation data are often of different temporal resolutions, however the input to most hydrological models, as is the case with the Dynamic TOPMODEL, requires those data at the same interval. This provides a method to resample a collection of such data to a single interval.

Value

The list of observations resampled at the new interval.

Examples

```
# Resample Brompton rainfall and PE data to 15 minute intervals
require(dynatopmodel)
data("brompton")

obs <- aggregate_obs(list("rain"=brompton$rain, "pe"=brompton$pe), dt=15/60)

# check totals for Sept - Oct 2012
sum(obs$rain*15/60, na.rm=TRUE)
sum(brompton$rain, na.rm=TRUE)
```

`approx.pe.ts` *Create sinsuiodal time series of potential evapotranspiration input*

Description

Create sinsuiodal time series of potential evapotranspiration input

Usage

```
approx.pe.ts(start, end, dt = 1, emin = 0, emax = 5/1000)
```

Arguments

`start`	Start time of returned series in a format that can be coerced into a POSIXct instance. Defaults to start of rainfall data
`end`	End time for returned series in a format that can be coerced into a POSIXct instance. Defaults to end of rainfall datA
`dt`	Time interval in hours
`emin`	Minimum daily PE total (m or mm)
`emax`	Maximum daily PE total (m or mm)

Details

Dynamic TOPMODEL requires a time series of potential evapotranspiration in order to calculate and remove actual evapotranpiration from the root zone during a run. Many sophisticated physical models have been developed for estimating PE and AE, including the Priestly-Taylor (Priestley and Taylor, 1972) and Penman-Monteith (Montieth, 1965) methods. These, however, require detailed meterological data such as radiation input and relative humidities that are, in general, difficult to obtain. Calder (1983) demonstrated that a simple approximation using a sinusoidal variation in potential evapotranspiration to be a good approximation to more complex schemes.

If the insolation is also taken to vary sinusoidally through the daylight hours then, ignoring diurnal meteorological variations, the potential evapotranspiration during daylight hours for each year day number can be calculated (for the catchment's latitude). Integration over the daylight hours allows the daily maximum to be calculated and thus a sub-daily series generated.

Value

Time series (xts) of potential evapotranspiration ([L]/[T]) covering the given time range and at the desired interval in m or mm/hr

References

Beven, K. J. (2012). Rainfall-runoff modelling : the primer. Chichester, UK, Wiley-Blackwell.

Calder, I. R. (1986). A stochastic model of rainfall interception. Journal of Hydrology, 89(1), 65-71.

Examples

```
## Not run:
# Create PE data for 2012 for use in the Brompton test case

require(dynatopmodel)

data("brompton")

# Generate time series at hourly and 15 minute intervals
pe.60 <- approx.pe.ts("2012-01-01", "2012-12-31", dt=1)
pe.15 <- approx.pe.ts("2012-01-01", "2012-12-31", dt=0.25)

# Check annual totals - should be around 900mm
sum(pe.60)*1000
sum(pe.15*0.25)*1000

# Check maximum daily total on the 1st of July
sum(pe.60["2012-07-01"])*1000
sum(pe.15["2012-07-01"]*0.25)*1000

## End(Not run)
```

`brompton` *Topographic and observation data for running Dynamic TOPMODEL.*

Description

Brompton is a small (approx 26km^2) agricultural catchment in N.Yorkshire, UK. Its eastern edges rise in the North Yorks Moors and it drains southwards, becoming North Beck before joining the Wiske in Northallerton.

In the late 19th century the area upstream of Water End was drained and turned over to arable cultivation and has since suffered from infrequent, but severe, flooding due to intense autumn rainfall from synoptic systems moving in from the North Sea and high antecedent soil moisture conditions. The last event that flooded the village was in November 2012; flooding was narrowly avoided in the storms of Decemember 2015.

The catchment exhibits high land-channel connectivity due to heavily-modified natural channels and extensive artificial drainage, both surface and subsurface. It has a homogenous land cover, with 95 class 1 arable grassland and crops. The terrain is undulating with slightly acid, base-rich loamy and clayey soils predominating. Distances from the channel appear to exert most influence over the catchment response. Areas close to the channel appear to drain very fast due their connection to the network the presence of field drainage. These have a steep transmissivity, hence a low value for the m parameter. Those areas further away are slower-draining, with a larger value of m. This may provides the long recession tail observed.

The results of a run included, brompton$storm.run, are the September 2012 storm event The simulation is for just over a week. Although giving a NSE of > 0.85, qualitatively the match is fairly poor, particulary at the start of the rising limb of the hydrograph. This may be due to an inaccurate rainfall record. It should be noted that the network width routing approach is unlikely to perform well in flood situations as it cannot deal with a variable wave velocity due to higher channel effciencies at deeper flow or the slowing down of the response as the downstream channels overbank and some flow is diverted via the much rougher riprarian areas.

In our current research will we expect it to be flattened when routing using the physical model under development. This takes into account the increased roughness for flow routed overbank and the much increased flow width across the flood plain. The user is encouraged to experiment with the discretisation and group parameters in order to achieve a better qualitative fit. For example, it is suggested that areas nearest the channel are under the influence of field drainage and this may be a better approach to capturing the catchment's behaviour.

Usage

```
data(brompton)
```

Format

An environment comprising the DEM, river network and processed data necessary to run the model. Includes a previously run example (storm.run).

Examples

```
require(dynatopmodel)
data(brompton)

require(raster)

# Show it
sp::plot(brompton$dem)
```

`build_chans` *Construct a raster of channel locations from vector or topographic data*

Description

The discretise and make.routing.table methods both require a raster defining the locations of the channel cells and the proportion of each river cell occupied by the channel. A detailed river network (DRN) may be available in vector format. If not, the channel location can be inferred from a spatially-distributed metric, typically the topographic wetness index.

Usage

```
build_chans(dem = NULL, drn = NULL, atb = NULL, chan.width = 5,
  buffer = 10, atb.thresh = 0.8, single.chan = TRUE)
```

Arguments

`dem`	Elevation raster (DEM) using a projected coordinate system (e.g UTM) and regular grid spacing. Not required if atb raster supplied.
`drn`	Detailed river network (DRN) in vector (ESRI Shapefile) format. Not required if atb raster supplied.
`atb`	Raster whose values provide a criteria for locating the channel. This is typically the value of the topographic wetness index (TWI) determined from the elevations. Should be in a projected coordinate system (e.g UTM) and regular grid spacing. For the TWI to be meaningful this raster should have a resolution of a least 30m. It can be calculated using the upslope.area method applied to the DEM and atb=T.
`chan.width`	Channel width, in m, which by default will be applied across entire network.
`buffer`	If using a vector input then buffer the DRN by this width to capture all river cells.
`atb.thresh`	If drn not supplied then this specifies the threshold value above which cells are identified as containing part of the channel network
`single.chan`	If using a vector input then individual reach IDs are ignored and the first raster layer returned contains either 1 for a river cell or NA for a non-river cell. Otherwise the discretise method will create an entry for ecah channel ID

Value

A two-band raster with the same dimensions as the elevation or ATB raster whose first layer comprises non-zero cells where identified with the channel and whose second layer holds the proportions of those cells occupied by the channel.

References

Kirkby, M. (1975). Hydrograph modelling strategies. In Peel, R., Chisholm, Michael, Haggett, Peter, & University of Bristol. Department of Geography. (Eds.). Processes in physical and human geography : Bristol essays. pp. 69-90. London: Heinemann Educational.

Examples

```
## Not run:
# Build channel raster in two ways and compare the results (each with a
# nominal 2m channel width). Artificial drainage in this catchment has
# apparently introduced many channels with dimensions below the scale. Their
# existence would not have been inferred simply from examining the topography.
# This high-network connectivity is suggested as a cause of the unexpectedly
# high responsiveness of the catchment given high antecedent moisture conditions.

require(dynatopmodel)
data("brompton")

# (1) Using the wetness index
```

```
a.atb <- upslope.area(brompton$dem, atb=TRUE)
chan.rast.1 <- build_chans(atb=a.atb$atb)

# (2) using the DRN
chan.rast.2 <- build_chans(dem=brompton$dem, drn=brompton$drn, buff=5, chan.width=2)

sp::plot(chan.rast.2[[1]], col="green", legend=FALSE)
sp::plot(chan.rast.1[[1]], col="blue", legend=FALSE, add=TRUE)
legend(fill=c("green", "blue"), legend=c("TWI", "DRN"), title="Method", x="bottomright")

## End(Not run)
```

`build_layers` *Construct basic landscape layer data for Dynamic TOPMODEL run*

Description

Given an elevation raster this function will create a basic multi-band raster that can be used to run Dynamic TOPMODEL after applying a suitable discretisation. It comprises the supplied elevations with the addition of upslope contributing area and topographic wetness index (TWI).

Usage

```
build_layers(dem, fill.sinks = TRUE, deg = 0.1)
```

Arguments

`dem`	Elevation raster using a projected coordinate system (e.g UTM) and regular grid spacing. Should have a resolution of a least 30m for the TWI to be meaningful.
`fill.sinks`	If TRUE (default) then run a sinkfill before calaculating the upslope area and TWI.
`deg`	Threshold intercell slope to determine sinks (degrees).

Value

A multi-band raster (stack) comprising, in order, the elevations, upslope area and topographic wetness index values.

Author(s)

Peter Metcalfe

Examples

```
## Not run:
require(dynatopmodel)
data("brompton")

# Upslope area and wetness index for Brompton catchment
layers <- build_layers(brompton$dem)

sp::plot(layers, main=c("Elevation AMSL (m)", "Upslope area (log(m^2/m))", "TWI ((log(m^2/m))"))

## End(Not run)
```

`build_routing_table` *Generate a network routing table*

Description

Generates a network width table for a catchment. When passed to the run.dtm routine this will be used to route channel flows to the outlet during a Dynamic TOPMODEL run.

Usage

```
build_routing_table(dem, chans = NULL, outlet = NULL, breaks = 5,
  len.fun = flow.lens)
```

Arguments

`dem`	Elevation raster using a projected coordinate system (e.g UTM) and a regular grid spacing. Areas outside the catchment should be set to NA
`chans`	Optional raster of the same dimensions and resolution as the DEM. Non-zero cells in this raster are considered to contain a river channel. If not supplied then flowpaths from the entire catchment area are considered.
`outlet`	Index of cell or cells identified with the catchment outlet
`breaks`	Number of distance intervals
`len.fun`	For large rasters the flow.len function can be very slow and many paths fail to reach a single outlet cell. This applies a simple straight line distance to the outlet to obtain a rough approximation.

Details

Dynamic TOPMODEL routes channel flow to the outlet by a network-width approach (see Beven, 2012, pp. 97-97). A time-delay histogram is produced using the table. When any flow is distributed to the channel "unit" it is immediately redistributed across future time steps according to the proportions found in the histogram. These flows are then added to future outputs from the model.

Value

A two-column data.frame. Its first column is the average flow distance to the outlet, in m, the second the proportions of the catchment channel network within each distance category.

Author(s)

Peter Metcalfe

References

Beven, K. J. (2012). Rainfall-runoff modelling : the primer. Chichester, UK, Wiley-Blackwell.

Examples

```
## Not run:
# Create a routing table for the Brompton test case and show histogram

data(brompton)

tab <- build_routing_table(brompton$dem,
  chans=brompton$reaches,
  breaks=5)
barplot(tab[,2]*100, xlab="Mean flow distance to outlet (m)",
ylab="Network Width %", names.arg=tab[,1])

## End(Not run)
```

`discretise` *Discrete a catchment into hydrological response units (HRUs)*

Description

Discrete a catchment into a set hydrological response units (HRUs) according to any number of landscape layers and cuts

Usage

```
discretise(layers, chans, cuts = list(a = 10), area.thresh = 2/100,
  order.by = names(cuts)[[1]], chan.width = 5)
```

Arguments

`layers`	A multi-band raster (stack) comprising the catchment data. This should be in a projected coordinate system (or none) and have reqular cells. The first layer should be the elevation raster, and subsequent (named) layers should supply the landscape data drawn in to create the discretisation

`chans`	Raster containing channel reach locations, of the same dimensions and resolution of the DEM and other catchment layers. The reaches should be numbered sequentially and any areas not containing part of the channel should be NA. If a second band is supplied with values 0-1 then this is taken to be the proportion of the corresponding non-zero cell occuppied by the channel. If this layer is not present then the proportion is infered from the channel width as p = min(1, chan.width/xres(dem))
`cuts`	A list of cuts of the form layer_name=number. Each name should correspond to a layer name in the layers parameter.
`area.thresh`	Minimum area for response units, expressed as a percentage of the catchment plan area, excluding channel cells. Areas smaller than this are aggregate dwith adjacent areas until exceeding the threshold aea
`order.by`	Name of layer whose values will be use to sort the response units, in decreasing order. Defaults to the name of the first cut
`chan.width`	Channel width, in same units as DEM. Only used if chans doesn't contain a layer to specify the proportion of each river cell comprised of the channel.

Details

This applies the given cuts to the supplied landscape layers to produce areal groupings of the catchment.

Value

A list comprising the following

weights Flux distribution (weighting) matrix. A ngroup x ngroup matrix defining the downslope flux distributions between groups, between land and the channel, and between channel reaches, where nh is the number of land discretisations identified by applying cuts to the catchment layers and nc the numbbr of channel reaches defined. The nth row gives the proportions of flow out of HRU #n to other response units and the channel. Row sums should thus always add to 1. The m th column gives the proportion of flow from the other response units into the m th group.

groups A data frame whose rows comprising the names, plan area and model parameters of each response unit. See Beven and Freer (2001) and Metcalfe et al (2015) for a description of these parameters

hru Multi-band raster comprising the original rasters that the specifed cuts were applied to produce the discretisation; the channel network;the resultant response unit locations

Examples

```
# Landcover and soils are fairly homogenous throughout the Brompton catchment;
# storm response of the appears to be mostly controlled by proximity to the
# channel network. A simple discretisation according to flow distance from the
# nearest channel thus appears to capture the dynamics during the  2012 event
# without introducing unnecessary complexity.
## Not run:
require(dynatopmodel)

data(brompton)
```

```
chans <- build.chans(brompton$dem, drn=brompton$drn, chan.width=2)
# sort by distance but want areas closest the channel to come first
layers <- addLayer(brompton$dem, 2000-brompton$flowdists)
disc <- discretise(layers, cuts=c(flowdists=5), chans=chans, area.thresh=2/100)

write.table(disc$groups, sep="\t", row.names=FALSE)

## End(Not run)
```

`disp.par` *Default list of parameters to control the graphical output during model simulation and via disp.output*

Description

list of parameters to control the graphical output during model simulation. Parameters with names corresponding to the graphical parameters returned by par() will be applied to the plot.

Usage

```
disp.par(...)
```

Arguments

`...`	Further named arguments supplied will overwrite default values

Value

List of parameters with default values. These include: graphics.show Whether to show graphic output. Default TRUE graphics.window.length Width of display window in hours. Default is 120 days. graphics.interval Interval between refreshing the graphical output, in hours. max.q Max discharge (mm/hr) for display max.rain Max rainfall (mm/hr) for display int.q Interval between ticks / line on the y axis, in mm/hr int.time Period between ticks on the time axis, a numerical value in hours or one of "day", "week", "month" prop Proportion of screen occupied by the rainfall hyteograph cex Overall scaling factor of the plot las.time Alignment of time axis labels xmar Size of margin on right and left of plot, inches ymar Size of margin above and below plot col Colours for plot lines: in order, simulated values, observed values

Note

In this version many of the options are now obselete. The most relevant are graphics.show and graphics.interval.

Author(s)

Peter Metcalfe

See Also

disp.output

Examples

```
# Enable graphics output and set display interval to 6 hours
par <- disp.par(graphics.show=TRUE,
      graphics.interval=6)
```

`disp_output` *Display output of a Dynamic TOPMODEL run*

Description

Simple output of the results of a simulation.

Usage

```
disp_output(qsim, rain, evap = NULL, qobs = NULL, tm = NULL, par = NULL,
  start = min(index(qsim)), end = max(index(qsim)), ...)
```

Arguments

`qsim`	Time series of simulated discharges.
`rain`	Time series of rainfall, at same interval as simulated values.
`evap`	Time series of evapotranspiration (optional), at same interval as simulated values.
`qobs`	Time series of evapotranspiration (optional), at same interval as simulated values.
`tm`	Display a vertical line at this time in the simulation. If NULL no line will be drawn.
`par`	Parameters controlling display output. A default set may be obtained through a call to disp.par.
`start`	Start time for plot in a format interpretable as POSIXct date time. Defaults start of simulated discharges.
`end`	End time for plot in a format interpretable as POSIXct date time. Defaults to end of simulated discharges.
`...`	Any further named parameters will be treated as graphics parameters and applied throughout the plot.

Details

This will render the hydrograph, any observations, actual evapotranpiration, if supplied, and the rainfall hyetograph.

Author(s)

Peter Metcalfe

See Also

disp.par

Examples

```
## Not run:
# Show the output of the storm simulation, overriding label colours and vertical axis limits.
require(dynatopmodel)

data(brompton)

x11()
with(brompton$storm.run, disp_output(evap=ae*1000,qobs=qobs*1000,
                                     qsim=qsim*1000, rain=rain*1000,
                                  max.q=2, cex.main=1, col.axis="slategrey", las.time=1))

## End(Not run)
```

`dynatopmodel` *Implementation of the Dynamic TOPMODEL hydrological model.*

Description

A native R implementation and enhancement of Dynamic TOPMODEL, Beven and Freers (2001) extension to the semi-distributed hydrological model TOPMODEL. It includes some digital terrain analysis functions for discretisation of catchments by topographic indexes and other geo-referenced layers supplying relevant landscape data.

TOPMODEL (Beven & Kirkby, 1979) is a well-established and widely used hydrological model that implements a spatial aggregation strategy ("discretisation") in order to reduce its computational demands. Hydrological similar areas identified by the discretisation procedure are referred to as hydrological response units (HRUs). Beven and Freer (2001) introduced a "dynamic" variant that addressed some of the limitations of the original TOPMODEL but which retained its computational and parametric efficiency. In particular, the original assumption of a quasi-steady water table was replaced by time-dependent kinematic routing between and within HRUs.

The new formulation allows a more flexible discretisation, variable upslope drainage areas and spatially variable physical properties, allowing the introduction of any type of landscape data to identify the HRUs. It retains the core dynamics of the FORTRAN implementation but makes use of data storage and vectorisation features of the R language to allow efficient scaling of the problem domain. The preprocessing routines supplied incorporate handling of geo-referenced spatial data to allows it to integrate with modern GIS through industry-standard file formats such as GEOTiff and ESRI Shapefiles.

References

Beven, K. J. and M. J. Kirkby (1979). A physically based variable contributing area model of basin hydrology. Hydrol. Sci. Bull 24(1): 43-69.

Beven, K. J. and J. Freer (2001). A Dynamic TOPMODEL. Hydrological Processes 15(10): 1993-2011.

Metcalfe, P., Beven, K., & Freer, J. (2015). Dynamic TOPMODEL: A new implementation in R and its sensitivity to time and space steps. Environmental Modelling & Software, 72, 155-172.

See Also

`discretise`

`run.dtm`

`NSE` *Nash Sutcliffe Efficiency of a model's output against observations*

Description

Returns the the NSE (NSE, Nash and Sutcliffe, 1970) of the simulated values against the given observations.

Usage

```
NSE(qsim, qobs, digits = 2)
```

Arguments

`qsim`	Time series or vector of simulated values.
`qobs`	Time series or vector of observations.
`digits`	No. DP in returned value.

Value

A number <= 1 indicating the goodness of fit of the simulated series against observations (1= perfect fit). Values of >0.8 are generally regardred as "behavioural".

Author(s)

Peter Metcalfe

References

Nash, J., & Sutcliffe, J. V. (1970). River flow forecasting through conceptual models part I-A discussion of principles. Journal of hydrology, 10(3), 282-290.

Examples

```
## Not run:
require(dynatopmodel)

data(brompton)

# Goodness of fit for the storm simulation

NSE(brompton$storm.run$qsim, brompton$storm.run$qobs)

## End(Not run)
```

`run.dtm` *Run Dynamic TOPMODEL using the catchment areal groupings (response units) for a discretisation.*

Description

Run Dynamic TOPMODEL using the catchment areal groupings (response units) for a discretisation.

Usage

```
run.dtm(groups, weights, rain, routing, qobs = NULL, qt0 = 1e-04,
  pe = NULL, dt = NULL, ntt = 1, ichan = 1, i.out = ichan[1],
  vchan = 1000, vof = 100, dqds = NULL, sim.start = NA, sim.end = NA,
  disp.par = disp.par())
```

Arguments

`groups`	Data frame of ngroup areal group definitions along with their hydrological parameters.
`weights`	The flux distribution (weighting) ngroup*ngroup matrix. Usually generated by the discretise method.
`rain`	A time series of rainfall data in m/hr. One column per gauge if multiple gauges used. Use aggregate_obs to apply a different time interval to this and the other observation data.
`routing`	Channel routing table comprises a two-column data.frame or matrix. Its first column should be average flow distance to the outlet in m, the second the proportions of the catchment channel network within each distance category. Can be generated by make.routing.table
`qobs`	Optional time series of observation data
`qt0`	Initial specific discharge (m/hr)
`pe`	Time series of potential evapotranspiration, at the same time step as rainfall data
`dt`	Time step (hours). Defaults to the interval used by the rainfall data

`ntt`	Number of inner time steps used in subsurface routing algorithm.
`ichan`	Integer index of the "channel" group. Defaults to 1
`i.out`	For multi-channel systems, the index of the outlet reach
`vchan`	Default channel routing velocity (m/hr)
`vof`	Default overland flow routing velocity (m/hr).
`dqds`	Function to supplies a custom flux-storage relationship as the kinematic wave celerity. If not supplied then exponential relationship used
`sim.start`	Optional start time for simulation in any format that can be coerced into a POSIXct instance. Defaults to start of rainfall data
`sim.end`	Optional end time of simulation in any format that can be coerced into a POSIXct instance. Defaults to end of rainfall datA
`disp.par`	List of graphical routing parameters. A set of defaults are retrieved by calling disp.par()

Details

The grouping (HRU) table may be generated by the discretise method and includes each indexed channel as separate group. See Metcalfe et al. (2015) for descriptions of the parameters maintained in this table.

Evapotranspiration input can be generated using the approx.pe.ts method

If disp.par$graphics.show = T then the output will be displayed graphically whilst the simulation is in progress. Otherwise simulated specific discharges

Value

A list containing run output and input data. These include

qsim: time series of specific discharges (m/hr) at the specified time interval. can be converted to absolutre discharges by multiying by catch.area

catch.area: the catchment area in m^2, calculated from the areas in the groups table

data.in: the parameters supplied as input to the call to run.dtm

sim.start: start of simulation

sim.end: end time of simulation

fluxes: a list comprising, for each response unit the specific base flows qbf, specific upslope inputs qin, drainage fluxes quz, and any overland flow qof, all in m/hr

storages: a list comprising, for each response unit, root zone and unsaturated storage and total storage deficit (all m)

Author(s)

Peter Metcalfe

References

Beven, K., & Freer, J. (2001). A dynamic topmodel. Hydrological processes, 15(10), 1993-2011.

Metcalfe, P., Beven, K., & Freer, J. (2015). Dynamic TOPMODEL: A new implementation in R and its sensitivity to time and space steps. Environmental Modelling & Software, 72, 155-172.

Examples

```
## Not run:
require(dynatopmodel)
data(brompton)
# September 2012 storm event
# The response is sensitive to the size of the channels, but many are small.
# Set an overall width of 2m.
chans <- build_chans(dem=brompton$dem, drn=brompton$drn, chan.width=2)

# discretisation by reverse distance from nearest channel. The raster brompton$flowdists
# gives the D8 flow pathway distance for every area in the catchment
layers <- addLayer(brompton$dem, 2000-brompton$flowdists)
disc  <- discretise(layers, cuts=c(flowdists=5), chans=chans, area.thresh=3/100)

Network routing table
routing <- build_routing_table(brompton$dem, chans)

# Here we apply the same parameter values to all groups. Suggest applying smaller m and td values to
# the closest areas to simulate a fast response due to the artificial drainage.
# It would also be possible to supply a custom transmissivity profile that has
# a discontinuity at the depth of the drains
groups <- disc$groups
groups$m <- 0.011
groups$td <-  42
# a very high transmissivity prevents saturation flow as there appears be little
groups$ln_t0 <- 18
groups$srz_max <- 0.1
# initial root zone storage
groups$srz0 <- 0.87
# quite slow channel flow, which might be expected with the shallow and reedy
# reaches in this catchment
groups$vchan <- 750

# Observations at a 15 minute time step
dt <- 0.25
obs <- list(rain=brompton$rain,
pe=brompton$pe,
qobs=brompton$qobs)
obs <- aggregate_obs(obs, dt=dt)

# parameters for graphics output
par <- disp.par(int.time=24)

# Note max.q in mm/hr
par$max.q <- 1000*max(obs$qobs, na.rm=TRUE)
sim.start <- "2012-09-23"
```

```
sim.end <- "2012-10-01"

# Ensure output goes to a new window
options("device"="X11")
# take initial discharge from the observations
qt0 <- as.numeric(obs$qobs[sim.start][1])

# Run the model across the September 2012 storm event using 2 inner time steps
and a 15 minute interval
 storm.run <- run.dtm(groups=groups,
   weights=disc$weights,
   rain=obs$rain,
   pe=obs$pe,
   qobs=obs$qobs,
   qt0=qt0,
   sim.start=sim.start,
   sim.end=sim.end,
   routing=routing,
   disp.par=par,
   ntt=2)
 # show run statistics
 cat("NSE=", NSE(storm.run$qsim, storm.run$qobs))
 cat("Time at peak =", format(time_at_peak(storm.run$qsim)))

## End(Not run)
```

`time_at_peak` *Time of maximum observation*

Description

Determine the time of the maximum item in the supplied time series.

Usage

```
time_at_peak(ts, icol = 1)
```

Arguments

`ts`	Time series
`icol`	Column index if a multi-column time series

Author(s)

Peter Metcalfe

Peter Metcalfe

Examples

```
require(dynatopmodel)

data(brompton)

with(brompton$storm.run, time_at_peak(qsim))
```

`time_to_peak` *Time between the peak rainfall and the peak discharge*

Description

Return the lag, in hours, between the peak in the rainfall record and that of the discharge response.

Usage

```
time_to_peak(rain, qsim, units = "hour")
```

Arguments

`rain`	Time series of rainfall.
`qsim`	Time series of discharges.
`units`	Units in which to return the value.

Author(s)

Peter Metcalfe

See Also

time_at_peak

Examples

```
require(dynatopmodel)

data(brompton)

with(brompton$storm.run, time_to_peak(rain, qsim))
```

`upslope.area` *Upslope contributing area and wetness index calculation*

Description

Determine upslope contributing area based on an elevation raster and, optionally, compute the topographic wetness index.

Usage

```
upslope.area(dem, log = TRUE, atb = FALSE, deg = 0.1, fill.sinks = TRUE)
```

Arguments

`dem`	Elevation raster (in m), using a projected coordinate system with identical x and y resolutions.
`log`	Return the natural log of the values.
`atb`	If TRUE, include both the upslope contributing area and the topographic wetness index $ln(a/tan(beta))$. Otherwise calculate just the upslope area.
`deg`	Minimum intercell slope to identify with a sink (degrees).
`fill.sinks`	Fill sinks before calculation using the threshold angle given by deg.

Note

This is a wrapper to the function implemented in the TOPMODEL package by Wouter Buytaert.

Author(s)

Peter Metcalfe and Wouter Buytaert

References

Quinn, P. F., Beven, K. J., & Lamb, R. (1995). The In (a/tan/beta) index: How to calculate it and how to use it within the Topmodel framework. Hydrological processes, 9(2), 161-182.

Examples

```
## Not run:
require(dynatopmodel)
data(brompton)

a.atb <- upslope.area(brompton$dem, atb=TRUE)
sp::plot(a.atb, main=c("Upslope area (log(m^2/m))", "TWI log(m^2/m)"))

## End(Not run)
```

Index